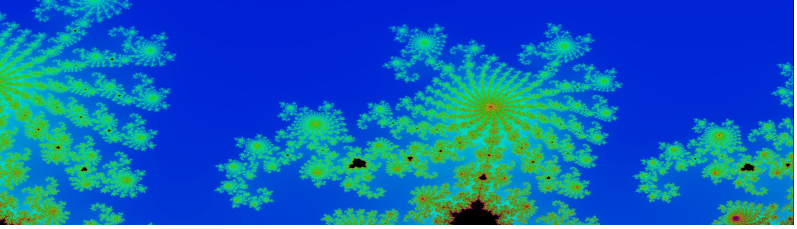


École doctorale
Mathématiques
et informatique

université de BORDEAUX

École doctorale numéro 39 : Mathématiques et Informatique

Université de Bordeaux

Manuscrit HDR

pour obtenir l'habilitation à diriger des recherches délivrée par

Université de Bordeaux

Spécialité "Mathématiques et Informatique"

présentée et soutenue publiquement le 28 novembre 2019 *par*

Pierrick LEGRAND

Artificial evolution, fractal analysis and applications

Jury

M. William B. Langdon,	Professeur	Rapporteur
Mme. Régine Le Bouquin Jeannès,	Professeur	Rapporteur
M. Jacques Levy-Vehel,	DR INRIA	Invité
M. Fabien Lotte,	DR INRIA	Examinateur
Mme. Evelyne Lutton,	DR INRA	Examinateur
M. Jérôme Saracco,	Professeur	Rapporteur

Université de Bordeaux
Inria Bordeaux Sud-Ouest
Institut de Mathématiques de Bordeaux
UMR CNRS 5251, Bordeaux, France

H
D
R

Contents

List of figures

List of Tables

Chapter 1

Introduction

Contents

This document is a selection of the research activities carried out since the defence of my doctoral thesis in December 2004. Since this is the presentation of an authorization to direct research, it seemed natural to me to choose to present as a priority work related to thesis or master's supervisions. This multidisciplinary work has, for the most part, been carried out as part of funded scientific projects. I will set out Section 1.1 some of these scientific projects, highlight the associated supervisions and try to make the link with what is presented in the rest of the document. Choices had to be made and I apologize to the (ex-)students and collaborators whose work I do not mention and who have done a significant amount of work. In the Section 1.2 I will present synthetically my research axes and give the plan of this manuscript which results from it Section 1.3.

1.1 Research projects

1.1.1 Psychology and Sound Interactions

The first project I was in charge of after my arrival at the University of Bordeaux was the **PSI project** (Psychology and Sound Interactions), a project that won the call for projects **Neuroinformatique of CNRS** in 2008 and continued until 2011. The objectives of this project were to:
1) Create a database of EEG signals obtained by acquisition on subjects in a given psychophysiological state. 2) Develop a method for classifying EEG signals according to the psychophysiological state of the subject. Define a distance on the classes obtained. 3) Implement a listening system with parameterized music generation. 4) Develop a prototype allowing feedback between the music generated and the psychophysiological state of a subject in order to bring him to a target state.
The work carried out was as follows:
Acquisition of EEG signals
The Neuroinformatics program has made it possible to acquire high quality equipment for the acquisition of EEG signals (a fixed system in a controlled environment and a portable system) DELTAMED. The experiments were monitored by **F. Faïta** assisted by **S. Bouaffre**. A database has been built.
Musical synthesis
The musical synthesis aspect was mostly handled by **P.H. Vulliard**, **J. Larralde** and **M. Desainte-Catherine**. The following work was carried out to define parameters for generating a wide spectrum of sounds and music. 1) Creation of rhythmic patterns and rhythms for melodies (Matlab). 2) Algorithm of tonal and non-tonal scale generation (Matlab). 3) Generation of tonal and modal melodies (Max). 4) Generation of improvisation from a melody (Ruby, TER engineer ENSEIRB, **Mathieu Carpentier**). 5) Generation of monophonic binaural beats + chords (Max). 6) Use of FM sounds + samples + granular synthesizer + natural environments + physical models (Max + Csound).
Prototype
The music feedback prototype - EEG is functional. It consists of a sound broadcasting system, a laptop computer, an EEG acquisition system.
To make this prototype work, the following technical difficulties had to be solved:
1) Generate parameters (from the Matlab software) allowing musical synthesis (in the MaxMsp software). 2) Establish a two-way communication Matlab / MaxMsp. 3) Establish a two-way communication Matlab / Coherence (proprietary acquisition software developed by the company DELTAMED) : Acquisition of real-time data in Matlab and sending of real-time markers on EEG signals in Coherence. 4) Programming of an evolutionary algorithm (under Matlab) to evaluate the synthesis parameters music thanks to the content of EEG signals (measured and processed in real time) and to generate new parameters adapted to this subject at this time.

This project formed the first foundations of the following projects and led to the creation of a working group on signal processing, neuroscience and EEG signal classification. It is therefore quite naturally that a project for the Aquitaine Region was submitted in order to continue the work on this theme and obtain thesis funding.

1.1.2 Regional PSI Project: Reduction of dimension in supervised learning. Applications to the study of brain activity

This project, of which I was the editor and lead/coordinator, took place from January 2010 to July 2014. Its purpose was to create a human-machine interface that would allow the subject's state of alertness to be modified using musical stimuli generated in real time. These stimuli are created according to the measurement obtained of the subject's state of alertness at each moment. The goal is therefore to be able to lead a person to a target psychophysiological state (so-called "relaxed" or "normal" state) depending on the desired objective with the help of synthesized music.

This project fully funded the thesis of **Laurent Vezard** (thesis supervised by **M. Chavent** (33%), **F. Faïta** (33 %), **P. Legrand** (34%)). The objective of the thesis supported by this regional project was to provide a method capable of predicting, using a minimum number of electrodes, an individual's state of alertness using his brain activity collected by electroencephalography. This thesis is part of the overall PSI project (see yellow box on the left of the Figure 1.1)

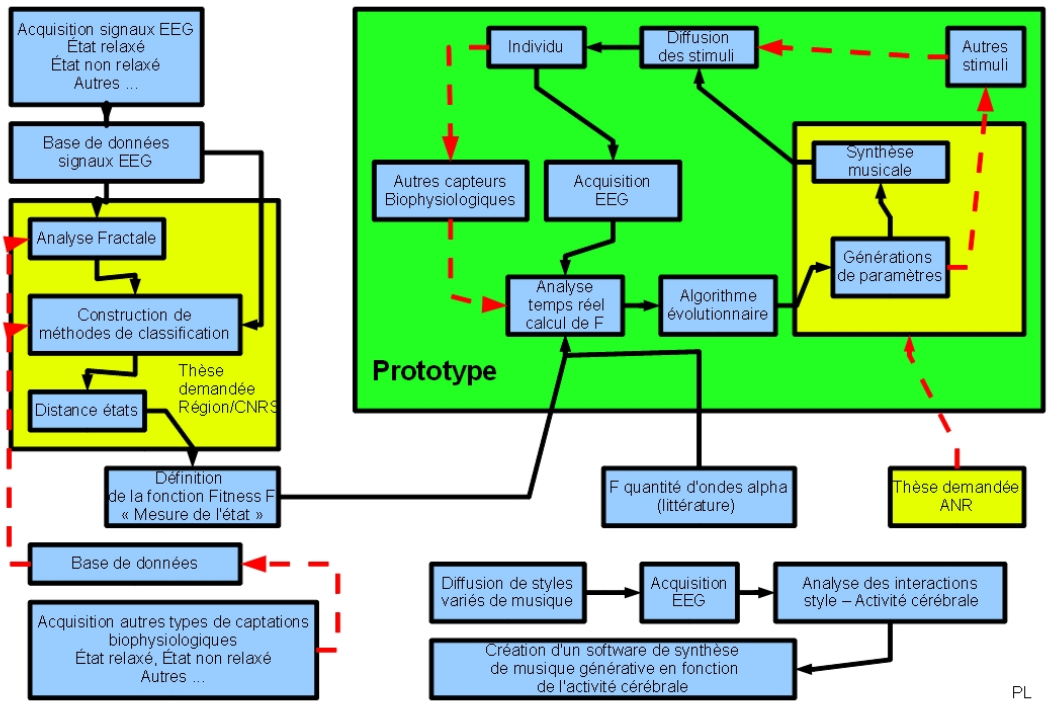


Figure 1.1 – CNRS Neuroinformatics Project and PSI Regional Project.

During part of the first year, a bibliographical work was carried out. This made it possible to review the current state of research on methods for processing data from electroencephalography and predicting states of vigilance. A data collection campaign was conducted to obtain EEG records for 44 subjects. We then processed the EEG signals. We based ourselves on the study of the CNV (Contingent Negative Variation) of the participants using the work of NAITOH et collab. [1971]; TECCE [1979] or TIMSIT-BERTHIER et collab. [1981] to

verify the effectiveness of the relaxation session on each of the subjects. In order to carry out this study, I developed a CNV analysis and visualization software, which allowed my collaborators to be able to use the data without having to deal with programming or signal processing considerations. This software is presented in Figure 1.2.

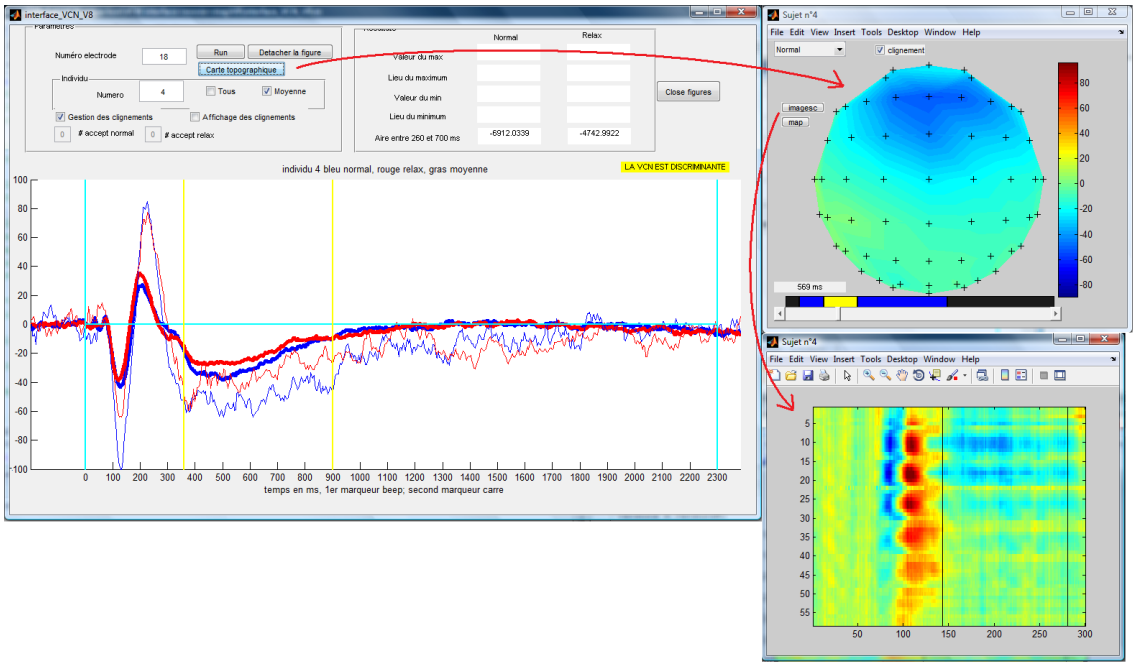


Figure 1.2 – Software for visualization and analysis of the Contingent Negative Variation Analysis. This software validated the effectiveness of the relaxation session on the subjects.

Participants for whom relaxation was considered ineffective were removed from the study. Thus, records of 13 subjects were retained after the CNV study. A criterion was extracted from the EEGs of these 13 subjects using a discrete wavelet transform (MALLAT [2008] and DAUBECHIES [1992]). A genetic algorithm was then used. The purpose of this algorithm was to select an electrode and a frequency range to use in order to discriminate between the two states of vigilance. This approach determined the most useful electrode for the classification task. Using the recording of this electrode, the prediction obtained has a reliability rate of 89.33%. The approach developed and the results obtained have been the subject of publications and conference presentations. The weakness of this approach is that too few records are available. This has an effect on the robustness of the estimate of the reliability of the prediction obtained. As a result, new recordings were required. In the second year of the thesis, the genetic algorithm developed in the first year was improved to include electrode combinations. When analysing the results obtained, we judged it necessary to obtain new signals in order to enlarge the available database. Thus, a new acquisition campaign was carried out. **Nidal El Yacoubi** and **Emilie Drouineau** joined the team as part of a two-month internship and participated in the data acquisition. This campaign collected EEG signals from an additional 14 participants. The same validation method used in the first year of the thesis (based on the CNV study) was applied to eliminate participants for whom relaxation had no effect. Thus, 6 subjects from this acquisition campaign have been retained. A part of this work is presented in Chapter 15.

Following a talk by **Fabien Lotte** about spatial filters and in particular the CSP (common spatial pattern), we thought it was possible to use this work in our study. Thus, the articles BLANKERTZ et collab. [2008]; LOTTE et GUAN [2011]; RAMOSER et collab. [2010] were studied in particular. This allowed us to understand how this method works. We have modified our genetic algorithm to include CSP and allow electrode combinations. The CSP is a method for constructing synthetic variables defined as linear combinations of the ini-

tial variables. A weight is therefore assigned to each of the original variables to create the synthetic variable. The use of a genetic algorithm was considered. This, combined with an evaluation based on the use of the CSP, selected the most useful subgroup of variables for the classification task. Thus, 9 electrodes were selected. The resulting correct classification rate is 71.59%. The approach has been published and presented in conferences. The weak point of this approach being the calculation time, two other iterative approaches have been proposed: backward and forward CSP. The results of these approaches seem to be very promising (reduced calculation time and correct classification rates comparable to that of the subgroup of electrodes extracted by the genetic algorithm).

During the third year of the thesis, iterative approaches using CSP (genetic algorithm, forward and backward CSP) were supplemented by a parsimonious version of the CSP algorithm. After having expressed the problems of optimizing the CSP as Main Component Analysis (PCA), we used the work of ZOU et collab. [2006] on the parsimonious PCA. Thus, a parsimonious version of the CSP was proposed. It allows to select a subgroup of electrodes for the calculation of CSP filters. The calculation time of this approach is the lowest of the different approaches proposed during this thesis. The results obtained are comparable to those of the genetic algorithm associated with CSP (73.11% with 13 electrodes).

Laurent Vezard's thesis consolidated the synergies around the research activities carried out on brain activity. The Chapter 15 of this manuscript is dedicated to a part of these thesis works.

1.1.3 European project IRSES FP7: Analysis and Classification of mental states of vigilance with evolutionary computation

Then, I decided to pursue the development of international collaborations that I had already initiated with Mexico and in particular with **Leonardo Trujillo**, a researcher at the ITT (Tijuana Institute of Technology). Indeed, we had already published several articles together, two of which will be presented in Chapters 12 and 13. In addition, I had recently joined the Tree-Lab laboratory (*TREE-LAB is part of the Cybernetics research line within the Engineering Science graduate program offered by the Department of Electric and Electronic Engineering at Tijuana's Institute of Technology (ITT), in Tijuana Mexico. TREE-LAB is mainly focused on scientific and engineering research within the intersection of broad scientific fields, particularly Computer Science, Heuristic Optimization and Pattern Analysis. In particular, specific domains studied at TREE-LAB include Genetic Programming, Classification, Feature Based Recognition, Bio-Medical signal analysis and Behavior-Based Robotics. Currently, TREE-LAB incorporates the collaboration of several top researchers, as well as the participation of graduate (doctoral and masters) and undergraduate students, from ITT. Moreover, TREE-LAB is actively collaborating with top researchers from around the world, including Mexico, France, Spain, Portugal and USA.*)

I therefore submitted and directed a European project and integrated Mexico into third countries around a project combining signal processing, artificial evolution and EEG signal classification. This project, which began in November 2013 and ended in October 2016, funded approximately 50 months of mobility for students and researchers from Mexico, Portugal, Spain and France and generated the scientific production of 31 journal articles, 2 books, 28 conference articles with publication of the proceedings and 1 book chapter. I have written a 34-page report, in English, on this project, available at the following address:
https://www.math.u-bordeaux.fr/~plegra100p/FINAL_REPORT_ACOBSEC/FINAL_REPORT_ACOBSEC_2016.pdf
Abstract of the proposal:
Over the last decade, Human-Computer Interaction (HCI) has grown and matured as a field. Gone are the days when only a mouse and keyboard could be used to interact with a computer. The most ambitious of such interfaces are Brain-Computer Interaction (BCI) systems. BCI's goal is to allow a person to interact with an artificial system using brain activity. A common approach towards BCI is to analyze, categorize and interpret Electroencephalography (EEG) signals in such a way that they

alter the state of a computer. ACoBSEC's objective is to study the development of computer systems for the automatic analysis and classification of mental states of vigilance; i.e., a person's state of alertness. Such a task is relevant to diverse domains, where a person is required to be in a particular state. This problem is not a trivial one. In fact, EEG signals are known to be noisy, irregular and tend to vary from person to person, making the development of general techniques a very difficult scientific endeavor. Our aim is to develop new search and optimization strategies, based on evolutionary computation (EC) and genetic programming (GP) for the automatic induction of efficient and accurate classifiers. EC and GP are search techniques that can reach good solutions in multimodal, non-differentiable and discontinuous spaces; and such is the case for the problem addressed here. This project combines the expertise of research partners from five converging fields: Classification, Neurosciences (University of Bordeaux), Signal Processing, Evolutionary Computation and Parallel Computing in Europe (France Inria, Portugal FCUL, Spain UNEX) and South America (Mexico ITT, CICESE). The exchange program goals and milestones give a comprehensive strategy for the strengthening of current scientific relations amongst partners, as well as for the construction of long-lasting scientific relationships that produce high quality theoretical and applied research.

Our aim was to develop new search and optimization strategies, based on evolutionary computation (EC) and genetic programming (GP) for the automatic induction of efficient and accurate classifiers. EC and GP are search techniques that can reach good solutions in highly multi-modal, nondifferentiable and discontinuous spaces; and such is the case for the problem addressed here: the detection of mental states of vigilance. This project combines the expertise of researcher partners from five converging fields: Classification, Neurosciences, Signal Processing, Evolutionary Computation and Parallel Computing, in Europe - France (Inria, University of Bordeaux), Portugal (BioISI) and Spain (UNEX) - and North America - Mexico (ITT and CICESE). The partners complement and enhance their respective disciplines, allowing for a strong multi-disciplinary collaboration and proposal development. The exchange program goals and milestones gave a comprehensive strategy for the strengthening of current scientific relations amongst the partners, and allowed the construction of long-lasting scientific relationships that produced high quality theoretical and applied research.

Within the framework of this project, several doctoral theses were defended, in particular those of **Yuliana Martinez**, **Enrique Naredo** and **Emigdio Z. Flores** that I had the chance to co-supervise. In the same way, this mobility program allowed me to participate in particular in the master's supervision of **Uriel Lopez Islas**, **Enrique Hernandez**, **Victor Raul Lopez** and **Luis Herrera** (which also contributed to the HUMO project see Section 1.1.4 and Figure 1.3).

The work done with these students has contributed to make the ACOBSEC project a success. It is therefore natural to pay tribute to their involvement by integrating our common, theoretical and applied contributions to this document. Thus, works carried out within the context of Yuliana Martinez's thesis will be presented in Chapters 4 and 3, others carried out within the context of Enrique Naredo's thesis Chapter 5, others carried out within the scope of Emigdo Z. Flores' thesis, Chapters 6, 7 and 16, others realized as part of Uriel Lopez-Islas' master Chapter 8.

The management of a European project is an enriching experience humanly, scientifically (and administratively). These advantages have not detracted from the desire to continue to set up projects on the Bordeaux campus as well. Thus I had the chance to have research projects selected through the ALBATROS GIS calls for projects.

1.1.4 HUMO Micro-Projects (Human monitoring) of GIS Albatros

From September 2015 to December 2017, I was scientific manager of the HUman MOnitoring (HUMO), 1, 2 and 3 projects. These projects were carried out in partnership with the IMS and ENSC as part of the ALBATROS GIS calls for projects (https://www.bordeaux-inp.fr/fr/gis-albatros). The HUMO project (HUman Monitoring - Interdisciplinary ap-

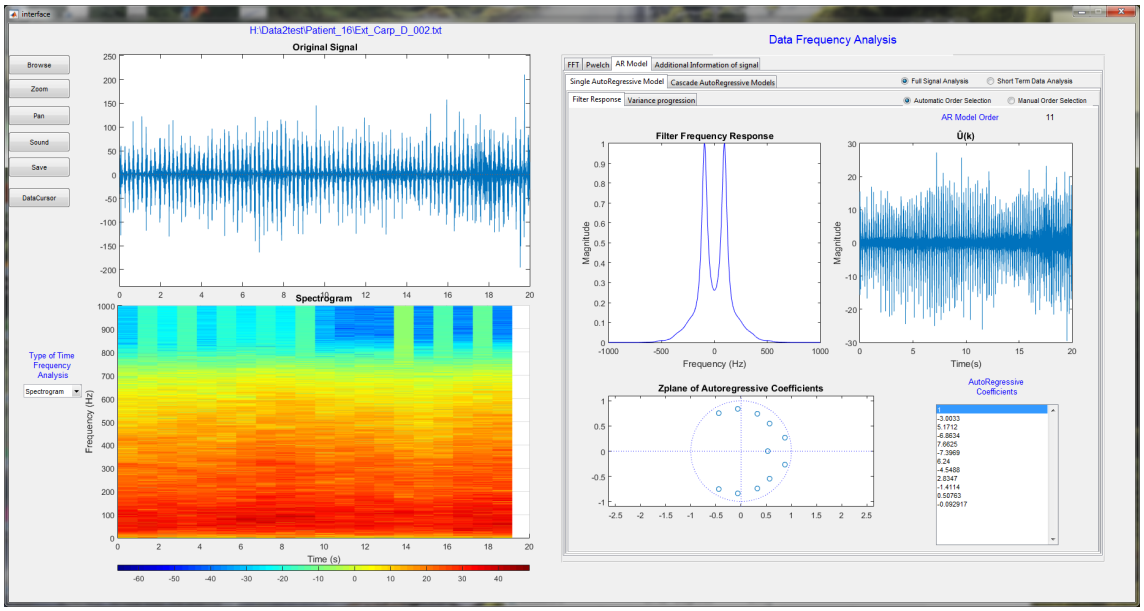


Figure 1.3 – Signal analysis software (L. Herrera, E. Grivel, P. Legrand). Software development for the analysis of biological signals. The objective of this Luis Herrera Master's internship (carried out jointly as part of the European IRSES ACOBSEC project and the HUMO project) was to program various time-frequency analysis methods under a Matlab environment and to design an ergonomic graphical interface to use these programs.

proach to the assessment of the cognitive state of the user) aims to validate a protocol for the collection, processing, and interpretation of physiological data for the assessment of the cognitive state of charge of the user (crew monitoring). This project is located at the interface of the human-system interaction theme and the signal and image theme within the GIS Albatros. It now brings together experts from different disciplinary fields, laboratories and research centers on the Bordeaux campus. This project was the opportunity to co-supervise (50%-50%) with **Eric Grivel** (IMS) the Enseirb engineering internship from **Vincent Lenhardt**.

One of the major difficulties of the project was to synchronize the acquisition of data from various sensors in time. It is a technological barrier on which I was particularly involved and which was lifted by building the following system (see Figure 1.4):

- Use of a Dell computer, dual intel xeon E5-2609 v3 processor (6 cores, 1.89Ghz, 32gigas of DDR4 RAM) with a MATLAB license with DAQ and ICT toolboxes.
- Acquisition EEG : use of a 16 channel gBSamp amplifier connected to an acquisition card or connected itself by USB to the acquisition computer.
- Acquisition of pupillometry data: use of a Tobii eye tracker connected in USB to the acquisition computer
- Acquisition of ECG and Electrodermal response: use of a BITalino card connected in bluetooth to the acquisition computer

Our work made it possible to control and synchronize all the acquisitions from the acquisition PC via Matlab (see Figure 1.5). To be more precise, it was necessary to interface Matlab and the NI card using DAQ toolbox functionalities for the EEG. It was necessary to calibrate, control the eye tracker using the Tobii APIs for Matlab and then extract the pupil diameters from the raw data. Finally, we used features from Matlab's ICT toolbox to connect the acquisition PC to the BITalino card using Bluetooth. Then via this Bluetooth connection we were able to retrieve the ECG and EDA data with Matlab. We therefore built a system

of parallel acquisitions of various physiological data, which provided relevant data for the rest of our study.

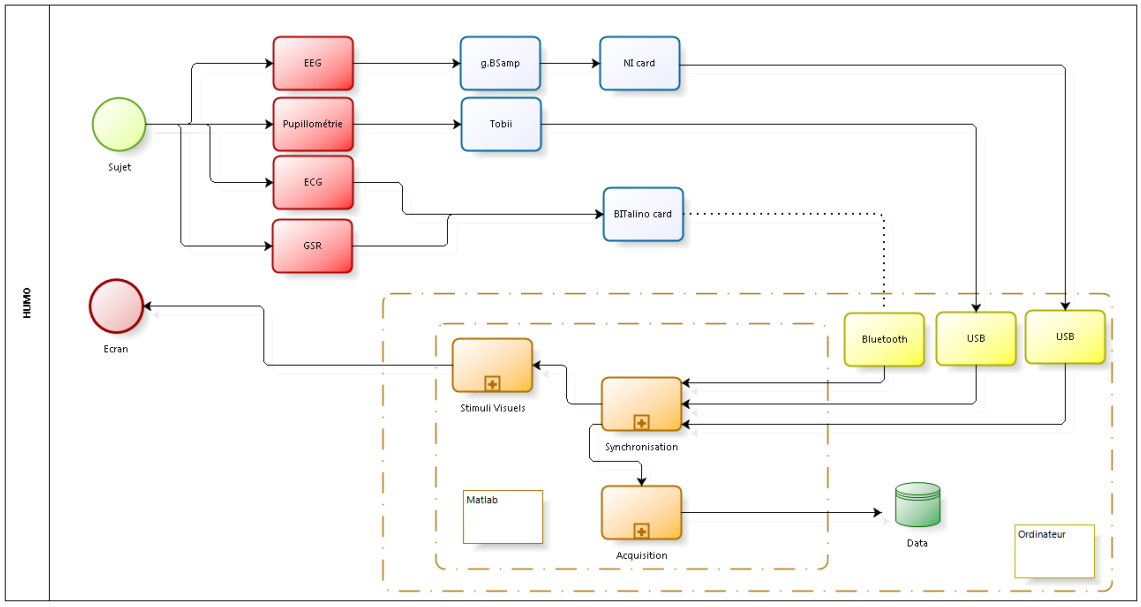


Figure 1.4 – HUMO acquisition protocol (P. Legrand)

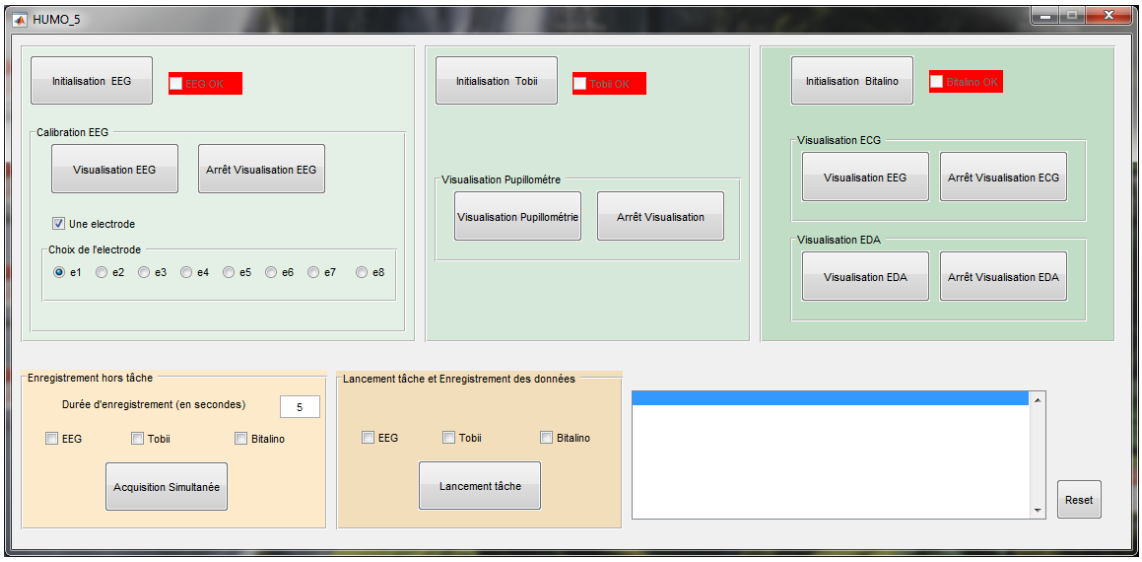


Figure 1.5 – Multi-sensor acquisition software HUMO (P. Legrand)

At the end of the HUMO project, a CIFRE THALES thesis funding was obtained (Director: Pierrick Legrand (34%), Co-Director **Eric Grivel** (33%), Co-supervisor : **Jean-Marc André** (33%)). This thesis, currently being carried out by **Bastien Berthelot** is entitled "Signal processing algorithms for extracting robust signatures on bio-signals". Some of the work carried out as part of this thesis is mentioned in Chapter 10. Two patents have been filed in the context of this thesis and are included in their "legal" version Chapter 11.

1.1.5 Micro-Project Micro-Doppler of GIS Albatros

From January 2017 to December 2017, I was the leader of the micro-project entitled "Contribution of time-frequency analysis and interpolation techniques for micro-Doppler estimation". This project was carried out in partnership with IMS and THALES.

The purpose of this micro-project was to analyze and process the micro-doppler phenomenon, induced by the mechanical vibrations or structural rotations of a marine object. This subject had in perspective the setup of a doctoral thesis. During this micro-project, a state of the art on methods of analysis of the micro-doppler, whatever the application context, with a particular focus on techniques of time-frequency analysis was performed. Existing methods were simulated and areas for improvement were proposed.

This project was the opportunity to co-supervise (50%-50%) with Eric Grivel (IMS) the contract of **Sabrina Machhour** (who was recruited at THALES at the end of this project). The work carried out was presented at the Eusipco 2018 conference MACHHOUR et collab. [2018]. The preparation of the envisaged thesis is currently being finalized (administrative locks removed and future doctoral student identified) but the theme has changed slightly.

1.2 Synthetic presentation of the research themes

My research focuses on two main areas: the measurement of signal regularity and evolutionary algorithms. In these two areas, I am interested in theoretical developments and implementation. Finally, I am interested in combining these tools to create other tools and to solve real problems (biomedical or other). The contributions are indicated in the following sections and the cited bibliography is given at the end of this introductory chapter.

1.2.1 Hölderian regularity, Hurst exponent, DFA and theoretical contributions

Many applications show that the characterization of local signal regularity obtained using wavelet techniques or fractal analysis tools is relevant for their description and processing LEGRAND [2009]. Although the analysis of the regularity of a signal and its estimation is a relatively recent problem in the field of signal processing, it is one of the fundamental characteristics of a signal even if there are an infinite number of signals with the same regularity at each point. A branch of fractal analysis, the Hölderian regularity analysis, offers mathematical tools to characterize the regularity of a signal LEGRAND [2009]. In particular, the Hölder exponent is particularly suitable for measuring the regularity of signals.

Several methods are available to estimate the Hölder exponent. The most natural, because it follows the definition of the exponent, is to study the oscillations around the considered point TRICOT [1995]. A second one is based on a theorem by Stéphane Jaffard and is based on a wavelet decomposition JAFFARD [2004].

1 - From these methods of estimation of the Hölder exponent, it was possible to construct denoising methods with asymptotic convergence rates similar to the most efficient methods in the literature LEGRAND [2004]; LEGRAND et LÉVY VÉHEL [2003a]; LÉVY VÉHEL et LEGRAND [2003].

2 - It has also been possible to develop an interpolation method that maintains Hölderian regularity regardless of the number of interpolations chosen LEGRAND et LÉVY VÉHEL [2003b]; LÉVY VÉHEL et LEGRAND [2006].

3 - These contributions have been integrated into the Fraclab toolbox LÉVY VÉHEL et LEGRAND [2004].

For a few years now I have also been interested in the estimation of the Hurst exponent. The Hurst exponent is directly linked to the Hölder exponent for monofractal signals. This exponent is used to measure the rate of decrease of the autocorrelation function as the delay increases. In particular, I am interested in the variants of the Detrended Fluctuation Analysis method.

Indeed, the detrended fluctuation analysis (DFA) is a well-established method to detect long-range correlations in time series. It has been used in a wide range of applications, from biomedical applications to signal denoising. It allows the Hurst exponent of a pure monofractal time series to be estimated. It operates as follows: after integration, the signal is

split into segments. Using a least-squares criterion, local trends are deduced. The resulting piecewise linear trend is then subtracted to the whole signal. The power of the residual is computed for different segment lengths and its log-log representation allows the Hurst exponent to be deduced.

The following contributions have been made and published (or are in the process of being published) as part of Bastien Berthelot's thesis (see Chapter 10).
The following contributions have been proposed:

- An alternative version of the DFA
- A common mathematical formalisation of the variants of the DFA
- An interpretation of the fluctuation functions of the DFA and variants as the result of a filtering BERTHELOT et collab. [2019b]
- A 2D Fourier Transform Based Analysis Comparing the DFA with the DMA BERTHELOT et collab. [2019c]
- A Filtering-based Analysis Comparing the DFA with variants for Wide Sense Stationary Processes BERTHELOT et collab. [2019a]

1.2.2 Evolutionary algorithms and theoretical contributions

Genetic Algorithms (GA) :
Genetic algorithms are optimization algorithms based on Darwinian theory of evolution (De Jong (1975) DE JONG [1975] and HOLLAND [1975]). The general idea is that a population of potential solutions will improve its characteristics over time through genetic mutations and crossovers in order to adapt as best as possible to its environment. The purpose of these algorithms is to optimize an evaluation function (fitness) on a research space. Individuals (called "parents" and corresponding to points in the research space) will be created to form a diverse population. They are represented by genomes (binary or real codes, of fixed or variable size). Using mutation and crossing operators, parents will give birth to children who will in turn be evaluated. The best individuals (including children and/or parents) will survive. The algorithm can, for example, be iterated until all individuals are identical (algorithm convergence).
Genetic Programming (GP) :
Genetic programming is quite similar to genetic algorithms. However, the research space is then a functional space. Indeed, in the context of genetic programming, we are no longer looking for a set of parameters to optimize a criterion but to build a function. Of the current algorithms, GP is one of the most advanced forms of evolutionary research LANGDON et POLI [2002]. In the classic GP form, each solution is represented as a tree that can represent a function. Trees are built using elements of two sets, a set of functions and a set of terminals. These two sets define the GP's research space.
The following contributions have been produced and published:

1 - Determining the difficulty of a problem is an important issue in the field of artificial evolution, and has been for years. From an algorithmic point of view, the difficulty of a problem can be associated with the execution time or memory required to find an optimal solution. As part of **Yuliana Martinez**'s thesis, we were interested in building models that can predict the performance of a GP-classifier without having to run the program or sample potential solutions in the research space MARTINEZ et collab. [2016]; TRUJILLO et collab. [2011, 2012b,c]. The principle is as follows: For a given classification problem, we apply a pre-processing step to simplify the feature extraction process. Then we perform the step of extracting the characteristics of the problem. Finally, we use a PEP (prediction of expected performance) model, which takes the characteristics of the problem as input and produces the predicted classification error on the test set as output. To build the PEP model, we used a

supervised learning method with a GP. Then, to refine this work, we developed an approach using several PEP models, each now becoming a specialized predictors of expected performance (SPEP) specialized for a particular group of problems. It appears that the PEP and SPEP models were able to accurately predict the performance of a GP-classifier and that the SPEP approach gave the best results MARTINEZ et collab. [2016]; TRUJILLO et collab. [2011]. See Chapter 3.

2 - Genetic programming has shown impressive results in various fields of application. However, there are still practical limitations to overcome, in particular, its computational cost, overfitting and excessive tree size increase (sometimes without correlation with the improvement of the quality of the individual, the so-called bloat). A particular feature of a GP is that it looks for symbolic expressions that best describe the relationships in a learning set of input and output pairs. These pairs are described in the literature as fitness-cases. Recently, researchers have proposed new approaches to avoid some of the problems mentioned above and have obtained interesting results by proposing methods for sampling fitness-cases to be considered during the evaluation of individuals. After an intensive comparison of various fitness case sampling methods, some of which we have proposed, it appeared through **Yuliana Martinez**'s thesis that the choice of fitness calculation method has an influence on the bloat, the calculation time and the quality of the result in terms of overfitting MARTINEZ et collab. [2017, 2016, 2014]. See Chapter 4.

3 - The Novelty search or NS is a unique approach in optimization where an explicit objective function is replaced by a measure of the "novelty" of the solution. Although the NS has been widely used in evolutionary robotics, its usefulness for classical machine learning problems had so far remained unexplored. It is this lack that motivated **Enrique Naredo**'s thesis. Indeed, it was discussed to design a NS-based GP (GP-NS) for common machine learning problems. The contributions resulting from this work are as follows: It has been shown that the NS can solve real and synthetic problems of classification, clustering and symbolic regression. In addition, a study on the bloat and research dynamics of GP-NS was conducted and two new high-performance versions of NS were proposed and tested. See Chapter 5.

4 - Genetic programming is a powerful tool but, as we said earlier, it has some disadvantages in its version usually described in the literature. During **Emigdio Z. Flores**' thesis, we were interested in improving the convergence speed of these algorithms and controlling the bloat by adding a local optimization step around individuals (trees) LEONARDO et collab. [2017]; Z-FLORES et collab. [2014, 2015]. Indeed, through the parameterization of trees (individuals) and local optimization, the search for the GP will converge more quickly towards high quality solutions. First, we simply added a parameter, a weight coefficient before each function of the set of functions (atoms available to build the tree). In a second step, we determined to which individuals and generations it was relevant to apply this local optimization. The results showed that the best strategy is to apply local optimization to all individuals or to a random sample of the best (in the sense of fitness) individuals in each generation. See Chapters 6 and 7 for more details in symbolic regression and binary classification.

5 - Genetic programming (GP) has been shown to be a powerful tool for automatic modeling and program induction. It is often used to solve difficult symbolic regression tasks, with many examples in real-world domains. However, the robustness of GP-based approaches has not been substantially studied. This task motivated the master and the PhD thesis of **Uriel Lopez**. In particular, this work deals with the issue of outliers, data in the training set that represent severe errors in the measuring process. In general, a datum is considered an outlier when it sharply deviates from the true behavior of the system of interest. GP practitioners know that such data points usually bias the search and produce inaccurate models. Therefore, this work presents a hybrid methodology based on the RAndom SAmpling Consensus (RANSAC) algorithm and GP, which we call RANSAC-GP. RANSAC is an approach to deal with outliers in parameter estimation problems, widely used in com-

puter vision and related fields. On the other hand, this work presents the first application of RANSAC to symbolic regression with GP, with impressive results. The proposed algorithm is able to deal with extreme amounts of contamination in the training set, evolving highly accurate models even when the amount of outliers reaches 90%. See Chapter 8.

1.2.3 Combination of tools, theoretical development and applications

By combining the tools presented above with more "conventional" signal and machine learning processing, it was possible to produce, among other works, the following contributions

1. Estimation of the Hölder exponent by genetic programming. Building local descriptors in an image. TRUJILLO et collab. [2010a, 2012a, 2010b, 2007].

2. Evolutionary denoising, image processing, haze removal. HERNANDEZ-BELTRAN et collab. [2016]; LEGRAND et collab. [2006].

3. Automatic fitting of cochlear implants. BOURGEOIS-REPUBLIQUE et collab. [2006]; COLLET et collab. [2006, 2009]; LEGRAND [2008a,b]; LEGRAND et collab. [2007].

4. Classification of EEG signals and psychophysiological states LEGRAND et collab. [2014]; VEZARD et collab. [2011, 2012a,b, 2014, 2013]; VÉZARD et collab. [2015]; Z-FLORES et collab. [2016].

5. Radar, micro-Doppler, signal processing and machine learning. CORRETJA et collab. [2011]; MACHHOUR et collab. [2018].

1.3 Organization of the document

This manuscript is organized as follows:
The part I includes a chapter of reminders on artificial evolution and then six chapters of contributions in the field of genetic programming: Chapter 3 *"Prediction of Expected Performance for a Genetic Programming Classifier"*, Chapter 4 *"A comparison of fitness-case sampling methods for genetic programming"*, Chapter 5 *"Evolving Genetic Programming Classifiers with Novelty Search"*, Chapter 6 *"Evaluating the Effects of Local Search in Genetic Programming"*, Chapter 7 *"A Local Search Approach to Genetic Programming for Binary Classification"* and finally Chapter 8 *"RANSAC-GP: Dealing with Outliers in Symbolic Regression with Genetic Programming"*.
The part II gives reminders on the estimation of Hölderian regularity Chapter 9 then chapters of contributions around the DFA: The Chapter 10 *"Theoretical comparison of the DFA and variants for the estimation of the Hurst exponent"* and the Chapter 11 describing two patents on this topic.
The part III contains contributions combining the tools previously mentioned in order to develop new tools such as in the Chapters 12 *"The Estimation of Hölderian Regularity using Genetic Programming"*, 13 *"Optimization of the Hölder Image Descriptor using a Genetic Algorithm"* Or contributions on the resolution of real problems in the biomedical field, such as in the Chapters 14 *"Interactive evolution for cochlear implants fitting"*, 15 *"Feature extraction and classification of EEG signals. The use of a genetic algorithm for an application on alertness prediction"* or 16 *"Regularity and Matching Pursuit Feature Extraction for the Detection of Epileptic Seizures"*.

References

BERTHELOT, B., E. GRIVEL, P. LEGRAND, J.-M. ANDRÉ, P. MAZOYER et T. FERREIRA. 2019a, «Filtering-based Analysis Comparing the DFA with the CDFA for Wide Sense Stationary Processes», dans *EUSIPCO 2019 - 27th European Signal Processing Conference*, A Coruna, Spain. URL https://hal.archives-ouvertes.fr/hal-02147655. 10

BERTHELOT, B., E. GRIVEL, P. LEGRAND, J.-M. ANDRE, P. MAZOYER et T. FERREIRA. 2019b, «Interpréter les Fonctions de Fluctuation du DFA et du DMA comme le Résultat d'un Filtrage», dans *GRETSI 2019*, Lille, France. URL https://hal.archives-ouvertes.fr/hal-02145464. 10

BERTHELOT, B., E. GRIVEL, P. LEGRAND, M. DONIAS, J.-M. ANDRÉ, P. MAZOYER et T. FERREIRA. 2019c, «2D Fourier Transform Based Analysis Comparing the DFA with the DMA», dans *EUSIPCO 2019 - 27th European Signal Processing Conference*, A Coruna, Spain. URL https://hal.archives-ouvertes.fr/hal-02147663. 10

BLANKERTZ, B., R. TOMIOKA, S. LEMM, M. KAWANABE et K. R. MÜLLER. 2008, «Optimizing spatial filters for robust EEG single-trial analysis», dans *IEEE Signal Proc. Magazine*, p. 581–607. 4

BOURGEOIS-REPUBLIQUE, C., P. COLLET, B. FRACHET, E. HARBOUN-COHEN, P. LEGRAND, V. PEAN, B. PHILIPPON et M. OUAYOUN. 2006, «Preliminary results on automatic cochlear implant fitting using an interactive evolutionary algorithm», dans *9th International Conference on Cochlear implants and related Sciences*, Vienne, Austria. URL https://hal.archives-ouvertes.fr/hal-00297457. 12

COLLET, P., P. LEGRAND, C. BOURGEOIS-REPUBLIQUE, V. PEAN et B. FRACHET. 2006, «Aide au paramétrage d'implants cochléaires par algorithme évolutionnaire interactif.», dans *Optimisation en traitement du signal et de l'image, Traité IC2*, Hermès-Lavoisier, p. Chapitre 13. URL https://hal.archives-ouvertes.fr/hal-00294843, chapitre 13: Aide au paramétrage d'implants cochléaires par algorithme évolutionnaire interactif. 12

COLLET, P., P. LEGRAND, C. BOURGEOIS-REPUBLIQUE, V. PEAN et B. FRACHET. 2009, «Using interactive evolutionary algorithms to help fit cochlear implants», dans *Optimization in Signal and Image Processing*, édité par P. Siarry, Iste-Wiley, p. 329–384. URL https://hal.archives-ouvertes.fr/hal-00409853. 12

CORRETJA, V., P. LEGRAND, E. GRIVEL et J. LÉVY-VEHEL. 2011, «Relevance of the Hölderian regularity-based interpolation for range-Doppler ISAR image post-processing.», dans *RADAR 2011: 6th International Conference on Radar*, Chengdu, China. URL https://hal.inria.fr/hal-00643369. 12

DAUBECHIES, I. 1992, *Ten Lectures on Wavelets*, SIAM. 4

DE JONG, K. 1975, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Theses, University of Michigan . 10

DELTAMED. URL http://www.natus.com/index.cfm?page=company_1&crid=129&contentid=223. 2

HERNANDEZ-BELTRAN, J. E., V. H. DÍAZ-RAMÍREZ, L. TRUJILLO et P. LEGRAND. 2016, «Restoration of degraded images using genetic programming», dans *Optics and photonics for information processing X*, vol. 9970, San Diego, United States. URL https://hal.inria.fr/hal-01389064. 12

HOLLAND, J. H. 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI. Second edition, 1992. 10

JAFFARD, S. 2004, «Wavelet techniques in multifractal analysis», cahier de recherche, PARIS UNIV (FRANCE). 9

LANGDON, W. B. et R. POLI. 2002, *Foundations of genetic programming*, Springer. 10

LEGRAND, P. 2004, *Débruitage et interpolation par analyse de la régularité Hölderienne. Application à la modélisation du frottement pneumatique-chaussée.*, Theses, Ecole Centrale de Nantes (ECN) ; Université de Nantes. URL https://tel.archives-ouvertes.fr/tel-00643450. 9

LEGRAND, P. 2008a, «Evolution interactive pour le réglage d'implants cochléaires.», dans *Congrès suisse des audioprothésistes*, Berne, Switzerland. URL https://hal.archives-ouvertes.fr/hal-00296788. 12

LEGRAND, P. 2008b, «Le réglage par algorithmes évolutionnaires des implants cochléaire», dans *Congrès international JAN09: 9e journée d'Analyse Numérique et d'Optimisation*, Mohammedia, Morocco. URL https://hal.archives-ouvertes.fr/hal-00385739. 12

LEGRAND, P. 2009, «Local Regularity and Multifractal methods for image and signal analysis», dans *Scaling, Fractals and Wavelets*, Abry P., Goncalves P., Lévy-Vehel J., p. 512 pages, chapitre 11. URL https://hal.archives-ouvertes.fr/hal-00294853. 9

LEGRAND, P., C. BOURGEOIS-REPUBLIQUE, V. PEAN, E. HARBOUN-COHEN, J. LÉVY VÉHEL, B. FRACHET, E. LUTTON et P. COLLET. 2007, «Interactive evolution for cochlear implants fitting», *Genetic Programming and Evolvable Machines*, vol. 8, nº 4, p. 319–354. URL https://hal.archives-ouvertes.fr/hal-00294838. 12

LEGRAND, P. et J. LÉVY VÉHEL. 2003a, «Local regularity-based image denoising», dans *ICIP03, IEEE International Conference on Image Processing*, vol. III, Barcelona, Spain, p. 377–380. URL https://hal.inria.fr/inria-00576475. 9

LEGRAND, P. et J. LÉVY VÉHEL. 2003b, «Local regularity-based interpolation», dans *WAVELET X, Part of SPIE's Symposium on Optical Science and Technology*, vol. 5207, SPIE, San Diego, United States. URL https://hal.inria.fr/inria-00576479. 9

LEGRAND, P., E. LUTTON et G. OLAGUE. 2006, «Evolutionary denoising based on an estimation of Hölder exponents with oscillations.», dans *EvoIASP 2006 - 8th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, LNCS*, vol. Applications of evolutionary computing, Springer, Budapest, Hungary, p. 520–524, doi:10.1007/11732242_49. URL https://hal.archives-ouvertes.fr/hal-00294873. 12

LEGRAND, P., L. VEZARD, M. CHAVENT, F. FAÏTA et L. TRUJILLO. 2014, «Feature extraction and classification of EEG signals. The use of a genetic algorithm for an application on alertness prediction.», dans *Guide to Brain-Computer Music Interfacing*, édité par E. Miranda, J. Castet et B. Knapp, Springer. URL https://hal.inria.fr/hal-01060317. 12

LEONARDO, T., E. Z-FLORES, P. S. JUAREZ SMITH, P. LEGRAND, S. SILVA, M. CASTELLI, L. VANNESCHI, O. SCHÜTZE et L. MUNOZ. 2017, «Local Search is Underused in Genetic Programming», dans *Genetic Programming Theory and Practice XIV*, édité par A. Arbor, Springer. URL https://hal.inria.fr/hal-01388426. 11

LÉVY VÉHEL, J. et P. LEGRAND. 2003, «Bayesian multifractal signal denoising», dans *ICASSP03, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong-Kong, China. URL https://hal.inria.fr/inria-00576482. 9

LÉVY VÉHEL, J. et P. LEGRAND. 2004, «Signal and Image processing with FracLab», dans *FRACTAL04, Complexity and Fractals in Nature, 8th International Multidisciplinary Conference*, vol. Thinking in Patterns : fractals and related phenomena in nature, Vancouver, Canada, p. 321–322. URL https://hal.inria.fr/inria-00576466. 9

LÉVY VÉHEL, J. et P. LEGRAND. 2006, «Hölderian regularity-based image interpolation», dans *ICASSP 06, International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, IEEE, Toulouse, France, p. 852–855, doi:10.1109/ICASSP.2006.1660788. URL https://hal.archives-ouvertes.fr/hal-00297218. 9

LOTTE, F. et C. GUAN. 2011, «Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms.», *IEEE Transactions on biomedical Engineering*, vol. 58 (2), p. 355–362. 4

MACHHOUR, S., E. GRIVEL, P. LEGRAND, V. CORRETJA et C. MAGNANT. 2018, «A Comparative Study of Orthogonal Moments for Micro-Doppler Classification», dans *EUSIPCO 2018 - 26th European Signal Processing Conference*, Rome, Italy. URL https://hal.archives-ouvertes.fr/hal-01795520. 9, 12

MALLAT, S. 2008, *A Wavelet Tour of Signal Processing.*, 3e éd., Academic Press. 4

MARTINEZ, Y., E. NAREDO, L. TRUJILLO, P. LEGRAND et U. LOPEZ. 2017, «A comparison of fitness-case sampling methods for genetic programming», *Journal of Experimental and Theoretical Artificial Intelligence*. URL https://hal.inria.fr/hal-01389047. 11

MARTINEZ, Y., L. TRUJILLO, P. LEGRAND et E. GALVAN-LOPEZ. 2016, «Prediction of Expected Performance for a Genetic Programming Classifier», *Genetic Programming and Evolvable Machines*, vol. 17, no 4, doi:10.1007/s10710-016-9265-9, p. 409–449. URL https://hal.inria.fr/hal-01252141. 10, 11

MARTINEZ, Y., L. TRUJILLO, E. NAREDO et P. LEGRAND. 2014, «A comparison of fitness-case sampling methods for symbolic regression with genetic programming», dans *EVOLVE 2014*, BEIJING, China, p. 201–212, doi:10.1007/978-3-319-07494-8_14. URL https://hal.inria.fr/hal-01060313. 11

NAITOH, P., L. C. JOHNSON et A. LUBIN. 1971, «Modification of surface negative slow potential (CNV) in the human brain after total sleep loss.», *Electroencephalography and Clinical Neurophysiology*, vol. 30, p. 17–22. 3

RAMOSER, H., J. MÜLLER-GERKING et G. PFURTSCHELLER. 2010, «Optimal spatial filtering of single trial EEG during imagined hand movement», *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no 4, p. 441–446. 4

TECCE, J. J. 1979, «A CNV rebound effect.», *Electroencephalography and Clinical Neurophysiology*, vol. 46, p. 546–551. 3

TIMSIT-BERTHIER, M., A. GERONO et H. MANTANUS. 1981, «Inversion de polarité de la variation contingente négative au cours d'état d'endormissement.», *EEG Neurophysiol*, vol. 11, p. 82–88. 3

TRICOT, C. 1995, *Curves and Fractal Dimension*, Springer-Verlag. 9

TRUJILLO, L., P. LEGRAND et J. LÉVY VÉHEL. 2010a, «The estimation of Hölderian regularity using genetic programming», dans *Genetic and Evolutionary Computation Conference (GECCO 2010). Best Paper Award in "Genetic Programming"*, vol. ISBN 978-1-4503-0072-8, Portland Oregon, United States, p. 861–868. URL https://hal.inria.fr/inria-00538943, best Paper Award in "Genetic Programming". 12

TRUJILLO, L., P. LEGRAND, G. OLAGUE et J. LÉVY-VEHEL. 2012a, «Evolving Estimators of the Pointwise Holder Exponent with Genetic Programming», *Information Sciences*, vol. 209, doi:10.1016/j.ins.2012.04.043, p. 61–79. URL https://hal.inria.fr/hal-00643387, submitted. 12

TRUJILLO, L., P. LEGRAND, G. OLAGUE et C. PÉREZ. 2010b, «Optimization of the Hölder Image Descriptor using a Genetic Algorithm», dans *GECCO 2010. Best paper award in "Real world applications"*., vol. ISBN 978-1-4503-0072-8, Portland Oregon, United States, p. 1147–1154. URL https://hal.inria.fr/inria-00534457, best paper award in "Real world applications". 12

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2011, «Predicting Problem Difficulty for Genetic Programming Applied to Data Classification», dans *Gecco 2011*, édité par U. Natalio KrasnogorUniversity of Nottingham, ACM New York, NY, USA ©2011, Dublin, Ireland, p. 1355–1362, doi:10.1145/2001576.2001759. URL https://hal.inria.fr/hal-00643358. 10, 11

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2012b, «A Comparative Study of an Evolvability Indicator and a Predictor of Expected Performance for Genetic Programming», dans *GECCO*, Philadelphie, United States. URL https://hal.inria.fr/hal-00757266. 10

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2012c, «A comparison of predictive measures of problem difficulty for classification with Genetic Programming», dans *ERA 2012*, Tijuana, Mexico. URL https://hal.inria.fr/hal-00757363. 10

TRUJILLO, L., G. OLAGUE, P. LEGRAND et E. LUTTON. 2007, «A new regularity based descriptor computed from local image oscillations.», *Optics Express*, vol. 15, nº 10, p. 6140–6145. URL https://hal.archives-ouvertes.fr/hal-00294862. 12

VEZARD, L., M. CHAVENT, P. LEGRAND, F. FAITA-AINSEBA et J. CLAUZEL. 2011, «Caractérisation d'états psychophysiologiques par classification de signaux EEG. Intégration de ces résultats dans le projet PSI.», dans *Journée AFIM : électroencéphalographie et composition musicale*, Talence, France. URL https://hal.inria.fr/hal-00649520. 12

VEZARD, L., P. LEGRAND, M. CHAVENT, F. FAITA-AINSEBA et J. CLAUZEL. 2012a, «Classification de données EEG par algorithme évolutionnaire pour l'étude d'états de vigilance», *Revue des Nouvelles Technologies de l'Information*. URL https://hal.inria.fr/hal-00643438. 12

VEZARD, L., P. LEGRAND, M. CHAVENT, F. FAITA-AINSEBA et J. CLAUZEL. 2012b, «Classification of EEG signals by an evolutionary algorithm», dans *COMPSTAT 2012*, Limassol, Cyprus. URL https://hal.inria.fr/hal-00757270. 12

VEZARD, L., P. LEGRAND, M. CHAVENT, F. FAITA-AINSEBA, J. CLAUZEL et L. TRUJILLO. 2014, «Classification of EEG signals by evolutionary algorithm», dans *Advances in Knowledge Discovery and Management Volume 4, Studies in Computational Intelligence*, vol. 527, édité par F. Guillet, B. Pinaud, G. Venturini et D. Zighed, Springer, p. 133–153, doi:10.1007/978-3-319-02999-3_8. URL https://hal.inria.fr/hal-00939850. 12

VEZARD, L., P. LEGRAND, M. CHAVENT, F. FAITA-AINSEBA et L. TRUJILLO. 2013, «Detecting mental states of alertness with genetic algorithm variable selection», dans *IEEE Congress on Evolutionary Computation (CEC) 2013*, CANCUN, Mexico, p. 1247 – 1254. URL https://hal.inria.fr/hal-00939851. 12

VÉZARD, L., P. LEGRAND, M. CHAVENT, F. FAÏTA-AÏNSEBA et L. TRUJILLO. 2015, «EEG classification for the detection of mental states», *Applied Soft Computing*, vol. 32, doi:10.1016/j.asoc.2015.03.028, p. 113–131. URL https://hal.inria.fr/hal-01207506. 12

Z-FLORES, E., L. TRUJILLO, O. SCHUETZE et P. LEGRAND. 2014, «Effects of local search in genetic programming», dans *EVOLVE 2014*, BEIJING, China. URL https://hal.inria.fr/hal-01060315. 11

Z-FLORES, E., L. TRUJILLO, O. SCHÜTZE et P. LEGRAND. 2015, «A Local Search Approach to Genetic Programming for Binary Classification», dans *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15*, GECCO '15 Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, doi:10.1145/2739480.2754797. URL https://hal.inria.fr/hal-01207504. 11

Z-FLORES, E., L. TRUJILLO, A. SOTELO, P. LEGRAND et L. CORIA. 2016, «Regularity and Matching Pursuit Feature Extraction for the Detection of Epileptic Seizures», *Journal of Neuroscience Methods*, vol. 266, doi:10.1016/j.jneumeth.2016.03.024, p. 107–125. URL https://hal.inria.fr/hal-01389051. 12

ZOU, H., T. HASTIE et R. TIBSHIRANI. 2006, «Sparse principal component analysis», *Journal of Computational and Graphical Statistics*, vol. 15, nº 2, p. 265–286. 5

Part I

Artificial Evolution

Chapter 2

Introduction

Contents

2.1 Evolutionary Algorithms

Genetic algorithms, genetic programming, evolution strategies are stochastic optimization techniques inspired by Darwin's theory of evolution DARWIN [1859]. These techniques are often referred to as evolutionary algorithms and involve mechanisms such as reproduction, mutation, selection and survival of the individuals best adapted to the environment.

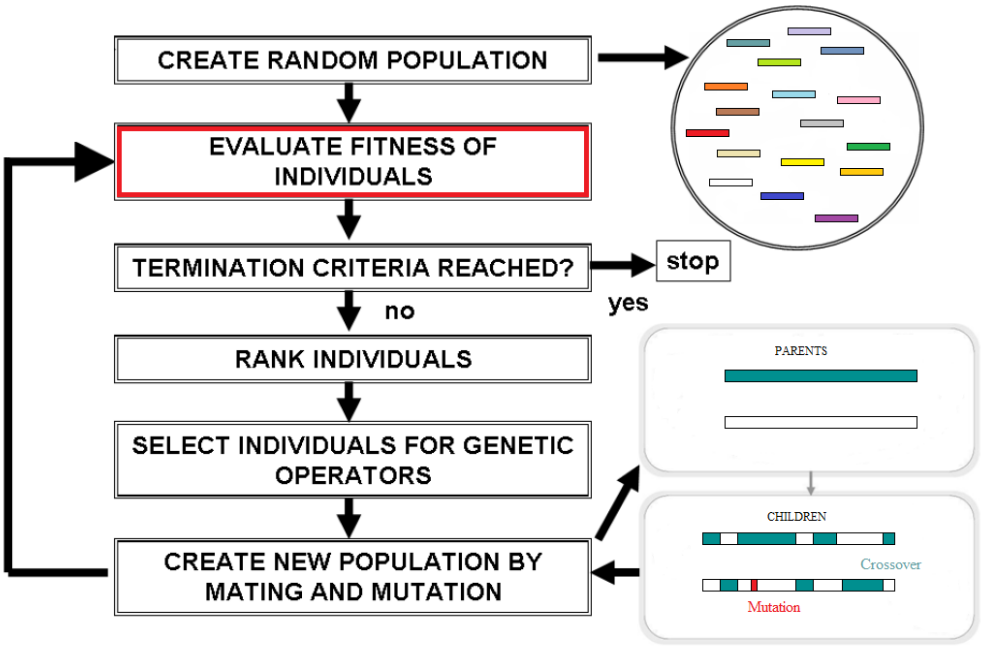


Figure 2.1 – Evolutionnary loop

The "classical" evolutionary loop is given in Figure 2.1. The potential solutions to an optimization problem are represented by individuals in a population. Initially this population is randomly generated.

The evolutionary engine, common to all types of evolutionary algorithms, includes the following elements: The **evaluation**, which will make it possible to estimate the quality of an individual, the **selection**, which will make it possible to select the best individuals, the **reproduction**, which consists in the application of genetic operators such as crossing and mutation (with the probabilities P_c and P_m), and the **replacement** which will make it possible to constitute the next generation.

The purpose of these algorithms is to optimize a function (called fitness) in a space where potential solutions can be found. The solutions (called individuals) correspond to points in the search space. A fixed number of solutions will be randomly generated, initializing the evolutionary algorithm. The solutions are represented by their genome (binary numbers or real numbers, with a fixed or variable size). Individuals are evaluated using the fitness function to assess the quality of the solution they offer. They are then selected based on this evaluation (using, for example, a series of tournaments). The selected individuals are called parents. These parents are used to generate new individuals through two basic genetic operations, crossover (crossing the genomes of two or more individuals) and mutation (random modification of one (or more) component(s) of the individual's genome). These newly generated individuals are called children because they share similarities (genetic) with the parents who were used to generate them. Finally, individuals are selected and replace the initial population (better individuals between parents and children or selection of children only or conservation of children and parents,...). The algorithm is iterated until a stopping criterion is reached; for example, when all individuals are identical (convergence

of the algorithm) or after a predetermined number of iterations.

To illustrate this process, we can refer to the Figure 2.2 which gives an example of maximization with an evolutionary algorithm. The objective is to find the $argmax$ of the f function on the interval $[0, 15]$. We initialize the population by randomly selecting values on the abscissa axis and then we make the population evolve in the hope of converging towards the solution.

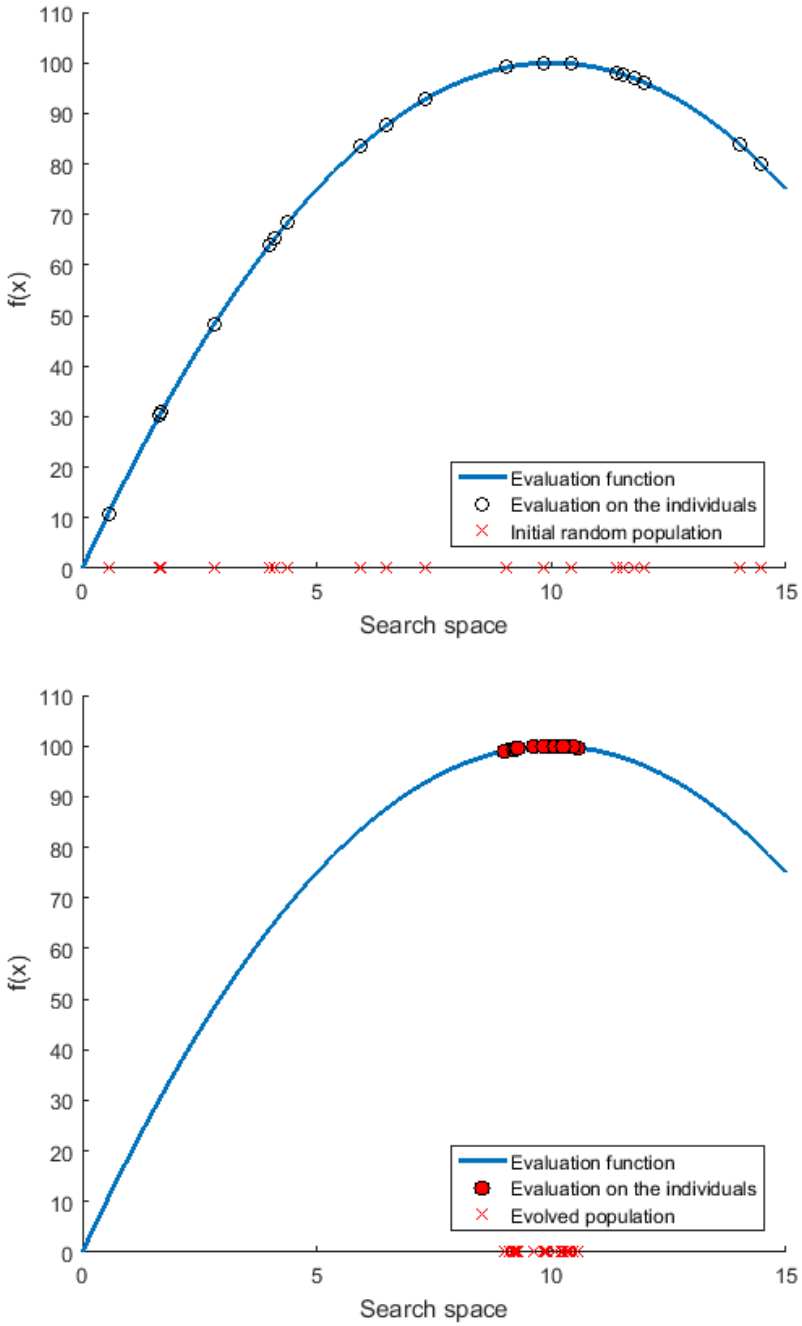


Figure 2.2 – Simple example of stochastic optimization with an evolutionary algorithm

Genetic operators depend on the choice of representation. The most traditional approaches will be given below. For a complete overview of evolutionary techniques, the reader may refer to LUTTON [2005].

2.1.1 Genetic Algorithms (GA)

Genetic algorithms are certainly optimization algorithms based on Darwinian evolutionary theory the most famous, thanks in particular to the work of John Holland HOLLAND [1975], Kenneth De Jong DE JONG [1975] and David Goldberg GOLDBERG [1989]. As mentioned above, the general idea is that a population of potential solutions will improve its characteristics over time through genetic mutations and crossovers in order to adapt as well as possible to its environment. The particularity of these "canonical" algorithms is that each individual was initially represented by a fixed-length binary string. So, the search is done in $\mathbb{N}^n$. This representation has often been extended to a discrete representation. Thus, for this type of algorithm, the most commonly used crossing will be the locus type crossing as shown in Figure 2.3. After individuals have been randomly selected to act as parents, one or more cut-off points are randomly selected from the parents' genome. Then the genes are swapped to create a child (or children).

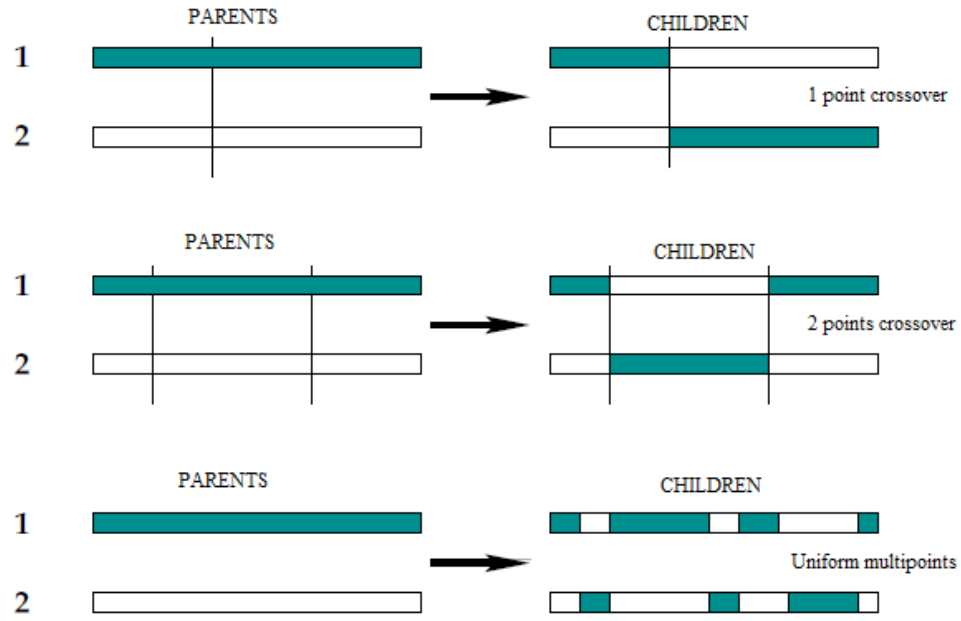


Figure 2.3 – Locus crossover (image courtesy of E. Lutton)

For this type of "canonical" algorithms, the mutation will consist in mutating a 1 into 0 and a 0 into 1 with a certain probability P_m (see Figure 2.4).

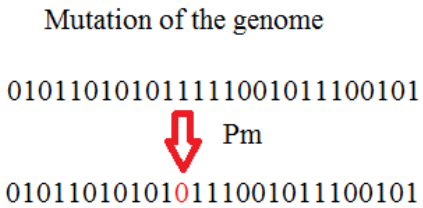


Figure 2.4 – Mutation for a canonical GA

2.1.2 Evolution Strategies (ES)

The term Evolution strategy will be used when the representation of the genes of individuals is real SCHWEFEL [1981]. Thus, in this approach, the search is performed in $\mathbb{R}^n$. Although it

is possible to apply "locus" crossover as described for genetic algorithms, the crossovers usually used in evolution strategies are barycentric crossover such as $\forall i \in \{1, ..., n\}, x_i^{children} = \alpha_i x_i^{father} + (1 - \alpha_i) x_i^{mother}$ with α_i random value in $[-\epsilon, 1+\epsilon]$. α_i can be the same for any i but not necessarily. The mutation commonly used in canonical algorithms of evolution strategies is a Gaussian mutation. It consists in taking a Gaussian random variable of standard deviation σ which will be added to a gene with a probability of P_m. $\forall i \in \{1, ..., n\}, x_i^{children} = x_i^{children} + N(0, \sigma)$. Adapting the two parameters P_m and σ requires a good knowledge of the optimization problem being addressed. Other types of mutations are sometimes used for evolutionary strategies, such as the uniform mutation in an interval $[x_i^{min}, x_i^{max}]$ or the self-adaptive log-normal mutation whose standard deviation σ is directly processed by the evolutionary algorithm by integrating it into the individual genome.

2.1.3 Genetic Programming (GP)

Of the current algorithms, genetic programming (GP) is one of the most advanced forms of evolutionary research. In the classic GP form, each solution is represented as a tree that can represent a function. Trees are built using elements of two sets: a set of functions and a set of terminals. These two sets define the GP's search space (see Figure 2.5).

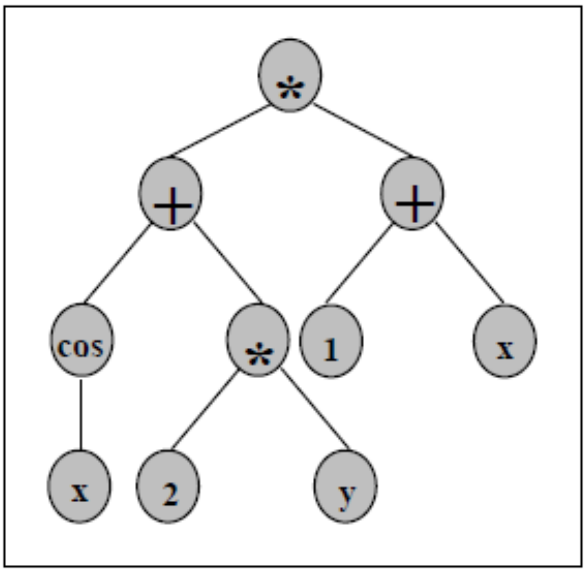


Figure 2.5 – GP tree, function $(cos(x) + 2y)(1 + x)$ (image courtesy of E. Lutton)

This section provides details on genetic programming and indicates the nuances inherent in its use for symbolic regression or classification.

GP can be understood as a generalization of the basic genetic algorithm, its main features can be summarized as follows POLI et collab. [2008]. First, GP was originally proposed as an EA that evolves simple programs, functions, operators, or in general symbolic expressions that perform some form of computation. GP is basically used to evolve solutions to different types of design problems, with examples as varied as quantum algorithms SPECTOR [2006], computer vision operators OLAGUE et TRUJILLO [2011] and satellite antennas HORNBY et collab. [2011]. Second, solutions are expressed as variable length structures, such as linked lists, parse trees or graphs POLI et collab. [2008]. These structures encode the syntax of each individual program. Therefore, in a canonical GP algorithm, search operators, such as crossover and mutation, perform syntactic variations on the evolving population (see Figures 2.6 et 2.7).

Third, by considering each individual in a GP run as a program, the evolutionary process is basically attempting to write the best program syntax that solves a given problem.

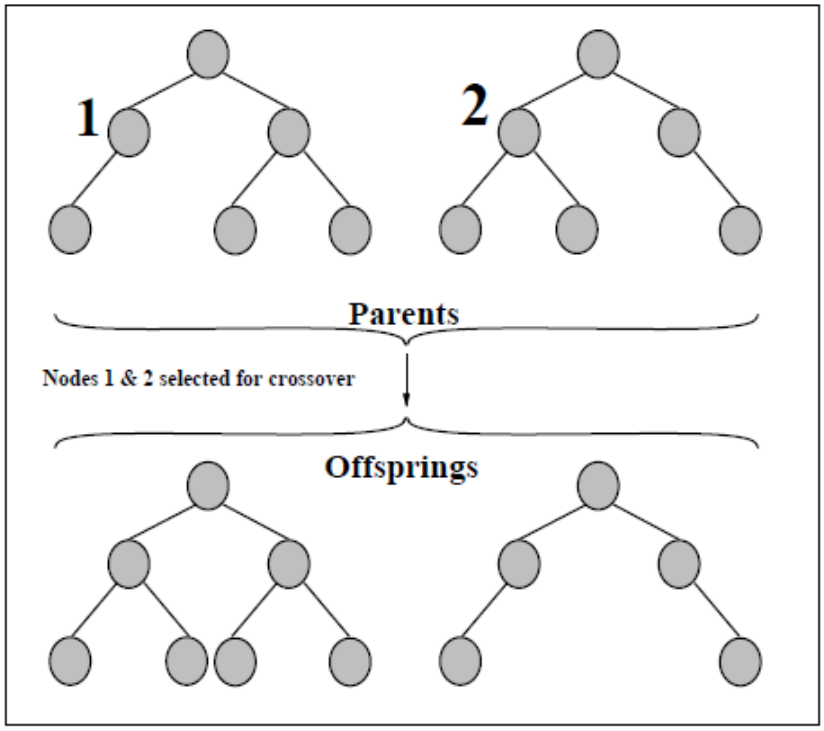


Figure 2.6 – GP crossover (image courtesy of E. Lutton)

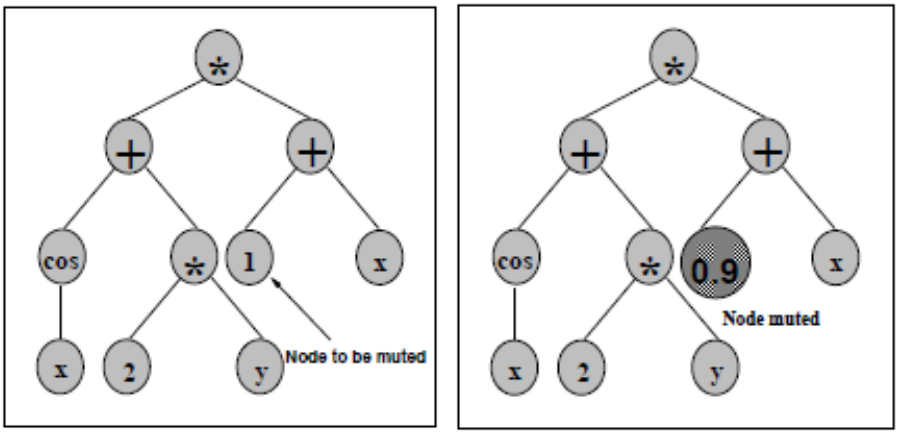


Figure 2.7 – GP mutation (image courtesy of E. Lutton)

Therefore, a finite set of basic symbols needs to be defined, which is called the primitive set $\mathbb{P}$. Within the primitive set there is a subset of basic operations or functions of different arity, called the function set F, and a subset of input variables, constants or zero arity functions called the terminal set T, such that $\mathbb{P} = F \cup T$.

2.1.3.1 GP for symbolic regression and for classification

Given the variety of possible GP configurations and applications, one can focuses on the problem of **symbolic regression** using a tree representation. In symbolic regression, the goal is to search for the symbolic expression $K^O : \mathbb{R}^p \to \mathbb{R}$ that best fits a particular training set $\mathbb{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n input/output pairs with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. In such problems, for instance, the function set can be defined as $F = \{+, -, /, *\}$ and the terminal set can be composed by each of the features in the input data, such that $T = \{x_i\}$ with $i = 1, \ldots, p$, but other terminal elements can be included such as integer or real-valued

constants.

The general symbolic regression problem can then be defined as

$$(K^O, \boldsymbol{\theta}^{\boldsymbol{O}}) \leftarrow \underset{K \in \mathbb{G}; \boldsymbol{\theta} \in \mathbb{R}^m}{arg\,min}\ f(K(\mathbf{x}_i, \boldsymbol{\theta}), y_i)\ with\ i = 1, \ldots, p\ , \tag{2.1}$$

where $\mathbb{G}$ is the solution or syntactic space defined by $\mathbb{P}$, f is the fitness function based on the difference between a program's output $K(\mathbf{x}_i, \boldsymbol{\theta})$ and the expected output y_i, such as the mean square error, and $\boldsymbol{\theta}$ is a particular parametrization of the symbolic expression K, assuming m real-valued parameters. This dual problem, of simultaneously optimizing syntax (structure) as well as its parametrization is also discussed in EMMERICH et collab. [2001]; LOHMANN [1991].

2.1.3.2 GP for classification

One can focuses on the problem of **classification** using a tree representation. In supervised learning, classification paradigm has several approaches. For this problematic, we begin defining it as the task of classifying the elements of a given set into groups on the basis of classification rule. Distinct label can be assigned to each of these groups (Class 1, Class 2 etc...). Later, for convenience, this classification problem can be transformed into a regression one by inserting a continuous function in order to get gradient information, and to be able to use a numerical optimizer.

2.2 Presentation of contributions

2.2.1 Difficulty prediction in evolutionary programming

Determining the difficulty of a problem is an important issue in the field of artificial evolution, and has been for years. From an algorithmic point of view, the difficulty of a problem can be associated with the execution time or memory required to find an optimal solution. As part of Yuliana Martinez's thesis, we were interested in building models that can predict the performance of a GP-classifier without having to run the program or sample potential solutions in the research space. The principle is as follows: for a given classification problem, we apply a pre-processing step to simplify the feature extraction process. Then we perform the step of extracting the characteristics of the problem. Finally, we use a PEP (prediction of expected performance) model, which takes the characteristics of the problem as input and produces the predicted classification error on the test set as output. To build the PEP model, we used a supervised learning method with a GP. Then, to refine this work, we developed an approach using several PEP models, each becoming then a specialized predictors of expected performance (SPEP) specialized for a particular group of problems. It appears that the PEP and SPEP models were able to accurately predict the performance of a GP-classifier and that the SPEP approach gave the best results MARTINEZ et collab. [2016]; TRUJILLO et collab. [2011]. These results will be presented in Chapter 3.

2.2.2 New fitness calculation methods

Genetic programming has shown impressive results in various fields of application. However, there are still practical limitations to overcome, in particular, its computational cost, overfitting and excessive tree size increase (sometimes without correlation with the improvement of the quality of the individual, the so-called bloat). A particular feature of a GP is that it looks for symbolic expressions that best describe the relationships in a learning set of input and output pairs. These pairs are described in the literature as fitness-cases. Recently, researchers have proposed new approaches to avoid some of the problems mentioned

above and have obtained interesting results by proposing methods for sampling fitness-cases to be considered during the evaluation of individuals. After an intensive comparison of various fitness case sampling methods, some of which we have proposed, it appeared through Yuliana Martinez's thesis that the choice of fitness calculation method has an influence on the bloat, the calculation time and the quality of the result in terms of overfitting MARTINEZ et collab. [2017, 2014]. These results will be presented in Chapter 4.

2.2.3 Genetic programming based on Novelty Search (NS)

The Novelty search or NS is a unique approach in optimization where an explicit objective function is replaced by a measure of the "novelty" of the solution. Although the NS has been widely used in evolutionary robotics, its usefulness for classical machine learning problems had so far remained unexplored. It is this lack that motivated Enrique Naredo's thesis. Indeed, he was asked to design a NS-based GP (GP-NS) for common machine learning problems. The contributions resulting from this collaboration are as follows: It has been shown that the NS can solve real and synthetic problems of classification, clustering and symbolic regression. In addition, a study on the bloat and research dynamics of GP-NS was conducted and two new high-performance versions of NS were proposed and tested NAREDO et collab. [2016]. These results will be presented in Chapter 5.

2.2.4 Local optimization in GP

Genetic programming is a powerful tool but, as we said earlier, it has some disadvantages in its version usually described in the literature. During Emigdio Z. Flores 's thesis, we were interested in improving the convergence speed of these algorithms and controlling the bloat by adding a local optimization step around individuals. Indeed, through the parameterization of trees (individuals) and local optimization, the search for the GP will converge more quickly towards high quality solutions. First, we simply added a parameter, a weight coefficient before each function of the set of functions (atoms available to build the tree). In a second step, we determined to which individuals and generations it was relevant to apply this local optimization. The results showed that the best strategy is to apply local optimization to all individuals or to a random sample of the best (in the sense of fitness) individuals in each generation TRUJILLO et collab. [2017]; Z-FLORES et collab. [2014, 2015]. These results will be presented in Chapter 6 and extended to the framework of the (binary) classification Chapter 7.

2.2.5 GP and outliers

Genetic programming (GP) has been shown to be a powerful tool for automatic modeling and program induction. It is often used to solve difficult symbolic regression tasks, with many examples in real-world domains. However, the robustness of GP-based approaches has not been substantially studied. In particular, the work carried out during the Master Thesis of Uriel Lopez Islas deals with the issue of outliers, data in the training set that represent severe errors in the measuring process. In general, a datum is considered an outlier when it sharply deviates from the true behavior of the system of interest. GP practitioners know that such data points usually bias the search and produce inaccurate models. Therefore, this work presents a hybrid methodology based on the RAndom SAmpling Consensus (RANSAC) algorithm and GP, which we called RANSAC-GP. RANSAC is an approach to deal with outliers in parameter estimation problems, widely used in computer vision and related fields. On the other hand, this work presents the first application of RANSAC to symbolic regression with GP, with impressive results. The proposed algorithm is able to deal with extreme amounts of contamination in the training set, evolving highly accurate

models even when the amount of outliers reaches 90%. These results will be presented in Chapter 8.

2.3 Organization of the part I

The rest of this part I contains six chapters of contributions in the field of genetic programming: Chapter 3 *"Prediction of Expected Performance for a Genetic Programming Classifier"*, Chapter 4 *"A comparison of fitness-case sampling methods for genetic programming"*, Chapter 5 *"Evolving Genetic Programming Classifiers with Novelty Search"*, Chapter 6 *"Evaluating the Effects of Local Search in Genetic Programming"*, Chapter 7 *"A Local Search Approach to Genetic Programming for Binary Classification"* and finally Chapter 8 *"RANSAC-GP: Dealing with Outliers in Symbolic Regression with Genetic Programming"*.

References

DARWIN, C. 1859, *On the Origin of Species by Means of Natural Selection*, Murray, London. Or the Preservation of Favored Races in the Struggle for Life. 22

DE JONG, K. 1975, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Theses, University of Michigan . 24

EMMERICH, M., M. GRÖTZNER et M. SCHÜTZ. 2001, «Design of graph-based evolutionary algorithms: A case study for chemical process networks», *Evol. Comput.*, vol. 9, nº 3, p. 329–354. 27

GOLDBERG, D. E. 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1re éd., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, ISBN 0201157675. 24

HOLLAND, J. H. 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI. Second edition, 1992. 24

HORNBY, G. S., J. D. LOHN et D. S. LINDEN. 2011, «Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission», *Evol. Comput.*, vol. 19, nº 1, p. 1–23. 25

LOHMANN, R. 1991, «Proceedings of parallel problem solving from nature (ppsn i)», dans *Proceedings from the 16th European Conference on Genetic Programming, EuroGP 2013, LNCS*, vol. 496, Springer-Verlag, p. 198–208. 27

LUTTON, E. 2005, «Darwinisme artificiel : une vue d'ensemble», *Technique et Science Informatique, TSI, Traitement du Signal, numéro spécial Méthodologie de la gestion intelligente des senseurs*, vol. 22 (4), p. 339–354. 23

MARTINEZ, Y., E. NAREDO, L. TRUJILLO, P. LEGRAND et U. LOPEZ. 2017, «A comparison of fitness-case sampling methods for genetic programming», *Journal of Experimental and Theoretical Artificial Intelligence*. URL https://hal.inria.fr/hal-01389047. 28

MARTINEZ, Y., L. TRUJILLO, P. LEGRAND et E. GALVAN-LOPEZ. 2016, «Prediction of Expected Performance for a Genetic Programming Classifier», *Genetic Programming and Evolvable Machines*, vol. 17, nº 4, doi:10.1007/s10710-016-9265-9, p. 409–449. URL https://hal.inria.fr/hal-01252141. 27

MARTINEZ, Y., L. TRUJILLO, E. NAREDO et P. LEGRAND. 2014, «A comparison of fitness-case sampling methods for symbolic regression with genetic programming», dans *EVOLVE 2014*, BEIJING, China, p. 201–212, doi:10.1007/978-3-319-07494-8_14. URL https://hal.inria.fr/hal-01060313. 28

NAREDO, E., L. TRUJILLO, P. LEGRAND, S. SILVA et L. MUNOZ. 2016, «Evolving Genetic Programming Classifiers with Novelty Search», *Information Sciences*, vol. 369, doi:10.1016/j.ins.2016.06.044, p. 347–367. URL https://hal.inria.fr/hal-01389049. 28

OLAGUE, G. et L. TRUJILLO. 2011, «Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming», *Image Vision Comput.*, vol. 29, nº 7, p. 484–498. 25

POLI, R., W. B. LANGDON et N. F. MCPHEE. 2008, *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd. 25

SCHWEFEL, H. 1981, *Numerical optimization of computer models*, Wiley, Chichester, WS, UK. 24

SPECTOR, L. 2006, *Automatic Quantum Computer Programming: A Genetic Programming Approach (Genetic Programming)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA. 25

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2011, «Predicting Problem Difficulty for Genetic Programming Applied to Data Classification», dans *Gecco 2011*, édité par U. Natalio KrasnogorUniversity of Nottingham, ACM New York, NY, USA ©2011, Dublin, Ireland, p. 1355–1362, doi:10.1145/2001576.2001759. URL https://hal.inria.fr/hal-00643358. 27

TRUJILLO, L., E. Z-FLORES, P. S. JUAREZ SMITH, P. LEGRAND, S. SILVA, M. CASTELLI, L. VANNESCHI, O. SCHÜTZE et L. MUNOZ. 2017, «Local Search is Underused in Genetic Programming», dans *Genetic Programming Theory and Practice XIV*, édité par A. Arbor, Springer. URL https://hal.inria.fr/hal-01388426. 28

Z-FLORES, E., L. TRUJILLO, O. SCHUETZE et P. LEGRAND. 2014, «Effects of local search in genetic programming», dans *EVOLVE 2014*, BEIJING, China. URL https://hal.inria.fr/hal-01060315. 28

Z-FLORES, E., L. TRUJILLO, O. SCHÜTZE et P. LEGRAND. 2015, «A Local Search Approach to Genetic Programming for Binary Classification», dans *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15*, GECCO '15 Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, doi:10.1145/2739480.2754797. URL https://hal.inria.fr/hal-01207504. 28

Chapter 3

Prediction of Expected Performance for a Genetic Programming Classifier

This chapter is related to the PhD thesis of Yuliana Martínez (ITT Tijuana) and has been published in Genetic Programming and Evolvable Machine, Springer Verlag, 2016, 17 (4), pp.409-449. Work carried out with Yuliana Martínez, Leonardo Trujillo and Edgar Galván-López.

Contents

Abstract

The study of problem difficulty is an open issue in Genetic Programming (GP). The goal of this work is to generate models that predict the expected performance of a GP-based classifier when it is applied to an unseen task. Classification problems are described using domain-specific features, some of which are proposed in this work, and these features are given as input to the predictive models. These models are referred to as predictors of expected performance (PEPs). We extend this approach by using an ensemble of specialized predictors (SPEP), dividing classification problems into specified groups and choosing the corresponding SPEP. The proposed predictors are trained using 2D synthetic classification problems with balanced datasets. The models are then used to predict the performance of the GP classifier on unseen real-world datasets that are multidimensional and imbalanced. Moreover, as we know, this work is the first to provide a performance prediction of the GP classifier on test data, while previous works focused on predicting training performance. Accurate predictive models are generated by posing a symbolic regression task and solving it with GP. These results are achieved by using highly descriptive features and including a dimensionality reduction stage that simplifies the learning and testing process. The proposed approach could be extended to other classification algorithms and used as the basis of an expert system for algorithm selection.

3.1 Introduction

Within the field of Evolutionary Computation (EC) EIBEN et SMITH [2003] it is not yet clear if a particular algorithm will perform well on a specific problem instance. The "No Free Lunch" (NFL) theorem WOLPERT et MACREADY [1997] has provided valuable theoretical and conceptual insights, broadly stating that all search algorithms on average are equivalent when they are evaluated over all possible problems. On the other hand, the NFL theorem does not apply to many common domains of genetic programming (GP) POLI et collab. [2009], a promising theoretical insight that drives research to develop the best possible GP-based search. Nevertheless, it is by now evident that most GP-based systems tend to perform well on some problem instances while failing on others, with little understanding as to why or when either of those two scenarios will arise GRAFF et collab. [2013a]; GRAFF et POLI [2008]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a,b].

The above issue can be described as the study of problem difficulty, which has been studied in different ways in EC and GP literature. Some methods focus on analyzing the properties of a problem's fitness landscape KINNEAR [1994]. This can be done in different ways, such as by defining specific classes of functions HE et collab. [2015] or by extracting high-level features GRAFF et collab. [2013a]; GRAFF et POLI [2008]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a,b] or statistical properties CLERGUE et collab. [2002]; GALVAN-LOPEZ et collab. [2008, 2010]; GOLDBERG [1987]; KIMURA [1983]; ROTHLAUF [2006]; VANNESCHI et collab. [2011]; VEREL et collab. [2003] of the fitness landscape. In the case of standard tree-based GP, where search operators are applied in syntax space, the concept of a fitness landscape is difficult to define given that there is no clear way of determining a general concept of neighborhood for GP representations that are usually highly redundant, which limits the usefulness of some of the above approaches. While some methods have been successfully applied to GP, these are mostly sampling approaches that attempt to infer specific types of structures within the underlying fitness landscape, such as: neutrality GALVAN-LOPEZ et collab. [2008]; GALVAN-LOPEZ et POLI [2006a,b]; POLI et GALVAN-LOPEZ [2012]; YU et MILLER [2001], locality GALVAN-LOPEZ et collab. [2010, 2011], ruggedness VANNESCHI et collab. [2011], deception TOMASSINI et collab. [2005], fitness distance correlation (FDC) CLERGUE et collab. [2002]; TOMASSINI et collab. [2005], fitness clouds VANNESCHI et collab. [2004] and negative slope coefficient (NSC) VANNESCHI et collab. [2006, 2007]. In this work, we refer to such methods as Evolvability Indicators (EIs), which are extensively reviewed in MALAN et ENGELBRECHT [2013] and discussed in the following

section.

One notable shortcoming of EIs is that they require an extensive sampling process of the search space in order to compute them ALTENBERG [1997]; QUICK et collab. [1998]; VANNESCHI et collab. [2003, 2009]. This is an important limitation: if we need to know when a particular problem is easy or difficult for an algorithm to solve it may just be easier to run the algorithm and observe its behavior and outcome. Therefore, some researchers have proposed predictive models that take the problem data (or a description of the data) as input, and produce as output a prediction of the expected performance, we will refer to such methods as Predictors of the Expected Performance (PEPs). Currently, the development of PEPs represents the minority of research devoted to problem difficulty in GP, with only a few recent works. In particular, Graff and Poli GRAFF et collab. [2013a]; GRAFF et POLI [2008, 2010, 2011]; GRAFF et collab. [2013b] have studied the development of such predictive models, for symbolic regression, Boolean and time-series problems. While their original work mostly focused on synthetic benchmarks GRAFF et POLI [2008], more recent contributions extended their approach to performance prediction in real-world problems GRAFF et collab. [2013a,b]. However, in their approach it is necessary to have an extensive knowledge of the real-world problems in advance. Furthermore, their models are intended to predict the performance of the best solution found by GP on the training set of data, they did not address the prediction of performance on unseen test cases.

This work is an extension of previous works TRUJILLO et collab. [2011a,b,c] where PEPs were first proposed for a GP classifier, making several methodological and experimental contributions. First, the PEP models are produced using only simple 2D synthetic datasets that are randomly generated. Second, the PEP models are used to predict the performance of the GP classifier on the test set of data, while previous works mostly focused on predicting performance on the training or learning set GRAFF et collab. [2013a]; GRAFF et POLI [2008, 2010, 2011]; GRAFF et collab. [2013b]. Third, accurate predictions are obtained on unseen real-world problems that are multidimensional and contain imbalanced data. On the other hand, previous works GRAFF et collab. [2013a]; GRAFF et POLI [2008, 2010, 2011]; GRAFF et collab. [2013b]; TRUJILLO et collab. [2011a,b,c] used the same type of problems (either synthetic or real) for both training and testing. Fourth, to increase PEP accuracy this chapter presents an ensemble approach using specialized PEP models called SPEPs. Each SPEP is trained to predict performance within a specific range of classification error. To do so, we use a two-tier approach, where each problem is first classified into a specific group, and then prediction is obtained from the corresponding SPEP which was trained for that group of problems. Finally, it is reasonable to state that the proposed approach could be applied to predict the performance of GP on other learning problems.

The remainder of this chapter proceeds as follows. Section 3.2 reviews related work and Section 3.3 provides a short survey of GP-based classification. The basic PEP strategy is outlined and evaluated in Section 3.4. Afterwards, Section 3.5 introduces the proposed ensemble strategy based on SPEPs and provides experimental results. Finally, Section 3.6 contains conclusions and future work.

3.2 Related Work

Determining problem difficulty has been an important issue in EC for several years MCCLYMONT et collab. [2012]. From an algorithmic perspective, problem difficulty can be related to the total runtime (or memory) required to find an optimal solution. Recently, He et al. HE et collab. [2015] took this view one step further, to analytically define broad classes of fitness functions which allowed them to demonstrate that easy functions define unimodal fitness landscapes, while hard functions define deceptive landscapes for a (1+1) ES. It is important to remember that the difficulty of a particular problem depends upon the solution method. Therefore, in what follows we will try to limit our overview to GP-related research.

3.2.1 Evolvability Indicators

The fitness landscape has dominated the way geneticists think about biological evolution and has been adopted by the EC community as a way to visualize evolution dynamics WRIGHT [1932]. Formally, a fitness landscape can be defined as a triplet (x, χ, f), where x is a set of configurations, χ is a notion of neighborhood, distance or accessibility on x, and f is a fitness function STADLER [2002]. The local and global structure of the fitness landscape describes the underlying difficulty of a search. However, in the case of standard GP LANGDON et POLI [2002] the concept of a fitness landscape is ill defined KINNEAR [1994]. To overcome this, some works have constructed synthetic problems; such as the Royal Tree problem PUNCH et collab. [1996] or the K-landscapes model VANNESCHI et collab. [2011], where the goal of the search is defined as a particular tree structure with a specific syntax. However, such models are not realistic since the space of possible programs is highly redundant LANGDON et POLI [2002] in most domains, and the goal is not a particular syntax but a particular expected output, also known as semantics MCPHEE et collab. [2008]; VANNESCHI et collab. [2014]. Therefore, some researchers have proposed variants of GP that explicitly account for program semantics. In semantic space the fitness landscape is clearly defined and unimodal. This has lead researchers to develop specialized search operators that modify program syntax while geometrically bounding the semantics of the generated offspring, this is known as geometric semantic GP (GSGP) MORAGLIO et collab. [2012]. Nevertheless, such approaches are still problematic since the size of the evolved programs grows exponentially with every generation, a limitation that is not easily solved SILVA et COSTA [2009]. This work will focus on measures of problem difficulty for standard GP systems [28], but could be applied to other supervised learning systems including GSGP.

In general, most meta-heuristics work under the assumption that the fitness of a candidate solution, a point on the fitness landscape, is positively correlated with the fitness of its (some) neighbors. Such a property can be defined as the *evolvability* of a landscape ALTENBERG [1994]; O'NEILL et collab. [2010]. EIs extract a numerical indicator of a specific property of the fitness landscape to provide a measure of the evolvability within the landscape. Malan et al. MALAN et ENGELBRECHT [2013] presents a comprehensive survey of EIs and other forms of fitness landscape analysis.

Those that have been studied in GP literature include neutrality GALVAN-LOPEZ et collab. [2008]; KIMURA [1983], locality GALVAN-LOPEZ et collab. [2010]; ROTHLAUF [2006], ruggedness KAUFFMAN et LEVIN [1987]; VANNESCHI et collab. [2011], fitness distance correlation (FDC) CLERGUE et collab. [2002]; JONES et FORREST [1995]; TOMASSINI et collab. [2005], fitness clouds VEREL et collab. [2003] and the negative slope coefficient (NSC) VANNESCHI et collab. [2004]. While these approaches can sometimes provide good estimates of problem difficult for GP, they suffer from two practical limitations. First, for each new problem instance they require a large amount of data, by sampling the search space or performing several runs. Second, they cannot estimate the actual quality of the solution found, which can be important if we want to choose the best algorithm to use for a new problem, and if such a choice must be made in real-time. Indeed, Malan et al. MALAN et ENGELBRECHT [2013] point out that a possible way forward is to build a mapping that can estimate algorithm performance based on a set of descriptive features of the problem, an approach that would provide a more practical measure of problem difficulty and allow us to choose the best algorithm for the specific task. Furthermore, Malan and Engelbrecht MALAN et ENGELBRECHT [2014] attempted to find a link between EIs and algorithm performance for particle swarm optimization.

3.2.2 Performance Prediction

PEPs predict the performance of a GP search on an unseen problem instance without performing the search or sampling the solution space. These models have been derived using

a machine learning approach GRAFF et collab. [2013a]; GRAFF et POLI [2008]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a,b]. The performance of GP on a set of problems and a description of those problems are used to pose a supervised learning task. A promising feature of PEPs is that they are not only useful for GP, they can also be used to predict the performance of other algorithms GRAFF et POLI [2010]; TRUJILLO et collab. [2011b].

Graff and Poli GRAFF et POLI [2010] proposed linear predictive models based on a sampling of the fitness landscape, given by

$$P(\mathbf{t}) \approx a_0 + \sum_{\mathbf{s} \in \mathcal{S}} a_{\mathbf{s}} \cdot d(\mathbf{s}, \mathbf{t}) \ , \tag{3.1}$$

where $P(\mathbf{t})$ is the predicted performance, $\mathbf{t}$ is the target functionality, $d(\mathbf{s}, \mathbf{t})$ is a distance measure[1], $\mathcal{S}$ is the set of all possible program outputs, which is also known as semantic space MORAGLIO et collab. [2012], and where each $\mathbf{s}$ represents the vector of program outputs obtained from the set of fitness cases used to define a particular problem, also known as the semantics of the program MCPHEE et collab. [2008]. In other words, Graff and Poli GRAFF et POLI [2010] derive PEPs by sampling semantic space $\mathcal{S}$. These models were tested on symbolic regression and 4-input Boolean problems with promising results.

The second and more recent approach towards building a PEP focuses on the problem data GRAFF et collab. [2013a]; GRAFF et POLI [2011]; GRAFF et collab. [2013b]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a, 2012, 2011b,c] and proceeds as follows. Assume we want to solve a supervised learning problem p with a GP search, where fitness is given by a cost function that must be minimized, such as an error measure. Lets define the performance of the GP algorithm as the associated error of the best solution found during training when it is evaluated on a particular set of fitness cases T, call this quantity $F_T(p)$. The goal is to predict $F_T(p)$, so first we construct a feature vector $\beta = (\beta_1, \beta_2, ...\beta_N)$ of N distinct features that describe the main properties of p. Then, a PEP is function K such that

$$F_T(p) \approx K(\beta) \ . \tag{3.2}$$

Notice that the form of K is not a priori restricted in any way. Graff and Poli GRAFF et POLI [2011] use a linear function similar to the one used in their previous work GRAFF et POLI [2010]. Using this approach the feature vector β should be designed specifically for the domain of p. For example, features designed for symbolic regression and Boolean problems are proposed in GRAFF et POLI [2011], and the results show that the predictive accuracy surpasses that of the fitness-based models proposed in GRAFF et POLI [2010]. However, their work did not scale well to real-world cases. For instance, in GRAFF et collab. [2013a,b] the authors built PEPs to predict performance on real-world problems, but require information obtained from runs performed on similar problem instances, models built with simpler synthetic problems could not be used. It was not trivial to map multidimensional problems to the proposed feature space since the training problems were much simpler with a small number of dimensions. It would be impractical to consider all possible dimensionalities during training. This is an important limitation in building PEPs, since it is not trivial to have all the possible versions of the same problem. Moreover, in the proposals made by Graff and Poli GRAFF et collab. [2013a]; GRAFF et POLI [2011]; GRAFF et collab. [2013b] the models predicted the performance of the GP system on the training set of fitness cases; i.e., T was the training set. While certainly of importance, performance on the training set may not be useful if the algorithm overfits the training examples, which happens often in real-world scenarios.

In previous work TRUJILLO et collab. [2011a,b,c], a similar approach was used to predict the performance of a GP-classifier using descriptive features that characterize the geometry of the data distribution in feature space. The PEPs where built using quadratic models

[1] Such a distance measure is a common fitness function for many application domains of GP, particularly for symbolic regression problems.

and non-linear GP models, the latter achieving the best performance on synthetic problems. However, it was not clear how well the PEPs generalized to unseen problem instances, particularly to real-world problems with imbalanced datasets and larger feature spaces than those used to train the models, a similar difficulty pointed out in GRAFF et collab. [2013a,b]. The current work extends our previous contributions by performing the learning process on 2D synthetic problems and testing on a wide variety of real-world datasets. Moreover, an important contribution of this work is that the PEP models are used to predict the performance of the best solution found by GP when it is evaluated on the test set of data. To achieve improved performance this work proposes a two-tiered ensemble approach using specialized PEP models and a preprocessing stage for dimensionality reduction.

3.3 Classification with GP

In machine learning one of the most common tasks is supervised classification KOTSIANTIS et collab. [2006]. The general task can be stated as follows. Given a pattern $\mathbf{x} \in \mathbb{R}^P$ assign the correct class label among C distinct classes $\omega_1, ..., \omega_C$, using a training set $\mathcal{T}$ of P-dimensional patterns with a known label. The idea is to build a mapping $g(\mathbf{x}) : \mathbb{R}^P \rightarrow C$, that assigns each pattern $\mathbf{x}$ to a corresponding class ω_i, where g is derived based on the evidence provided by $\mathcal{T}$. GP has been widely used to address this problem MUNOZ et collab. [2015]; SOTELO et collab. [2013]; TRUJILLO et collab. [2011a]; Z-FLORES et collab. [2015]; ZHANG et SMART [2004, 2006]. In general, GP can be applied to classification following three general approaches:

1. Feature selection and construction MUHARRAM et SMITH [2005]; MUNOZ et collab. [2015]; SHERRAH et collab. [1997]; TRUJILLO et collab. [2011a]; ZHANG et SMART [2006].

2. Model extraction BENTLEY [2000]; TANIGAWA et ZHAO [2000]; TSAKONAS [2006]; Z-FLORES et collab. [2015]; ZHANG et SMART [2004].

3. Learning ensemble classifiers HENGPRAPROHM et CHONGSTITVATANA [2008]; IMAMURA et collab. [2003]; LANGDON et POLI [2002].

Feature selection and construction is also known as preprocessing of the problem data. These approaches use GP to either select the most interesting problem features or to construct new features that simplify the classification problem. These techniques are often described as either filter GUO et collab. [2005]; MUHARRAM et SMITH [2005]; TRUJILLO et collab. [2011a]; ZHANG et SMART [2006] or wrapper approaches MUNOZ et collab. [2015]; SHERRAH et collab. [1997]; SMITH et BULL [2005]. In the former, feature construction is done independently of the model used to build the classifier, while in the latter fitness assignment is based on the performance of a classifier. On the other hand, model extraction with GP is used to build specific types of classifiers, such as decision trees TANIGAWA et ZHAO [2000]; TSAKONAS [2006], classification rules BENTLEY [2000]; QING-SHAN et collab. [2007] and discriminant functions ZHANG et SMART [2004]. Finally, ensemble classifiers are used to improve the quality of the classification task by using not only a single classifier, but a group of them, each one providing a different output HENGPRAPROHM et CHONGSTITVATANA [2008]; IMAMURA et collab. [2003]; LANGDON et POLI [2002].

3.3.1 Probabilistic Genetic Programming Classifier

In this work we derive PEPs for the Probabilistic Genetic Programming Classifier (PGPC) proposed in ZHANG et SMART [2006], a feature construction method. PGPC was chosen due to its simplicity and strong performance on real-world problems SOTELO et collab. [2013]; while other GP-based classifiers could have been used this is left as future work. In PGPC,

GP is used to evolve a mapping $g(\mathbf{x}) : \mathbb{R}^P \rightarrow \mathbb{R}$ that transforms each input pattern $\mathbf{x}$ into a point on the real line. Furthermore, it is assumed that the behavior of g can be modeled using multiple Gaussian distributions, each corresponding to a single class ZHANG et SMART [2006]. The distribution of each class $\mathcal{N}(\mu, \sigma)$ is derived from the examples provided for it in set $\mathcal{T}$, by computing the mean μ and standard deviation σ of the outputs obtained from g on these patterns. Then, from the distribution $\mathcal{N}$ of each class a fitness measure can be derived using Fisher's linear discriminant; for a two class problem it proceeds as follows. After the Gaussian distribution $\mathcal{N}$ for each class are derived, a distance is required. In ZHANG et SMART [2006], Zhang and Smart propose a distance measure between both classes as

$$d = 2 * \frac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2} , \tag{3.3}$$

where μ_1 and μ_2 are the means of the Gaussian distribution of each class, and σ_1 and σ_2 their standard deviations. When this measure tends to 0, it is the worst case scenario because the mappings of both classes overlap completely, and when it tends to ∞, it represents the optimal case with maximum separation. To normalize the above measure, the fitness for an individual mapping g is given by

$$f_d = \frac{1}{1 + d} . \tag{3.4}$$

After executing PGPC, the best individual found determines the parameters for the Gaussian distribution $\mathcal{N}_i$ associated to each class. Then, a new test pattern $\mathbf{x}$ is assigned to class i when $\mathcal{N}_i$ gives the maximum probability; performance is measured by the total classification error (CE) on test data.

3.4 PEP: Predictor of Expected Performance

The general goal of this work is to build models that can predict the performance of a GP-classifier (PGPC) without executing the search or sampling the problem's search space. The general proposal is depicted in Figure 3.1, where for a given classification problem we do the following. First, apply a preprocessing step to simplify the feature extraction process and deal with multidimensional representations. Second, perform feature extraction to obtain an abstraction of the problem. Third, use a PEP model that takes as input the extracted features and produces as output the predicted classification error (PCE) on the testing set.

Moreover, to derive the PEP models we use a supervised learning methodology, depicted in Figure 3.2. This process takes as input a set of synthetic classification problems $\mathcal{Q}$ and produces as output the PEP model as follows:

1. Compute the average classification error (CEμ) on the test data by PGPC for each $p \in \mathcal{Q}$.

2. Apply a preprocessing for dimensionality reduction using principal component analysis (PCA), and take the first m principal components to represent the problem data.

3. Perform feature extraction on the transformed data using statistical and complexity measures to build a feature vector β for each $p \in \mathcal{Q}$.

4. Finally, using the set of feature vector/performance pairs $\{(\beta_i$,CE$\mu_i)\}$ formulate a supervised symbolic regression problem and solve it using GP.

3.4.1 Synthetic Classification Problems

A set of synthetic classification problems was generated to learn our PEP models. Specifically, 500 binary classification problems were generated using Gaussian mixture models

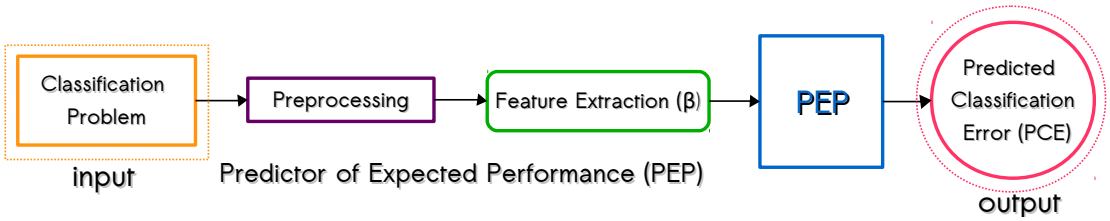


Figure 3.1 – Block diagram of the proposed PEP approach. Given a classification problem, the goal is to predict the performance of the GP classifier on the test data, in this case PGPC.

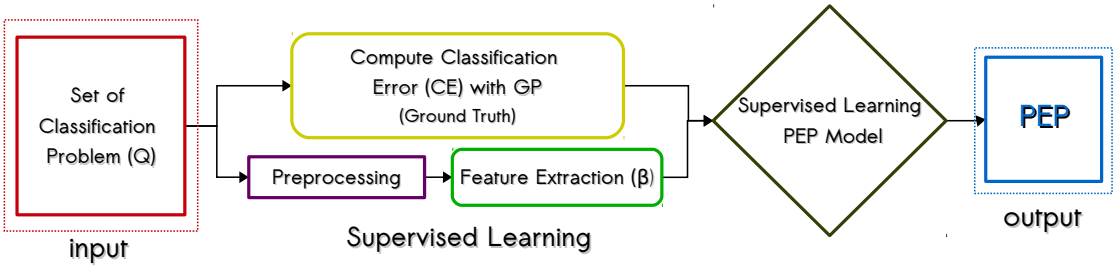


Figure 3.2 – The methodology used to build the PEP model. Given a set $\mathcal{Q}$ of synthetic classification problems: (1) compute the CEμ of PGPC on all problems; (2) apply a preprocessing for dimensionality reduction; (3) extract the feature vector β from the problem data; and (4) learn the predictive model using GP.

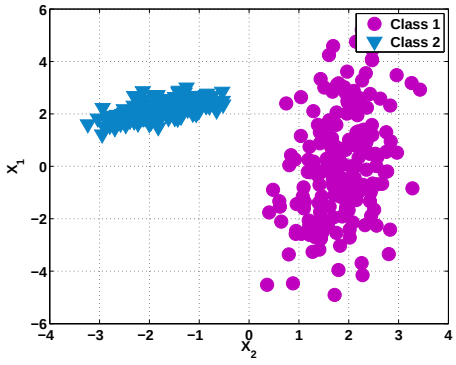
(GMMs) with either unimodal or multimodal classes, with different amounts of class overlap. All class samples lie within the closed 2-D interval $x, y \in [-10, 10]$, and 200 sample points were randomly generated for each class. The parameters for the GMM of each class were randomly chosen using a uniform distribution in the following ranges:

1. Number of Gaussian components: $\{1, 2, 3\}$.
2. Median of each Gaussian component for each feature dimension: $[-3, 3]$.
3. Each element of the covariant matrix of each Gaussian component: $(0, 2]$.
4. The rotation angle of each covariance matrix: $[0, 2\pi]$.
5. Proportion of samples generated with each Gaussian component: $[0, 1]$.

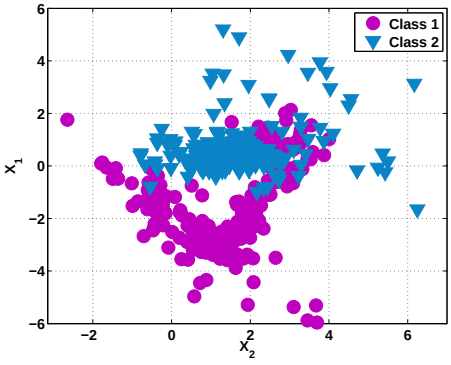
3.4.2 PGPC Classification Error

For each problem $p \in \mathcal{Q}$ we perform 30 runs of PGPC randomly choosing the training and testing sets. Then, the mean classification error CEμ is computed by the average of the test performance achieved by the best solutions in each run. The parameters of the PGPC system are given in Table 3.1, a tree-based GP algorithm with dynamic depth bloat control SILVA et COSTA [2009], implemented using Matlab and the GPLAB toolbox SILVA et ALMEIDA [2003]. Figure 3.3 presents some examples, showing the problem data, the CEμ achieved by PGPC and the standard deviation σ over all runs. The problems are ordered from the lowest CEμ (easiest problem) to the highest CEμ (hardest problem).

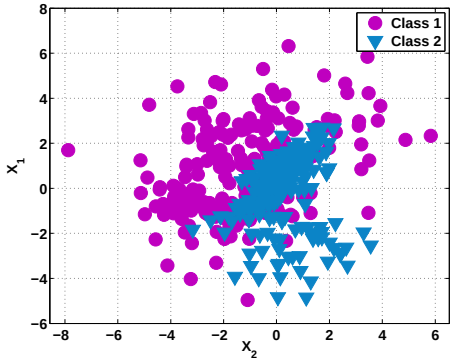
Figure 3.4 summarizes PGPC performance over all 500 synthetic problems in $\mathcal{Q}$. Figure 3.4(a) plots the CEμ for each problem, ordered from the lowest to the highest error. On the other hand, Figure 3.4(b) shows an histogram of PGPC performance, quantifying how many problems are solved with a particular CEμ. We arbitrarily set a threshold such that problems in the range $0 \leq$CE$\mu \leq 0.15$ are considered “easy" and the rest are “hard". From this perspective the plot reveals that randomly generated problems produce a biased distribution, where most problems are easy to solve. Since we intend to use this set to pose a supervised



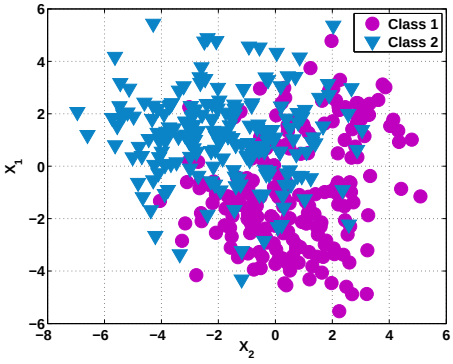
(a) [CE$\mu = 0$ $\sigma = 0$]



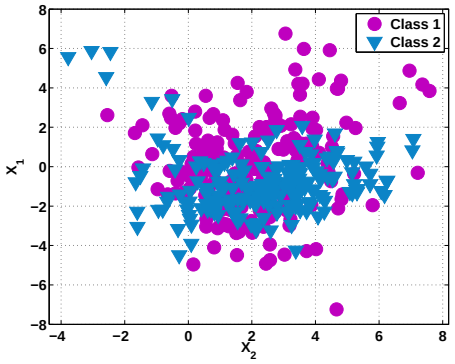
(b) [CE$\mu = 0.14$ $\sigma = 0.03$]



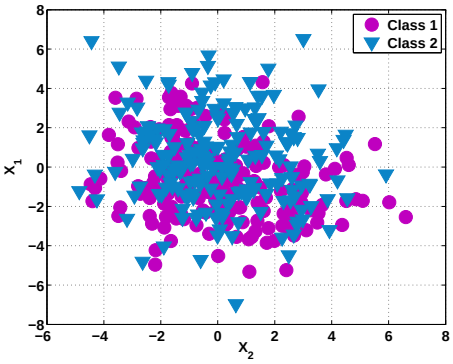
(c) [CE$\mu = 0.17$ $\sigma = 0.03$]



(d) [CE$\mu = 0.21$ $\sigma = 0.03$]



(e) [CE$\mu = 0.36$ $\sigma = 0.04$]



(f) [CE$\mu = 0.46$ $\sigma = 0.04$]

Figure 3.3 – The scatter plots show examples of synthetic classification problems, specifying the CEμ and standard deviation σ achieved by PGPC. These ordered from the lowest CEμ (easiest) to the highest CEμ (hardest).

Table 3.1 – Parameters for the PGPC algorithm.

Parameter	Description
Population size	200 individuals
Generations	200 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.8$; Mutation $p_\mu = 0.2$
Function set	$\{+, -, *, /, \sqrt{}, sin, cos, log, x^y, \lvert\cdot\rvert, if\}$
Terminal set	$\{x_1, ..., x_i, ..., x_P\}$ Where each x_i is a dimension of the data patterns $\mathbf{x} \in \mathbb{R}^P$
Bloat control	Dynamic depth control
Initial dynamic depth	6 levels
Hard maximum depth	20 levels
Selection	Tournament Size 3
Survival	Keep best elitism
Training Data	70%
Testing Data	30%
Runs	30

learning problem, this would induce an unwanted bias. Therefore, we subsample $\mathcal{Q}$ to get a more balanced distribution over CEμ. The new set consists of 300 problems, and Figure 3.5 summarizes PGPC performance over this new set $\mathcal{Q}'$. Notice that the performance plot for $\mathcal{Q}' \subset \mathcal{Q}$ is similar to the one obtained for $\mathcal{Q}$ (see Figure 3.5(a)), but now the distribution over CEμ is flat (Figure 3.5(b)), providing a more balanced learning set.

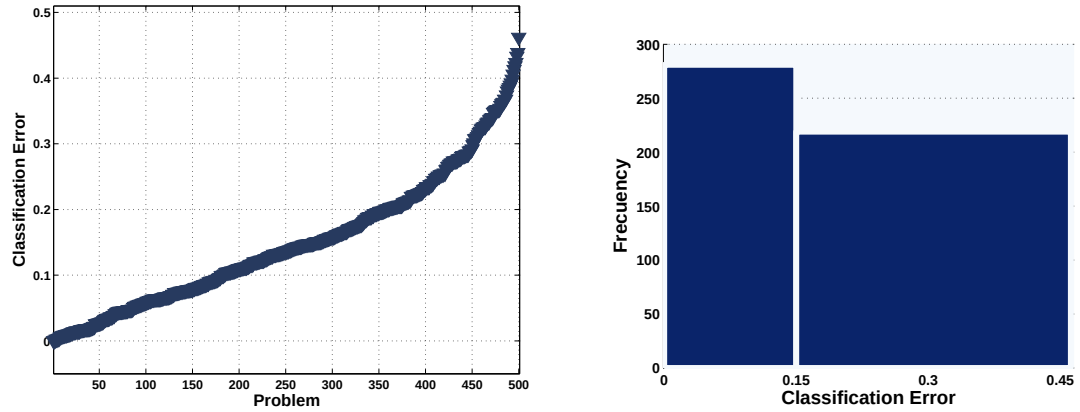


Figure 3.4 – Performance of PGPC over all 500 synthetic problems in $\mathcal{Q}$; where: (a) shows the CEμ for each problem, ordered from the easiest to the hardest; and (b) shows the histogram over CEμ.

3.4.3 Preprocessing

Previous work has found that PEP models can predict GP performance accurately for small scale synthetic problems GRAFF et POLI [2008, 2010, 2011]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a, 2012, 2011b,c], but accuracy degrades for real-world problems with high dimensional data GRAFF et collab. [2013a,b]. This is due to the fact that feature extraction (the next step in the PEP approach) fails at extracting meaningful information in high dimensional spaces GRAFF et collab. [2013a,b]. To deal with this issue, we apply a dimensionality reduction preprocessing of the problem data using PCA DUDA et collab. [2000]. We propose to take the first m principal components to represent the data of each problem.

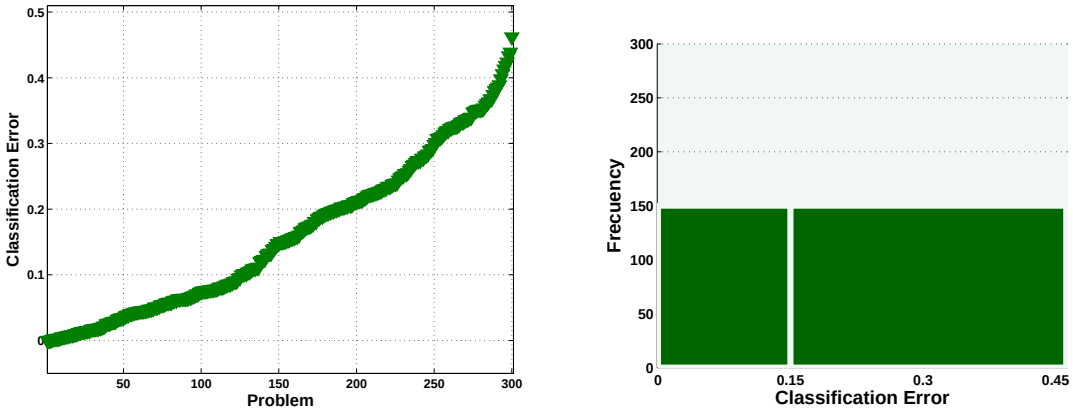


Figure 3.5 – Performance of PGPC over all 300 synthetic problems in $\mathcal{Q}' \subset \mathcal{Q}$; where: (a) shows the CEμ for each problem, ordered from the easiest to the hardest; and (b) shows the histogram over CEμ.

In particular, we set $m = 2$ in all experiments reported here. In this way, all problems are reduced to the same number of dimensions used in the synthetic training set.

3.4.4 Feature Extraction

The goal of this step is to extract a set of descriptive measures from each problem. In this work, we use a subset of the features proposed in SOHN [1999] and HO et BASU [2002]. Those works attempted to develop meta-representations of classification problems. A wider set of features was previously tested in TRUJILLO et collab. [2011a, 2012, 2011b,c], but the present work only uses those features that showed the highest correlation with PGPC CEμ. We also propose three new measures based on the Canberra distance; each measure is presented next.

3.4.4.0.1 Geometric mean (SD): measures the homogeneity of covariances MICHIE et collab. [1994]; SOHN [1999]. This quantity is related to a test of the hypothesis that all populations have a common covariance structure; i.e.. to the hypothesis $H_0 : \sum_1 = \sum_2$, which can be tested via Box's M test statistic (MTS), that can be re-expressed as

$$SD = exp\left\{\frac{MTS}{m\sum_{i=1}^{C}(n_i - 1)}\right\} \tag{3.5}$$

where C is the number of classes, n_i is the number of the instances for i-th class and m is the number of features. The SD is strictly greater than unity if the covariances differ, and is equal to unity if and only if the MTS is zero.

3.4.4.0.2 Feature Efficiency (FE): measures the amount by which each feature dimension contributes to the separation of both classes. This measure is computed for the $i-th$ feature by

$$FE_i = \left(1 - \frac{\eta_i}{tp}\right) \tag{3.6}$$

where η_i represent the number of points inside the overlapping region and tp is the total number of sample points; as seen in Figure 3.6(a). Finally, we define $FE = max(\{FE_i\})$ with $i = [1, m]$ for any given problem.

3.4.4.0.3 Class Distance Ratio (CDR): compares the dispersion within the classes to the gap between the classes HO et BASU [2002]. For each data sample, compute the Euclidean distance to its nearest neighbor within the class (intraclass distance) and nearest-neighbor from the other class (interclass distance), as shown in Figure 3.6(b). The CDR is the ratio of the averages of all intraclass and interclass distances.

3.4.4.0.4 Volume of Overlap Region (VOR): provides an estimate of the amount of overlap between both classes in feature space HO et BASU [2002]. The VOR is computed by finding, for each feature, the maximum and minimum value of each class and then calculating the length of the overlap region. The length obtained from each feature is then multiplied to measure the overlapping region, as depicted in Figure 3.6(c). The VOR is zero when there is at least one dimension in which the two classes do not overlap.

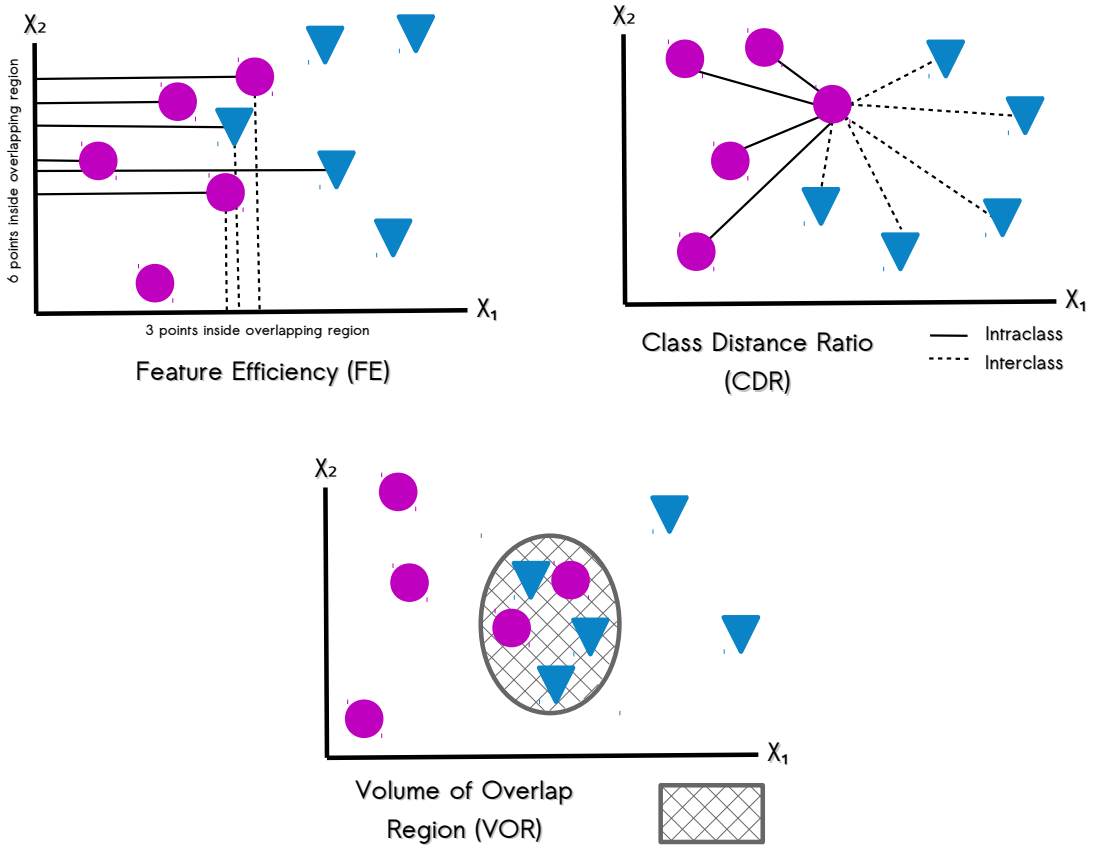


Figure 3.6 – These figures depict the complexity features used to describe each classification problem as suggested in HO et BASU [2002], where: (a) Feature Efficiency (FE); (b) Class Distance Ratio (CDR); and (c) Volume of Overlap Region (VOR).

3.4.4.0.5 Canberra Distance (CD): provides a numerical measure of the distance between pairs of points in a vector space. Suppose a problem has m features, we take a rank statistic of the samples of each class, call it x_i for class 1 and y_i for class 2, for the i-th feature. This produces two vectors $\mathbf{x}$ and $\mathbf{y}$, such that $\mathbf{x} = (x_1, ..., x_m)$ and $\mathbf{y} = (y_1, ..., y_m)$. The CD is given by:

$$CD(\mathbf{x}, \mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i| + |y_i|}. \tag{3.7}$$

In this work, we use the CD to describe the distance between both classes using three rank statistics: (1) CD-1 uses the 1st quartile; (2) CD-2 uses the median; and (3) CD-3 uses the 3rd

Table 3.2 – Parameters for the GP used to derive PEP models for PGPC algorithm.

Parameter	Description
Population size	200 individuals
Generations	100 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.8$; Mutation $p_\mu = 0.2$
Hard maximum depth	12 levels
Selection	Tournament Size 3
Survival	Keep best elitism
Runs	100

quartile.

The meta-representations discussed above, helps to minimize the information about each problem. Now, analyzing the algorithmic complexity (Big O notation) of the measures, these do not represent an important computational cost. For instance, the FE, VOR, CD-1, CD-2 and CD-3 features mainly depend on a sorting process, which can have a complexity of $O(n \log n)$. Moreover, the SD relies on computing the covariance matrix of the data which has a complexity of $O(n^2)$. Similarly, to compute the CDR feature we need to do all pairwise comparisons, which also has a complexity of $O(n^2)$.

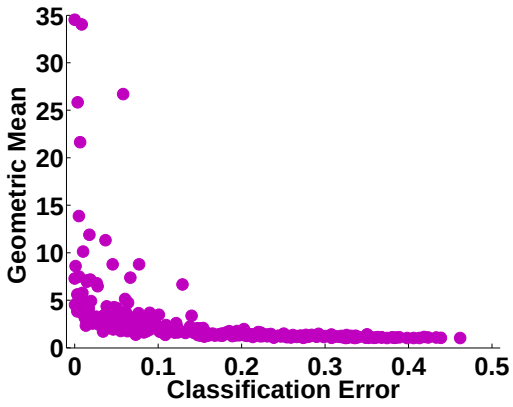
Figure 3.7 provides a visual description of the descriptive power of each feature. The figure shows scatter plots where each point corresponds to a single problem $p \in \mathcal{Q}'$, the x-axis is a particular feature (SD, FE, CDR, VOR, CD-1, CD-2 and CD-3) and the y-axis is the associated CEμ. The legend of each plot also gives the Pearson's correlation coefficient ρ. It is evident that all of the chosen features are correlated with PGPC performance, in particular FE, VOR, CDR, CD-1 and CD-3 show the highest correlation.

3.4.5 Supervised Learning of PEP Models

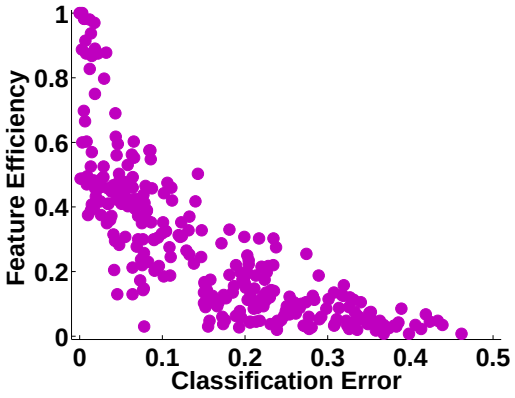
It is now possible to pose a symbolic regression problem using the set $T = \{(\beta_i, \text{CE}\mu_i)\}$ with $i = 1, .., |\mathcal{Q}'|$, where the goal is to evolve a model K that can predict each CEμ_i using β_i as input. Previous works have used several types of linear models GRAFF et POLI [2011]; MARTINEZ et collab. [2012]; TRUJILLO et collab. [2011a, 2012, 2011b,c], but TRUJILLO et collab. [2011a,b,c] showed that non-linear models evolved with GP achieved higher prediction accuracy.

Therefore, in this work we use a tree-based GP, configured with the parameters given in Table 3.2. Three versions of the problem are posed, each with a different terminal set defined as subsets of all extracted features(4F, 5F, 7F) as specified in Table 3.3. Set 4F uses the features with the four highest correlation coefficients (FE, CDR, VOR and CD-1), set 5F uses the features with the five highest correlation coefficients (SD, FE, CDR, VOR and CD-1), and 7F uses all of the seven features. The function set is defined as $F = \left\{+, -, *, /, \sqrt{}, sin, cos, log, x^y, |\cdot|, if\right\}$. Finally the fitness function is computed by the root mean squared error (RMSE) between the predicted CE and the true CEμ_i, given by

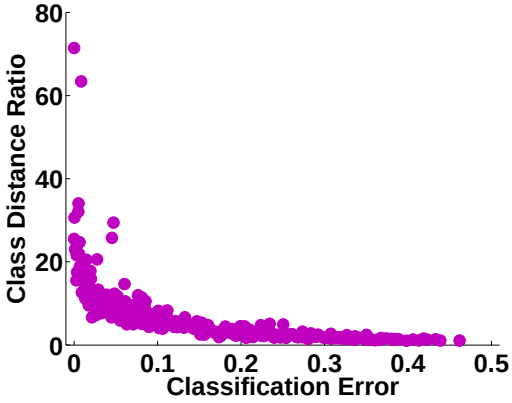
$$f(K) = \sqrt{\frac{\sum_{i=1}^{n}(K(\beta_i) - CE\mu_i)^2}{n}}. \tag{3.8}$$



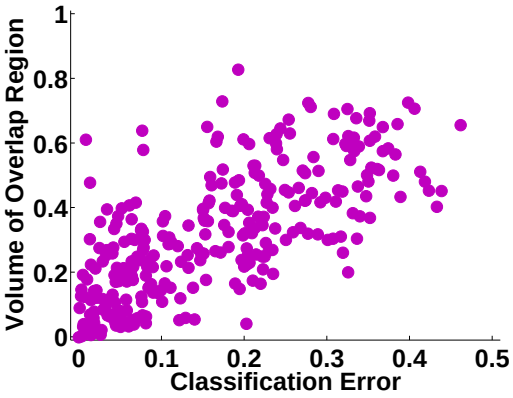
(a) [SD: $\rho = -0.42$]



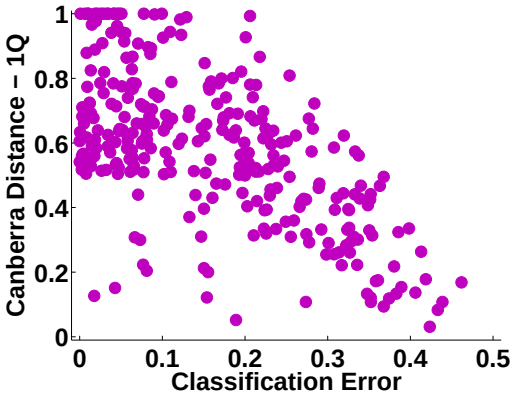
(b) [FE: $\rho = -0.78$]



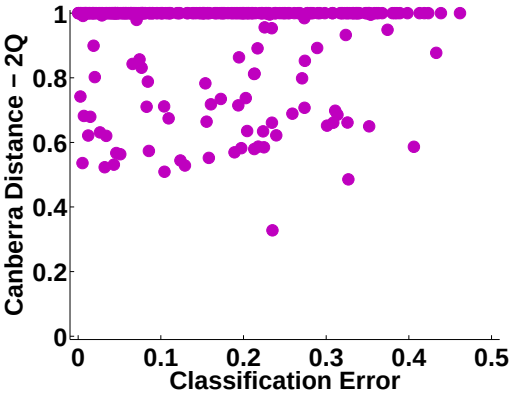
(c) [CDR: $\rho = -0.62$]



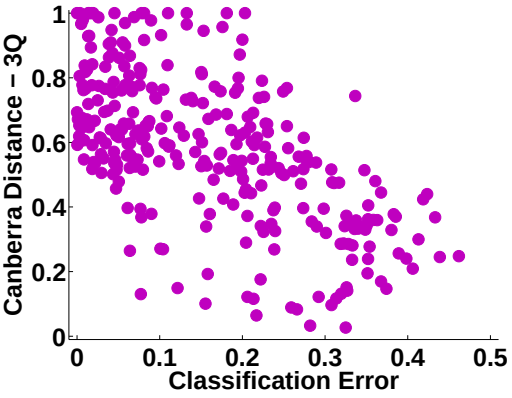
(d) [VOR: $\rho = 0.72$]



(e) [CD-1: $\rho = -0.62$]



(f) [CD-2: $\rho = -0.03$]



(g) [CD-3: $\rho = -0.61$]

Figure 3.7 – The scatter plots show the relationship between the CEμ (x-axis) and each descriptive feature (y-axis) for all problems $p \in \mathcal{Q}'$, where ρ specifies Pearson's correlation coefficient.

Table 3.3 – Three different features sets used as terminal elements for the symbolic regression GP algorithm.

Feature vector β	
4F	FE, CDR, VOR and CD-1
5F	SD, FE, CDR, VOR and CD-1
7F	SD, FE, CDR, VOR, CD-1, CD-2 and CD-3

Table 3.4 – Prediction performance of the evolved PEPs applied on the synthetic problems using each feature set (4F, 5F and 7F, see Table 3.3). Performance is given based on the RMSE and Pearson's correlation coefficient, with bold indicating the best performance.

	median ***training*** RMSE	median ***testing*** RMSE	***best*** RMSE	***best*** correlation
PEP-4F	0.0320	0.0375	0.0318	0.9634
PEP-5F	**0.0317**	**0.0362**	**0.0295**	**0.9688**
PEP-7F	0.0326	0.0364	0.0317	0.9636

3.4.6 Testing the PEP models

For each version of the symbolic regression problem defined above (with different feature sets), we performed 100 runs using two different test scenarios: (1) train and test the PEP models using only synthetic problems; and (2) train with synthetic problems and test with real-world problems. In the first scenario, we use 70% of the problems for training and the rest for testing, generating a random partition of the set of problems $\mathcal{Q}'$ for each run. This is the simplest scenario, since both the training and testing problems are generated in the same manner. In the second scenario, we use test the PEP models trained with synthetic problems and evaluate their predictions on many real-world datasets, a more challenging scenario since the real-world problems have high dimensional data, imbalanced classes and different data distributions.

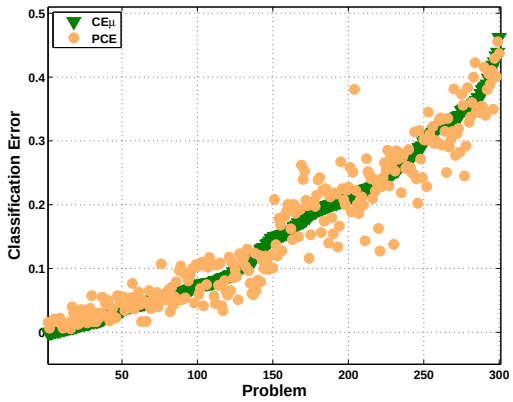
3.4.6.1 Testing on Synthetic Classification Problems

Table 3.4 summarizes the performance of the evolved PEPs, showing the median of the RMSE of the best solution found in each run for the training and testing sets, as well as the RMSE and Pearson's correlation coefficient ρ of the best solution found. The table presents three rows of results, one for each feature set (PEP-4F, PEP-5F and PEP-7F). The numerical results are encouraging, suggesting that the PEP models can accurately predict PGPC performance. Moreover, there is a very small difference between training and testing performance, suggesting that the PEP models are not overfitted.

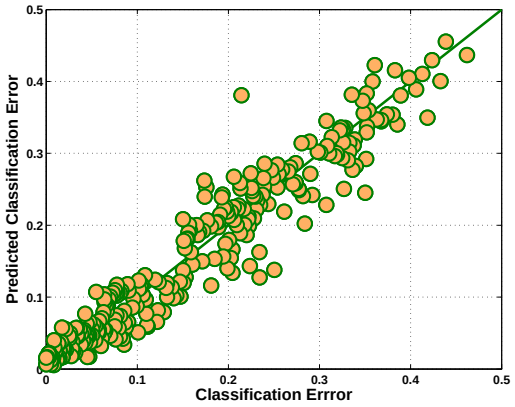
Figure 3.8 shows plots in three rows, where in each row we plot each feature set (PEP-4F, PEP-5F and PEP-7F). The plots on the left column show the PCE of the best PEP model and the true CEμ for all synthetic problems, specifying the RMSE. The plots on the right column show the CEμ and PCE as scatter plots, specifying the Pearson's correlation coefficient ρ. The evolved PEPs produce highly accurate predictions with all feature sets.

3.4.6.2 Testing on Real-World Classification Problems

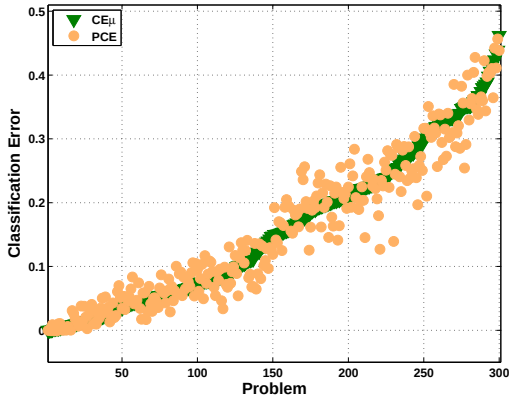
This section presents the results of testing the best evolved PEPs to predict the testing error of PGPC on real-world problems. To this end, twenty-two real-world datasets are chosen from the University of California Irvine (UCI) machine learning repository LICHMAN [2013], summarized in Table 3.5. Since our PEPs only consider binary classification, we use these datasets to build 40 binary classification problems. The problems are summarized in Table



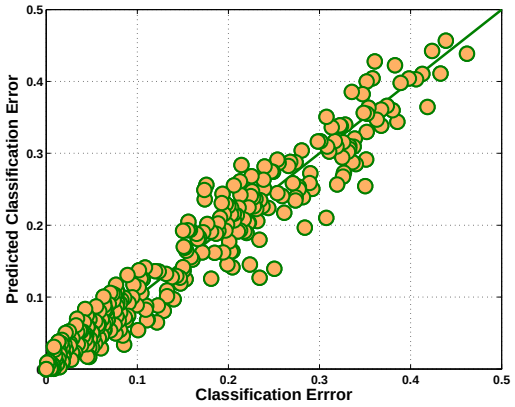
(a) [*PEP-4F:* RMSE = 0.0318]



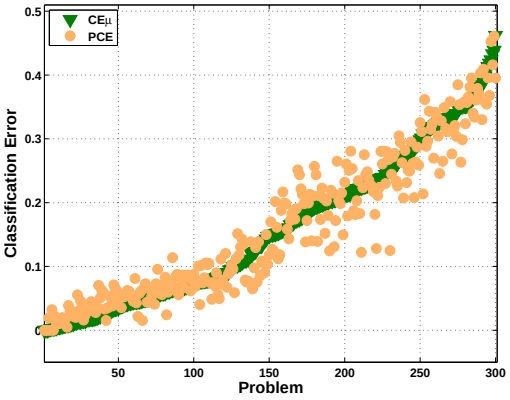
(b) [*PEP-4F:* $\rho = 0.9634$]



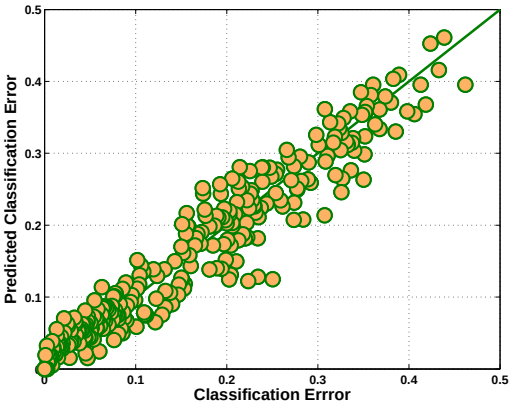
(c) [*PEP-5F:* RMSE = 0.0295]



(d) [*PEP-5F:* $\rho = 0.9688$]



(e) [*PEP-7F:* RMSE = 0.0317]



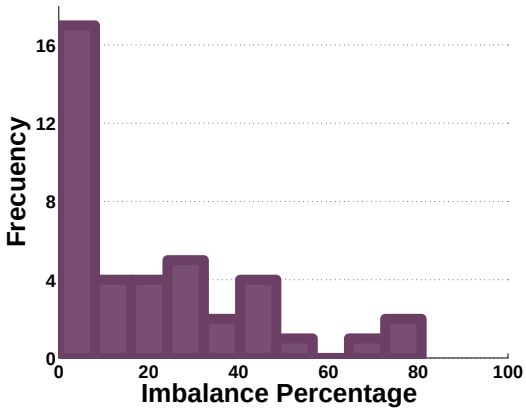
(f) [*PEP-7F:* $\rho = 0.9636$]

Figure 3.8 – Figures showing for synthetic problems, the performance prediction of the best PEP models evolved with the different feature set, each row belongs to each feature set: PEP-4F(top), PEP-5F(middle) and PEP-7F(bottom). The plots on the left column show the PCE of the best solution and the know CEμ, specifying the corresponding RMSE. The right column shows scatter plots of the PCE and the CEμ, specifying Pearson's correlation coefficient ρ.

3.6, specifying the name of the dataset and the classes used to define each problem, the number of total samples and the imbalance percentage of each problem computed as $\frac{a-b}{c}$ where a and b are respectively the number of samples in the minority and majority class, and c is the total number of samples. Notice that the synthetic problems used to train the PEPs are completely balanced and relatively small problems in terms of number of samples, while the real-world problems are considerably more varied. In particular, considering class imbalance Figure 3.9 shows a histogram of the number of problems with different amounts of imbalance percentage.

Table 3.5 – Real-world datasets from the UCI machine learning repository used in this work.

No.	Problem	Classes	Features	Description
1	*Balance scale*	3	4	Balance scale weight and distance database.
2	*Breast cancer wisconsin*	2	8	Original Wisconsin Breast Cancer Database.
3	*Breast tissue*	6	9	Dataset with electrical impedance measurements of freshly excised tissue samples from the breast.
4	*Cardiotocography*	3	23	Fetal cardiotocograms (CTGs) were automatically processed and the respective diagnostic features measured.
5	*EEG eye state*	2	15	All data is from one continuous EEG measurement with the Emotiv EEG Neuroheadset.
6	*Fertility*	2	10	100 volunteers provide a semen sample analyzed according to the WHO 2010 criteria.
7	*Glass*	6	10	From USA Forensic Science Service; 6 types of glass.
8	*Indian liver patient*	2	32	This data set contains 416 liver patient records and 167 non liver patient records.
9	*Ionosphere*	2	32	Classification of radar returns from the ionosphere.
10	*Iris*	3	4	The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
11	*parkinsons*	2	22	Oxford Parkinson's Disease Detection Dataset.
12	*Pima indians diabetes*	2	8	From National Institute of Diabetes and Digestive and Kidney Diseases.
13	*Retinopathy*	2	19	This dataset contains features extracted from the Messidor image set to predict whether an image contains signs of diabetic retinopathy or not.
14	*Red wine*	6	11	The goal is to model wine quality based on physicochemical tests.
15	*Seed*	3	7	The examined group comprised kernels belonging to three different varieties of wheat.
16	*Sonarall*	2	60	The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock.
17	*Tae*	3	5	The data consist of evaluations of teaching performance; scores are "low", "medium", or "high".
18	*Vertebral-column 2C*	2	6	Biomedical data set built by Dr. Henrique da Mota.
19	*Vertebral-column 3C*	3	6	Biomedical data set built by Dr. Henrique da Mota.
20	*White wine*	6	11	The goal is to model wine quality based on physicochemical tests.
21	*Wine*	3	13	Using chemical analysis determine the origin of wines.
22	*Zoo*	7	3	Artificial, 7 classes of animals.



IMBALANCED PROBLEMS

1. The PEP models were trained with balanced problems.
2. The real-world test problems show a varied amount of imbalanced cases.

Figure 3.9 – Histogram of imbalance percentage for the 40 real-world classification problems.

Before testing the evolved PEP models, we compute the CEμ achieved by PGPC using 30 independent runs. PGPC performance is summarized in Figure 3.10, showing: (a) the CEμ for each problem and (b) the histogram over CEμ. Figures 3.11 presents scatter plots

Table 3.6 – The 40 real-world binary classification problems based on the UCI datasets.

No.	Problem	Classes	Instances	Imbalance %
1	*Balance scale*	1-3	576	0
2	*Breast cancer wisconsin*	1-2	699	31
3	*Breast tissue*	1-2	36	17
4	*Breast tissue*	1-3	39	8
5	*Breast tissue*	1-4	37	14
6	*Breast tissue*	2-3	33	9
7	*Breast tissue*	2-4	31	3
8	*Breast tissue*	3-4	34	6
9	*Cardiotocography*	1-2	1950	70
10	*Cardiotocography*	1-3	1831	81
11	*Cardiotocography*	2-3	471	26
12	*EEG eye state*	1-2	8388	17
13	*Fertility*	1-2	100	76
14	*Glass*	1-2	146	4
15	*Glass*	1-6	99	41
16	*Glass*	2-6	105	45
17	*Indian liver patient*	1-2	579	43
18	*Ionosphere*	1-2	351	28
19	*Iris*	1-2	100	0
20	*Iris*	1-3	100	0
21	*Iris*	2-3	100	0
22	*Parkinsons*	1-2	195	51
23	*Pima indians diabetes*	1-2	768	30
24	*Red wine*	5-6	1319	3
25	*Retinopathy*	1-2	1151	6
26	*Seeds*	1-2	140	0
27	*Seeds*	1-3	140	0
28	*Seeds*	2-3	140	0
29	*Sonarall*	1-2	208	7
30	*Tae*	1-2	99	1
31	*Tae*	1-3	101	3
32	*Tae*	2-3	102	2
33	*Vertebral column 2C*	1-2	310	35
34	*Vertebral column 3C*	1-2	210	43
35	*Vertebral column 3C*	1-3	160	25
36	*Vertebral column 3C*	2-3	250	20
37	*White wine*	5-6	3655	20
38	*Wine*	1-2	130	9
39	*Wine*	1-3	107	10
40	*Zoo*	1-2	61	34

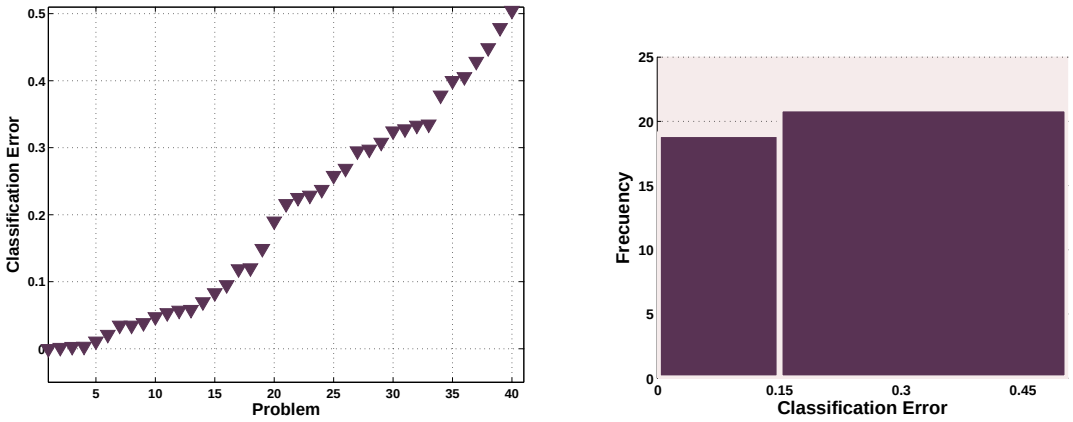


Figure 3.10 – Performance of PGPC on the 40 real-world classification problems; where: (a) shows the CEμ for each problem; and (b) shows the histogram over CEμ.

Table 3.7 – Prediction performance of the evolved PEPs applied on the real-world problems using each feature set (4F, 5F and 7F, see Table 3.3). Performance is given based on the RMSE and Pearson's correlation coefficient, with bold indicating the best performance.

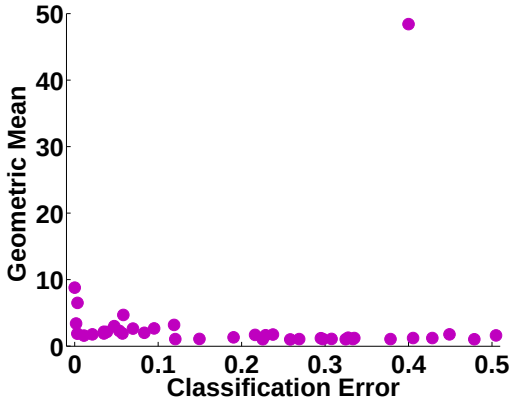
	median RMSE	***best*** RMSE	***best*** correlation
PEP-4F	**0.1522**	**0.0828**	0.8634
PEP-5F	0.1583	0.0929	**0.8823**
PEP-7F	0.1676	0.0930	0.8046

of each descriptive feature (x-axis) and the CEμ (y-axis) of each problem, specifying the corresponding Pearson's correlation coefficient ρ in the legend of each plot. The figures show that the best correlated features with CEμ are FE and CD-1, respectively with ρ values of -0.73 and -0.71. The rest of the features do not show particularly good correlation values, with SD clearly being the worst.

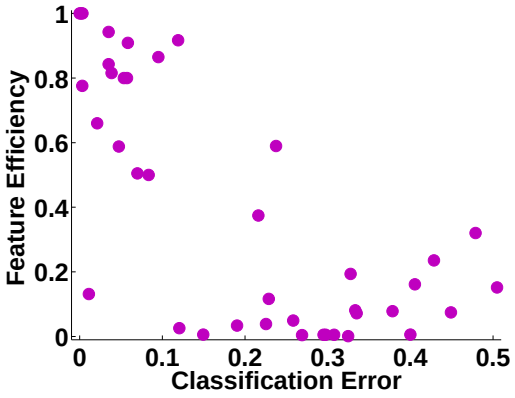
These results are different to what was observed on the synthetic problems. While VOR, CDR and CD-3 showed absolute correlation values above 0.6 on synthetic datasets, they were all below 0.44 on the real-world problems. This difference was particularly marked for SD, on synthetic problems the correlation coefficient was -0.42 but on real-world problems it is $\rho = 0.09$. In fact, only FE and CD-1 showed a good correlation on both sets.

Table 3.7 summarizes the performance of the evolved PEPs applied on the real-world problems, showing the median of the RMSE of the best solution found, as well as the RMSE and Pearson's correlation coefficient ρ of the best solution. The table presents three rows of results, one for each feature set (PEP-4F, PEP-5F and PEP-7F). In this case, the best performance is achieved by PEP-4F, which was unexpected. However, if we contrast the results with those achieved on the set of synthetic problems, shown in Table 3.4, a performance drop-off is evident, based on both median and best performance.

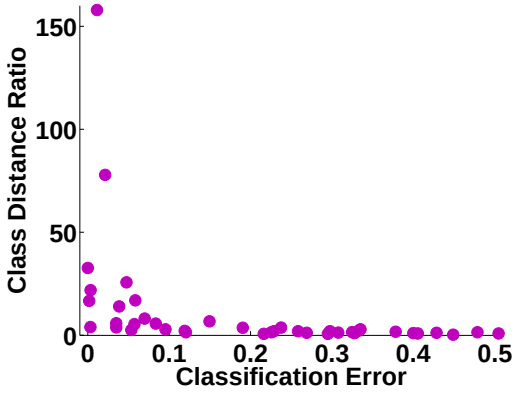
Figure 3.12 shows three rows of plots, one for each feature set (PEP-4F, PEP-5F and PEP-7F). The figures on the left column show the PCE of the best PEP model and the true CEμ for all real-world problems, specifying the RMSE. The figures on the right column show the CEμ and PCE as scatter plots, specifying the Pearson's correlation coefficient ρ. Again, these figures reveal that the evolved PEP models provide less accurate prediction on real-world problems.



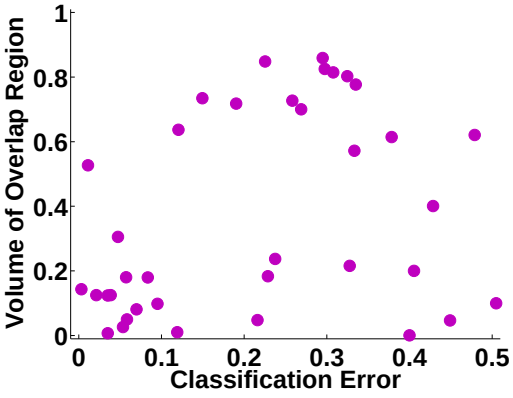
(a) [SD: $\rho = 0.09$]



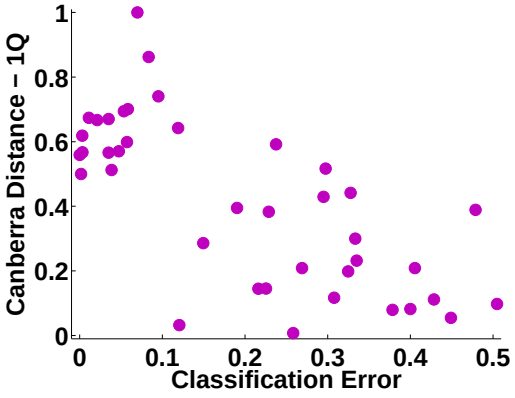
(b) [FE: $\rho = -0.73$]



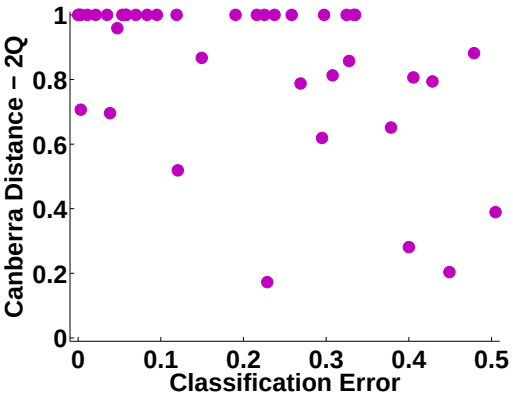
(c) [CDR: $\rho = -0.40$]



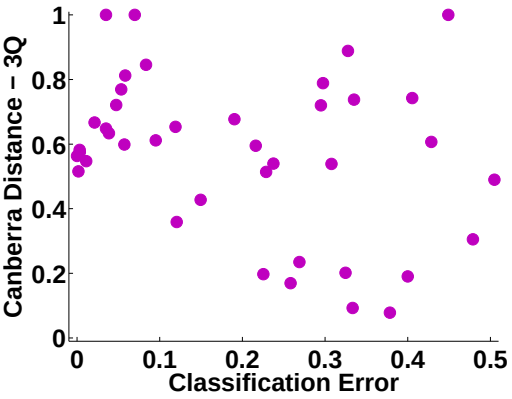
(d) [VOR: $\rho = 0.43$]



(e) [CD-1: $\rho = -0.71$]

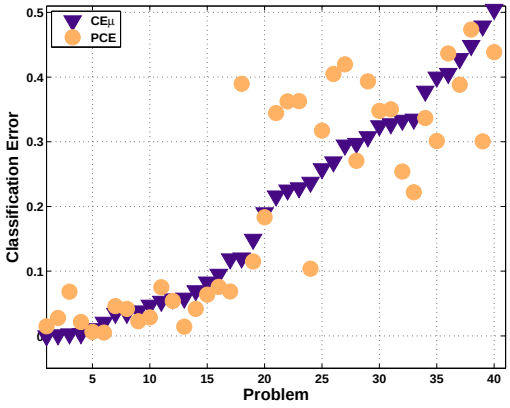


(f) [CD-2: $\rho = -0.46$]

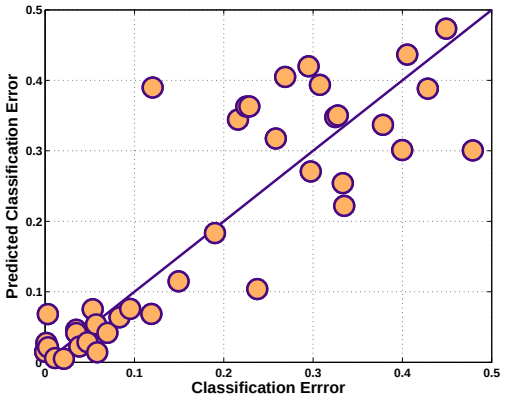


(g) [CD-3: $\rho = -0.30$]

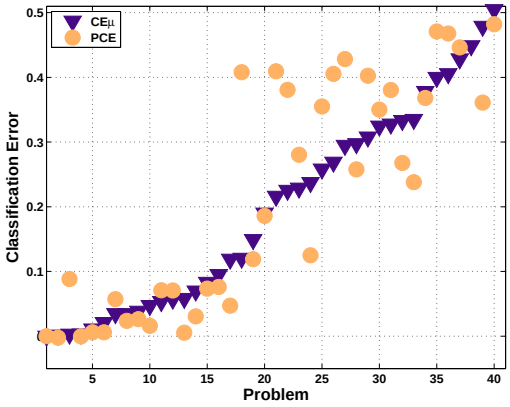
Figure 3.11 – Scatter plots showing for the real-world problems the relationship between the CEμ (x-axis) and each descriptive feature (y-axis). The legend specifies Pearson's correlation coefficient ρ.



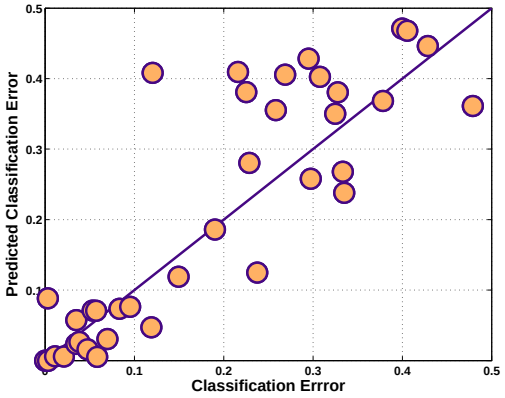
(a) [*PEP-4F:* RMSE = 0.0828]



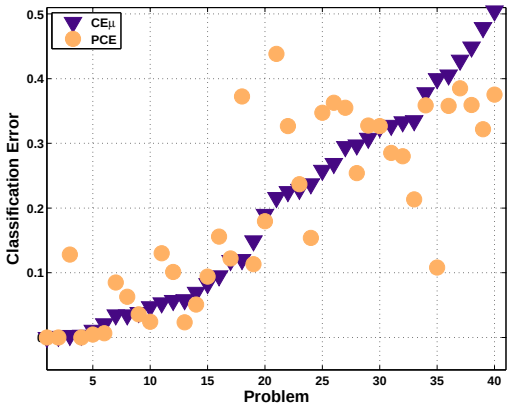
(b) [*PEP-4F:* $\rho = 0.8634$]



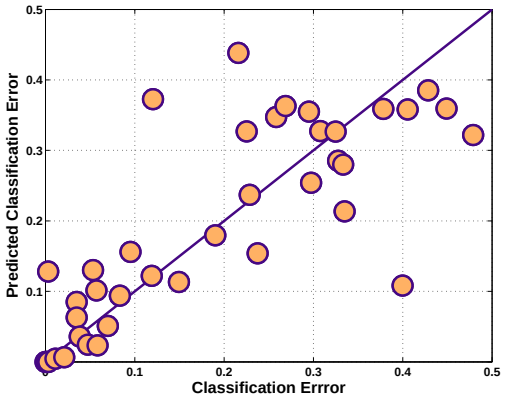
(c) [*PEP-5F:* RMSE = 0.0929]



(d) [*PEP-5F:* $\rho = 0.8823$]



(e) [*PEP-7F:* RMSE =0.0930]



(f) [*PEP-7F:* $\rho = 0.8046$]

Figure 3.12 – Figures showing for real-world problems, the performance prediction of the best PEP models evolved with the different feature set, each row belongs to each feature set: PEP-4F (top), PEP-5F (middle) and PEP-7F (bottom). The plots on the left column show the PCE of the best solution and the know CEμ, specifying the corresponding RMSE. The right column shows scatter plots of the PCE and the CEμ, specifying Pearson's correlation coefficient ρ.

3.5 SPEP: Specialist Predictors of Expected Performance

The above results are encouraging, but for a real-world application even small improvements in the quality of the predictions could have non-negligible effects. Therefore, in this section we propose an ensemble approach using several PEP models, each one referred to as an SPEP. We propose an ensemble approach for two main reasons. First, previous works suggest that ensemble-based modeling can improve performance in a variety of scenarios FOLINO et collab. [2010]; ZHOU [2012]. Second, an ensemble approach allows us to obtain two types of predictions, a numerical prediction of the expected performance and a categorical or fuzzy prediction based on the chosen ensemble component used to compute the final prediction. The proposal is depicted in Figure 3.13, an extension of the basic PEP approach shown in Figure 3.1. However, in the SPEP approach before computing the PCE for a given problem, each problem is classified into a specific group using its corresponding feature vector β. Each group is associated to a particular SPEP in the ensemble, hence if a problem is classified into the *i-th* group then the *i-th* SPEP is used to compute the predicted PGPC performance on the test set.

To implement this approach, several design choices must be specified. First, how to define a meaningful grouping of problems. Second, train SPEPs that are specialized for each group in order to build the ensemble. Third, choose the correct SPEP for a particular problem by determining its group membership. Each of these issues are discussed next.

3.5.1 Grouping Problems based on PGPC Performance and Training SPEPs

The proposal is to group problems based on the performance of PGPC given by CEμ. This can be seem as a categorical prediction, where problems are grouped into general groups of different difficulty; e.g. easy and hard problems. In particular, we propose two different groupings based on CEμ, using either two or three groups as shown in Figure 3.14. The groups were chosen in such a way that the number of (synthetic) problems in each group would be approximately the same, in this way posing a balanced classification task for the SPEP approach. Figure 3.14 show the ranges of PGPC performance for each group and the number of synthetic problems (Figure 3.14(a)) and real-world problems (Figure 3.14(b)) that fall within each group. The plots on the top divide the problems into two groups, while the plots on the bottom divide the problems into three. Finally, for clarity, since the two group division requires two SPEPs, we refer to a solution for this task as an Ensemble-2, while a solution for the three group task is referred to as an Ensemble-3.

For each group an SPEP is trained using the same strategy described in the previous section for PEPs. Except that instead of using all of the synthetic problems, each SPEP is trained using the subset of synthetic problems from the corresponding group, as depicted in Figure 3.14. Since we are interested in presenting the best possible prediction of PGPC performance on real-world problems, we must select the best predictive models. Therefore, the testing phase is performed using two subsets of the real-world problems, one for validation and other for testing.

3.5.2 SPEP Selection

As depicted in Figure 3.13, in order to choose an SPEP we must first classify each problem to its corresponding group. This is a straightforward classification task, solved using each problem's feature vector β as the decision variables. Several classification algorithms are tested DUDA et collab. [2000], namely:

1. Euclidean distance classifier (EDC).

2. Mahalanobis distance classifier (MDC).

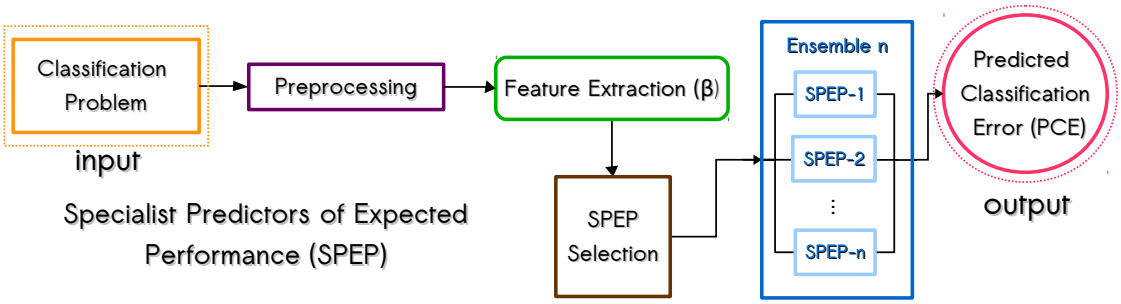


Figure 3.13 – Block diagram showing the proposed SPEP approach. The proposed approach is an extension of the basic PEP approach of Figure 3.1, with the additional ensemble approach, where problems are first classified into prespecified groups and based on this a corresponding specialized model (SPEP) is chosen to compute the PCE of PGPC on the test set.

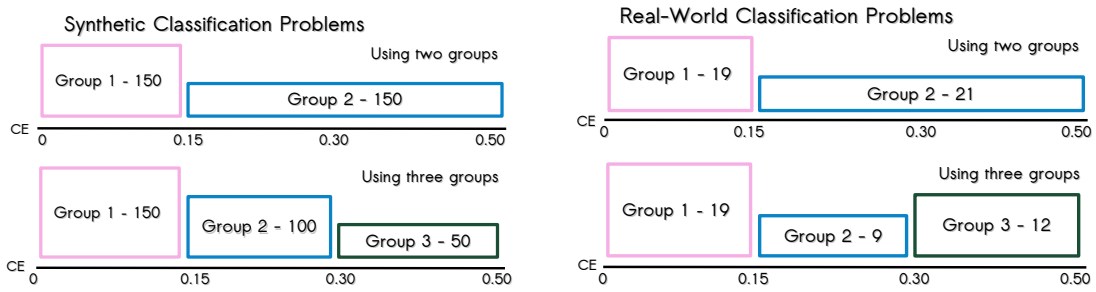


Figure 3.14 – The proposed groupings of classification problems used with the SPEP approach, showing the ranges of PGPC performance and the number of problems in each group.

3. Naive Bayes classifier (NBC).

4. Support Vector Machine (SVM), with Gaussian Radial Basis Function kernel and a default scaling factor of 1.

5. K-Nearest Neighbor (KNN), using K $= 5$ neighbors.

6. Treebagger Classifier (TBC), using 3 trees.

7. Probabilistic Genetic Programming Classifier (PGPC), parameters on Table 3.1.

Moreover, the classification task is posed using different subsets of the features in β as previously described in Table 3.3. We apply all classifiers using all subsets of features on both the two-group and three-group classification tasks.

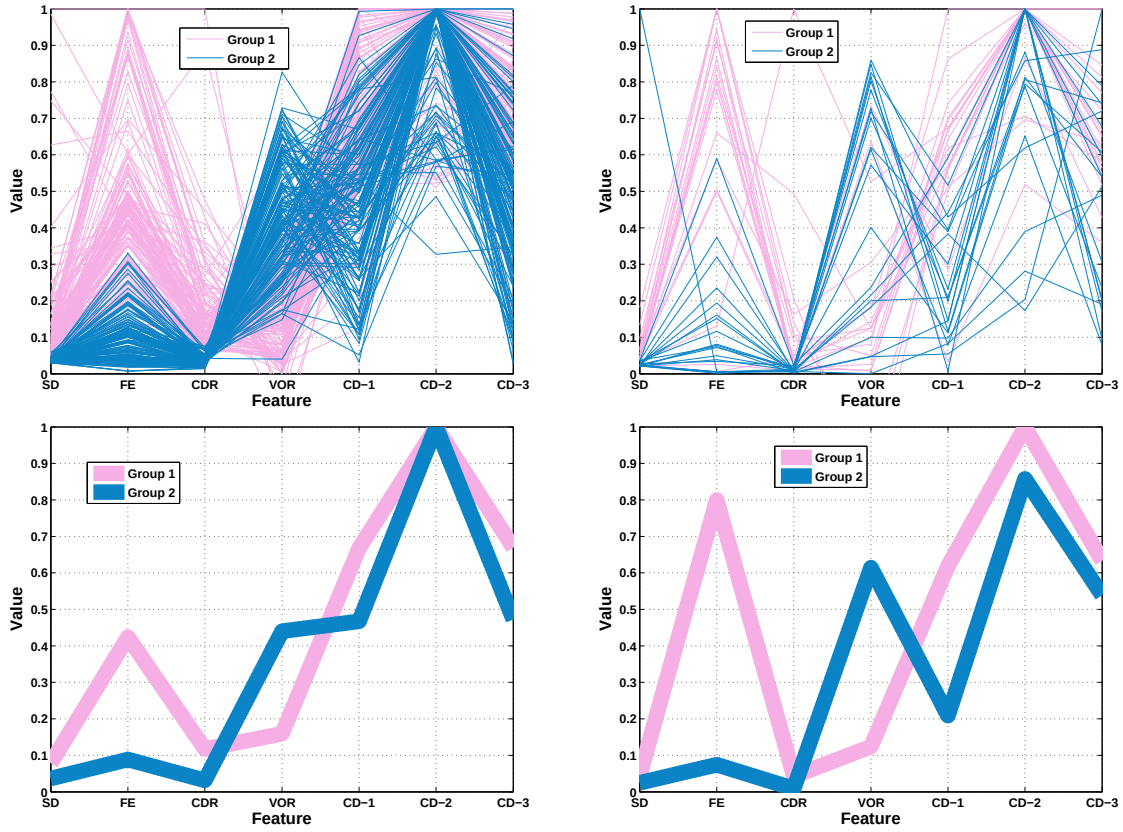
As done for the PEP models, in all cases the complete set of synthetic problems $\mathcal{Q}'$ is used to train the classifiers. The testing phase is performed with two sets, 10% of the real-world problems are used as a validation set while the remaining 90% of real-world problems are used for testing. After performing 100 independent runs, the best solution is chosen based on its validation set performance, and methods are compared based on the performance on the testing set. If several solutions achieve the best validation set performance, then the final solution used in the ensemble is randomly chosen.

3.5.3 Evaluation of SPEP Ensembles

This section presents the performance of the evolved SPEP models, and the performance of the complete ensembles, using both the true problem groups (an oracle approach, where the correct SPEP is always chosen) and the predicted group by the best classifier (a more realistic testing scenario).

3.5.3.1 Ensemble-2 Solutions

To visualize the underlying difficulty of choosing the correct SPEP for a given problem (i.e., determining the group to which it belongs to) Figure 3.15 presents a parallel coordinate plot dividing the problems into two groups, where each coordinate is given by a feature in β. Plots are shown for synthetic (Figure 3.15(a)) and real-world problems (Figure 3.15(b)). The plots on the top show a single line for each problem, while the plots at the bottom show the median values for each group. For clarity in the parallel plots, the features SD and CDR were rescaled to values between $[0, 1]$.



(a) [Synthetic Problems] (b) [Real-World Problems]

Figure 3.15 – Parallel coordinate plots dividing the problems into two groups, where each coordinate is given by a feature in β. Plots are shown for synthetic (a) and real-world problems (b). The plots on the top show a single line for each problem, while the plots at the bottom show the median values for each group.

The table 3.8 summarizes the performance of the best SPEP models used to build the Ensemble-2 solution. The first columns specifies the feature subset used from β. The second column specifies the evaluated SPEP, SPEP-1 was trained with synthetic problems from the first group while SPEP-2 was trained with problems from the second group. The training RMSE is given in column 3. Every SPEP was tested on real-world problems from both groups, to illustrate the performance difference and specialization of each model; this is specified in the forth column. The final column gives the testing RMSE on each group.

The results show that the SPEP models are specialized to their groups, achieving error values below 0.1 when tested using problems from their groups, while performing worse when tested on problems from the other group. In general, performance on testing set is good, particularly if we compare with the results achieved by the PEP models from the preceding section. Finally, performance is similar for all feature sets when considering testing performance, with the best performance on Group 1 achieved by using the set 4F and the best performance on Group 2 with set 5F.

Table 3.8 – RMSE of the best evolved SPEP models, using different feature sets (first column). Performance is given based on training and testing set. Moreover, each SPEP$-i$ corresponds to the $i-th$ problem group but is tested on both problem groups, as specified in the fourth column. Bold indicates the best performance on each group.

	SPEP	*training*	*Testing group*	*testing*
4F	*SPEP-1*	0.0201	1	**0.0315**
			2	0.2470
	SPEP-2	0.0341	1	0.1445
			2	0.0919
5F	*SPEP-1*	0.0195	1	0.0380
			2	0.1819
	SPEP-2	0.0380	1	0.1119
			2	**0.0832**
7F	*SPEP-1*	0.0212	1	0.0469
			2	0.2096
	SPEP-2	0.0332	1	0.1586
			2	0.1014

Table 3.9 – Performance on the SPEP selection problem for all tested classifiers, showing the median classification error from 100 independent runs. The performance is given on the training and testing sets. Bold text indicates the best performance on each feature set.

Algorithm		EDC	MDC	NBC	SVM	KNN	TBC	PGPC
4F	*training*	0.1533	0.0567	0.0200	0.0233	0.0133	0.0067	0.0100
	testing	0.2500	0.1389	0.1111	0.1111	0.1389	0.1111	**0.0833**
5F	*training*	0.1533	0.0567	0.0200	0.0200	0.0200	0.0067	0.0100
	testing	0.2778	0.1389	0.1389	0.1389	0.1667	0.1389	**0.1111**
7F	*training*	0.1533	0.0467	0.0200	0.0033	0.0200	0.0067	0.0100
	testing	0.2778	0.1389	0.1389	0.2500	0.1667	0.1111	**0.0972**

The results in Table 3.8 represent the best possible performance if the correct problem group is chosen, but also confirm that if the correct group is not chosen than prediction accuracy can decline. Table 3.9 summarizes the performance of all of the tested classifiers for the two-group case, showing the median classification error achieved on the training and testing sets. On these tests, PGPC achieves the best performance based on test error.

Table 3.10 shows the performance of the best classifier obtained from each method and chosen based on the validation set. In this table performance is given using all real-world problems. Again, PGPC clearly outperforms all other variants, with the best performance achieved using feature set 7F with a classification error of 0.0250.

It is now possible to evaluate the performance of the complete Ensemble-2 solutions, using the best evolved SPEPs and the best classifier. These results are summarized in Table 3.11, specifying the RMSE and Pearson's correlation coefficient when evaluated on the synthetic and real-world problem sets. These tests show that the Ensemble-2 solutions can achieve low predictive error and a high correlation with the true PGPC performance, for both synthetic and real-world problems. In particular, using feature set 5F correlation on synthetic problems is close to unity, while performance on the real-world problems show the lowest error and approximately 0.9 correlation.

Focusing on the real-world problems, Figure 3.16 summarize the performance of the Ensemble-2 predictors using each feature set (each row of the figure). The column on the left shows plots of the ground truth CEμ of each problem (triangles) and the Ensemble-2 prediction PCE, specifying the corresponding RMSE. These plots show three types of PCE: (1) correctly classified problems for which the appropriate SPEP was selected (CC-PCE); (2) misclassified problems for which an incorrect SPEP was selected (MC-PCE); and (3) for the misclassified problems the oracle SPEP prediction (O-PCE), which is the PCE produced by the correct SPEP. The column on the right of the Figure 3.16 presents scatter plots of the true

Table 3.10 – Performance on the SPEP selection problem for all tested classifiers, showing the classification error of the best solution found, evaluated over all real-world problems, with bold indicating the best performance on each feature set.

Feature Set	EDC	MDC	NBC	SVM	KNN	TBC	PGPC
4F	0.2500	0.1250	0.1000	0.1000	0.1250	0.1250	**0.0500**
5F	0.2750	0.1250	0.1250	0.1250	0.1500	0.1250	**0.1000**
7F	0.2750	0.1250	0.1250	0.2500	0.1500	0.1250	**0.0250**

Table 3.11 – Ensemble-2 prediction accuracy using each feature set (4F, 5F and 7F), using the best evolved SPEPs and the best classifiers with each feature set. Performance is given based on the RMSE and Pearson's correlation coefficient when evaluated on the synthetic and real-world problem sets; with bold indicating the best performance.

Feature Set	Synthetic		Real-world	
	RMSE	correlation	RMSE	correlation
4F	0.0284	0.9709	0.0818	0.8717
5F	0.0302	**0.9984**	**0.0736**	**0.8981**
7F	**0.0276**	0.9728	0.0897	0.8514

CEμ and the PCE, using the same notation and specifying Pearson's correlation coefficient ρ.

These plots provide a graphical confirmation of the quality of the performance prediction. It is important to highlight the impact of a misclassified problem, shown as a black circle, compared to the prediction on the same problem if the correct SPEP had been chosen (O-PCE). For all problems for which the correct SPEP was chosen the PCE is highly correlated with the ground truth with only marginal differences in most cases.

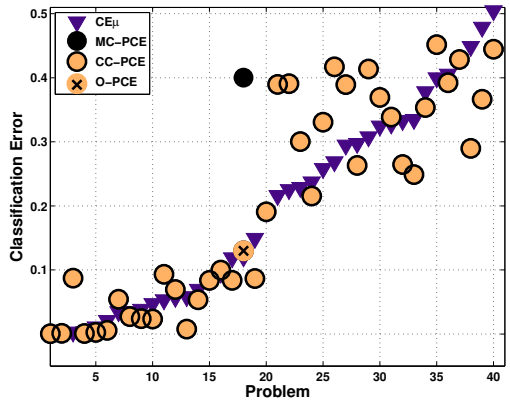
3.5.3.2 Ensemble-3 Solutions

Figure 3.17 presents a parallel coordinate plot dividing the problems into three groups, where each coordinate is given by a feature in β. Plots are shown for synthetic (Figure 3.17(a)) and real-world problems (Figure 3.17(b)). The plots on the top show a single line for each problem, while the plots at the bottom show the median values for each group. For clarity, features SD and CDR were rescaled to values between $[0, 1]$.

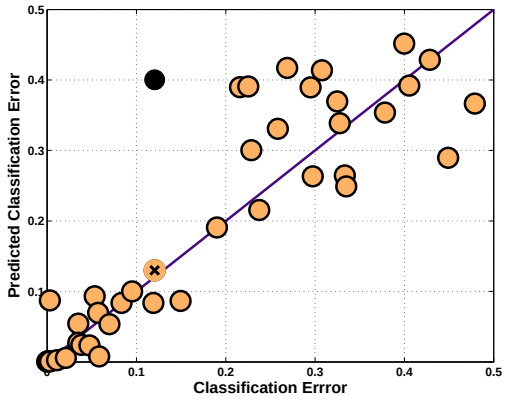
The table 3.12 summarizes the performance of the best SPEP models used to build the Ensemble-3 solution. The first column, from left to right, specifies the feature subset used from β. The second column specifies the evaluated SPEP, SPEP-1 was trained with synthetic problems from the first group, SPEP-2 with problems from the second group and SPEP-3 with problems from the third group. The third column shows the training RMSE, the fourth column shows the testing group and the final columns shows the testing RMSE.

Again, the results show that the SPEP models are specialized to their respective groups. Performance on the testing set is better than the simple PEP models, but worse than the Ensemble-2 solution presented before. All feature sets produce similar performance on testing set problems, with the best performance on Group 1 and Group 2 achieved by using set 4F, and the best performance on Group 3 with set 5F.

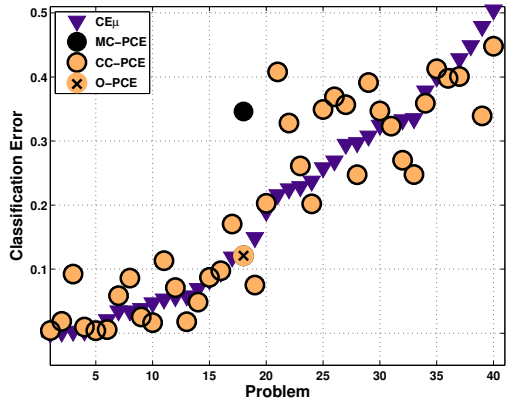
The results summarized in Table 3.12 represent the best possible performance if the correct problem group is chosen, but also confirm that if the correct group is not chosen than prediction accuracy can decline. Table 3.13 summarizes the performance of all of the tested classifiers for the three-group case, showing the median classification error achieved on the training and testing sets. On these tests, TBC achieves the best median performance. Table 3.14 focuses on the performance of the best classifier evaluated over all real-world problems. Again, TBC outperforms all other variants, with the best performance achieved using



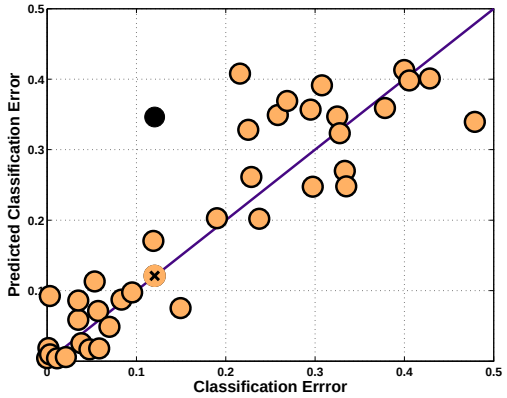
(a) [*4F:* RMSE = 0.0818]



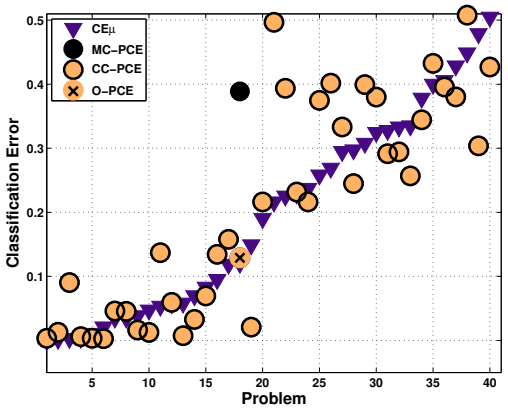
(b) [*4F:* $\rho = 0.8717$]



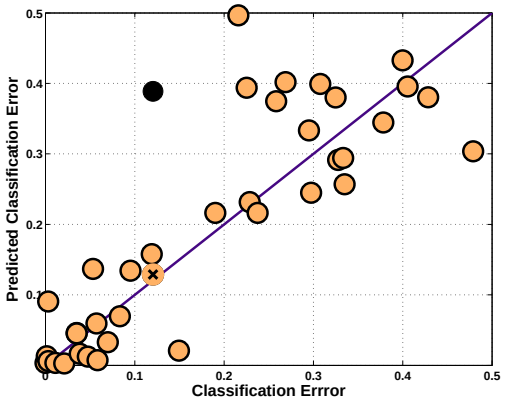
(c) [*5F:* RMSE = 0.0736]



(d) [*5F:* $\rho = 0.8981$]

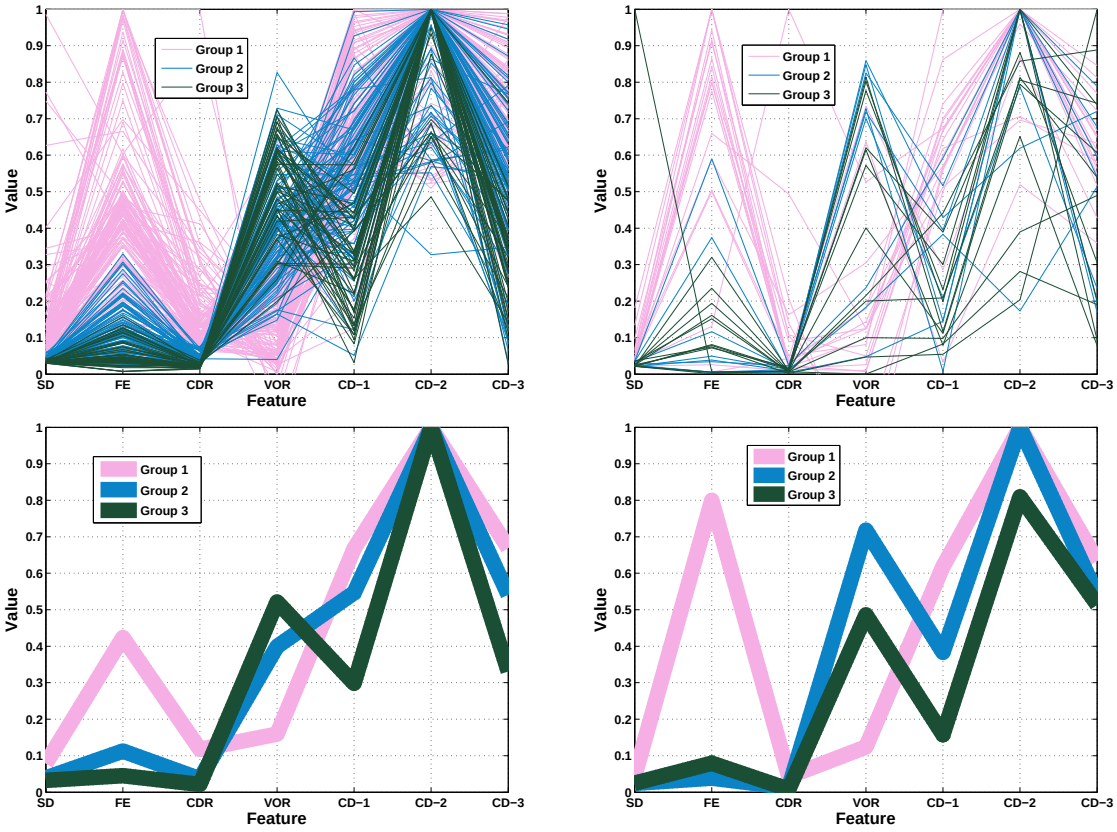


(e) [*7F:* RMSE = 0.0897]



(f) [*7F:* $\rho = 0.8514$]

Figure 3.16 – Performance prediction of the best Ensemble-2 solutions for each feature set: 4F (top), 5F (middle) and 7F (bottom). The left column of plots shows the ground truth CEμ of each problem (triangles) and the corresponding PCE (circles), specifying the RMSE of the ensemble. The right column shows scatter plots between the CEμ and the corresponding PCE, specifying Pearson's correlation coefficient ρ. The PCE is presented in three different cases: (a) the PCE of a correctly classified problem (CC-PCE, circle); (b) the PCE of a misclassified problem (MC-PCE, dark circle); and (c) the oracle PCE of a misclassified problem using the correct SPEP (O-PCE, circle with a cross).



(a) [Synthetic Problems] (b) [Real-World Problems]

Figure 3.17 – Parallel coordinate plots dividing the problems into three groups, where each coordinate is given by a feature in β. Plots are shown for synthetic (a) and real-world problems (b). The plots on the top show a single line for each problem, while the plots at the bottom show the median values for each group.

Table 3.12 – RMSE of the best evolved SPEP models, using different feature sets (first column). Performance is given based on training and testing set. Moreover, each SPEP$-i$ corresponds to the $i-th$ problem group but is tested on all problem groups, as specified in column 4. Bold text indicates best performance on each group.

	SPEP	*training*	*Testing group*	*testing*
4F	*SPEP3-1*	0.0201	1	**0.0315**
			2	0.1312
			3	0.2767
	SPEP3-2	0.0303	1	0.1883
			2	**0.0302**
			3	0.1459
	SPEP3-3	0.0264	1	0.3955
			2	0.1349
			3	0.0532
5F	*SPEP3-1*	0.0195	1	0.0380
			2	0.2076
			3	0.1602
	SPEP3-2	0.0313	1	0.0931
			2	0.0380
			3	0.1245
	SPEP3-3	0.0294	1	0.2691
			2	0.1250
			3	**0.0525**
7F	*SPEP3-1*	0.0212	1	0.0469
			2	0.1723
			3	0.2391
	SPEP3-2	0.0285	1	0.1096
			2	0.0352
			3	0.1719
	SPEP3-3	0.0277	1	0.1339
			2	0.1133
			3	0.0531

Table 3.13 – Performance on the SPEP selection problem for all tested classifiers, showing the median classification error from 100 independent runs. The performance is given on the training and testing sets, with bold indicating the best performance on each feature set.

Algorithm		EDC	MDC	NBC	SVM	KNN	TBC	PGPC
4F	*training*	0.2533	0.1967	0.0833	0.1067	0.0467	0.0167	0.0633
	testing	0.4722	0.3056	0.3889	0.3611	0.3056	**0.2778**	0.3611
5F	*training*	0.2500	0.1933	0.0833	0.1033	0.0533	0.0200	0.0667
	testing	0.5000	**0.3056**	0.4167	0.3611	0.3333	**0.3056**	0.3333
7F	*training*	0.2467	0.1867	0.0800	0.0533	0.0567	0.0167	0.0667
	testing	0.5000	**0.3056**	0.3889	0.4444	0.3333	0.3333	0.3333

feature set 5F with a classification error of 0.1750.

It is now possible to evaluate the performance of the complete Ensemble-3 solutions, using the best evolved SPEPs and the best classifier. These results are summarized in Table 3.15, specifying the RMSE and Pearson's correlation coefficient when evaluated on the synthetic (training) and real-world (validation and testing) problem sets. These tests show that the Ensemble-3 solutions can achieve low predictive error and a high correlation with the true PGPC performance, for both synthetic and real-world problems. In all feature sets the correlation on synthetic problems is above 0.97, while the best performance on the real-world problems is achieved using set 5F based on RMSE and set 7F based on correlation.

Focusing on the real-world problems, Figure 3.18 summarize the performance of the Ensemble-3 predictors using each feature set (each row of the figure). These plots illustrate the performance of the achieved prediction. As in the Ensemble-2 case, it is important to highlight the impact of misclassified problems (shown as a black circle) compared to the

Table 3.14 – Performance on the SPEP selection problem for all tested classifiers, showing the classification error of the best solution found, evaluated over all real-world problems, with bold indicating the best performance on each feature set.

Feature Set	EDC	MDC	NBC	SVM	KNN	TBC	PGPC
4F	0.4750	0.3000	0.4000	0.3500	0.3000	**0.2250**	0.2500
5F	0.5000	0.3000	0.4250	0.3500	0.3250	**0.1750**	0.2500
7F	0.5000	0.3000	0.3750	0.4500	0.3250	**0.2500**	0.3000

Table 3.15 – Ensemble-3 prediction accuracy using each feature set (4F, 5F and 7F), using the best evolved SPEPs and the best classifiers with each feature set. Performance is given based on the RMSE and Pearson's correlation coefficient when evaluated on the synthetic and real-world problem sets; with bold indicating the best performance.

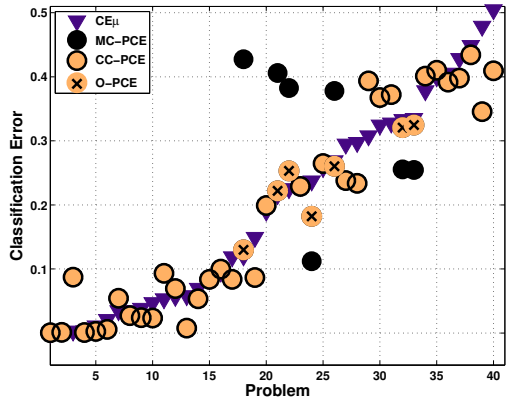
Feature Set	Synthetic		Real-world	
	RMSE	correlation	RMSE	correlation
4F	0.0288	0.9704	0.0808	0.8685
5F	0.0300	0.9687	**0.0775**	0.8707
7F	**0.0285**	**0.9714**	0.0786	**0.8736**

prediction on the same problem if the correct SPEP had been chosen (O-PCE). In this case we can see more misclassifications. The reason is evident in Figure 3.17, since Group 2 and Group 3 are not so easily differentiated. However, the impact of the misclassified problems is not as large as it is for the Ensemble-2 solution, given the comparatively similar RMSE of both the Ensemble-3 and the Ensemble-2 solutions.

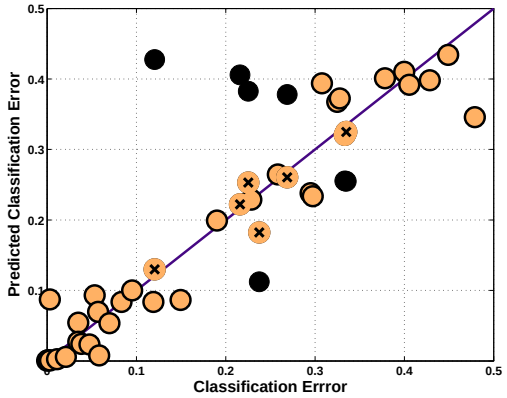
3.5.4 Discussion

This work presents three approaches towards solving the performance prediction problem using the general PEP approach: a single PEP, an Ensemble-2 solution (2 SPEPs) and an Ensemble-3 solution (3 SPEPs). Table 3.16 presents a comparison of the best results of each solution evaluated on the real-world test cases. While all solutions achieve comparable results, it is clear that the Ensemble-2 solution achieves the lowest RMSE and the highest correlation, particularly when using set 5F. These results provide two important insights. First, that the ensemble approach is justified in this domain, with both ensembles outperforming the single PEP models. Second, that grouping the problem into useful subsets based on performance can be solved using two broad categories, what might be considered as *easy* and *difficult* problems. However, differentiating problems further becomes difficult given the underlying distribution of problems within feature space, as shown in Figure 3.17 and confirmed by the lower performance of the Ensemble-3 solution.

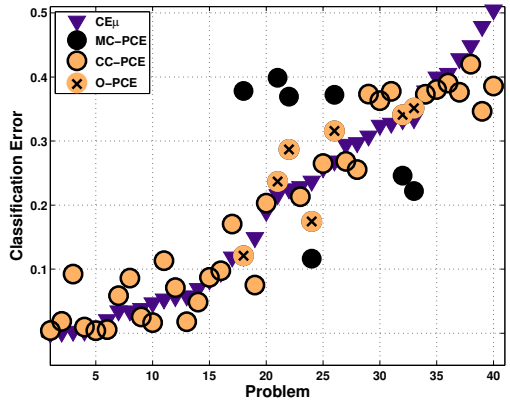
Before concluding lets discuss some additional observations, starting with the relative importance of each feature used to predict performance. Since all PEPs and SPEPs where generated using symbolic regression with GP, we use statistics over the GP runs to measure the importance of each feature. Figure 3.19 shows two plots that quantify the frequency of feature use when the models were evolved using the complete feature set (7F) over 100 independent runs. Figure 3.19(a) is a bar plot where the frequency is given by summing the number of times that each feature appeared as a terminal element in the best symbolic regression solutions from each run. Figure 3.19(b) plots the median of the number of times that each feature appears in the best solution from each run. In this plot each line corresponds to either a single PEP or a particular SPEP from each ensemble; for instance, for the Ensemble-2 solutions there are two SPEPs labeled as Ensemble-2-1 and Ensemble-2-2, and similarly for the Ensemble-3 models. Notice that in this plot the lines for SPEP Ensemble-2-1 and SPEP Ensemble-3-1 overlap since they correspond to the same problem group.



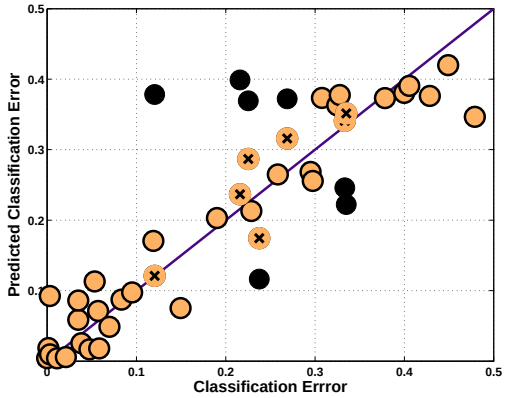
(a) [*4F:* RMSE = 0.0808]



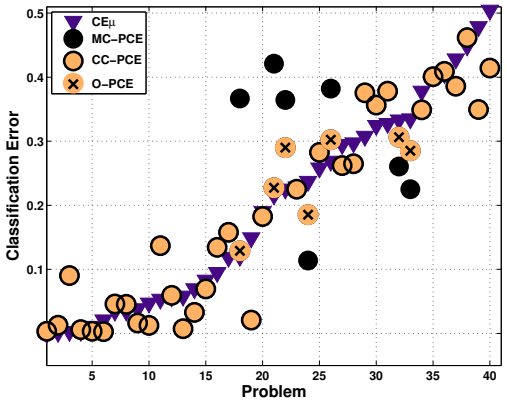
(b) [*4F:* $\rho = 0.8685$]



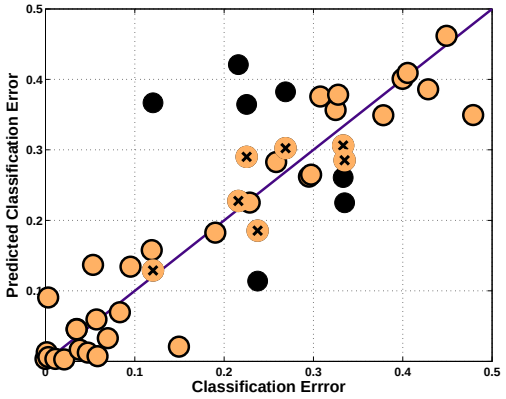
(c) [*5F:* RMSE = 0.0775]



(d) [*5F:* $\rho = 0.8707$]



(e) [*7F:* RMSE = 0.0786]



(f) [*7F:* $\rho = 0.8736$]

Figure 3.18 – Performance prediction of the best Ensemble-3 solutions for each feature set: 4F (top), 5F (middle) and 7F (bottom). The left column of plots shows the ground truth CEμ of each problem (triangles) and the corresponding PCE (circles), specifying the RMSE of the ensemble. The right column shows scatter plots between the CEμ and the corresponding PCE, specifying Pearson's correlation coefficient ρ. The PCE is presented in three different cases: (a) the PCE of a correctly classified problem (CC-PCE, circle); (b) the PCE of a misclassified problem (MC-PCE, dark circle); and (c) the oracle PCE of a misclassified problem using the correct SPEP (O-PCE, circle with a cross).

Table 3.16 – A comparison of each predictor approach; where bold indicates best performance.

	PEP		SPEP Ensemble-2		SPEP Ensemble-3	
	RMSE	correlation	RMSE	correlation	RMSE	correlation
4F	0.0828	0.8634	0.0818	0.8717	0.0808	0.8685
5F	0.0929	0.8823	**0.0736**	**0.8981**	0.0775	0.8707
7F	0.0930	0.8046	0.0897	0.8514	0.0786	0.8736

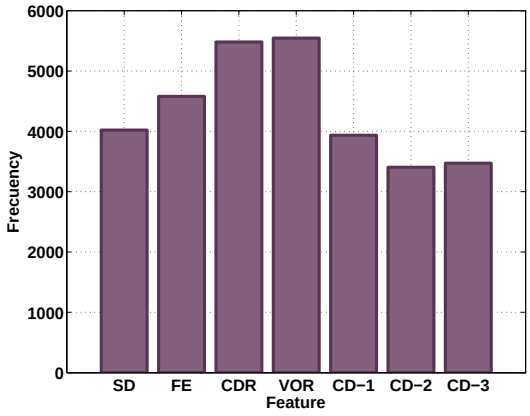
Figure 3.19 reveals some interesting facts of how the symbolic regression system performs feature selection. As shown in Figure 3.7, the feature with higher correlation to PGPC performance are FE, VOR, CDR and CD-1, in that order. However, if we consider all evolved models (Figure 3.19(a)) FE is not the most widely used feature, the evolved models consistently select VOR and CDR at a higher frequency. On the other hand, the less correlated features SD, CD-2 and CD-3 are indeed used less by GP.

If we consider feature frequency in finer detail by comparing the frequency in the PEP models with the frequency in each SPEP, some interesting trends appear, as shown in Figure 3.19(b). In this case it is clear that some features are better predictors of PGPC performance on particular problem groups. For instance, CDR and VOR are the most used by the PEP models. On the other hand, FE is used with a higher frequency when predicting performance on easier problems (Ensemble-2-1, Ensemble-3-1) than for the hardest problems (Ensemble-3-3). This is also the case for CDR and slightly for CD-2. Conversely, while CD-3 is rarely used in PEP models, it appears to be very useful in predicting performance on the most difficult problems (Ensemble-3-3) and the easiest (Ensemble-2-1 and Ensemble-3-1).

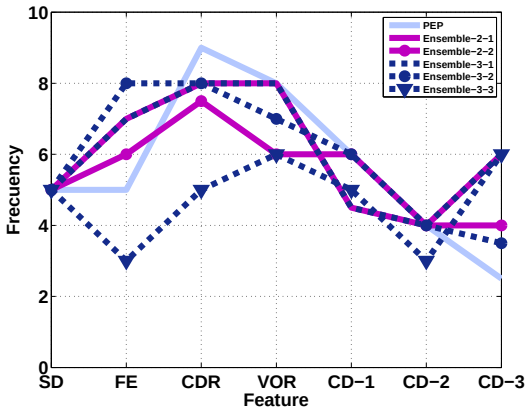
It is also instructive to determine if the dimensionality reduction applied as preprocessing has a negative effect with regards to performance prediction. Our proposal is to use the first two principal components of the data, in order to simplify the description of the real-world problems. However, it is not clear if the percentage of the variance described by such few components is enough to properly characterize the problems. To analyze this, Figure 3.20 presents scatter plots of all the real-world problems $p \in \mathcal{Q}'$, showing the percentage of the total variance of the data explained by the first two principal components (x-axis) and the prediction error (PE) (y-axis) computed as the absolute difference between CEμ and PCE. In particular, Figure 3.20(a) is based on the PEP-4F model while Figure 3.20(b) is based on the SPEP-2-5F model. The legend of each plot specifies the computed Pearson's correlation coefficient ρ between both measures. Notice that there is no significant correlation, suggesting that when the PEP or SPEP models fail at given accurate predictions, it is not due to the proposed preprocessing step.

Finally, an implicit goal of the PEP and SPEP models is to obtain accurate performance predictions in a fraction of the time required to obtain those same estimates by actually performing the GP runs. Pragmatically, one way to validate if this goal is achieved is to calculate the running time for all problems, based on the employed PGPC implementation and the complete SPEP process. These experiments were conducted using MATLAB r2013a and the GPLAB toolbox running on a PC with Ubuntu 12.04 LTS using an Intel RXeon(R) CPU E3-1270 v3 @ 3.50GHz x 8 processor with 15.6 GB of RAM. In these tests, the minimum amount of time required to compute CEμ (30 runs of PGPC) was 3360.96 seconds, while the maximum amount of time required to compute the PCE (running the SPEP process) was 11.22 seconds. These results clearly show that PEP and SPEP models can be used in real-world scenarios to obtain both accurate and efficient estimations of GP performance [2]

[2]It is important to state that our PGPC and SPEP implementations were not implemented in any optimal way, and that running times with other implementations might be substantially different. Nonetheless, we believe that these results give a sufficiently accurate estimate of the possible usefulness of our proposed methodology.

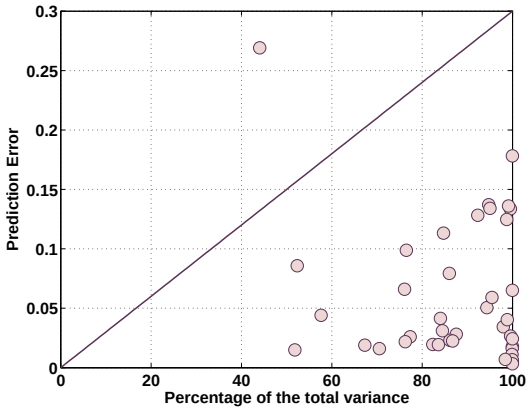


(a) [7F]

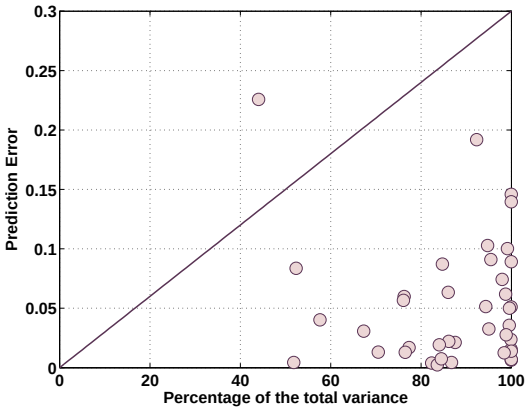


(b) [7F]

Figure 3.19 – Feature selection by the symbolic regression GP used to evolve all PEP and SPEP models, showing usage frequency over 100 runs: (a) bar plot of the total number of times that each feature appeared as a terminal element in the best models; and (b) median of the number of times that each feature appeared in each tree.



(a) [PEP-4F: $\rho = -0.16$]



(b) [SPEP-2-5F: $\rho = -0.08$]

Figure 3.20 – Scatter plots show the relationship between the percentage of the total variance explained by two principal components (x-axis) and the prediction error (y-axis), for all problems $p \in \mathcal{Q}'$, where the prediction error is the absolute difference between the CEμ and PCE, figure on the left show the PEP-4F model and figure on the right SPEP-2-5F, where ρ specifies Pearson's correlation coefficient.

3.6 Conclusions

This work presents three main contributions. First, extensions of the PEP approach originally proposed in TRUJILLO et collab. [2011a,b,c], by adding new descriptive measures and testing the PEP models built with synthetic classification problems over a more challenging scenario, performance prediction on real-world classification problems with different amounts of features and class imbalance. To achieve the latter we included a preprocessing step for dimensionally reduction, something that previous proposals lacked. Second, the proposed models predict the performance of the GP classifier when they are evaluated on the test set of fitness cases, while previous works focused on predicting training performance. For real-world scenarios, predicting the test performance of a learning algorithm is more relevant since overfitting can appear on difficult problem instances. Third, this work presents a new proposal using an ensemble of SPEPs, where the problems are separated into groups and specialized models were built for each group, improving the prediction accuracy on unseen real-world problems.

The main conclusions derived from this work are the following. First, the proposed dimensionality reduction was successful, it allowed us to learn the predictive models using simple 2D synthetic problems and apply them on real-world problems with considerably more dimensions. Second, the evolved PEP and SPEP models were able to accurately predict PGPC performance on imbalanced datasets, without the need of using imbalanced data during the training phase. Third, the new descriptive measures proposed in this work (CD-1, CD-2, CD-3) complemented the problem descriptors used in previous works to help improve predictive accuracy. Some of the proposed features (CD-1) were among the most correlated with PGPC performance; their usefulness was confirmed when analyzing the feature selection performed by GP. However, it's important to note that all measures were used in most evolved PEPs, this mean that all measures contribute to performance prediction even if some features exhibited very small amounts of correlation with PGPC performance. Finally, our ensemble proposal provides two general perspectives of the prediction task: categorical and numerical prediction. Where, a categorical prediction is used to select specific SPEPs, while the numerical prediction is given by the chosen SPEP. While not explored in this work, the categorical prediction might be sufficient for some applications, such as in fuzzy inference systems.

Finally, possibles future work derived from this research includes the following. The problem features used in this work produced good results, but defining the optimal set of features is still an open question. We will also use this methodology for many classifiers, deriving one PEP for each classifier thus allowing us to create an expert system for classifier selection. Another possibility is to use the PEPs within a wrapper approach, where the PEP model could be used as a surrogate fitness function for GP-based classifiers. Moreover, these methodologies could be extended to predict the performance of a GP-based symbolic regression system, building PEP models using a set of descriptive measures that can characterize symbolic regression problems accurately. To do so, a proper dimensionality reduction step must be developed.

References

ALTENBERG, L. 1994, *The evolution of evolvability in genetic programming*, MIT Press, Cambridge, MA, USA, p. 47–74. 34

ALTENBERG, L. 1997, «Fitness distance correlation analysis: an instructive counterexample», dans *In Proceedings of the Seventh International Conference on Genetic Algorithms*, édité par T. Back, Morgan Kaufmann, San Francisco, CA, USA, p. 57–64. 33

BENTLEY, P. J. 2000, «"Evolutionary, my dear Watson" Investigating Committee-based Evo-

lution of Fuzzy Rules for the Detection of Suspicious Insurance Claims», dans *Genetic and Evolutionary Computation Conf.(GECCO-2000)*, p. 702–709. 36

CLERGUE, M., P. COLLARD, M. TOMASSINI et L. VANNESCHI. 2002, «Fitness distance correlation and problem difficulty for genetic programming», dans *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, New York, USA, 9-13 July 2002*, p. 724–732. 32, 34

DUDA, R. O., P. E. HART et D. G. STORK. 2000, *Pattern Classification (2Nd Edition)*, Wiley-Interscience, ISBN 0471056693. 40, 52

EIBEN, A. E. et J. E. SMITH. 2003, *Introduction to Evolutionary Computing*, SpringerVerlag, ISBN 3540401849. 32

FOLINO, G., C. PIZZUTI et G. SPEZZANO. 2010, «An ensemble-based evolutionary framework for coping with distributed intrusion detection», *Genetic Programming and Evolvable Machines*, vol. 11, doi:10.1007/s10710-010-9101-6, p. 131–146. 52

GALVAN-LOPEZ, E., S. DIGNUM et R. POLI. 2008, «The effects of constant neutrality on performance and problem hardness in gp», dans *Proceedings of the 11th European conference on Genetic programming*, EuroGP'08, Springer-Verlag, Berlin, Heidelberg, p. 312–324. 32, 34

GALVAN-LOPEZ, E., J. MCDERMOTT, M. O'NEILL et A. BRABAZON. 2010, «Defining locality in genetic programming to predict performance», dans *IEEE Congress on Evolutionary Computation*, p. 1–8. 32, 34

GALVAN-LOPEZ, E., J. MCDERMOTT, M. O'NEILL et A. BRABAZON. 2011, «Defining locality as a problem difficulty measure in genetic programming», *Genetic Programming and Evolvable Machines*, vol. 12, n° 4, p. 365–401. 32

GALVAN-LOPEZ, E. et R. POLI. 2006a, «An empirical investigation of how and why neutrality affects evolutionary search», dans *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, ACM, New York, NY, USA, ISBN 1-59593-186-4, p. 1149–1156. 32

GALVAN-LOPEZ, E. et R. POLI. 2006b, «Some steps towards understanding how neutrality affects evolutionary search», dans *Parallel Problem Solving from Nature - PPSN IX, Lecture Notes in Computer Science*, vol. 4193, édité par T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervos, L. Whitley et X. Yao, Springer Berlin Heidelberg, p. 778–787. 32

GOLDBERG, D. E. 1987, «Simple genetic algorithms and the minimal, deceptive problem», dans *Genetic algorithms and simulated annealing*, édité par L. Davis, Research Notes in Artificial Intelligence, Pitman, London, p. 74–88. 32

GRAFF, M., H. J. ESCALANTE, J. CERDA-JACOBO et A. A. GONZALEZ. 2013a, «Models of performance of time series forecasters», *Neurocomputing*, vol. 122, n° 0, p. 375 – 385, ISSN 0925-2312. Advances in cognitive and ubiquitous computing Selected papers from the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012). 32, 33, 35, 36, 40

GRAFF, M. et R. POLI. 2008, «Practical model of genetic programming's performance on rational symbolic regression problems», dans *EuroGP*, p. 122–133. 32, 33, 35, 40

GRAFF, M. et R. POLI. 2010, «Practical performance models of algorithms in evolutionary program induction and other domains», *Artif. Intell.*, vol. 174, n° 15, p. 1254–1276. 33, 35, 40

GRAFF, M. et R. POLI. 2011, «Performance models for evolutionary program induction algorithms based on problem difficulty indicators», dans *Proceedings of the 14th European conference on Genetic programming*, EuroGP'11, Springer-Verlag, Berlin, Heidelberg, p. 118–129. 33, 35, 40, 43

GRAFF, M., R. POLI et J. J. FLORES. 2013b, «Models of performance of evolutionary program induction algorithms based on indicators of problem difficulty», *Evolutionary Computation*, vol. 21, nº 4, p. 533–560. 33, 35, 36, 40

GUO, H., L. JACK et A. NANDI. 2005, «Feature generation using genetic programming with application to fault classification», *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, nº 1, p. 89–99, ISSN 1083-4419. 36

HE, J., T. CHEN et X. YAO. 2015, «On the easiest and hardest fitness functions», *Evolutionary Computation, IEEE Transactions on*, vol. 19, nº 2, p. 295–305. 32, 33

HENGPRAPROHM, S. et P. CHONGSTITVATANA. 2008, «A genetic programming ensemble approach to cancer microarray data classification», dans *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on*, p. 340–340. 36

HO, T. K. et M. BASU. 2002, «Complexity measures of supervised classification problems», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, nº 3, p. 289–300. xi, 41, 42

IMAMURA, K., T. SOULE, R. HECKENDORN et J. FOSTER. 2003, «Behavioral diversity and a probabilistically optimal gp ensemble», *Genetic Programming and Evolvable Machines*, vol. 4, nº 3, p. 235–253, ISSN 1389-2576. 36

JONES, T. et S. FORREST. 1995, «Fitness distance correlation as a measure of problem difficulty for genetic algorithms», dans *Proceedings of the 6th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 184–192. 34

KAUFFMAN, S. et S. LEVIN. 1987, «Towards a general theory of adaptive walks on rugged landscapes», *Journal of Theoretical Biology*, vol. 128, nº 1, p. 11 – 45. 34

KIMURA, M. 1983, *The neutral theory of molecular evolution*, Cambridge University Press. 32, 34

KINNEAR, K. E. 1994, «Fitness landscapes and difficulty in genetic programming», dans *Proceedings of the First IEEE Conference on Evolutionary Computing*, IEEE Press, Piscataway, NY, p. 142–147. 32, 34

KOTSIANTIS, S. B., I. D. ZAHARAKIS et P. E. PINTELAS. 2006, «Machine learning: A review of classification and combining techniques», *Artif. Intell. Rev.*, vol. 26, nº 3, p. 159–190, ISSN 0269-2821. 36

LANGDON, W. B. et R. POLI. 2002, *Foundations of genetic programming*, Springer. 34, 36

LICHMAN, M. 2013, «UCI machine learning repository», URL `http://archive.ics.uci.edu/ml`. 45

MALAN, K. et A. P. ENGELBRECHT. 2014, «Particle swarm optimisation failure prediction based on fitness landscape characteristics», dans *2014 IEEE Symposium on Swarm Intelligence, SIS 2014, Orlando, FL, USA, December 9-12, 2014*, p. 149–157. 34

MALAN, K. M. et A. P. ENGELBRECHT. 2013, «A survey of techniques for characterising fitness landscapes and some possible ways forward», *Inf. Sci.*, vol. 241, p. 148–163, ISSN 0020-0255. 32, 34

MARTINEZ, Y., L. TRUJILLO, E. GALVAN-LOPEZ et P. LEGRAND. 2012, «A comparison of predictive measures of problem difficulty for classification with genetic programming», dans *ERA 2012*, Tijuana, Mexico. 32, 35, 40, 43

MCCLYMONT, K., D. WALKER et M. DUPENOIS. 2012, «The lay of the land: a brief survey of problem understanding», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, ACM, New York, NY, USA, p. 425–432. 33

MCPHEE, N., B. OHS et T. HUTCHISON. 2008, «Semantic building blocks in genetic programming», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 4971, édité par M. O'Neill, L. Vanneschi, S. Gustafson, A. Esparcia Alcazar, I. De Falco, A. Della Cioppa et E. Tarantino, Springer Berlin Heidelberg, p. 134–145. 34, 35

MICHIE, D., D. J. SPIEGELHALTER, C. C. TAYLOR et J. CAMPBELL, éd.. 1994, *Machine learning, neural and statistical classification*, Ellis Horwood, Upper Saddle River, NJ, USA. 41

MORAGLIO, A., K. KRAWIEC et C. G. JOHNSON. 2012, «Geometric semantic genetic programming», dans *Parallel Problem Solving from Nature - PPSN XII - 12th International Conference, Taormina, Italy, September 1-5, 2012, Proceedings, Part I*, p. 21–31. 34, 35

MUHARRAM, M. et G. SMITH. 2005, «Evolutionary constructive induction», *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, nº 11, p. 1518–1528, ISSN 1041-4347. 36

MUNOZ, L., S. SILVA et L. TRUJILLO. 2015, «M3GP - multiclass classification with GP», dans *Genetic Programming - 18th European Conference, EuroGP 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings*, p. 78–91. 36

O'NEILL, M., L. VANNESCHI, S. GUSTAFSON et W. BANZHAF. 2010, «Open issues in genetic programming», *Genetic Programming and Evolvable Machines*, vol. 11, nº 3-4, p. 339–363. 34

POLI, R. et E. GALVAN-LOPEZ. 2012, «The effects of constant and bit-wise neutrality on problem hardness, fitness distance correlation and phenotypic mutation rates», *Evolutionary Computation, IEEE Transactions on*, vol. 16, nº 2, p. 279–300. 32

POLI, R., M. GRAFF et N. F. MCPHEE. 2009, «Free lunches for function and program induction», dans *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, ACM, New York, NY, USA, p. 183–194. 32

PUNCH, B., D. ZONGKER et E. GOODMAN. 1996, «Advances in genetic programming», chap. The Royal Tree Problem, a Benchmark for Single and Multiple Population Genetic Programming, MIT Press, Cambridge, MA, USA, ISBN 0-262-01158-1, p. 299–316. 34

QING-SHAN, C., GG DE-FU, W. LI-JUN et C. HUO-WANG. 2007, «A modified genetic programming for behavior scoring problem», dans *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, p. 535–539. 36

QUICK, R., V. RAYWARD-SMITH et G. SMITH. 1998, «Fitness distance correlation and ridge functions», dans *Parallel Problem Solving from Nature PPSN V*, *Lecture Notes in Computer Science*, vol. 1498, édité par A. Eiben, T. Back, M. Schoenauer et H.-P. Schwefel, Springer Berlin Heidelberg, p. 77–86. 33

ROTHLAUF, F. 2006, *Representations for Genetic and Evolutionary Algorithms*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, ISBN 354025059X. 32, 34

SHERRAH, J. R., R. E. BOGNER et A. BOUZERDOUM. 1997, «The evolutionary pre-processor: Automatic feature extraction for supervised classification using genetic programming», dans *In Proc. 2nd International Conference on Genetic Programming (GP-97*, Morgan Kaufmann, p. 304–312. 36

SILVA, S. et J. ALMEIDA. 2003, «GPLAB - A Genetic Programming Toolbox for MATLAB», *In Gregersen L (ed), Proceedings of the Nordic MATLAB Conference*, p. 273—278. 38

SILVA, S. et E. COSTA. 2009, «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories», *Genetic Programming and Evolvable Machines*, vol. 10, nº 2, p. 141–179. 34, 38

SMITH, M. et L. BULL. 2005, «Genetic programming with a genetic algorithm for feature construction and selection», *Genetic Programming and Evolvable Machines*, vol. 6, nº 3, p. 265–281, ISSN 1389-2576. 36

SOHN, S. Y. 1999, «Meta analysis of classification algorithms for pattern recognition», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, nº 11, p. 1137–1144. 41

SOTELO, A., E. GUIJARRO, L. TRUJILLO, L. N. CORIA et Y. MARTINEZ. 2013, «Identification of epilepsy stages from ecog using genetic programming classifiers», *Comp. in Bio. and Med.*, vol. 43, nº 11, p. 1713–1723. 36

STADLER, P. 2002, «Fitness landscapes», dans *Biological Evolution and Statistical Physics*, *Lecture Notes in Physics*, vol. 585, édité par M. Lassig et A. Valleriani, Springer Berlin Heidelberg, p. 183–204. 34

TANIGAWA, T. et Q. ZHAO. 2000, «A study on efficient generation of decision trees using genetic programming», dans *Proc. Genetic and Evolutionary Computation Conference (GECCO'2000), Las Vegas*, Morgan Kaufmann, p. 1047–1052. 36

TOMASSINI, M., L. VANNESCHI, P. COLLARD et M. CLERGUE. 2005, «A study of fitness distance correlation as a difficulty measure in genetic programming», *Evol. Comput.*, vol. 13, nº 2, p. 213–239, ISSN 1063-6560. 32, 34

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2011a, «Predicting problem difficulty for genetic programming applied to data classification», dans *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, ACM, New York, NY, USA, p. 1355–1362. 32, 33, 35, 36, 40, 41, 43, 64

TRUJILLO, L., Y. MARTINEZ, E. GALVAN-LOPEZ et P. LEGRAND. 2012, «A comparative study of an evolvability indicator and a predictor of expected performance for genetic programming», dans *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012, Companion Material Proceedings*, p. 1489–1490. 35, 40, 41, 43

TRUJILLO, L., Y. MARTINEZ et P. MELIN. 2011b, «Estimating classifier performance with genetic programming», dans *Proceedings of the 14th European conference on Genetic programming*, EuroGP'11, Springer-Verlag, Berlin, Heidelberg, p. 274–285. 32, 33, 35, 40, 41, 43, 64

TRUJILLO, L., Y. MARTINEZ et P. MELIN. 2011c, «How many neurons?: A genetic programming answer», dans *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '11, ACM, New York, NY, USA, p. 175–176. 33, 35, 40, 41, 43, 64

TSAKONAS, A. 2006, «A comparison of classification accuracy of four genetic programming-evolved intelligent structures», *Information Sciences*, vol. 176, nº 6, p. 691 – 724, ISSN 0020-0255. 36

VANNESCHI, L., M. CASTELLI et L. MANZONI. 2011, «The k landscapes: A tunably difficult benchmark for genetic programming», dans *Proceedings of the 13th Annual Conference on*

Genetic and Evolutionary Computation, GECCO '11, ACM, New York, NY, USA, p. 1467–1474. 32, 34

VANNESCHI, L., M. CASTELLI et S. SILVA. 2014, «A survey of semantic methods in genetic programming», *Genetic Programming and Evolvable Machines*, vol. 15, nº 2, p. 195–214, ISSN 1573-7632. 34

VANNESCHI, L., M. CLERGUE, P. COLLARD, M. TOMASSINI et S. VEREL. 2004, «Fitness clouds and problem hardness in genetic programming», dans *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'04*, p. 690–701. 32, 34

VANNESCHI, L., M. TOMASSINI, P. COLLARD et M. CLERGUE. 2003, «Fitness distance correlation in genetic programming: a constructive counterexample», dans *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2003, 8 - 12 December 2003, Canberra, Australia*, p. 289–296. 33

VANNESCHI, L., M. TOMASSINI, P. COLLARD et S. VEREL. 2006, «Negative slope coefficient: A measure to characterize genetic programming fitness landscapes», dans *Genetic Programming, 9th European Conference, EuroGP 2006, Budapest, Hungary, April 10-12, 2006, Proceedings*, p. 178–189. 32

VANNESCHI, L., M. TOMASSINI, P. COLLARD, S. VÉREL, Y. PIROLA et G. MAURI. 2007, «A comprehensive view of fitness landscapes with neutrality and fitness clouds», dans *Proceedings of the 10th European conference on Genetic programming*, EuroGP'07, Springer-Verlag, Berlin, Heidelberg, p. 241–250. 32

VANNESCHI, L., A. VALSECCHI et R. POLI. 2009, «Limitations of the fitness-proportional negative slope coefficient as a difficulty measure», dans *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, ACM, New York, NY, USA, ISBN 978-1-60558-325-9, p. 1877–1878. 33

VEREL, S., P. COLLARD et M. CLERGUE. 2003, «Where are bottlenecks in NK fitness landscapes?», dans *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2003, 8 - 12 December 2003, Canberra, Australia*, p. 273–280. 32, 34

WOLPERT, D. et W. MACREADY. 1997, «No free lunch theorems for optimization», *IEEE Transactions on Evolutionary Computation*, vol. 1, nº 1, p. 67–82. 32

WRIGHT, S. 1932, «The roles of mutation, inbreeding, crossbreeding and selection in evolution», *Proceedings of the Sixth International Congress of Genetics*, vol. 1, p. 356–66. 34

YU, T. et J. MILLER. 2001, «Neutrality and the evolvability of boolean function landscape», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 2038, édité par J. Miller, M. Tomassini, P. Lanzi, C. Ryan, A. Tettamanzi et W. Langdon, Springer Berlin Heidelberg, p. 204–217. 32

Z-FLORES, E., L. TRUJILLO, O. SCHÜTZE et P. LEGRAND. 2015, «A local search approach to genetic programming for binary classification», dans *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, GECCO '15, ACM, New York, NY, USA, ISBN 978-1-4503-3472-3, p. 1151–1158. 36

ZHANG, M. et W. SMART. 2004, «Multiclass object classification using genetic programming», dans *Applications of Evolutionary Computing*, *Lecture Notes in Computer Science*, vol. 3005, édité par G. Raidl, S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, C. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. Smith et G. Squillero, Springer Berlin Heidelberg, p. 369–378. 36

ZHANG, M. et W. SMART. 2006, «Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification», *Pattern Recogn. Lett.*, vol. 27, nº 11, p. 1266–1274. 36, 37

ZHOU, Z.-H. 2012, *Ensemble Methods: Foundations and Algorithms*, 1re éd., Chapman & Hall/CRC, ISBN 1439830037, 9781439830031. 52

Chapter 4

A comparison of fitness-case sampling methods for genetic programming

This chapter is related to the PhD thesis of Yuliana Martínez (ITT Tijuana) and has been published in Journal of Experimental & Theoretical Artificial Intelligence, Taylor & Francis, 2017, 29 (6), pp.1203-1224. Work carried out with Yuliana Martínez, Enrique Naredo, Leonardo Trujillo and Uriel López.

Contents

Abstract

Genetic programming (GP) is an evolutionary computation paradigm for automatic program induction. GP has produced impressive results but it still needs to overcome some practical limitations, particularly its high computational cost, overfitting and excessive code growth. Recently, many researchers have proposed fitness-case sampling methods to overcome some of these problems, with mixed results in several limited tests. This chapter presents an extensive comparative study of four fitness-case sampling methods, namely: Interleaved Sampling, Random Interleaved Sampling, Lexicase Selection and Keep-Worst Interleaved Sampling. The algorithms are compared on 11 symbolic regression problems and 11 supervised classification problems, using 10 synthetic benchmarks and 12 real-world datasets. They are evaluated based on test performance, overfitting and average program size, comparing them with a standard GP search. Comparisons are carried out using non-parametric multigroup tests and post hoc pairwise statistical tests. The experimental results suggest that fitness-case sampling methods are particularly useful for difficult real-world symbolic regression problems, improving performance, reducing overfitting and limiting code growth. On the other hand, it seems that fitness-case sampling cannot improve upon GP performance when considering supervised binary classification.

4.1 Introduction

Genetic programming (GP) is an evolutionary computation paradigm that is generally used to solve supervised machine learning problems [KOZA, 2010]. A distinctive feature of GP is that it searches for symbolic expressions that best describes the relationship between a set of training input/output pairs, which in GP literature are referred to as fitness-cases. Normally, a GP algorithm uses the entire training set to compute the fitness of each individual in the evolving population, a common strategy for most supervised learning approaches. However, recent works have reported improved performance when the entire training set is not used, and a different subset of fitness-cases is used at each generation instead [DOUCETTE et HEYWOOD, 2008; GATHERCOLE et ROSS, 1994a,b; GIACOBINI et collab., 2002; GONÇALVES et SILVA, 2013; LASARCZYK et collab., 2004; MARTÍNEZ et collab., 2014]. Moreover, while each proposal was motivated by different goals and assumptions, all works produced quite similar algorithms, sharing the same high level strategy of dynamically focusing on a different subset of fitness-cases to determine fitness as the evolutionary search progresses. Hence, we will refer to these algorithms as fitness-case sampling[1] methods. This chapter presents a comprehensive evaluation of different fitness-case sampling algorithms using a large set of benchmark and real-world problems, to determine what improvements they are able to provide over the standard GP approach. This work is an extension of the previous findings reported in [MARTÍNEZ et collab., 2014], substantially extending the experimental evaluation and providing strict statistical comparisons. The goal is to provide insights regarding the performance of these techniques on common machine learning problems, particularly symbolic regression and supervised classification, measuring test performance, overfitting and bloat.

The remainder of this chapter is organized as follows. Section 4.2 reviews the fitness-case sampling techniques evaluated in this chapter. The experimental work is presented in Section 4.3 where the main results are discussed and analyzed. Finally, concluding remarks and future work are outlined in Section 4.4.

4.2 Previous Work

GP is widely used to generate mathematical functions, or operators, that solve symbolic regression and classification problems, which can be stated as follows. The goal is to search for the symbolic expression $K : \mathbb{R}^p \rightarrow \mathbb{R}$ that best fits a particular training set $\mathbb{T} = \{I_1, I_2, \ldots, I_n\}$, where $I_i = (x_i, y_i)$ of n input-output pairs with $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$ for symbolic regression, while $y_i \in \{w_1, ..., w_m\}$ for a classification problem with m classes (each w_i represents a class label), stated as

$$K \leftarrow \underset{K \in \mathbb{G}}{arg\,min}\ f(K(x_i), y_i)\ with\ i = 1, \ldots, n\ , \tag{4.1}$$

where $\mathbb{G}$ is the solution or syntactic space defined by the primitive set $\mathbb{P}$ of functions and terminals, f is the fitness function which is based on the difference between a program's output $K(x_i)$ and the desired output y_i. Each input-output pair I_i is referred to as a fitness-case.

As stated above, GP traditionally uses the entire set of fitness-cases $\mathbb{T}$ to determine a program's fitness. Recent works, however, have focused on evolving solutions by only using a subset of fitness-cases $\mathbb{S}_g \subseteq \mathbb{T}$ to determine the fitness of each solution candidate K at a given generation g; in some cases reducing the number of considered fitness-cases to a single one. This, of course, can produce one obvious benefit, a reduction in computational cost during fitness evaluation, which is usually the main computational bottleneck in a GP system. However, the motivation for most approaches has been varied. For instance, some

[1] Fitness-case sampling refers to the selection of a subset of fitness-cases, which is mostly the case for the methods presented in this study. Though, there is the possibility to consider the entire training set.

methods focus on reducing computational costs during fitness evaluation [GATHERCOLE et ROSS, 1994a,b] or reducing overfitting and improving generalization [GONÇALVES et SILVA, 2013], others have addressed the issue of problem modality [SPECTOR, 2012] or attempted to promote novel solutions that can solve every fitness-case [MARTÍNEZ et collab., 2013], particularly the most difficult ones [GATHERCOLE et ROSS, 1994a,b; MARTÍNEZ et collab., 2014]. In what follows, we review these methods and describe their main details.

4.2.1 Dynamic Training Subset Selection

Dynamic Training Subset Selection (DTSS) was proposed in [GATHERCOLE et ROSS, 1994a,b], and probably should be considered as the first fitness-case sampling method proposed in GP literature. In their original work, GATHERCOLE et ROSS [1994a,b] developed their method for classification problems, outperforming the basic GP approach. In each generation g, DTSS performs two passes through the entire training set of fitness-cases. It assigns a weight w_i to each fitness-case I_i computed by

$$w_i(g) = D_i(g)^d + A_i(g)^a \tag{4.2}$$

where D is a function that measures the difficulty of I_i, A is an age factor that measures the number of generations g since I_i was last selected (D and A are reset to zero once I_i is selected), while a and d are parameters that need to be set by the user. Afterward, a total of S fitness cases are selected using a probability that is proportional to their associated weight, given by $P = \frac{w_i}{\sum_{j=1}^{n} w_j}$ with n the total number of fitness-cases. In their original work, D_i represented the total number of times that a particular fitness-case was misclassified by the individuals in the population. While the authors reported strong results on some tests, DTSS has not been extensively benchmarked on several problems. Moreover, there are several shortcomings with DTSS. In particular, it requires several parameters to be tuned; the difficulty score is not trivially extrapolated to other problems, particularly symbolic regression; and, it is more computationally costly than standard GP since at every generation the algorithm requires two passes over the entire training set of fitness-cases.

4.2.2 Interleaved Sampling and related methods

Interleaved Sampling (IS) [GONÇALVES et SILVA, 2011] is a deterministic-based sampling method, which uses the entire training set to compute fitness in some generations and uses a single fitness-case in others. This approach was motivated by the idea of balancing learning and overfitting through the interleaving of fitness-cases, attempting to elude local optima. Determining in which generation to use a single fitness-case and in which one to use all of them, is an integral part of the algorithm design. In [GONÇALVES et SILVA, 2013], the authors present two variants that achieved the best results, IS and Random IS (RIS). IS uses the entire training set in odd numbered generations, and uses a single randomly chosen fitness-case on even numbered generations; see Algorithm 1. RIS, on the other hand, uses a probabilistic decision at the beginning of each generation to determine if all of the fitness-cases are used or a single randomly chosen fitness-case; see Algorithm 2. In [GONÇALVES et SILVA, 2013], RIS exhibited the best performance with the probability of using a single fitness-case set to $\delta = 0.75$. These methods are related to the approach presented by LASARCZYK et collab. [2004] that also selects a different subset of fitness-cases at each generation.

4.2.3 Lexicase Selection

SPECTOR [2012] proposed the concept of modality to describe problems for which an optimal solution must exhibit different modes of operations; i.e., solutions must exhibit distinct behaviors based on contextual information that is provided implicitly by the input data

Algorithm 1 Interleaved Sampling (IS)

Deterministically interleaving between using a single fitness case or the entire training set at each generation:

(1) Initialize:

  (a) Entire training set $\mathbb{T} = \{I_1, I_2 ... I_n\}$.

(2) First generation:

  (a) Evaluate population using $\mathbb{T}$.

(3) Loop on the remaining generations:

  (a) Odd generations:
    - Generate an integer random number r, where $r \in [1, n]$.
    - Evaluate population using a single fitness case I_r.

  (b) Even generations:
    - Evaluate population using the entire training set $\mathbb{T}$.

Algorithm 2 Random Interleaved Sampling (RIS)

Stochastically interleaving between using a single fitness case or the entire training set at each generation:

(1) Initialize:

  (a) Entire training set $\mathbb{T} = \{I_1, I_2 ... I_n\}$.

  (b) Interleave probability $\delta \in [0, 1]$.

(2) Loop for every generation:

  (a) Generate a real random number $\phi \in [0, 1]$.

  (b) If $\phi \leq \delta$ then
    - Generate an integer random number r, where $r \in [1, n]$.
    - Evaluate population using a single fitness case I_r.

  (c) Otherwise
    - Evaluate population using the entire training set $\mathbb{T}$.

[TRUJILLO et collab., 2013]. SPECTOR [2012] points out that, in general, GP practitioners usually address problems with a static environment (unimodal problems), hence, GP evolves solution programs which usually perform similar actions for all possible inputs. However, real-world problems will normally require more complex solutions, that change their behavior depending on context.

To solve such problems, SPECTOR [2012] presents the Lexicase Selection (LEX) method for parent selection, which allows each fitness-case to possibly be the main source of selective pressure at any given parent selection event. Traditionally parent selection methods consider fitness as the performance of each individual over the entire training set, in other words, fitness in the traditional approach is a measure which summarize the performance of each individual on the entire training set. The goal of these traditional methods is to find

Algorithm 3 Lexicase Selection (LEX)

For parent selection:

(1) Initialize:

  (a) Set **parent candidates** to be the entire population $\mathbb{K} = \{K_1, K_2...K_m\}$.

  (b) Set **fitness cases** to be the entire training set $\mathbb{T} = \{I_1, I_2, ..., I_n\}$ randomly ordered.

(2) Loop:

  (a) Set **parent candidates** to be the subset of the current candidates that have exactly the best fitness of any individual currently in **parent candidates** for the first case in **fitness cases**.

  (b) If **parent candidates** or **fitness cases** contains just a single element then return the first individual in **parent candidates**.

  (c) Otherwise remove the first case from cases and go to Loop.

general solutions. LEX, on the other hand, focuses on individual fitness-cases and the strategy is to choose specialized parents which can pass on these specialized abilities to their offspring, and at the end of the search construct a general solution. During evolution, LEX selects parents by starting with a pool of candidate parents, and removing candidates based on the performance achieved on a single fitness-case. LEX is elitist, all of the individuals that do not achieve the best performance are removed. This process is repeated using another fitness-case, until only one individual remains, which is then returned as the selected parent. In the basic implementation, the initial pool of candidates is composed by the entire population and the fitness-cases are ordered randomly each time a parent is selected. LEX resembles a lexicographic ordering of a character string, where the first fitness-case has the largest effect in choosing the parent, then the next fitness-case acts as tie-breaker, and so on. This means that each fitness-case has a chance to be the deciding factor in determining which individuals are used to produce offspring at any given parent selection event; see Algorithm 3.

4.2.4 Keep Worst Interleaved Sampling

Based on [GONÇALVES et SILVA, 2013; MARTÍNEZ et collab., 2013] we proposed a new fitness-case sampling method called Keep-Worst Interleaved Sampling (KW-IS) [MARTÍNEZ et collab., 2014]. This method is based on the general methodology of the Novelty Search-based $\epsilon - descriptor$, proposed in [MARTÍNEZ et collab., 2013], but it is also common in other learning paradigms such as AdaBoost, for example, where solution design is adjusted based on the most difficult training data samples. KW-IS is similar to IS, using the entire set of fitness-cases in some generations, just like IS would. However, in the remaining generations, fitness-cases are not chosen randomly. Instead, the goal is to bias selective pressure towards individuals that exhibit good performance on the most difficult fitness-cases. Therefore, the fitness-cases are ordered based on difficulty. Afterwards, the $\rho\%$ most difficult fitness-cases are used to determine fitness in the next generation; see Algorithm 4.

The best performance of this parameter was achieved with $\rho = 90\%$. For symbolic regression problems the difficulty of a single fitness-case is given by the average absolute error of the entire population in a given generation. Where we take the fitness-cases that have a larger error than the error found in the ρ percentile. On the other hand, for classification the difficulty of a fitness-case is computed as done in DTSS, by the total number of individuals in the population that misclassified it. Therefore, in the same way we will take the

Algorithm 4 Keep Worst Interleaved Sampling (KW-IS)

Deterministically interleaving between using a subset of the most difficult or all fitness-cases at each generation:

(1) Initialize:

  (a) Entire training set $\mathbb{T} = \{I_1, I_2 ... I_n\}$.

  (b) Percentile value error $\rho \in [0, 1]$.

(2) First generation:

  (a) Evaluate population using $\mathbb{T}$.

  (b) Construct the subset φ of the *most difficult fitness-cases* using the ρ value as threshold.

(3) Loop on the remaining generations:

  (a) Odd generations:

    - Evaluate population using φ.

  (b) Even generations:

    (a) Evaluate population using $\mathbb{T}$.

    (b) Construct the subset φ of the *most difficult fitness-cases* using the ρ value as threshold.

fitness-cases that have been misclassified by ρ percent of the population.

KW-IS is closely related to DTSS, since it also focuses on reducing the size of the training set used in each generation by concentrating on a small subset of the most difficult fitness-cases. However, KW-IS requires less parameters to tune, it only performs a single pass of the training set $\mathbb{T}$ at each generation g by relying on the estimated difficulty given in generation $g - 1$, and it can be used directly on symbolic regression problems. Therefore, only KW-IS is included in the experimental tests performed in the present work.

4.3 Experiments

As stated before, the goal of this chapter is to provide a comprehensive evaluation of the fitness-case sampling methods described above. In all of the experiments, we use a tree-based GP algorithm with standard subtree crossover and subtree mutation, as originally proposed by KOZA [2010]. With this GP system, we test IS, RIS, LEX and KW-IS on two of the most common problems on which GP is used, symbolic regression and classification, using benchmark and real-world problems. Moreover, a standard GP search (GP-STD) is included as the control method.

None of the algorithms have been extensively studied or compared besides our previous study reported in [MARTÍNEZ et collab., 2014], however that work only performed an informal comparison (without statistical tests) and only considered symbolic regression problems. The algorithms are compared based on the following criteria. First, performance of the best solution found during training evaluated on the test set, a standard evaluation measure. Second is overfitting, here measured by the difference between the training error of the best solutions and its respective test error. Finally, one of the most important open issues with GP is the bloat phenomenon, where the average size of the population increases disproportionately relative to the improvements achieved in terms of fitness [SILVA

Table 4.1 – Five symbolic regression problems, originally published in [UY et collab., 2011], and suggested as benchmark regression problems in [MCDERMOTT et collab., 2012] and [MARTÍNEZ et collab., 2013].

Problem	Function	Fitness/Test cases
f_1-Benchmark	$x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
f_2-Benchmark	$x^5 + x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
f_3-Benchmark	$sin(x^2) * cos(x) - 1$	20 random points $\subseteq [-1, 1]$
f_4-Benchmark	$log(x+1)+log(x^2+1)$	20 random points $\subseteq [0, 2]$
f_5-Benchmark	$2sin(x) * cos(y)$	100 random points $\subseteq [-1, 1] \times [-1, 1]$

et COSTA, 2009]. Therefore, we also compare the methods based on the average size of the individuals in the final population, where size is given by the number of nodes of each tree.

In all problems a total of thirty independent runs are performed, with different and randomly chosen training and testing sets. Results are analyzed using rank statistics and nonparametric tests, since machine learning algorithms do not tend to produce normally distributed samples [DERRAC et collab., 2011; TRAWINSKI et collab., 2012]. Therefore, the Friedman multiple comparison test [FRIEDMAN, 1937] is used to compare all of the algorithms on each problem, and a post-hoc procedure is used to perform pairwise comparisons using the Bonferroni-Holm correction [HOLM, 1979] of the p-value (considering five methods), under the null hypothesis that each pair of samples share equal medians. The results of all tests are presented in tables with the corresponding p-values. Moreover, a summary of the medians of each measure (test fitness, overfitting and average size) are presented, using bold to indicate that a given method achieved the best performance on a given problem. For instance, if the performance of all methods are presented in bold, this means that no statistical difference was detected. Similarly, if one (or several) result(s) is (are) in bold, this means that the method(s) achieved significantly different performance compared to the other (non-bold) methods. Box plots are used to graphically illustrate the behavior of each method. Finally, all algorithms were implemented using the GPLab Matlab toolbox [SILVA et ALMEIDA, 2003] and all statistical test were performed using the Matlab statistical toolbox.

4.3.1 Symbolic Regression Problems

Five benchmark problems are used for symbolic regression, originally published in [UY et collab., 2011], and suggested in [MCDERMOTT et collab., 2012] and [MARTÍNEZ et collab., 2013]; these problems are summarized in Table 4.1. The experimental parameters used with these problems are given in Table 4.2, and fitness is calculated as the root mean square error between predicted and expected outputs specified in the training set.

Moreover, to get a better assessment of each method in more difficult scenarios, six real-world problems are used to evaluate the algorithms; these are: Toxicity, Plasma Protein Binding, Bioavailability, Concrete, Housing and Yacht. The first three problems are described in [GONÇALVES et SILVA, 2013] and the remaining three are from the University of California, Irvine (UCI) machine learning repository [LICHMAN, 2013], which are characterized by a high-dimensionality and a difficult to model behavior. The experimental parameters used are the same as those in [GONÇALVES et SILVA, 2013], summarized in Table 4.3. Fitness is calculated as the root mean square error between predicted and expected outputs, and the data set is randomly divided before each run, using 50% for training and 50% for testing.

For the benchmark problems, the five box plots in Figure 4.1 show the performance of the best individual from each run evaluated with the test set. Figure 4.2 shows the overfitting results for each method, and Figure 4.3 summarizes the effect on bloat for each method, cap-

Table 4.2 – GP parameters used for the benchmark symbolic regression problems.

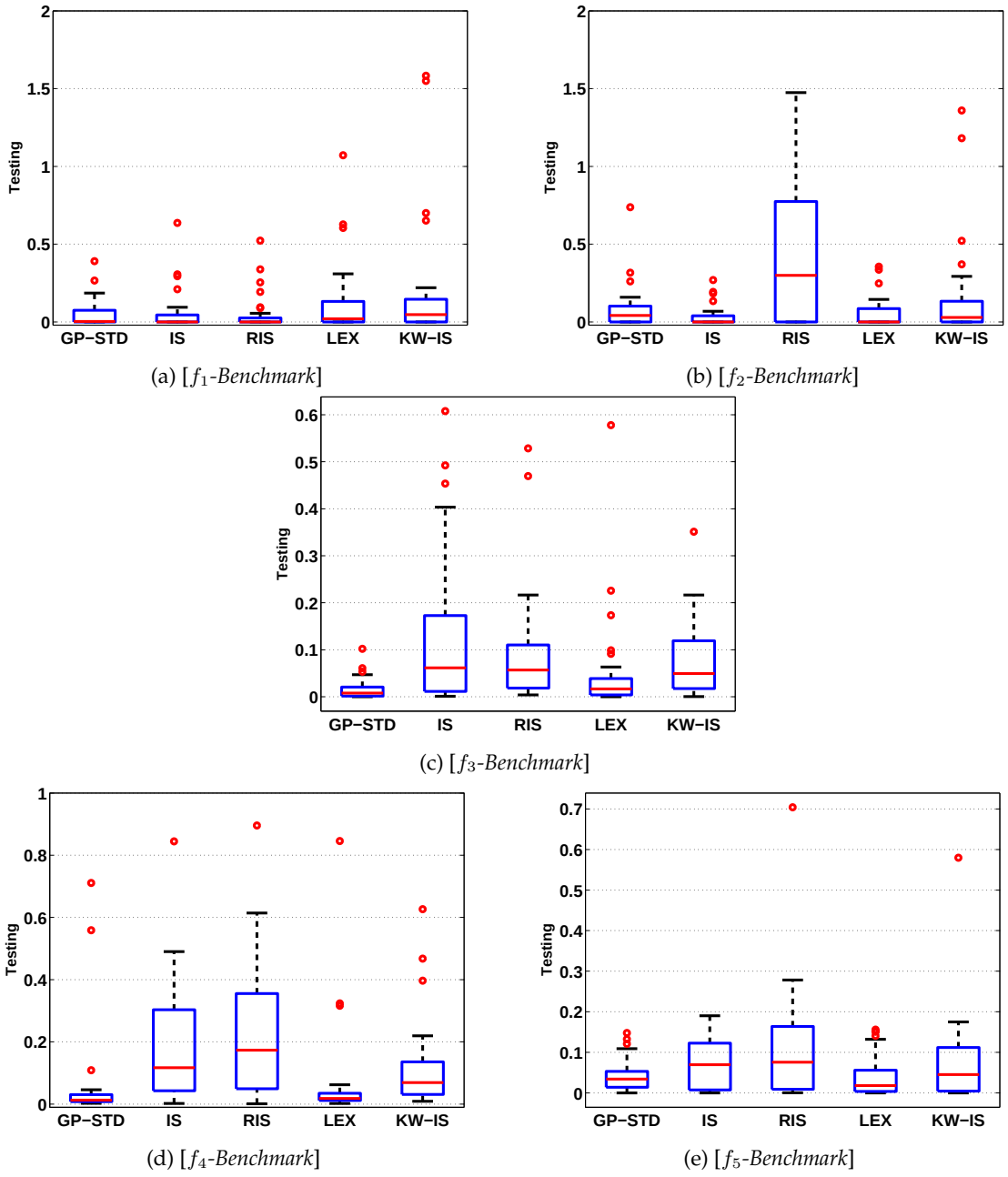
Parameter	**Description**
Population size	200 individuals
Generations	100 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.9$, Mutation $p_\mu = 0.05$
Function set	$(+, -, \times, \div, sin, cos, exp, log)$.
Terminal set	$x, 1$ for single variable problems and x, y for bivariable problem.
Maximum tree depth	20 levels
Selection	Tournament selection of size 3
Elitism	Best individual always survives

Table 4.3 – GP parameters used for symbolic regression real-world problems.

Parameter	**Description**
Population size	500 individuals
Generations	200 generations
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth
Operator probabilities	Crossover $p_c = 0.9$, Mutation $p_\mu = 0.05$
Function set	$\{+, -, \times, \div\}$
Terminal set	Input variables, constants $-1.0, -0.5, 0, 0.5$ and 1.0
Maximum tree depth	17 levels
Selection	Tournament selection of size 10
Elitism	Best individual always survives

tured by the average size of the population in the final generation. Similar plots are shown for the real-world problems in Figures 4.4, 4.5 and 4.6, for test performance, overfitting and average population size, respectively.

To summarize these results, a numerical comparison of the methods is provided in Table 4.4, showing the median values of each method, on each problem and for each measure. Here, bold indicates statistically different results with respect to the other (non-bold) values. Tables 4.5 and 4.6 show the p-values with the Bonferroni-Holm correction for the pairwise comparisons on each problem where bold values indicate that the null hypothesis is rejected at the $\alpha = 0.05$ significance level. The results on symbolic regression reveal several interesting trends.

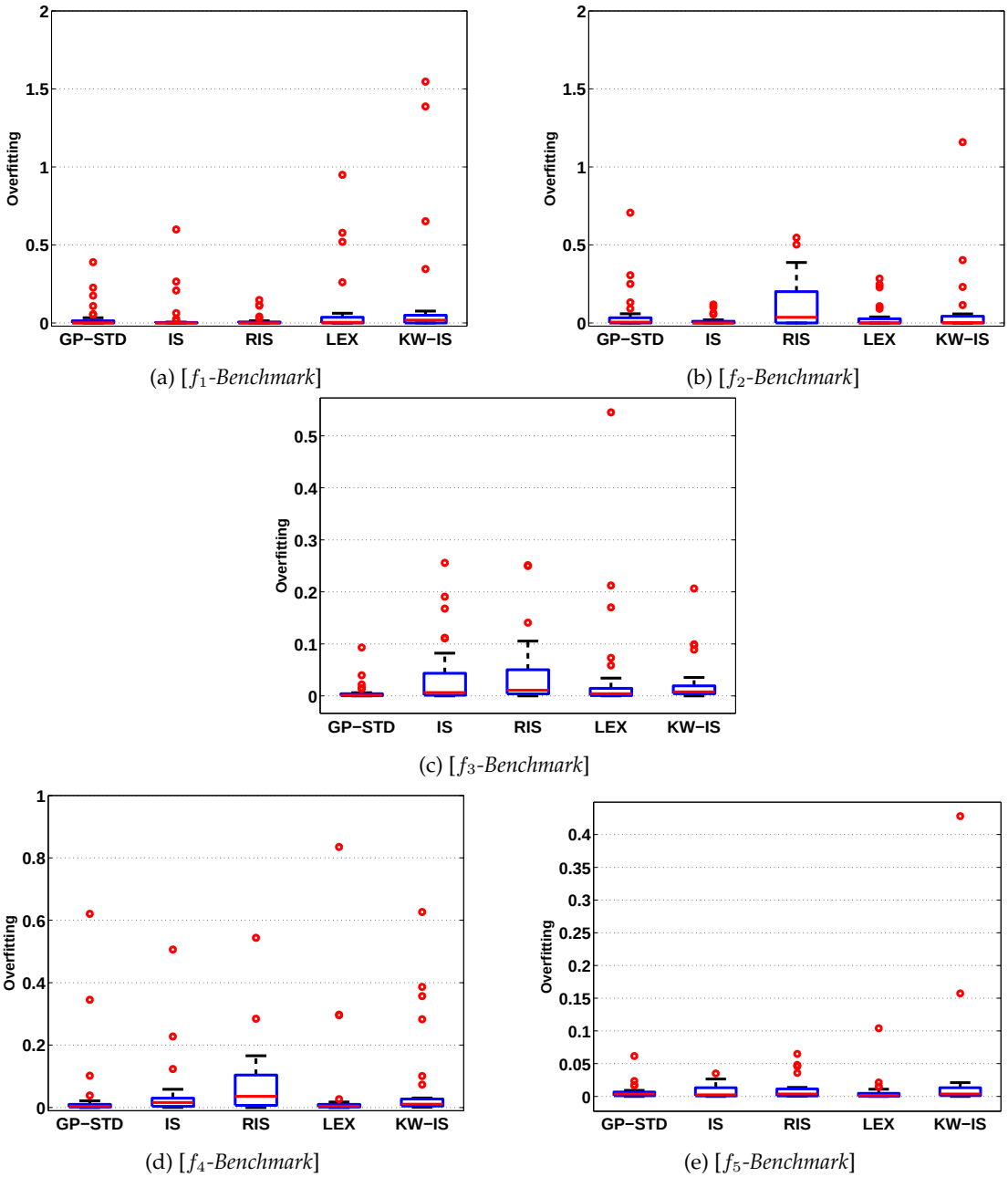


(a) [f_1-*Benchmark*] (b) [f_2-*Benchmark*]

(c) [f_3-*Benchmark*]

(d) [f_4-*Benchmark*] (e) [f_5-*Benchmark*]

Figure 4.1 – Box plot comparison about the test performance of the methods, from the best solution found for each benchmark symbolic regression problem over all thirty runs.

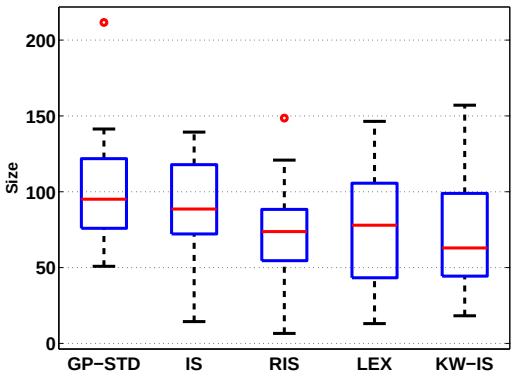


(a) [f_1-*Benchmark*] (b) [f_2-*Benchmark*]

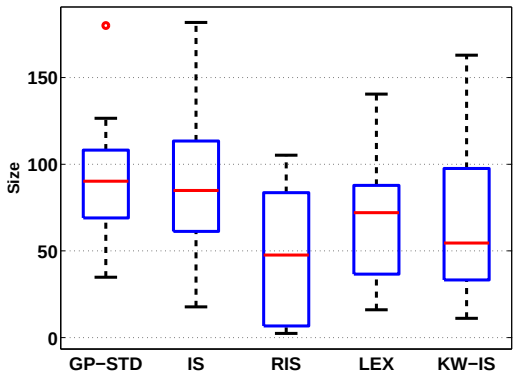
(c) [f_3-*Benchmark*]

(d) [f_4-*Benchmark*] (e) [f_5-*Benchmark*]

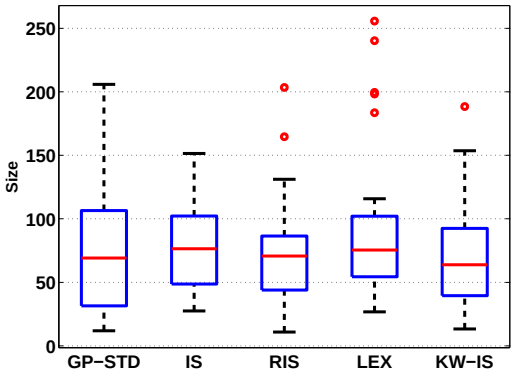
Figure 4.2 – Box plot comparison about the overfitting performance of the methods, from the best solution found for each benchmark symbolic regression problem over all thirty runs.



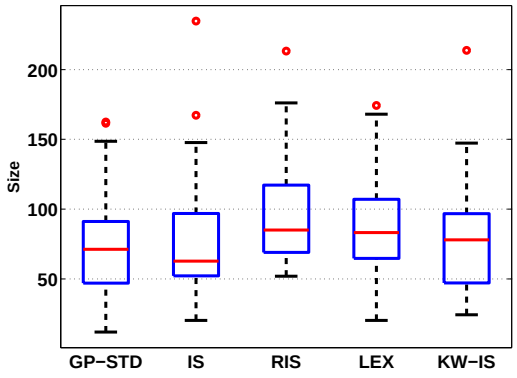
(a) $[f_1 - Benchmark]$



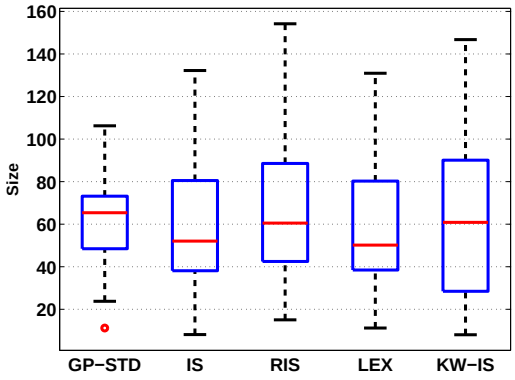
(b) $[f_2 - Benchmark]$



(c) $[f_3 - Benchmark]$

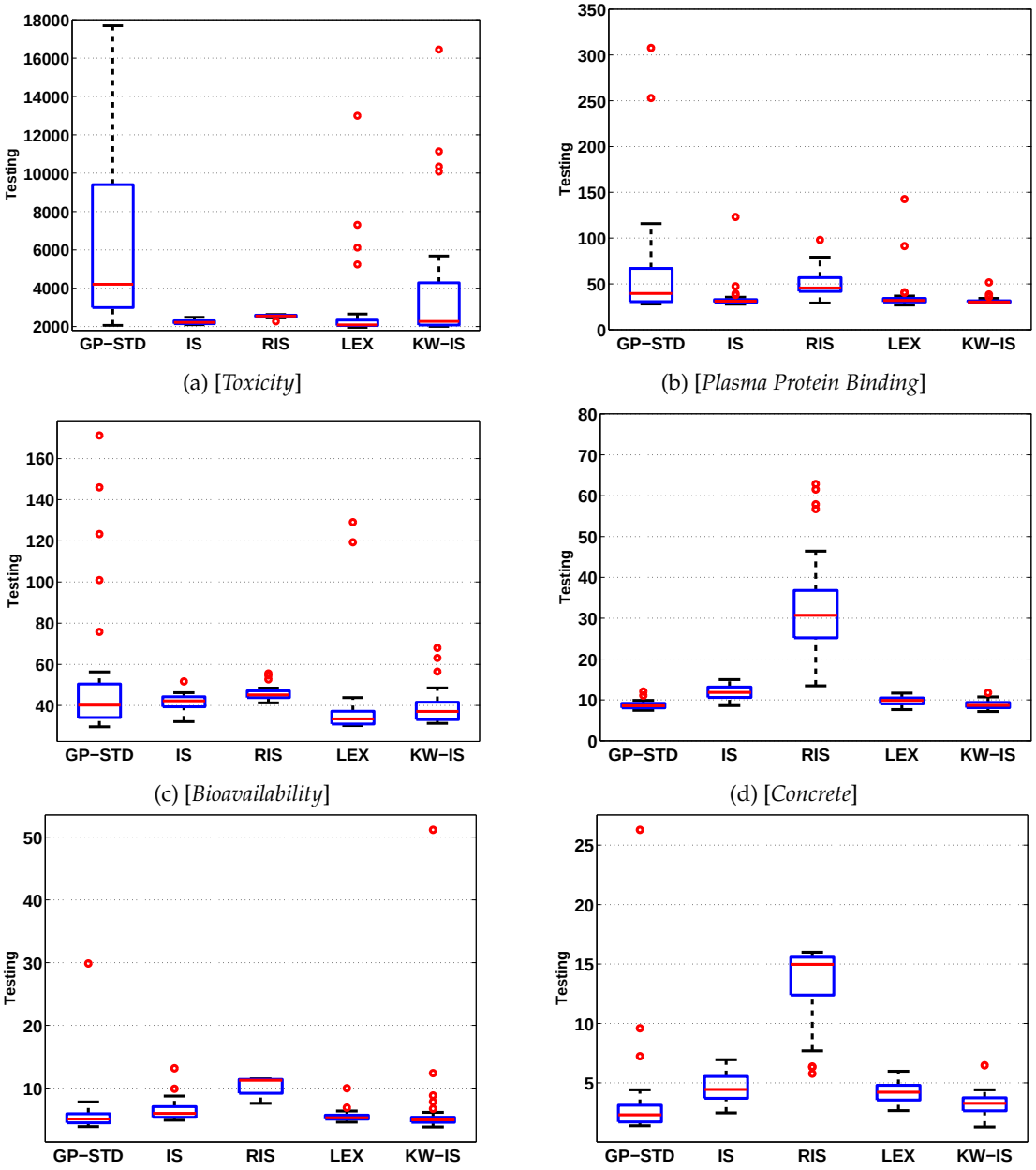


(d) $[f_4 - Benchmark]$



(e) $[f_5 - Benchmark]$

Figure 4.3 – Box plot comparison about the average size performance of the methods, from the solutions found for each benchmark symbolic regression problem over all thirty runs.

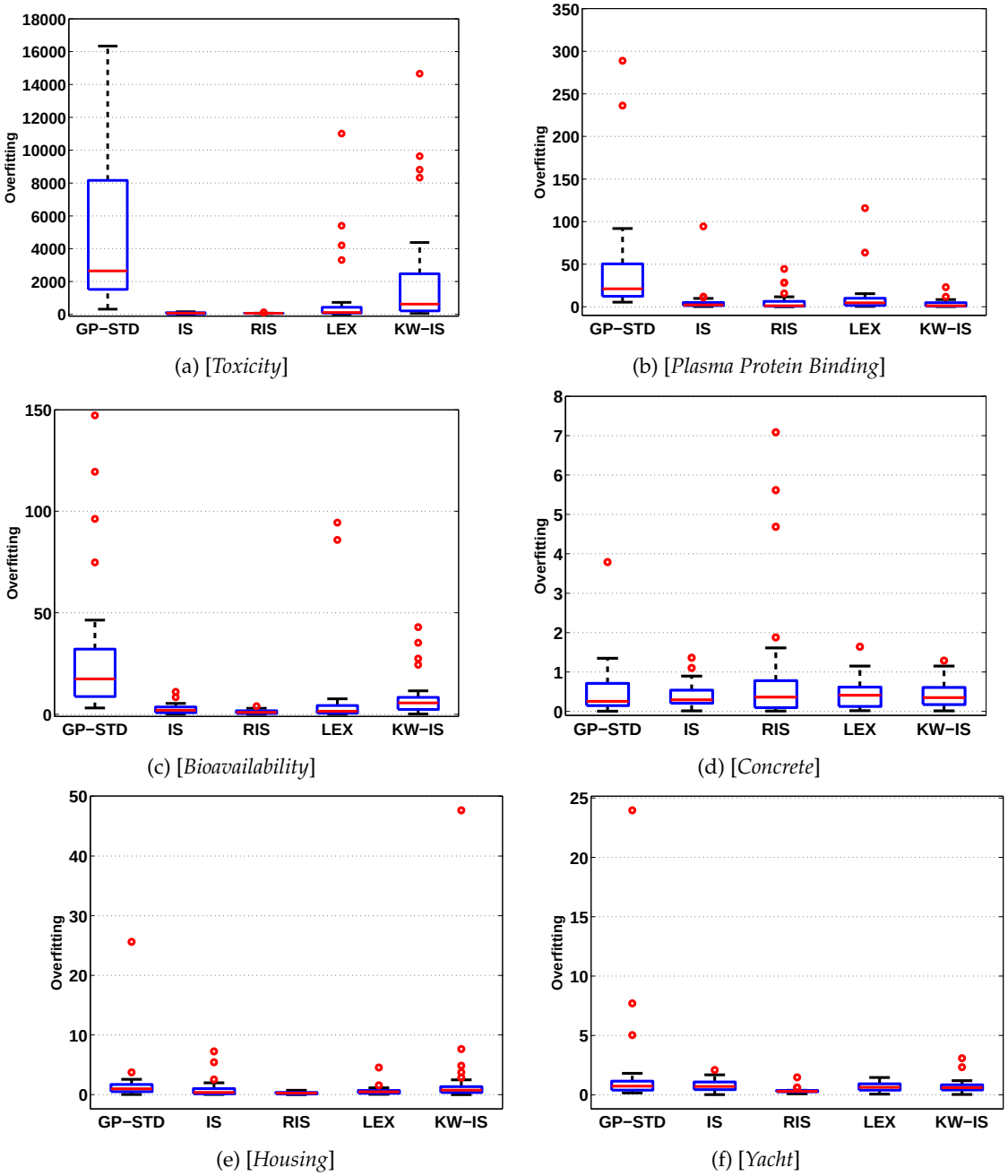


(a) [*Toxicity*] (b) [*Plasma Protein Binding*]

(c) [*Bioavailability*] (d) [*Concrete*]

(e) [*Housing*] (f) [*Yacht*]

Figure 4.4 – Box plot comparison about the test performance of the methods, from the best solution found for each real-world regression problem over all thirty runs.

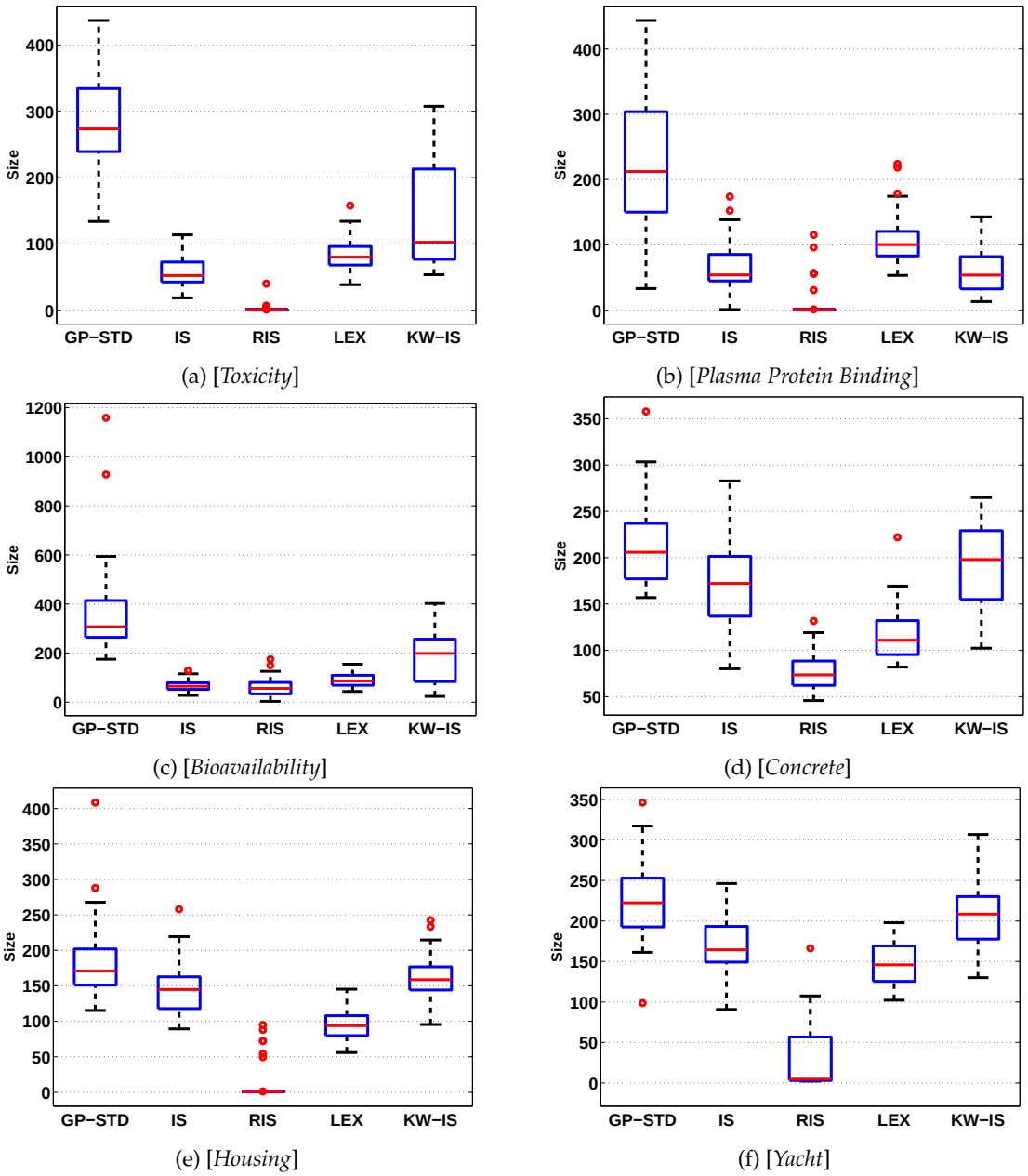


(a) [*Toxicity*] (b) [*Plasma Protein Binding*]

(c) [*Bioavailability*] (d) [*Concrete*]

(e) [*Housing*] (f) [*Yacht*]

Figure 4.5 – Box plot comparison about the overfitting performance of the methods, from the best solution found for each real-world regression problem over all thirty runs.



(a) [*Toxicity*] (b) [*Plasma Protein Binding*]

(c) [*Bioavailability*] (d) [*Concrete*]

(e) [*Housing*] (f) [*Yacht*]

Figure 4.6 – Box plot comparison about the average size performance of the methods, from the solutions found for each real-world regression problem over all thirty runs.

Table 4.4 – Median of 30 executions for testing, overfitting and size; bold indicates best.

	GP-STD	IS	RIS	LEX	KW-IS
Testing					
f1-Benchmark	**0.0031**	**0.0000**	**0.0000**	**0.0207**	**0.0468**
f2-Benchmark	**0.0418**	**0.0000**	**0.2990**	**0.0000**	**0.0286**
f3-Benchmark	**0.0079**	0.0616	0.0569	**0.0166**	0.0496
f4-Benchmark	**0.0126**	0.1172	0.1733	**0.0180**	0.0691
f5-Benchmark	**0.0342**	**0.0695**	**0.0756**	**0.0180**	**0.0455**
Toxicity	4206.99	**2217.21**	2555.48	**2089.20**	**2267.04**
Plasma Protein Binding	39.47	**31.21**	45.47	**32.00**	**30.31**
Bioavailability	**40.16**	42.23	45.15	**33.49**	**37.12**
Concrete	**8.56**	11.84	30.72	9.87	**8.67**
Housing	**5.08**	5.96	11.26	5.26	**4.93**
Yacht	**2.31**	4.45	14.96	4.21	**3.27**
Overfitting					
f1-Benchmark	**0.0010**	**0.0000**	**0.0000**	**0.0039**	**0.0184**
f2-Benchmark	**0.0052**	**0.0000**	**0.0366**	**0.0000**	**0.0023**
f3-Benchmark	**0.0013**	**0.0065**	0.0111	**0.0044**	0.0078
f4-Benchmark	**0.0039**	**0.0159**	0.0362	**0.0041**	**0.0109**
f5-Benchmark	**0.0036**	**0.0025**	**0.0036**	**0.0011**	**0.0036**
Toxicity	2645.66	**81.56**	**65.03**	115.32	629.16
Plasma Protein Binding	21.19	**2.08**	**1.21**	**4.67**	**1.25**
Bioavailability	17.40	**2.01**	**0.94**	**1.50**	5.64
Concrete	**0.2541**	**0.2950**	**0.3609**	**0.4100**	**0.3486**
Housing	0.9882	**0.3800**	**0.2519**	**0.4918**	0.7668
Yacht	0.7244	0.6957	**0.2884**	0.6293	0.5997
Size					
f1-Benchmark	**95**	**89**	**74**	**78**	**63**
f2-Benchmark	90	85	**48**	**72**	**55**
f3-Benchmark	**69**	**76**	**71**	**75**	**64**
f4-Benchmark	**71**	**63**	**85**	**83**	**78**
f5-Benchmark	**65**	**52**	**61**	**50**	**61**
Toxicity	274	52	**1**	80	102
Plasma Protein Binding	212	54	**1**	100	54
Bioavailability	308	**65**	**57**	87	199
Concrete	206	172	**74**	111	198
Housing	171	145	**1**	94	159
Yacht	222	164	**5**	146	209

Table 4.5 – Results of the Friedman test for the symbolic regression problems (part 1), showing the p-value after the Bonferroni-Holm correction for each pairwise comparison; bold indicates that the test rejects the null hypothesis at the $\alpha = 0.05$ significance level.

		Testing					*Overfitting*					*Size*				
		GP-STD	IS	RIS	LEX	KW-IS	GP-STD	IS	RIS	LEX	KW-IS	GP-STD	IS	RIS	LEX	KW-IS
f1-Benchmark	GP-STD	-	1.5948	1.2921	1.7988	2.3515	-	0.5431	0.8648	1.2842	1.4126	-	2.0000	0.1059	1.0089	0.2561
	IS	-	-	1.3662	2.3515	0.1674	-	-	1.4300	0.5431	0.1674	-	-	0.5431	1.0089	0.8648
	RIS	-	-	-	1.7988	0.1235	-	-	-	1.4126	**0.0250**	-	-	-	2.0000	1.0933
	LEX	-	-	-	-	2.1190	-	-	-	-	1.4300	-	-	-	-	1.3956
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f2-Benchmark	GP-STD	-	**0.0481**	0.0847	0.3527	1.0000	-	**0.0314**	0.9519	0.1867	1.0933	-	0.9304	**0.0102**	0.5431	0.5431
	IS	-	-	**0.0067**	0.5809	0.3593	-	-	**0.0016**	0.4073	1.0933	-	-	**0.0102**	0.7206	0.7206
	RIS	-	-	-	0.2876	0.3787	-	-	-	0.4073	0.2876	-	-	-	0.9304	0.4752
	LEX	-	-	-	-	0.7063	-	-	-	-	0.9304	-	-	-	-	0.8200
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f3-Benchmark	GP-STD	-	**0.0279**	**0.0091**	0.7206	**0.0006**	-	0.0847	**0.0102**	1.0089	**0.0314**	-	3.5750	1.9133	3.5750	2.8600
	IS	-	-	0.9304	0.4073	0.9304	-	-	1.0089	1.0933	0.9304	-	-	2.7913	2.1450	1.4413
	RIS	-	-	-	0.0741	0.8200	-	-	-	0.8648	1.0933	-	-	-	1.4413	1.4300
	LEX	-	-	-	-	0.7206	-	-	-	-	0.9304	-	-	-	-	1.2971
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f4-Benchmark	GP-STD	-	**0.0244**	**0.0026**	0.5466	**0.0244**	-	0.4752	**0.0314**	0.9304	0.8200	-	2.1450	1.2971	1.2971	2.8600
	IS	-	-	0.5765	**0.0026**	0.5466	-	-	0.7206	0.4752	0.9304	-	-	1.3666	1.1530	2.8600
	RIS	-	-	-	**0.0209**	0.5765	-	-	-	**0.0026**	0.7206	-	-	-	2.0000	0.6789
	LEX	-	-	-	-	**0.0023**	-	-	-	-	0.2277	-	-	-	-	1.0089
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f5-Benchmark	GP-STD	-	2.1866	0.6789	1.4300	1.7428	-	2.3260	2.1450	1.2971	2.1190	-	2.1450	2.0000	2.4599	4.2900
	IS	-	-	1.4300	2.1866	2.3260	-	-	2.3260	0.2846	2.1450	-	-	2.4599	4.2900	1.4413
	RIS	-	-	-	2.3260	1.9133	-	-	-	2.1866	2.0000	-	-	-	3.5750	2.8600
	LEX	-	-	-	-	1.8608	-	-	-	-	2.1866	-	-	-	-	3.2565
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4.6 – Results of the Friedman test for the symbolic regression problems (part 2), showing the p-value after the Bonferroni-Holm correction for each pairwise comparison; bold indicates that the test rejects the null hypothesis at the $\alpha = 0.05$ significance level.

Toxicity	GP-STD	-	**0.0001**	**0.0005**	**0.0209**	0.0529	-	**0.0000**	**0.0000**	**0.0004**	**0.0318**	-	**0.0000**	**0.0000**	**0.0000**	**0.0002**
	IS	-	-	**0.0000**	0.0529	0.7150	-	-	0.7150	0.1358	**0.0000**	-	-	**0.0000**	**0.0020**	**0.0000**
	RIS	-	-	-	**0.0018**	0.5466	-	-	-	**0.0041**	**0.0000**	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	0.0854	-	-	-	-	**0.0013**	-	-	-	-	**0.0285**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Plasma Protein Binding	GP-STD	-	0.3394	0.8200	0.3394	**0.0244**	-	**0.0000**	**0.0005**	**0.0005**	**0.0000**	-	**0.0001**	**0.0000**	**0.0003**	**0.0000**
	IS	-	-	**0.0000**	1.4300	1.4300	-	-	0.4324	0.1423	0.9304	-	-	**0.0003**	**0.0020**	0.7150
	RIS	-	-	-	**0.0005**	**0.0000**	-	-	-	0.2716	0.9304	-	-	-	**0.0000**	**0.0001**
	LEX	-	-	-	-	0.1708	-	-	-	-	**0.0209**	-	-	-	-	**0.0008**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bioavailability	GP-STD	-	0.4652	0.4324	0.3394	0.4324	-	**0.0000**	**0.0000**	**0.0001**	**0.0001**	-	**0.0000**	**0.0000**	**0.0000**	**0.0061**
	IS	-	-	**0.0081**	**0.0081**	**0.0209**	-	-	0.1138	0.2883	**0.0013**	-	-	0.1441	**0.0318**	**0.0061**
	RIS	-	-	-	**0.0000**	**0.0023**	-	-	-	0.2883	**0.0000**	-	-	-	**0.0051**	**0.0018**
	LEX	-	-	-	-	0.3394	-	-	-	-	0.1138	-	-	-	-	0.1358
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Concrete	GP-STD	-	**0.0000**	**0.0000**	**0.0105**	1.0000	-	3.2565	1.4413	1.4300	2.8600	-	**0.0318**	**0.0000**	**0.0000**	0.1441
	IS	-	-	**0.0000**	**0.0000**	**0.0000**	-	-	3.2565	2.4599	2.8600	-	-	**0.0000**	**0.0041**	0.0569
	RIS	-	-	-	**0.0000**	**0.0000**	-	-	-	2.1450	2.7913	-	-	-	**0.0001**	**0.0000**
	LEX	-	-	-	-	0.1358	-	-	-	-	2.4599	-	-	-	-	**0.0003**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Housing	GP-STD	-	0.0529	**0.0000**	1.4300	1.4300	-	0.8648	**0.0001**	0.0847	1.3956	-	**0.0030**	**0.0000**	**0.0000**	0.1358
	IS	-	-	**0.0004**	0.0529	**0.0209**	-	-	1.3956	0.9304	0.8648	-	-	**0.0000**	**0.0002**	0.4652
	RIS	-	-	-	**0.0000**	**0.0000**	-	-	-	0.4752	**0.0091**	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	**0.0423**	-	-	-	-	0.7206	-	-	-	-	**0.0000**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Yacht	GP-STD	-	**0.0004**	**0.0000**	**0.0010**	0.1358	-	2.1450	**0.0023**	2.7913	2.0000	-	**0.0001**	**0.0000**	**0.0000**	0.7150
	IS	-	-	**0.0000**	0.1441	**0.0004**	-	-	**0.0000**	2.7913	2.8600	-	-	**0.0000**	0.0569	**0.0105**
	RIS	-	-	-	**0.0000**	**0.0000**	-	-	-	**0.0023**	**0.0021**	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	**0.0030**	-	-	-	-	2.8600	-	-	-	-	**0.0041**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

First, considering test performance the following can be observed. For all benchmark problems none of the fitness-case sampling methods outperform GP-STD. In fact, only LEX can compare on test performance, while RIS achieves the weakest results of all methods. On the other hand, for the real-world problems GP-STD is outperformed by at least 2 fitness-case sampling methods, and in two problems by six of them. In these problems, LEX and KW-IS clearly show the best performance, while IS outperforms GP-STD on two of the six problems (Toxicity and Plasma). Again, RIS clearly achieves the lowest performance of all fitness-case sampling methods.

Based on overfitting, IS shows the best performance, on both the synthetic benchmarks and the real-world problems. Similarly, RIS shows strong performance on the real-world cases. The performance of IS and RIS is consistent with those reported in [GONÇALVES et SILVA, 2013]. On the other hand, GP-STD and LEX do not seem to overfit on synthetic problems, but clearly do so on the real-world cases. In particular, GP-STD clearly shows the worst performance among all methods on the real-world problems. KW-IS shows the weakest performance among all fitness-case sampling methods.

Finally, if we consider size, some interesting results can be seen. On synthetic problems there appears to be small differences between the methods, where we can only see a statistically significant difference in the second problem. However, on the real-world cases more interesting results are obtained. First, the reason for the bad test performance of RIS can be attributed to the fact that the evolved trees are basically a single terminal (variable) in three of the six cases. Second, GP-STD consistently produces the larger trees, in some cases one order of magnitude larger than the fitness-case sampling methods. Third, if we consider size and test fitness, then we can say that IS, LEX and KW-IS produce smaller trees with better performance than GP-STD.

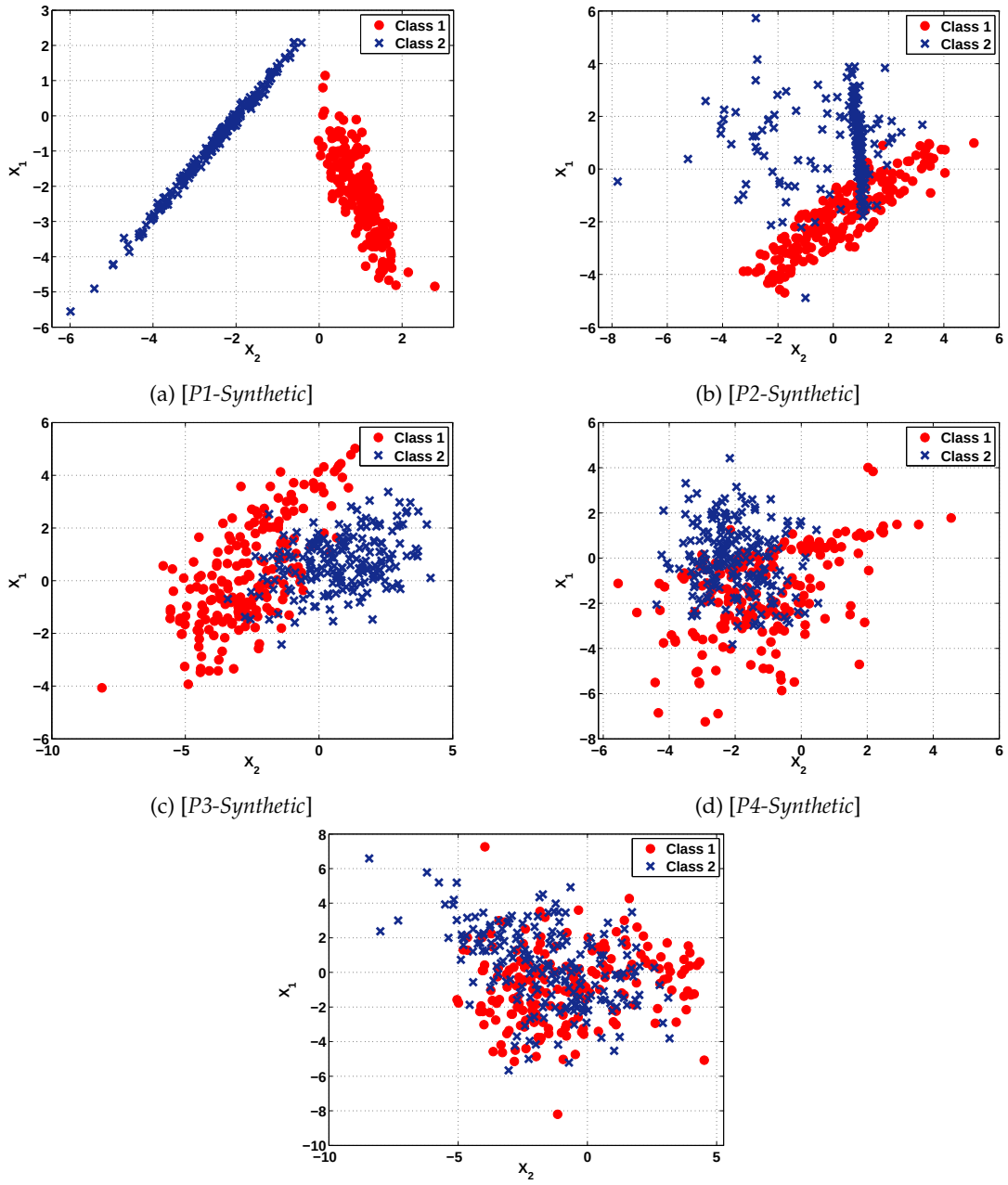
4.3.2 Classification Problems

Synthetic binary classification problems were randomly generated using Gaussian mixture models (GMM's). Examples of the classification problems generated are shown in Figure 4.7, which depicts sample points of two different classes (circles and crosses) scattered over the $\mathbb{R}^2$ plane. Problems are generated with unimodal or multimodal classes, with different amounts of class overlap. All class samples lie within the closed 2-D interval $x, y \in [-10, 10]$, and 200 sample points were randomly generated for each class. The parameters for the GMM of each class were also randomly chosen using the following ranges of values:

1. Number of Gaussian components: $\{1, 2, 3\}$.
2. Median of each Gaussian component for each feature dimension: $[-3, 3]$.
3. Each element of the covariant matrix of each Gaussian component: $(0, 2]$.
4. The rotation angle of each covariance matrix: $[0, 2\pi]$.
5. The proportion of sample points generated with each Gaussian component: $[0, 1]$.

Afterward, five problems were chosen, shown in Figure 4.7, that are used to evaluate the algorithms tested here. Additionally, six real-world problems from the University of California, Irvine (UCI) machine learning repository were chosen [LICHMAN, 2013], summarized in Table 4.7. These problems have a more complex nature and therefore will be interesting test cases for the sampling methods.

In this work, a GP classifier called static range selection (SRS) was used [ZHANG et SMART, 2006]. For a two class problem and real-valued GP outputs, the SRS decision rule is simple: if the program output for input pattern x is greater than zero then the pattern is labeled as belonging to class A, otherwise its labeled as a class B pattern. The experimental parameters are given in Table 4.8, and fitness is given by the classification error.



(a) [*P1-Synthetic*]

(b) [*P2-Synthetic*]

(c) [*P3-Synthetic*]

(d) [*P4-Synthetic*]

(e) [*P5-Synthetic*]

Figure 4.7 – Synthetic binary classification problems randomly generated using Gaussian mixture models with different amounts of class overlap, scattered over 2 dimensional space.

For the synthetic problems, Figures 4.8, 4.9 and 4.10 illustrate the performance of each method on test fitness, overfitting and average size. Similarly, Figures 4.11, 4.12 and 4.13 show the same for the real-world problems.

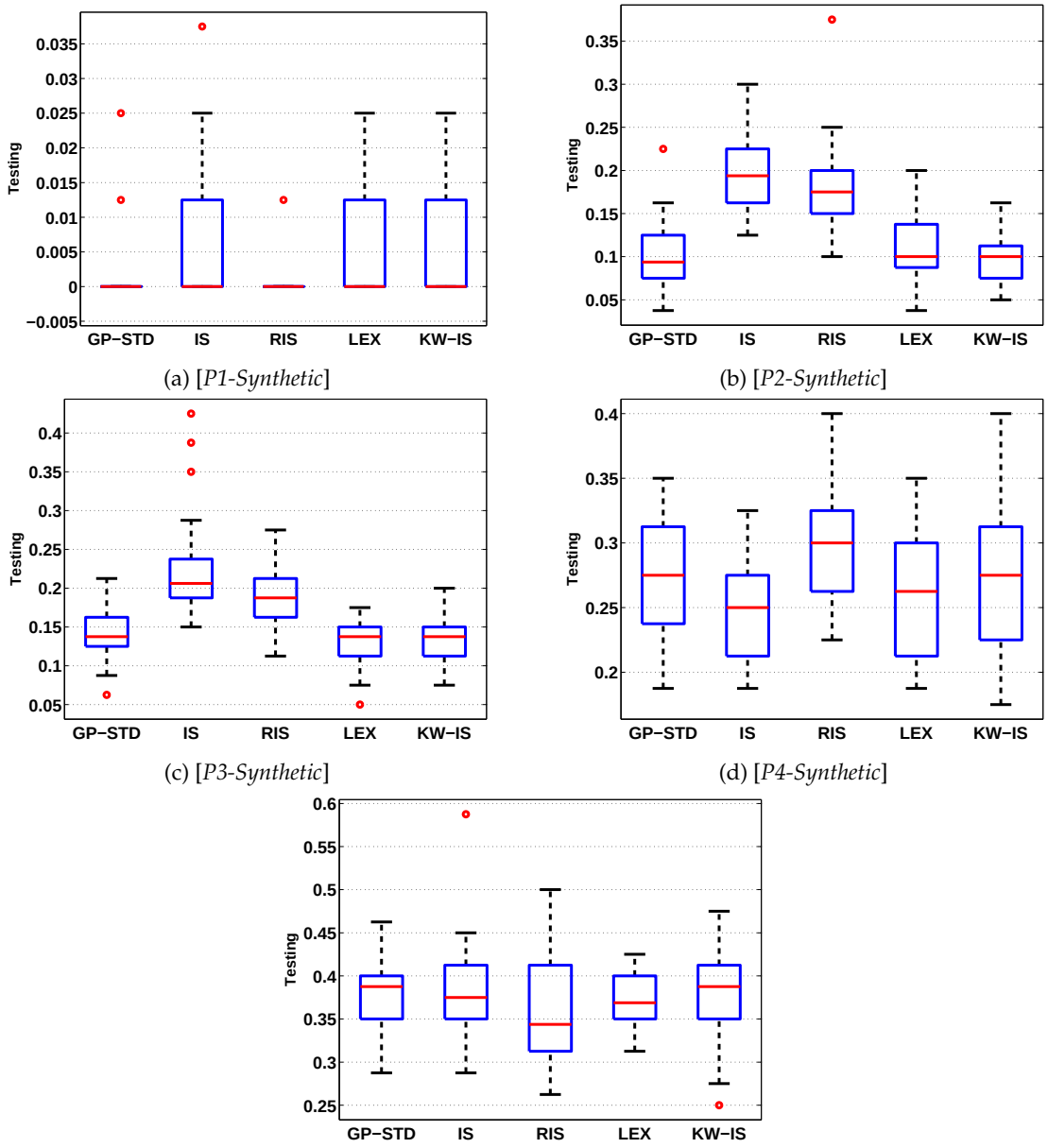
To summarize these results, a numerical comparison of the methods is provided in Table 4.9, showing the median values of each method, on each problem and for each measure. Here, bold indicates statistically different results with respect to the other (non-bold) values. Tables 4.10 and 4.11, show the p-values with the Bonferroni-Holm correction for the pairwise comparisons on each problem where bold values indicate that the null hypothesis is rejected at the $\alpha = 0.05$ significance level.

Table 4.7 – Real-world classification problems from Irvine (UCI) machine learning repository [LICHMAN, 2013].

No.	Problem	Classes	Features	Instances	Description
1	*Breast Cancer Wisconsin*	2	8	699	Original Wisconsin Breast Cancer Database.
2	*Parkinson's*	2	22	195	Oxford Parkinson's Disease Detection Dataset
3	*Pima Indians Diabetes*	2	8	768	From National Institute of Diabetes
4	*Indian Liver Patient*	2	10	579	Contains 416 liver patient records and 167 non liver patient records
5	*Retinopathy*	2	19	1151	Contains features extracted from the Messidor image set, to predict signs of diabetic retinopathy or not
6	*Vertebral Column 2C*	2	6	310	Biomedical data set built by Dr. Henrique da Mota

Table 4.8 – GP parameters used for the classification problems.

Parameter	Description
Runs	30
Size of population	200 individuals
Generations	100 generations
Initialization	*Ramped Half-and-half* with mamixum depth level of 6
Operator probabilities	Crossover $p_c = 0.8$, mutation $p_\mu = 0.2$
Function set	$(+, -, \times, \div, \sin, \cos, \exp, \log, if)$
Terminal set	input variables
Maximum tree depth	20 levels
Selection	Size 3 tournament
Elitism	Best individual always survives



(a) [*P1-Synthetic*]

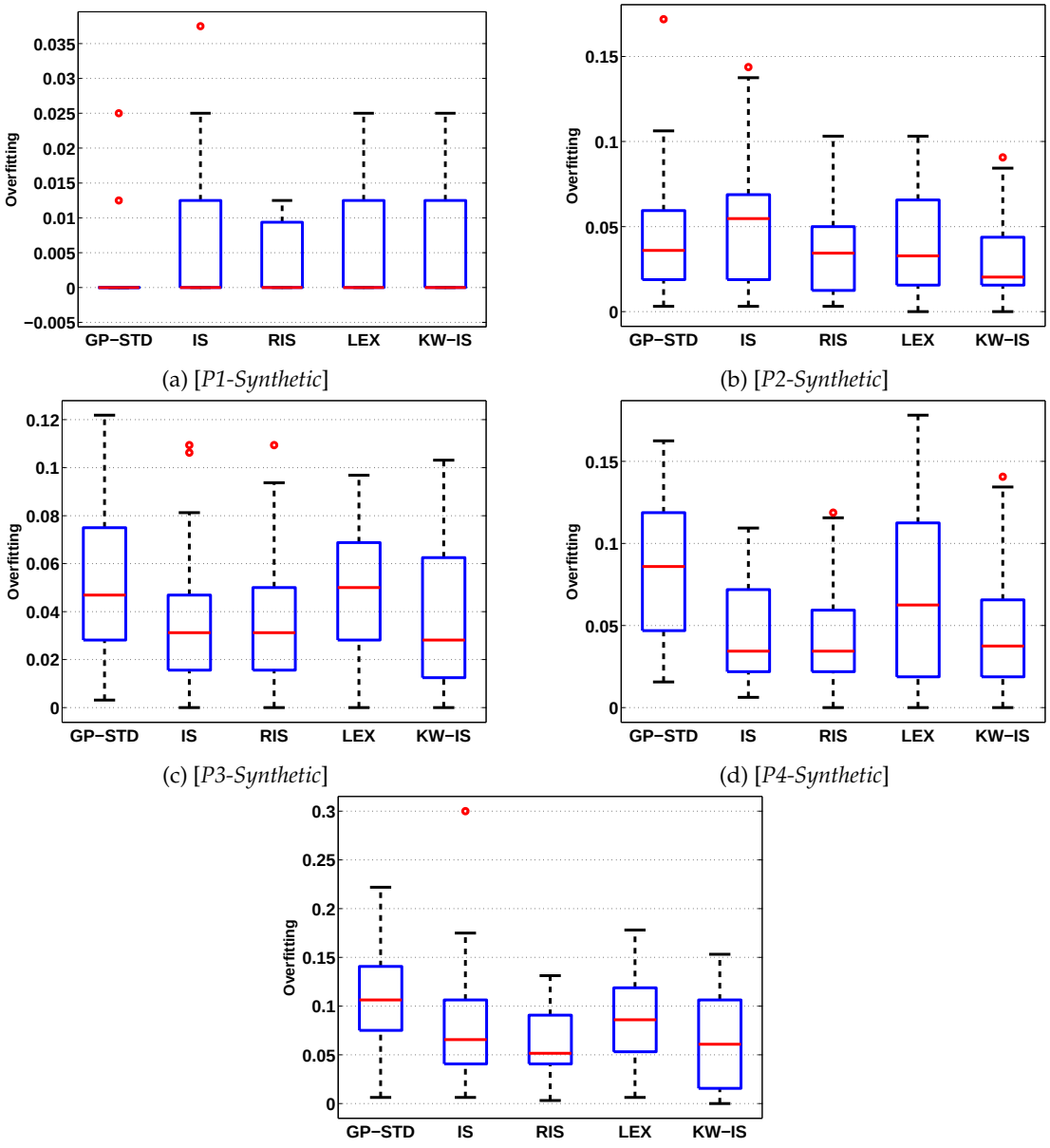
(b) [*P2-Synthetic*]

(c) [*P3-Synthetic*]

(d) [*P4-Synthetic*]

(e) [*P5-Synthetic*]

Figure 4.8 – Box plot comparison about the test performance of the methods, from the best solution found for each synthetic classification problem over all thirty runs.



(a) [*P1-Synthetic*]

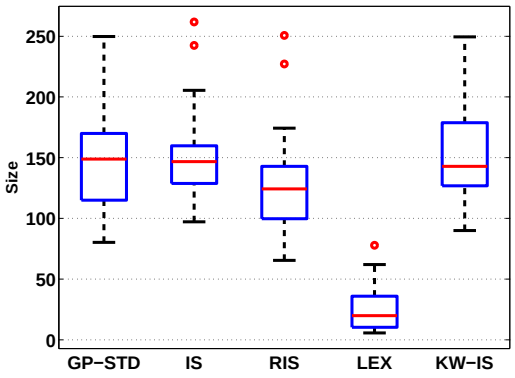
(b) [*P2-Synthetic*]

(c) [*P3-Synthetic*]

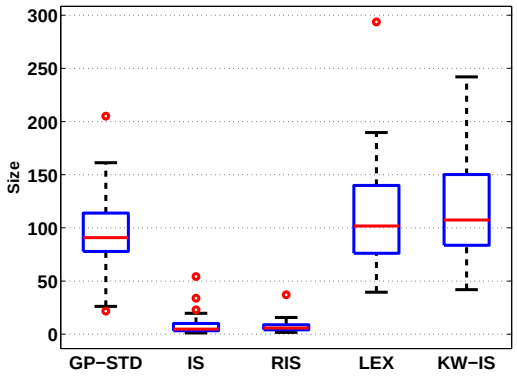
(d) [*P4-Synthetic*]

(e) [*P5-Synthetic*]

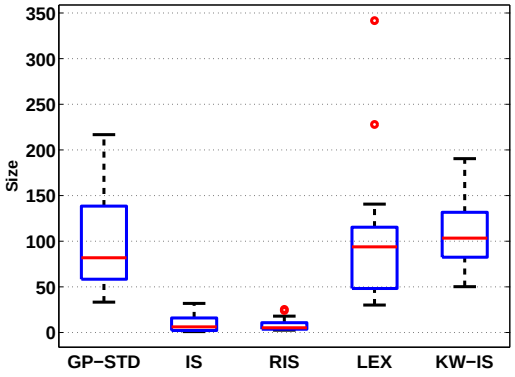
Figure 4.9 – Box plot comparison about the overfitting performance of the methods, from the best solution found for each synthetic classification problem over all thirty run.



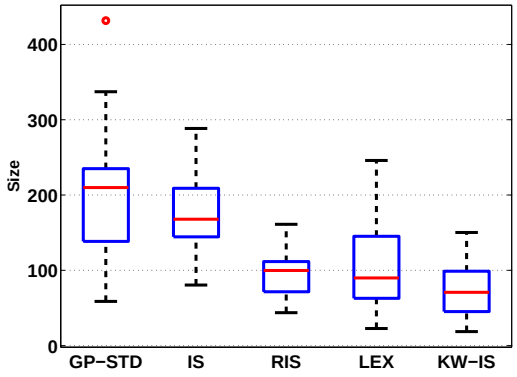
(a) [*P1-Synthetic*]



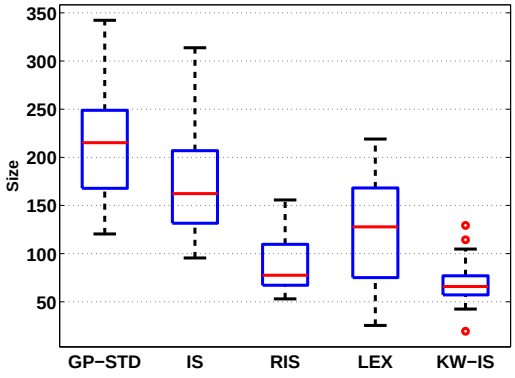
(b) [*P2-Synthetic*]



(c) [*P3-Synthetic*]

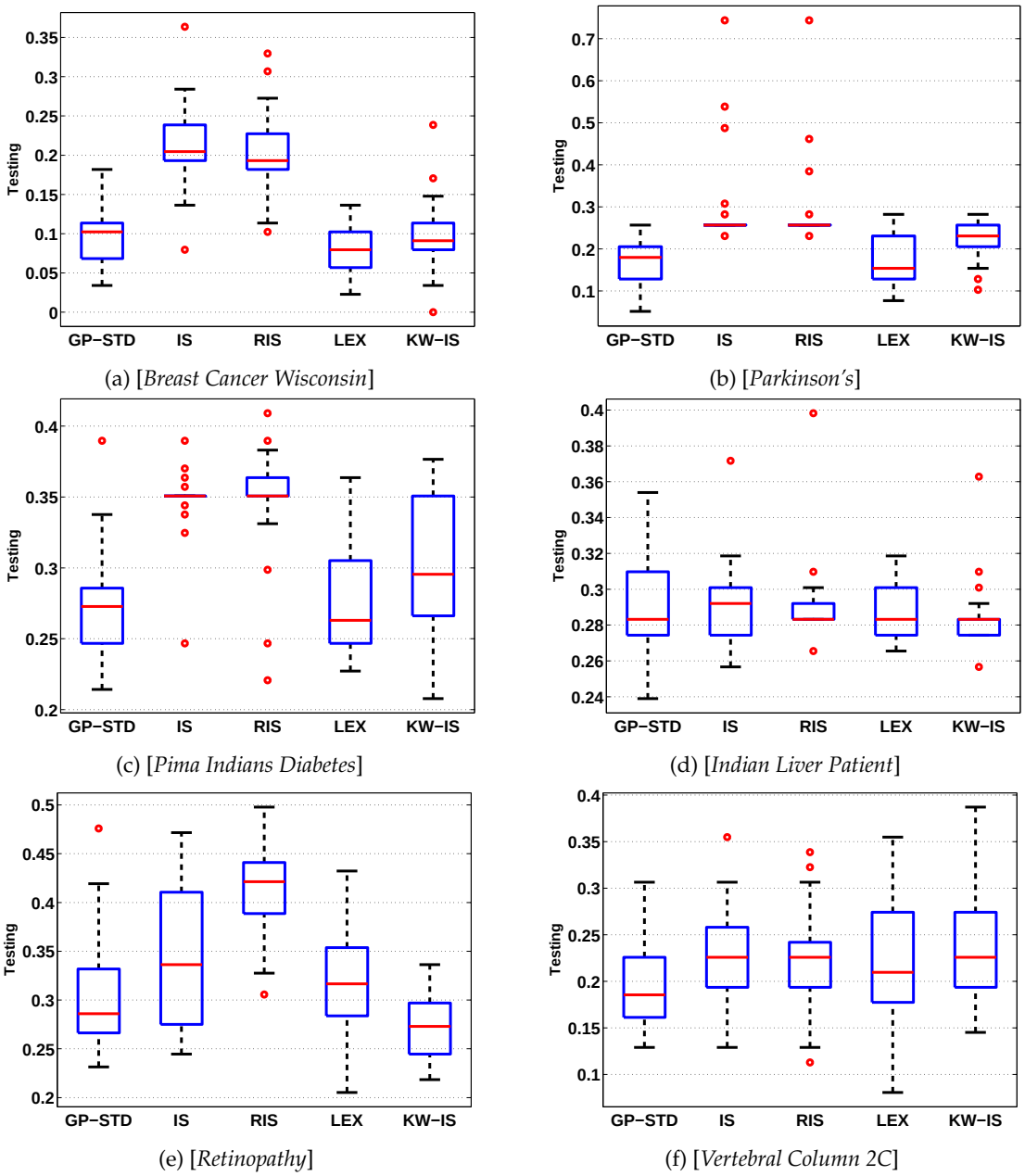


(d) [*P4-Synthetic*]



(e) [*P5-Synthetic*]

Figure 4.10 – Box plot comparison about the average size performance of the methods, from the solutions found for each synthetic classification problem over all thirty run.

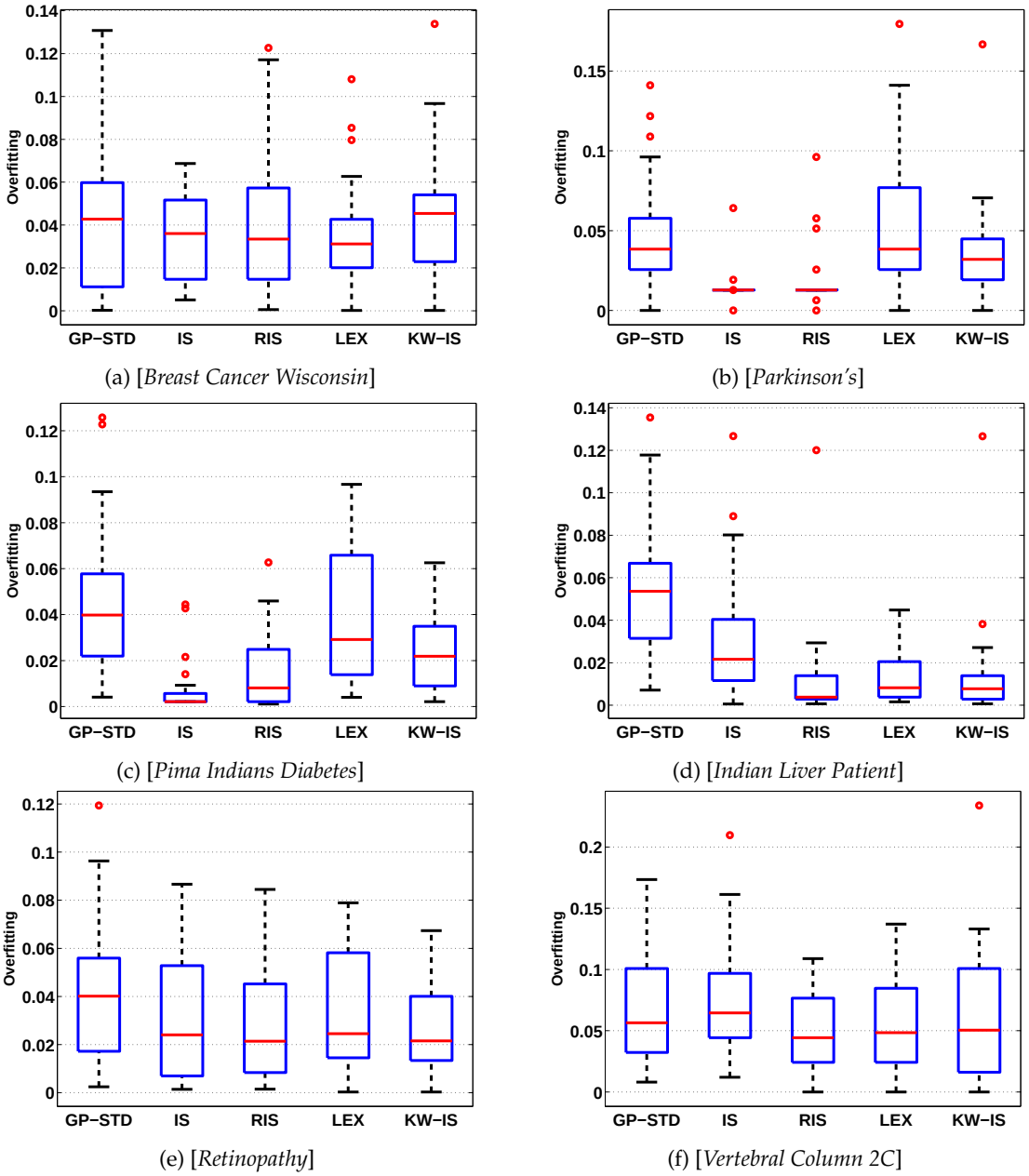


(a) [*Breast Cancer Wisconsin*] (b) [*Parkinson's*]

(c) [*Pima Indians Diabetes*] (d) [*Indian Liver Patient*]

(e) [*Retinopathy*] (f) [*Vertebral Column 2C*]

Figure 4.11 – Box plot comparison about the test performance of the methods, from the best solution found for each real-world classification problem over all thirty run.

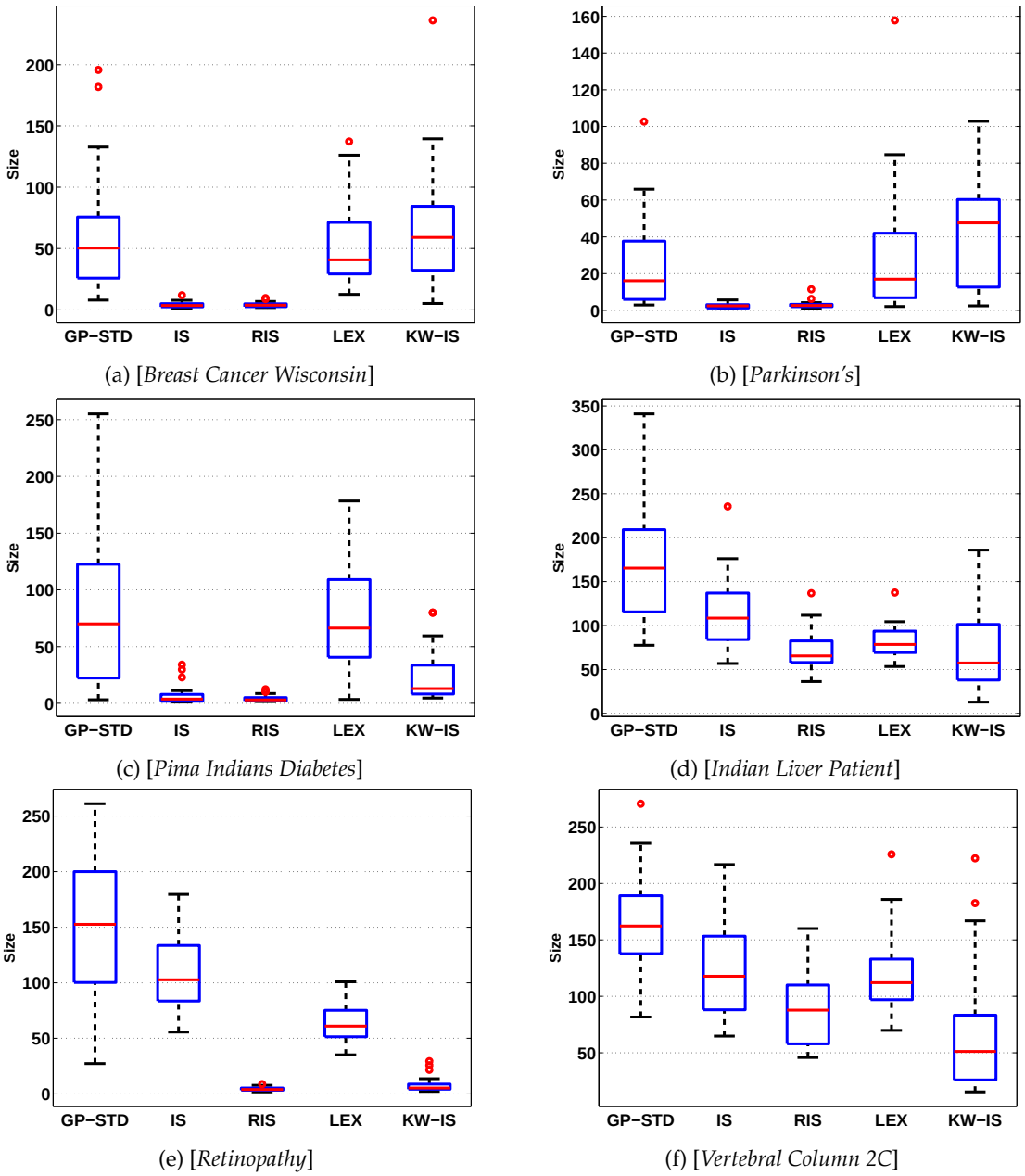


(a) [*Breast Cancer Wisconsin*] (b) [*Parkinson's*]

(c) [*Pima Indians Diabetes*] (d) [*Indian Liver Patient*]

(e) [*Retinopathy*] (f) [*Vertebral Column 2C*]

Figure 4.12 – Box plot comparison about the overfitting performance of the methods, from the best solution found for each real-world classification problem over all thirty run.



(a) [*Breast Cancer Wisconsin*] (b) [*Parkinson's*]

(c) [*Pima Indians Diabetes*] (d) [*Indian Liver Patient*]

(e) [*Retinopathy*] (f) [*Vertebral Column 2C*]

Figure 4.13 – Box plot comparison about the average size performance of the methods, from the solutions found for each real-world classification problem over all thirty run.

Table 4.9 – Median of 30 executions over testing, overfitting and size; bold indicates best.

	GP-STD	IS	RIS	LEX	KW-IS
Testing					
P1-Synthetic	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
P2-Synthetic	**0.0938**	0.1937	0.1750	**0.1000**	**0.1000**
P3-Synthetic	**0.1375**	0.2062	0.1875	**0.1375**	**0.1375**
P4-Synthetic	**0.2750**	**0.2500**	0.3000	**0.2625**	**0.2750**
P5-Synthetic	**0.3875**	**0.3750**	**0.3438**	**0.3688**	**0.3875**
Breast Cancer Wisconsin	**0.1023**	0.2045	0.1932	**0.0795**	**0.0909**
Parkinson's	**0.1795**	0.2564	0.2564	**0.1538**	0.2308
Pima Indians Diabetes	**0.2727**	0.3506	0.3506	**0.2630**	**0.2955**
Indian Liver Patient	**0.2832**	**0.2920**	**0.2832**	**0.2832**	**0.2832**
Retinopathy	**0.2860**	0.3362	0.4214	0.3166	**0.2729**
Vertebral Column 2C	**0.1855**	**0.2258**	**0.2258**	**0.2097**	**0.2258**
Overfitting					
P1-Synthetic	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
P2-Synthetic	**0.0359**	**0.0547**	**0.0344**	**0.0328**	**0.0203**
P3-Synthetic	**0.0469**	**0.0312**	**0.0312**	**0.0500**	**0.0281**
P4-Synthetic	**0.0859**	**0.0344**	**0.0344**	**0.0625**	**0.0375**
P5-Synthetic	0.1063	**0.0656**	**0.0516**	**0.0859**	**0.0609**
Breast Cancer Wisconsin	**0.0428**	**0.0360**	**0.0335**	**0.0312**	**0.0454**
Parkinson's	0.0385	**0.0128**	**0.0128**	0.0385	0.0321
Pima Indians Diabetes	0.0398	**0.0021**	**0.0081**	0.0291	0.0219
Indian Liver Patient	0.0536	**0.0216**	**0.0038**	0.0082	0.0077
Retinopathy	**0.0401**	**0.0241**	**0.0214**	**0.0245**	**0.0215**
Vertebral Column 2C	**0.0565**	**0.0645**	**0.0444**	**0.0484**	**0.0504**
Size					
P1-Synthetic	149	147	124	**20**	143
P2-Synthetic	91	**5**	**6**	102	107
P3-Synthetic	82	**6**	**5**	94	103
P4-Synthetic	210	168	**100**	**90**	**71**
P5-Synthetic	215	162	**78**	128	**66**
Breast Cancer Wisconsin	50	**3**	**4**	41	59
Parkinson's	16	**2**	**3**	17	48
Pima Indians Diabetes	70	**4**	**3**	66	13
Indian Liver Patient	165	108	**65**	79	**57**
Retinopathy	153	103	**4**	61	5
Vertebral Column 2C	162	118	88	112	**51**

Table 4.10 – Results of the Friedman test for the classification problems (part 1), showing the p-value after the Bonferroni-Holm correction for each pairwise comparison; bold indicates that the test rejects the null hypothesis at the $\alpha = 0.05$ significance level.

		Testing					*Overfitting*					*Size*				
		GP-STD	IS	RIS	LEX	KW-IS	GP-STD	IS	RIS	LEX	KW-IS	GP-STD	IS	RIS	LEX	KW-IS
P1-Synthetic	GP-STD	-	0.7494	1.8512	2.0269	1.3515	-	0.8326	1.8674	2.4323	1.8020	-	2.1450	0.1423	**0.0000**	2.1450
	IS	-	-	0.3251	1.8674	2.0269	-	-	1.2025	1.7789	2.1929	-	-	**0.0209**	**0.0000**	2.0000
	RIS	-	-	-	0.7494	0.6661	-	-	-	2.4323	1.9953	-	-	-	**0.0000**	0.1423
	LEX	-	-	-	-	1.5260	-	-	-	-	1.2342	-	-	-	-	**0.0000**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P2-Synthetic	GP-STD	-	**0.0000**	**0.0003**	1.0595	0.8655	-	0.2846	2.8600	2.8600	1.7054	-	**0.0000**	**0.0000**	0.5765	0.8200
	IS	-	-	0.7117	**0.0000**	**0.0000**	-	-	0.6661	1.6399	0.6110	-	-	0.9304	**0.0000**	**0.0000**
	RIS	-	-	-	**0.0000**	**0.0000**	-	-	-	1.3555	2.1450	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	1.0595	-	-	-	-	2.3260	-	-	-	-	0.9304
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P3-Synthetic	GP-STD	-	**0.0001**	**0.0125**	1.7324	1.7324	-	1.1530	0.6789	2.1450	0.6789	-	**0.0000**	**0.0000**	0.9304	0.2716
	IS	-	-	0.7117	**0.0000**	**0.0000**	-	-	1.7054	1.1530	2.1450	-	-	1.0000	**0.0000**	**0.0000**
	RIS	-	-	-	**0.0001**	**0.0002**	-	-	-	1.3666	1.8608	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	1.1549	-	-	-	-	1.0089	-	-	-	-	0.2716
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P4-Synthetic	GP-STD	-	2.0155	0.9140	2.2548	1.1274	-	0.2561	**0.0006**	0.9682	0.2561	-	0.2716	**0.0000**	**0.0013**	**0.0001**
	IS	-	-	**0.0016**	2.2548	1.6911	-	-	1.3956	1.3956	1.0933	-	-	**0.0001**	**0.0004**	**0.0001**
	RIS	-	-	-	**0.0481**	0.4703	-	-	-	0.4752	0.9304	-	-	-	1.0000	0.2716
	LEX	-	-	-	-	2.1638	-	-	-	-	0.8648	-	-	-	-	0.2037
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P5-Synthetic	GP-STD	-	2.8185	1.5492	1.5492	1.3057	-	0.4752	**0.0349**	1.3666	0.0953	-	**0.0174**	**0.0000**	**0.0000**	**0.0000**
	IS	-	-	2.8185	1.7324	1.6948	-	-	1.3666	0.5680	1.1549	-	-	**0.0000**	**0.0174**	**0.0000**
	RIS	-	-	-	2.5966	1.3057	-	-	-	0.2277	1.3956	-	-	-	0.1441	0.1358
	LEX	-	-	-	-	2.3099	-	-	-	-	1.3956	-	-	-	-	**0.0139**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4.11 – Results of the Friedman test for the classification problems (part 2), showing the p-value after the Bonferroni-Holm correction for each pairwise comparison; bold indicates that the test rejects the null hypothesis at the $\alpha = 0.05$ significance level.

Breast Cancer Wisconsin	GP-STD	-	**0.0000**	**0.0000**	0.5223	0.8415	-	3.5750	3.0000	3.2565	2.7332	-	**0.0000**	**0.0000**	0.5466	0.5765
	IS	-	-	0.7063	**0.0000**	**0.0000**	-	-	3.0000	2.0000	2.7332	-	-	0.5466	**0.0000**	**0.0000**
	RIS	-	-	-	**0.0000**	**0.0000**	-	-	-	2.4599	3.5750	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	0.5338	-	-	-	-	3.2565	-	-	-	-	0.5765
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Parkinson's	GP-STD	-	**0.0000**	**0.0000**	1.1860	0.0558	-	**0.0021**	**0.0244**	1.1860	1.3956	-	**0.0000**	**0.0001**	0.2716	0.2883
	IS	-	-	1.1860	**0.0000**	**0.0003**	-	-	1.3956	**0.0001**	**0.0003**	-	-	0.7150	**0.0000**	**0.0000**
	RIS	-	-	-	**0.0000**	**0.0006**	-	-	-	**0.0244**	**0.0408**	-	-	-	**0.0000**	**0.0000**
	LEX	-	-	-	-	**0.0494**	-	-	-	-	0.5765	-	-	-	-	0.2716
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Pima Indians Diabetes	GP-STD	-	**0.0001**	**0.0004**	1.0000	0.2037	-	**0.0000**	**0.0005**	1.0000	0.2716	-	**0.0001**	**0.0000**	0.5466	**0.0030**
	IS	-	-	0.2037	**0.0003**	**0.0096**	-	-	0.2477	**0.0000**	**0.0005**	-	-	0.5466	**0.0000**	**0.0013**
	RIS	-	-	-	**0.0000**	0.1643	-	-	-	**0.0061**	0.2840	-	-	-	**0.0000**	**0.0004**
	LEX	-	-	-	-	0.0789	-	-	-	-	0.5466	-	-	-	-	**0.0013**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Indian Liver Patient	GP-STD	-	2.7425	2.0493	1.6830	1.4536	-	**0.0244**	**0.0000**	**0.0000**	**0.0001**	-	**0.0061**	**0.0000**	**0.0000**	**0.0000**
	IS	-	-	3.1897	2.2548	1.4536	-	-	0.0529	0.1358	0.0529	-	-	**0.0018**	**0.0061**	**0.0423**
	RIS	-	-	-	3.7213	1.0829	-	-	-	**0.0244**	**0.0473**	-	-	-	0.1358	0.4652
	LEX	-	-	-	-	3.7213	-	-	-	-	0.2733	-	-	-	-	**0.0423**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Retinopathy	GP-STD	-	0.9304	**0.0005**	0.8200	0.3787	-	3.2565	0.6789	2.1866	0.6789	-	**0.0212**	**0.0000**	**0.0002**	**0.0000**
	IS	-	-	0.0529	0.9304	**0.0374**	-	-	1.4300	3.5750	3.2565	-	-	**0.0000**	**0.0000**	**0.0000**
	RIS	-	-	-	**0.0001**	**0.0000**	-	-	-	3.5750	2.8600	-	-	-	**0.0000**	**0.0212**
	LEX	-	-	-	-	**0.0374**	-	-	-	-	2.1450	-	-	-	-	**0.0000**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vertebral Column 2C	GP-STD	-	0.8520	0.8520	1.3555	0.4109	-	3.5750	0.1059	3.0000	1.1530	-	0.2883	**0.0001**	**0.0174**	**0.0005**
	IS	-	-	2.2485	1.4300	1.9039	-	-	0.6110	1.9133	3.5750	-	-	**0.0318**	0.2883	**0.0000**
	RIS	-	-	-	2.3099	2.1164	-	-	-	2.7913	3.0000	-	-	-	**0.0174**	**0.0061**
	LEX	-	-	-	-	2.3099	-	-	-	-	2.0000	-	-	-	-	**0.0005**
	KW-IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

First, the test performance for the binary synthetic classification problems shows that all methods find solutions that reach a perfect score on the first problem, but IS, LEX andKW-IS show wider variance than GP-STD and RIS. When increasing the difficulty of the problem it seems that the sampling methods improve a bit the performance, particularly on P4 where IS shows the best performance, as well as RIS for P5. For the real world problems, three algorithms consistently show the best results, GP-STD, LEX and KW-IS. On the other hand, RIS and IS are clearly the weakest methods, in some cases their performance shows twice the error as the best methods (*Breast Cancer Wisconsin* and *P2-Synthetic*). Therefore, these results suggest that no performance improvement is obtained by using fitness-case sampling methods on real-world problems.

Based on overfitting, from the analysis for the synthetic classification problems we can observe that IS and RIS show the worst performance, while for the real-world problems IS and RIS show the best results. However, given their worse performance based on test fitness for the real-world problems, they should not be preferred. Therefore, the best choice considering overfitting performance for the real-world problems is either LEX or KW-IS.

Finally, based on size, from the analysis for the synthetic classification problems we can observe that LEX gets the shortest trees, while IS and RIS also perform well. On the real-world problems IS and RIS evolve the smaller trees, but the improvement in program size is not justified given their poor performance. On the other hand, LEX and KW-IS sometimes evolve smaller trees than GP-STD, but in other cases their performance is similar or a bit worse.

Given these results, it can be stated that LEX and KW-IS do not improve upon GP-STD, but they do not compromise performance either, while IS and RIS negatively effect classifier performance.

4.4 Conclusions

This work presents the first extensive comparative study between fitness-case sampling methods for GP, that use only a subset of training instances in each generation to reduce computational cost and possibly improve generalization or reduce bloat. In particular, four methods are evaluated Interleaved Sampling (IS), Random Interleaved Sampling (RIS), Lexicase Selection (LEX) and Keep-Worst IS (KW-IS), all of them compared with a standard GP search.

Experimental work is extensive, considering symbolic regression with 5 synthetic problems and 6 real-world problems, as well as supervised classification with 5 synthetic problems and 6 real-world datasets. The algorithms were compared using three performance measures: test error, overfitting and average program size. Statistical comparisons were carried out using a non-parametric multigroup test, and post hoc non-parametric comparisons.

The results are illustrative and can be summarized as follows. For symbolic regression the conclusions are dependent on the type of problem. For the simpler benchmark problems, none of the sampling methods outperform standard GP, and only LEX achieves equal performance on all problems. However, when we increase difficulty and consider the real-world problems, then we see the added benefit of the sampling approaches, with three of the methods (LEX, KW-IS and IS) significantly improving upon the standard GP approach. The sampling techniques also exhibit substantially smaller amounts of overfitting and also tend to produce smaller trees than standard GP. Based on these results, it is clear that for difficult real-world symbolic regression, fitness-case sampling can help improve performance, reduce overfitting and reduce code growth. Furthermore, based on the presented experimental work, the best methods to use are LEX and KW-IS, with IS also exhibiting strong results, and RIS showing the weakest performance.

On the other hand, when we consider classification problems, the results are not convincing in favor of the fitness-case sampling methods. In all problems, either synthetic or

real-world, none of the tested algorithms could improve upon the performance of standard GP. In fact only two methods achieved the same performance, with LEX and KW-IS never producing worse results. Moreover, while IS and RIS exhibited the smallest amount of overfitting, this result was not satisfactory since their test performance was significantly worse than GP, LEX and KW-IS on most problems. Finally, an unexpected result was that the fitness-case sampling methods showed the same amount of code growth than standard GP, except for IS and RIS.

In conclusion, the main recommendations that can be drawn from these results are the following. First, LEX, KW-IS and IS are useful fitness-case sampling methods that can improve GP performance on difficult real-world symbolic regression problems. Second, for classification tasks standard GP is still recommended, LEX and KW-IS will not degrade or improve performance, while IS and RIS do not seem to be appropriate choices for this domain. Therefore, a real-world GP-based tool should include as a configurable option the use of either LEX or KW-IS in both problem domains studied here, while IS is also a good choice for real-world symbolic regression tasks.

Finally, future work related to this work will focus on studying the behavior of the sampling methods on real-world symbolic regression problems where the training set is contaminated by *outliers*. It seems that these sampling methods could be used to help identify outliers in the data and help guide the GP search to the desired solution.

Acknowledgement(s)

Funding for this work was provided by CONACYT Basic Science Research Project No. 178323, DGEST (México) Research Project 5414.14-P, FP7-PEOPLE-2013-IRSES project ACOBSEC financed by the European Commission with contract No. 612689 and CONACYT Project FC-2015-2/944 “Aprendizaje evolutivo a gran escala".

References

DERRAC, J., S. GARCIA, D. MOLINA et F. HERRERA. 2011, «A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.», *Swarm and Evolutionary Computation*, vol. 1, nº 1, p. 3–18. 77

DOUCETTE, J. et M. I. HEYWOOD. 2008, «Gp classification under imbalanced data sets: Active sub-sampling and auc approximation», dans *Proceedings of the 11th European Conference on Genetic Programming*, EuroGP'08, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-78670-8, 978-3-540-78670-2, p. 266–277. 72

FRIEDMAN, M. 1937, «The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance», *Journal of the American Statistical Association*, vol. 32, nº 200, p. 675–701, ISSN 01621459. 77

GATHERCOLE, C. et P. ROSS. 1994a, «Dynamic training subset selection for supervised learning in genetic programming», dans *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature*, PPSN III, Springer-Verlag, London, UK, UK, ISBN 3-540-58484-6, p. 312–321. 72, 73

GATHERCOLE, C. et P. ROSS. 1994b, «Some training subset selection methods for supervised learning in genetic programming, presented at ecai'94 workshop on applied genetic and other evolutionary algorithms», . 72, 73

GIACOBINI, M., M. TOMASSINI et L. VANNESCHI. 2002, «Limiting the number of fitness cases in genetic programming using statistics», dans *Proceedings of the 7th International*

Conference on Parallel Problem Solving from Nature, PPSN VII, Springer-Verlag, London, UK, UK, ISBN 3-540-44139-5, p. 371–380. 72

GONÇALVES, I. et S. SILVA. 2011, «Experiments on controlling overfitting in genetic programming», dans *Proceedings in Artificial Intelligence, 15th Portuguese Conference on Artificial Intelligence, EPIA 2011, October 10-13.*, Lecture Notes in Computer Science, Springer, ISBN 978-3-642-24768-2. 73

GONÇALVES, I. et S. SILVA. 2013, «Balancing learning and overfitting in genetic programming with interleaved sampling of training data», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 7831, édité par K. Krawiec, A. Moraglio, T. Hu, A. Etaner-Uyar et B. Hu, Springer Berlin Heidelberg, ISBN 978-3-642-37206-3, p. 73–84. 72, 73, 75, 77, 88

HOLM, S. 1979, «A simple sequentially rejective multiple test procedure», *Scandinavian Journal of Statistics*, vol. 6, p. 65–70. 77

KOZA, J. 2010, «Human-competitive results produced by genetic programming», *Genetic Programming and Evolvable Machines*, vol. 11, nº 3, p. 251–284. 72, 76

LASARCZYK, C. W. G., P. W. G. DITTRICH et W. W. G. BANZHAF. 2004, «Dynamic subset selection based on a fitness case topology», *Evol. Comput.*, vol. 12, nº 2, p. 223–242, ISSN 1063-6560. 72, 73

LICHMAN, M. 2013, «UCI machine learning repository», URL `http://archive.ics.uci.edu/ml`. xxiv, 77, 88, 90

MARTÍNEZ, Y., E. NAREDO, L. TRUJILLO et E. GALVÁN-LÓPEZ. 2013, «Searching for novel regression functions», dans *2013 IEEE Congress on Evolutionary Computation*, ISSN 1089-778X, p. 16–23, doi:10.1109/CEC.2013.6557548. xxiv, 73, 75, 77

MARTÍNEZ, Y., L. TRUJILLO, E. NAREDO et P. LEGRAND. 2014, «A comparison of fitness-case sampling methods for symbolic regression with genetic programming», dans *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, *Advances in Intelligent Systems and Computing*, vol. 288, édité par A.-A. Tantar et collab., Springer International Publishing, p. 201–212. 72, 73, 75, 76

MCDERMOTT, J., D. R. WHITE, S. LUKE, L. MANZONI, M. CASTELLI, L. VANNESCHI, W. JASKOWSKI, K. KRAWIEC, R. HARPER, K. DE JONG et U.-M. O'REILLY. 2012, «Genetic programming needs better benchmarks», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, ACM, New York, NY, USA, p. 791–798. xxiv, 77

SILVA, S. et J. ALMEIDA. 2003, «GPLAB - A Genetic Programming Toolbox for MATLAB», *In Gregersen L (ed), Proceedings of the Nordic MATLAB Conference*, p. 273—278. 77

SILVA, S. et E. COSTA. 2009, «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories», *Genetic Programming and Evolvable Machines*, vol. 10, nº 2, p. 141–179. 76

SPECTOR, L. 2012, «Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, ACM, ISBN 978-1-4503-1178-6, p. 401–408. 73, 74

TRAWINSKI, B., M. SMETEK, Z. TELEC et T. LASOTA. 2012, «Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms.», *Applied Mathematics and Computer Science*, vol. 22, nº 4, p. 867–881. 77

TRUJILLO, L., L. SPECTOR, E. NAREDO et Y. MARTÍNEZ. 2013, «A behavior-based analysis of modal problems», dans *Genetic and Evolutionary Computation Conference, GECCO '13, Amsterdam, The Netherlands, July 6-10, 2013, Companion Material Proceedings*, p. 1047–1054. 74

UY, N. Q., N. X. HOAI, M. O'NEILL, R. I. MCKAY et E. GALVÁN-LÓPEZ. 2011, «Semantically-based crossover in genetic programming: application to real-valued symbolic regression», *Genetic Programming and Evolvable Machines*, vol. 12, nº 2, p. 91–119. xxiv, 77

ZHANG, M. et W. SMART. 2006, «Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification», *Pattern Recogn. Lett.*, vol. 27, nº 11, p. 1266–1274. 88

Chapter 5

Evolving Genetic Programming Classifiers with Novelty Search

This chapter is related to the PhD thesis of Enrique Naredo (ITT Tijuana) and has been published in Information Sciences, Elsevier, 2016, 369, pp.347-367. Work carried out with Enrique Naredo, Leonardo Trujillo, Sara Silva and Luis Muñoz.

Contents

Abstract

Novelty Search (NS) is a unique approach towards search and optimization, where an explicit objective function is replaced by a measure of solution novelty. However, NS has been mostly used in evolutionary robotics (ER) while its usefulness in classic machine learning problems has not been explored. This work presents a NS-based Genetic Programming (GP) algorithm for supervised classification. Results show that NS can solve real-world classification tasks, the algorithm is validated on real-world benchmarks for binary and multiclass problems. These results are made possible by using a domain-specific behavior descriptor. Moreover, two new versions of the NS algorithm are proposed, Probabilistic NS (PNS) and a variant of Minimal Criteria NS (MCNS). The former models the behavior of each solution as a random vector and eliminates all of the original NS parameters while reducing the computational overhead of the NS algorithm. The latter uses a standard objective function to constrain and bias the search towards high performance solutions. This chapter also discusses the effects of NS on GP search dynamics and code growth. Results show that NS can be used as a realistic alternative for supervised classification, and specifically for binary problems the NS algorithm exhibits an implicit bloat control ability.

5.1 Introduction

Evolutionary algorithms (EAs) are a broad family of search and optimization algorithms that are based on a simplified model of Neo-Darwinian evolution EIBEN et SMITH [2007], achieving impressive results in many domains KOZA [2010]. The bio-inspired origins of EAs suggest a substantial difference with respect to traditional optimization approaches. However, EAs are guided by an objective function and specially designed search operators just like most optimization algorithms LUKE [2013]. The use of an objective function in EAs is a key difference with respect to natural evolution, which is an open-ended process that lacks a predefined purpose.

Open-ended artificial evolution does not use an objective function to drive the search, at least not an explicit one. An important feature of open-ended systems is the continuous emergence of novelty BANZHAF [2014a,b]. In fact, some of the earliest EAs were open-ended DAWKINS [1996], but they have mostly been used in specialized domains such as artificial life OFRIA et WILKE [2004] and interactive search KOWALIW et collab. [2012]. Only recently has open-ended search been proposed to solve mainstream problems, one promising algorithm is Novelty Search (NS) proposed by Lehman and Stanley LEHMAN et STANLEY [2008]. NS was conceived to overcome deception in evolutionary robotics (ER) LEHMAN et STANLEY [2008, 2010a, 2011a], a common issue in most challenging problems WHITLEY [1991].

Lehman and Stanley relate deception with problem hardness, stating that "[a] deceptive problem is one in which a reasonable EA will not reach the desired objective in a reasonable amount of time" LEHMAN et STANLEY [2011a] (p.193). The core idea behind NS is that using an objective function to determine fitness in challenging problems may mislead the search and prevent it from reaching a global optima. Therefore, the proposal of NS is to abandon the objective function as the source of selective pressure, and instead determine selective pressure based on the novelty or "uniqueness" of each individual by considering a description of the behavior each individual exhibits. From the NS perspective a behavior refers to a description of the interaction between a candidate solution and its domain-specific context.

NS has achieved promising results in different areas of ER WOOLLEY et STANLEY [2012], such as navigation GOMES et collab. [2013]; LEHMAN et STANLEY [2008, 2010a,b, 2011a]; URBANO et LOUKAS [2013]; URBANO et collab. [2014], morphology design LEHMAN et STANLEY [2011b] and gait control LEHMAN et STANLEY [2011a]. Despite the growing evidence that NS can be used as an alternative to traditional objective-based search (OS), we conjecture that it is not yet widely used for the following reasons. First, most work on NS has been

limited to ER, providing little insight regarding the competence of NS in other areas, particularly in common machine learning problems. Second, NS introduces several additional algorithm parameters that must be heuristically tuned. Third, NS relies on a kernel method to estimate the uniqueness of each new solution based on its dissimilarity with previously generated solutions. Such an approach leads to a high computational overhead, which is normally solved with additional heuristics. Finally, NS has been shown to struggle when behavioral space is large KISTEMAKER et WHITESON [2011], the search for specific behaviors in these cases can become very slow while the algorithm explores many uninteresting solutions. To address this problem, Lehman and Stanley proposed an extension to NS called Minimal Criteria NS (MCNS) LEHMAN et STANLEY [2010b], where a solution is considered to be novel only if it is unique and satisfies some domain-specific minimal criteria reducing in this way the portion of behavioral space that should be explored.

The present work builds on previous contributions to extend the NS paradigm. Firstly, we apply NS on supervised classification with genetic programming (GP) and propose a behavior descriptor for evolved GP classifiers, whereas previous works on NS have focused mainly on ER. The NS approach is tested on twelve real-world datasets, considering binary and multiclass problems and using two different GP-based classifiers. Secondly, an extension to the basic NS algorithm is proposed, where the novelty of a solution is estimated probabilistically by modelling each behavior as a random vector. The proposed strategy is called probabilistic novelty search (PNS), which reduces the computational cost of the original NS algorithm, and all the parameters introduced by NS are eliminated. Thirdly, several NS variants are extensively tested and compared, including NS, MCNS and PNS. Results show that NS-based GP can perform competitively relative to a standard OS, while endowing the search with implicit bloat control in some cases. Preliminary results of this research were presented in MARTÍNEZ et collab. [2013]; NAREDO et TRUJILLO [2013]; NAREDO et collab. [2013]; TRUJILLO et collab. [2013a,b]; however, those works only studied the general applicability of the original NS algorithm on synthetic pattern recognition problems without considering any algorithmic improvements or real-world scenarios. Nonetheless, those works served as a proof-of-concept for the proposed approach, which is fully explored and evaluated in the current chapter. In summary, the work presented here will help establish NS as a viable alternative for GP-based systems.

The remainder of this chapter is organized as follows. Section 5.2 provides the required background for this work, an overview of GP is given and the concept of behaviors in GP is introduced, discussing how it relates to objective-based fitness and semantics as understood within GP literature. Section 5.3 describes the NS algorithm and the proposed MCNS variant. Section 5.4 presents our basic approach towards applying NS with a GP-based classifier. Afterwards, the proposed PNS is described in Section 5.5. The experimental setup and results are presented in Section 5.6. Finally, Section 5.7 contains conclusions and future work.

5.2 Background

This section introduces GP, analyzes the search spaces used by GP, and introduces as well the concept of behavior in GP which can be related with an open-ended algorithm such as NS.

5.2.1 GP Search

One of the central challenges of computer science was stated in the 1950s by Arthur Lee Samuel KOZA [1992]; "how can computers be made to do what needs to be done, without being told exactly how to do it?". The evolutionary computation paradigm called GP is a generalization of genetic algorithms (GA) that provides a noteworthy proposal to address

this challenge. It is able to create computer programs from a high-level problem statement, a process which is also called program synthesis or automatic program induction KOZA [1992].

GP is a domain-independent evolutionary method intended to solve problems without requiring the user to know or specify the form or structure of the solution in advance POLI et collab. [2008]. GP is inspired in natural genetic operations, applying similar operations to computer programs, such as crossover (sexual recombination), mutation and reproduction. Computer programs can be represented in several ways, but since trees can be easily evaluated in a recursive manner this is the traditional representation used in the literature KOZA [1992]. GP-trees are composed by two different types of nodes known as functions and terminals. Function are internal tree nodes that represent the primitive operations used to construct more complex programs, such as standard arithmetic operations, programming structures, mathematical functions, logical functions, or domain-specific functions KOZA [1992]. Terminal nodes are at the leaves of the GP-trees and usually correspond to the independent variables of the problem, zero-arity functions or random constants. There are other GP versions that use different representations, such as linear genetic programming BRAMEIER et BANZHAF [2010] or cartesian genetic programming PETER [2000]. Other examples include μGP SQUILLERO [2005] that evolves programs in a given assembly language, while other authors have developed special languages for GP-based evolution such as Push, which is used to implement the PushGP system SPECTOR et ROBINSON [2002].

Like other EAs, GP starts with a population of randomly created individuals, which in this case represent programs or mathematical functions. Inspired in the Darwinian principle of natural selection GP traditionally applies the maxima of "survival of the fittest" to guide the search. Every generation, the individuals in the population are evaluated through an objective function to determine their fitness, usually measuring how close a given computer program is to achieving the desired goal. The fitness score is then used in the selection process to choose individuals as parents to generate the next generation of candidate solutions. Chosen parents are passed through genetic-like operations to share genetic information (between two parents) or to mutate any of their genetic material. There are three termination criteria traditionally used in GP, first two are related with respect to an upper limit reached: a maximum number of either generations or evaluations of the fitness function. The third criterion is to stop the search when the chance of improvement in the next generations is excessively low. After at least one of the termination criteria is satisfied, the best program produced during the whole run, usually named as the best-so-far, is returned as the result of the run.

While the objective function is usually used to guide the search, other approaches are possible. Moreover, the process by which each individual is evaluated reveals that the search is concurrently exploring several spaces. These issues are discussed next.

5.2.2 Search Spaces in GP

It is known that many EAs concurrently sample three different spaces during a search: genotypic, phenotypic and objective space. In the case of traditional tree-based GP, the genotypic space corresponds to the syntactic space and selective pressure is given in objective space by the objective function designed for the problem. Typically, the genotype of a computer program K is given by its syntax (GP-tree), as shown in Figure 5.1. Syntactic space contains all possible computer programs which can be composed by the set of functions and terminals used in the search. The syntax receives as input the information from the vector of terminals $\mathbf{x}$. In GP, the phenotype corresponds to the algorithm which is implicitly implemented by the syntax; though, the phenotype is rarely analyzed explicitly by most GP systems MCDERMOTT et collab. [2011] since extracting it is not a trivial task.

While these spaces have been the focus of most GP research, other spaces have recently been used to develop new GP-based algorithms. For instance, the computer program K

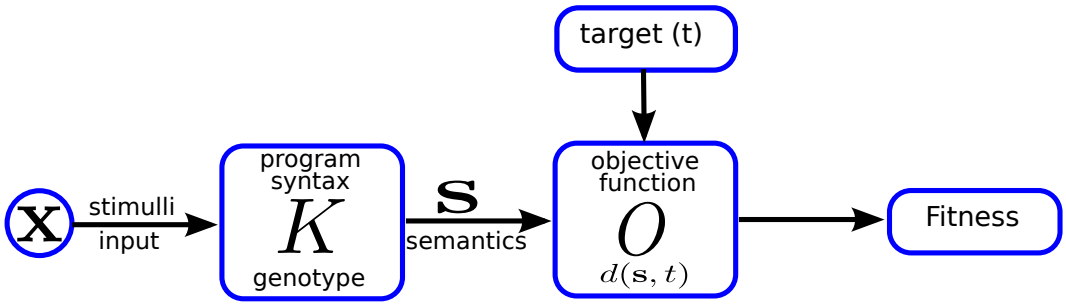


Figure 5.1 – The traditional program evaluation process used in GP, where K is the genotype of a computer program, $\mathbf{s}$ is the program output vector called semantics, O is the objective function which typically is defined as a distance $d(\mathbf{s}, t)$ between semantics and the expected target, obtaining then the fitness score assigned to the program K.

produces an output vector, named recently in GP literature as semantics MORAGLIO et collab. [2012]. Afterwards, the output semantics are transformed into a single quality measure by the objective function O which assigns fitness to K.

Formally, semantics can be defined as in MORAGLIO et collab. [2012]. Given a set of n fitness cases in a training set $T = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, the semantics of a program K is the corresponding set of outputs it produces, expressed as $\mathbf{s} = (K_j(\mathbf{x}_1), ..., K_j(\mathbf{x}_n))$. If we consider real-valued outputs, then $\mathbf{s} \in \mathbb{R}^n$. In this case, the objective function is usually defined as a distance $d(\mathbf{s}, \mathbf{t})$ between a program's semantics $\mathbf{s}$ and the desired target $\mathbf{t} = (y_1, ..., y_n)$, see Figure 5.1. The most interesting feature about semantic space is that it defines a unimodal fitness landscape. Through semantics, researchers have proposed new search operators that help improve the GP search. For instance, Beadle and Johnson BEADLE et JOHNSON [2008] proposed Semantically Driven Crossover (SDC) that promotes semantic diversity. Uy et al. UY et collab. [2011] proposed four different variants of semantically aware crossover operators for symbolic regression. Moraglio et al. MORAGLIO et collab. [2012] proposed Geometric Semantic GP (GSGP), designing special syntactic search operators that generate offspring that are mapped to geometrically bounded regions in semantic space. Recently, GSGP has been enhanced with local search mechanisms that improve performance and convergence CASTELLI et collab. [2015]. However, in some domains the target is known beforehand, (the optimal output -semantics- might not be known), instead what is measured is the effect that the output has on an external system, what we are referring to as the domain context.

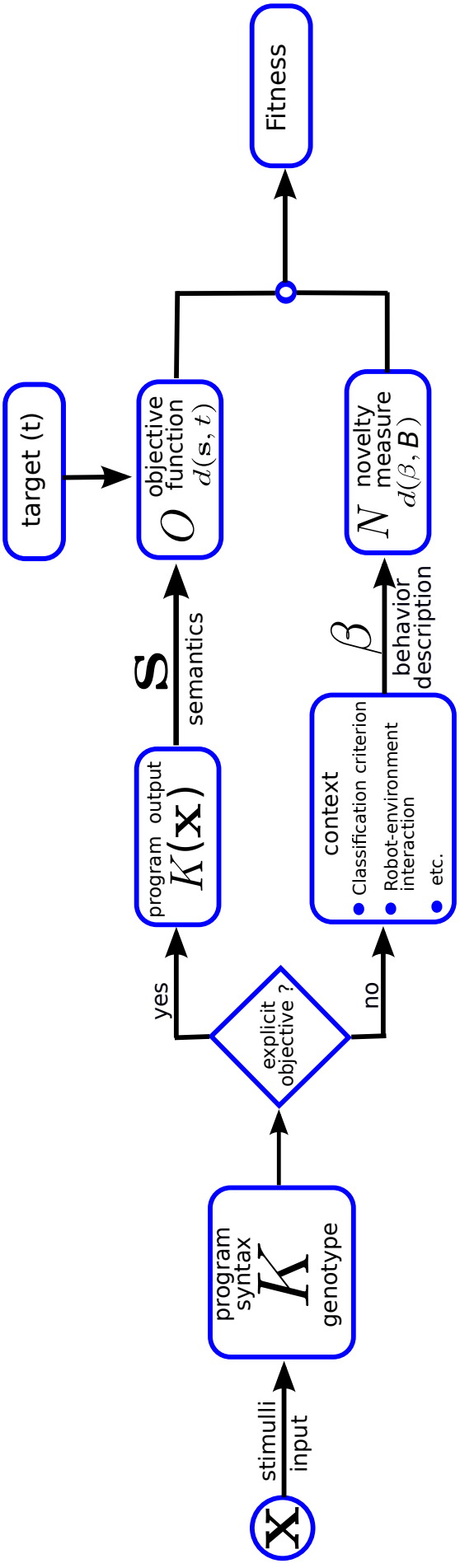


Figure 5.2 – The basic program evaluation process used in GP considering the choice of using the objective explicitly or implicitly. First option (yes), computes fitness through a distance function between the computer program s (semantics) and the expected target t, named as objective-based fitness. Second option (no), computes fitness through a behavior description β, then applying a distance function between each current behavior β and a set of behaviors B composed by current behaviors and previous behaviors to assign a novelty score as fitness, named as novelty-based fitness.

For example, consider the static range selection (SRS) GP classifier ZHANG et SMART [2006] (discussed in Section 5.4), which functions as follows. For a two class $\{\omega_1, \omega_2\}$ problem and real-valued GP outputs, the SRS classifier applies a simple classification rule $\mathcal{R}$: if the program output for input pattern $\mathbf{x}$ is greater than a threshold r, then the pattern is labeled as belonging to class ω_1, otherwise it is labeled as a class ω_2 pattern. In this case, while the semantics of two programs might be different (maybe substantially), they can still producing the same classification performance.

Indeed, these issues have led some authors to pose that the different mappings found and sometimes used in the GP process are the representation of different phenotypes mappings between them. For instance, MCDERMOTT et collab. [2011] suggests that the behavior of each component mapping can be studied separately. In this work, however, we take advantage of recent findings gained from the field of ER, to focus on the aggregate behavior of all these mappings and analyze the total effect that a given program has on a specific problem. The concept of behaviors has been widely used in robotics, particularly emphasized in the seminal works by Brooks from the 1980's BROOKS [1999], making it easy to integrate this concept in ER. In behavior-based robotics, for instance, an individual's behavior is described at a higher level of abstraction than the semantics of a particular controller, accounting not only for program output but the context in which the output was produced. The relation between input and semantic space can be injective or non-injective, while behaviors can allow for multivalued mappings or many-to-many relations between inputs and behavior space.

In ER, where GP can be used to evolve controllers that interface directly with a robot's actuators, the fitness functions used normally account for the observed high-level interactions between the robot and its environment. In fact, in a broad survey of types of fitness functions used in ER, Nelson et al. NELSON et collab. [2009] found that much of the research introduces *a priori* human knowledge when selecting a fitness function to solve a given problem. Nelson et al. group the fitness functions into seven classes, called training data, behavioral, functional incremental, tailored, environmental, competitive and aggregate fitness functions, ordered based on the amount of *a priori* knowledge incorporated into the function. The first class of fitness functions coincides with the most common approach taken in GP, where training data is used, the highest amount of *a priori* knowledge about the problem, the same as depicted in Figure 5.1. However, as Nelson et al. stated, this type of fitness functions require specific knowledge of what should be the optimal output, something that in many cases is not feasible or might be even unnecessary, as we argued in the example of the SRS classifier. Therefore, all other classes of fitness functions in ER, each to a different extent, incorporate the concept of behavior.

Since we can find some disagreement about the interpretation of the concept of behavior that we can find in different areas, we agree with the definition given in LEVITIS et collab. [2009] from the behavioral biology perspective, which states that a "behavior is the internally coordinated responses (actions or inactions) of whole living organisms (individuals or groups) to internal and/or external stimuli, excluding responses more easily understood as developmental changes" (p. 108). Considering the above definition, in this work we understand the concept of behavior as a measurable description about the internally coordinated external responses of a computer program K to internal and/or external stimuli $\mathbf{x}$ within a given environment or context. The behavior produced by a particular solution K is captured by a domain dependent descriptor β. In the ER case, context is given by the robot morphology and parts of the environment that cannot be sensed, while the inputs are the robot sensors and the outputs of the controller interface directly with the actuators. In this case the robot behavior descriptor can include such quantities as the robot position LEHMAN et STANLEY [2008], the robot velocity NELSON et collab. [2009] or patterns generated in the robot's path TRUJILLO et collab. [2008]. Conversely, for the classification problem described above, the context is provided by the specified classification rule $\mathcal{R}$, while one way to describe a classifier behavior can be related to the accuracy of the classifier on the training

Figure 5.3 – Conceptual view of how the performance of a program can be analyzed. At one extreme we have objective-based analysis, a coarse view of performance. Semantics lies at another extreme, where a high level of detail is sought. Finally, behavior analysis provides a variable scale based on how the problem context is considered.

instances. Afterward, the observed behavior β can be used to compute fitness by a function that considers the objective either explicitly or implicitly. The explicit approach is the customary way to compute fitness, using an objective function measuring how close the observed behavior is to a particular goal. The implicit approach can be to compute fitness based on the novelty or uniqueness of each behavior, such as in NS. The concept of behavior is graphically depicted in Figure 5.2, along with the manner in which behavioral information is included in the computation of the objective and fitness functions.

We propose that the behavior-based perspective should be seen as part of a scale of different forms of analyzing performance. In essence, the objective function, semantics, and behaviors are different levels of abstraction of a program's performance as shown in the Figure 5.3. At one extreme form of analysis, an objective function provides a coarse grained look at performance, a single value (for each criterion) that attempts to capture a global description of performance. At the other end of the analysis scale, semantics describe program performance in great detail, considering the raw outputs. On the other hand, behavior descriptors should be considered to be situated between fitness and semantics, providing a finer or coarser level of description depending on how behaviors are meaningfully characterized within a particular domain.

Moreover, the behavior-based approach allows us to modify the fitness computation in other ways. In this work we study the NS algorithm LEHMAN et STANLEY [2008]; STANLEY et LEHMAN [2015] that focuses selective pressure based on the concept of solution novelty. In particular, since NS was conceived for ER problems it measures novelty in behavioral space. This work extends previous NS-based contributions, to apply NS in the common machine learning task of supervised classification.

5.3 Novelty Search

NS introduces a new perspective to guide an evolutionary search, inspired by the open-ended nature of biological evolution STANLEY et LEHMAN [2015]. Lehman and Stanley conjectured that the objective function does not necessarily reward stepping stones in the search space that will ultimately lead to the desired goal, particularly in challenging problems LEHMAN et STANLEY [2008]. NS measures progress by focusing on the uniqueness or novelty of each new individual, which is a dynamic measure that depends on the search progress at any given generation.

Instead of designing an objective function that summarizes the performance of each individual, to use NS successfully the concept of uniqueness must be grounded in some way.

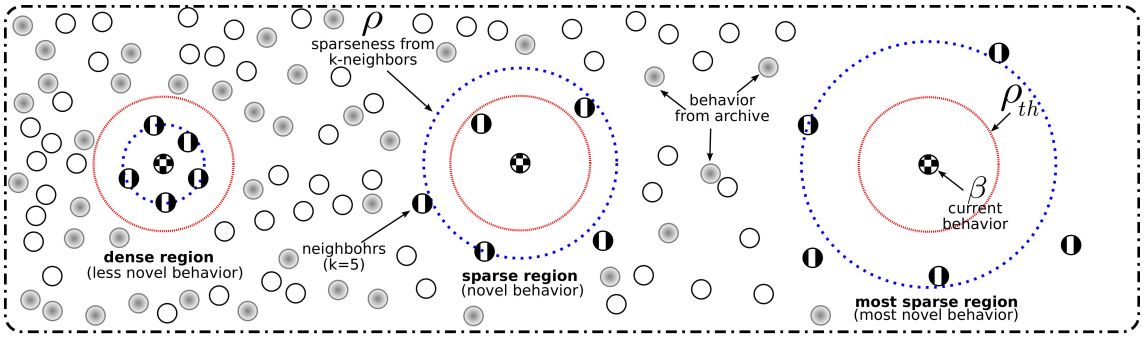


Figure 5.4 – The original NS proposed in LEHMAN et STANLEY [2008] uses a measure of local sparseness ρ around each individual behavior β within behavioral space to estimate its novelty, considering the current population and novel solutions from previous generations stored in an archive by mean of a threshold which depends on the sparseness measure ρ_{th} of the current individual behavior β. The figure shows three different scenarios for an individual's behavior β, where the cases are sorted from the most dense region (less novel) to the most sparse one (the most novel).

The uniqueness of a solution must be measured against the rest of the evolved solutions. For instance, solutions can be compared based on their genotype or phenotype. Instead, NS is based on the concept of behavioral space, where a behavior is characterized by a vector β that describes the way an agent K (or GP individual) acts in response to a series of stimuli (input) within a particular context. The authors proposed a measure of sparseness ρ around each individual K described by its behavior descriptor β to measure novelty, given by

$$\rho(\beta) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{d}(\beta, \alpha_i) \tag{5.1}$$

where α_i is the ith-nearest neighbor of β in behavioral space with respect to a distance or similarity measure $\mathrm{d}(.,.)$, and the number of neighbors k is an algorithm parameter LEHMAN et STANLEY [2008], ultimately the inverse of this value is taken since we want to maximize the novelty. Given this definition, when the average distance is large then the individual is located within a sparse region of behavioral space, and it is located in a dense region if the measure is small, see Figure 5.4. The original NS proposal considers the current population and an archive of individuals to compute sparseness to avoid backtracking. An individual is added to the archive if its sparseness satisfies a certain threshold condition ρ_{th}, which is the second NS parameter. Several papers have suggested implementing the archive as a FIFO queue of size q, this alleviates the cost of computing sparseness but adds another parameter.

5.3.1 Minimal Criteria Novelty Search

Lehman and Stanley proposed an extension to NS that evolves solutions more efficiently in very large behavioral spaces LEHMAN et STANLEY [2010b], called Minimal Criteria Novelty Search (MCNS). The main idea behind MCNS is that novelty should be preserved as long as it satisfies some minimal criteria (MC) for selection. Those individuals that meet the MC preserve their novelty score $\rho(\beta_i)$, and individuals that do not satisfy the MC will receive a penalized score, given by

$$\rho_{MCNS}(\beta_i) = \begin{cases} \rho(\beta_i), & \text{if the MC are satisfied} \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

For instance, for navigation in an unenclosed maze where the behavioral space can be very large a simple MC is that each individual must stay within the maze LEHMAN et STANLEY

[2010b]. The MC can be used as a tool to discard unfeasible solutions during the search, thereby directing the search towards the most promising regions in the search space URBANO et collab. [2014].

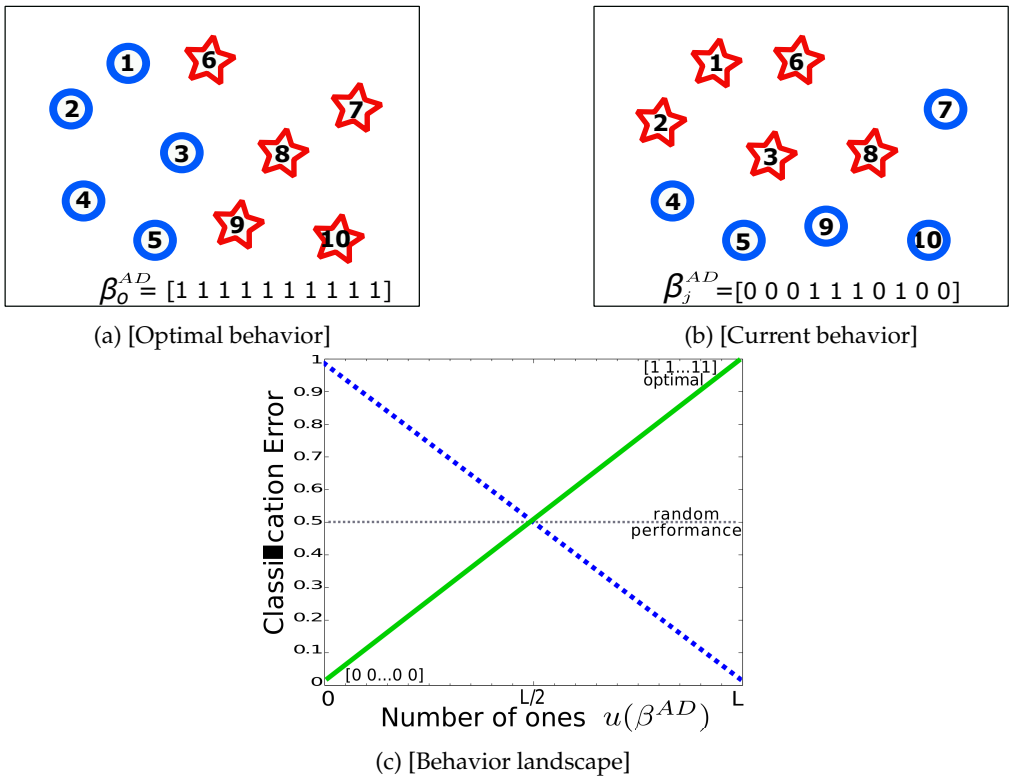
On the other hand, several works have attempted to combine novelty with an objective function. For instance, CUCCU et collab. [2011] and DOUCETTE et HEYWOOD [2010] proposed an arithmetic combination of both measures, and a novelty-based multiobjectivisation is proposed in MOURET [2011]. An extension of MCNS is proposed in GOMES et collab. [2012] named as Progressive Minimal Criteria NS (PMCNS), where the objective function is used as the basis for the MC using Equation 5.2. PMCNS applies a dynamic threshold, starting typically at zero and then increased according with a percentil P-th of the fitness scores in the current population ($0 \leq P < 100$). In this work, we propose an extension of PMCNS where the the dynamical threshold is computed with respect to the best-so-far solution, this version is named as MCNS$_{bsf}$ (best-so-far) and evolves the population through novelty but constrains the search by penalizing individuals that do not meet a MC that is based on the objective function. Considering a minimization problem, the MC of Equation 5.2 is satisfied if and only if $F(K_i) \leq F(K_{bsf})(p+1)$, where $F()$ is the objective function that assigns a quality score to each individual K_i, K_{bsf} is the best solution found so far and $p \in [0, 1]$. We choose a dynamic MC as a proportion of $F(K_{bsf})$ because if we use a static threshold we would need to set this threshold differently for each problem. For instance, if the value is too high then none of the individuals will satisfy the MC, which was observed in preliminary tests. Conversely, if it is too low then all individuals will satisfy the MC and it would become useless.

The proposed approach is a particular instance from a broader group of methods, where the MC is set proportionally to some statistic over the current population. In this case it is the fitness (based on the objective function value) of the best solutions found, but the method could set the MC based on the median, the mean or any other statistic such as in PMCNS. However, in this work we only consider the best solution since it provides the greediest approach to possibly improve convergence speed which is the original goal of MCNS, and it is the main difference with respect to PMCNS. Other variants will be studied in future work. Drawbacks of the proposal is that it introduces a new parameter p, in addition to those already used by NS. Moreover, if many individuals in a given generation are assigned a novelty measure of 0, the worst case, then the search might lack a sufficient gradient and could proceed randomly.

5.4 NS for Supervised Classification

To apply NS successfully, a behavior descriptor must first be proposed KISTEMAKER et WHITESON [2011]. For instance, in a maze navigation problem Lehman and Stanley, used the final robot position as the behavior descriptor LEHMAN et STANLEY [2008, 2010a,b, 2011a]. In this work, our main goal is to apply NS in GP-based classification. In particular, we use two GP classifiers: a simple binary classifier based on a static threshold ZHANG et SMART [2006] and a recently proposed multiclass approach INGALALLI et collab. [2014]; MUÑOZ et collab. [2015]. Both algorithms are wrapper methods that evolve features transformations of the form $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$, such that each input pattern $\mathbf{x} \in \mathbb{R}^n$ is transformed into a new feature vector $\mathbf{z} \in \mathbb{R}^p$ that is easier to classify using a predefined classification rule $\mathcal{R}$ which provides the context in this domain. For binary classification problems $\mathcal{R}$ defines a fixed threshold ZHANG et SMART [2006] while for multiclass classification problems $\mathcal{R}$ is based on the minimum Mahalanobis distance MUÑOZ et collab. [2015].

The proposed descriptor considers the accuracy of a classifier at a fine scale. Here we consider supervised classification problems specified by a training set T which contains sample patterns from each class. The accuracy descriptor (AD) is constructed in the following way. If $T = \{\mathbf{x}_1, ..., \mathbf{x}_L\}$, then the behavior descriptor for each GP individual K_i is a



(a) [Optimal behavior]

(b) [Current behavior]

(c) [Behavior landscape]

Figure 5.5 – Graphical depiction of the Accuracy Descriptor (AD): (a) shows the optimal behavior β_o^{AD}; (b) shows a possible behavior β_j^{AD}; and (c) shows the underlying fitness landscape based on behavioral space, where solid line represents a typical fitness landscape where a function $u(\beta^{AD})$ returns the number of ones, with a binary string full of ones, and dotted line represents the mirror fitness landscape built by the opposite behavior.

binary vector $\beta_i^{AD} = (\beta_1, \beta_2, ...\beta_j.., \beta_L)$ of size L, where each vector element β_j is set to 1 if the classification rule $\mathcal{R}$ correctly classifies the transformed features $\mathbf{z}_j$ corresponding to the training data $\mathbf{x}_j$ based on the output provided by K_i, and is set to 0 otherwise. The proposed descriptor is illustrated in Figure 5.5.

To understand the underlying behavioral space specified by AD, let's consider a binary classification problem. The AD descriptor specifies β as a binary vector, and suppose that function $u(\beta)$ returns the number of 1s in β. Let K_O be the *optimal* transformation that achieves a perfect accuracy on the training set based on $\mathcal{R}$. The AD of K_O is given by β^{AD} where $u(\beta^{AD}) = L$. For a two-class problem an equally useful solution is to take the opposite (complement) behavior and invert the classification, such that a 1 is converted to a 0 and vice-versa. This mirror behaviors is a β^{AD} with $u(\beta^{AD}) = 0$. These two optima are shown in the underlying behavior fitness landscape depicted in Figure 5.5(c), when fitness is given by the classification accuracy. For a problem with a reasonable degree of difficulty, the initial generations of a GP search should be expected to contain close to random classifiers, with roughly a 50% accuracy. Therefore, early population will mostly exhibit behavior descriptors with approximately equal proportions of zeros and ones. Then, NS will necessarily explore towards two opposite points in behavioral space.

Just like in semantic space, the specified fitness landscape based on behaviors defines a clear global optima Beadle et Johnson [2008, 2009]; Krawiec et Pawlak [2013]; Uy et collab. [2011]. However, for supervised classification semantic space would not be useful, since the global semantic optima is not known beforehand, it necessarily depends upon the training set T and the specified classification rule $\mathcal{R}$. It is important to mention that Moraglio et collab. [2012] also proposed geometric operators for GP-based classifiers based

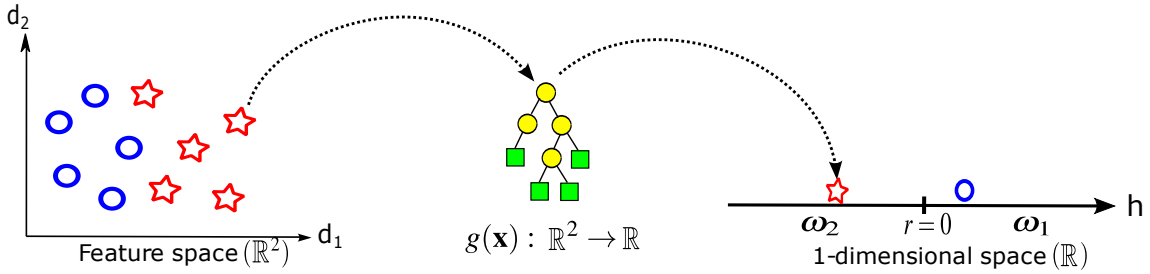


Figure 5.6 – Graphical depiction of SRS-GPC for a binary classification dataset described by 2 features, where GP evolves mapping functions $g(\mathbf{x})$ to map from the feature space $\mathbb{R}^2$ to 1-dimensional space $\mathbb{R}$. The classification rule $\mathcal{R}$ is defined by mean of a threshold r, which divides the 1-dimensional space into 2 regions and then each region is related with either class.

on semantics, however that formulation is not applicable for wrapper methods such as MUÑOZ et collab. [2015]; ZHANG et SMART [2006]. While semantic approaches have been widely used for symbolic regression CASTELLI et collab. [2015], they have not been fully studied in supervised classification. Finally, given the above binary descriptor, a natural distance function to applying NS with Equation 5.1 is the Hamming distance, that counts the number of bits that differ between two binary vectors.

5.4.1 Binary Classifier: Static Range Selection

For binary classification, we use the Static Range Selection GP Classifier (SRS-GPC) described by Zhang and Smart ZHANG et SMART [2006]. In a binary problem, a pattern $\mathbf{x} \in \mathbb{R}^n$ has to be classified as belonging to one of either two classes, ω_1 or ω_2. In this method, the goal is to evolve a mapping $g : \mathbb{R}^n \rightarrow \mathbb{R}$. The classification rule $\mathcal{R}$ states that pattern $\mathbf{x}$ is labeled as belonging to class ω_1 if $g(\mathbf{x}) > r$, and belongs to ω_2 otherwise. The fitness function is simple, it consists on maximizing the total classification accuracy after $\mathcal{R}$ is applied, normally setting the decision boundary to $r = 0$; the process is depicted in Figure 5.6.

5.4.2 Multiclass Classifier: M3GP

To test the NS approach on multiclass problems we choose the recently proposed Multi-dimensional Multiclass GP with multidimensional populations (M3GP) MUÑOZ et collab. [2015]. The goal of M3GP is to evolve a mapping $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$, where p is determined by the evolutionary process. The classification rule $\mathcal{R}$ used by M3GP consists of the following steps: (1) build M p-dimensional Gaussian clusters, one for each class, using the training set; (2) classify each training instance based on the minimum Mahalanobis distance to each class cluster, this provides the total accuracy used as fitness measure. For testing, the clusters found during training are used to determine class membership using the same distance measure. Results reported with M3GP are quite competitive with state-of-the-art methods, such as Random Forests BREIMAN [2001] and Random Subspaces BRYLL et collab. [2003]; HO [1998], providing a more advanced GP-based classification algorithm compared with SRS-GPC. Nonetheless, given the proposed behavior descriptor we can use the same NS algorithms in both cases. In essence, each classifier provides a different context for the outputs generated by the GP trees, just like previous works in ER that solved different navigation tasks in different environments using the same descriptor and NS algorithm LEHMAN et STANLEY [2008, 2010a,b, 2011a].

Summarizing, SRS-GPC and M3GP are used to evaluate the NS approach, respectively on binary and multiclass problems. For both algorithms the traditional approach will hereafter be referred to as objective-based search (OS).

5.5 Probabilistic NS

As stated in Section 5.3, NS suffers from some shortcomings which are addressed by the proposal developed in this section. In particular, computing novelty using Equation 5.1 can lead to several problems. First, it is not evident which value for the number of neighbors k will provide the best performance. Second, the computation of novelty based on sparseness showed in Equation 5.1, has a time complexity of $O((m+q)^2)$ where m is the size of population and q is the archive size, which will grow unbounded if it is not implemented as a FIFO queue LEHMAN et STANLEY [2008, 2010a,b, 2011a]. Third, it is no evident which individuals should be stored in the archive to avoid backtracking.

To overcome these issues, a probabilistic approach towards computing the novelty of each solution is proposed. The behavior descriptor β of a GP classifier is modeled as a binary random vector of size n, with n the number of fitness cases and an estimation of its probability mass function $P(\beta)$ is used to compute the novelty ϕ of each solution behavior β , given by

$$\phi(\beta) = \frac{1}{P(\beta)} \ , \tag{5.3}$$

such that the novelty of each solution is inversely proportional to the probability of producing it during the search. In this way, measuring novelty is accomplished without the need of empirically tuning any additional parameters in the GP search. The time complexity of computing ϕ is negligible once $P(\beta)$ is known and does not require the use of an external archive. To simplify this process further, we make the naive Bayesian assumption that the individual dimensions i in the behavior descriptor $\beta_{j,i}$ are independent; i.e. that the performance of a classifier K_j on a particular training sample is independent with its performance on any other. Under this assumption, ϕ can be computed as

$$\phi(\beta_j) = \frac{1}{\prod_{i=1}^{n} P_i(\beta_{j,i})} \ , \tag{5.4}$$

where $P(\beta_i)$ represents the probability mass function (pmf) of the i-th component β_i of β. Therefore, the problem then becomes estimating each $P(\beta_i)$ during the search, which is accomplished as follows. First, let B^t represent the behavior matrix of generation t, given by the Equation 5.5, such that B^t contains all behaviors $\beta^t_{j,i}$ from each individual j corresponding to the i-th fitness case at a generation t, with m individuals in the population and n fitness cases.

$$B^t = \begin{bmatrix} \boldsymbol{\beta}^t_1 \\ \vdots \\ \boldsymbol{\beta}^t_j \\ \vdots \\ \boldsymbol{\beta}^t_m \end{bmatrix} = \begin{bmatrix} \beta^t_{1,1} & \cdots & & \cdots & \beta^t_{1,n} \\ \vdots & \ddots & & & \vdots \\ & & \beta^t_{j,i} & & \\ \vdots & & & \ddots & \vdots \\ \beta^t_{m,1} & \cdots & & \cdots & \beta^t_{m,n} \end{bmatrix} , \tag{5.5}$$

$$\delta^t = \sum_{j=1}^{m} \beta^t_j = \begin{bmatrix} \delta^t_1 & \cdots & & \cdots & \delta^t_n \end{bmatrix} . \tag{5.6}$$

A second step is to compute the frequency of different behaviors in the first generation ($t = 0$) for each fitness case as shown in Equation 5.6. The frequency of 1s as behavior description related with a particular fitness case is computed by summing over each column of B^t, given by ${}_1\delta^{t=0}_i = \sum_{j=1}^{n}(\beta^{t=0}_{j,i} = 1)$, and the frequency of 0s is expressed as the complement ${}_0\delta^{t=0}_i = m - ({}_1\delta^{t=0}_i)$. The accumulated frequencies of 1's and 0's, are computed iteratively every generation t by; ${}_1\widehat{\delta}^t_i = {}_1\widehat{\delta}^{t-1}_i + {}_1\delta^t_i$, and ${}_0\widehat{\delta}^t_i = (t+1)m - {}_1\widehat{\delta}_i$, respectively.

Knowing the frequency of the different behaviors, the probability $P_i(\beta_{j,i})^t$ can be estimated by

$$P_i(\beta^t_{j,i} = 1) = \frac{{}_1\widehat{\delta}_i}{m(t+1)}, \tag{5.7}$$

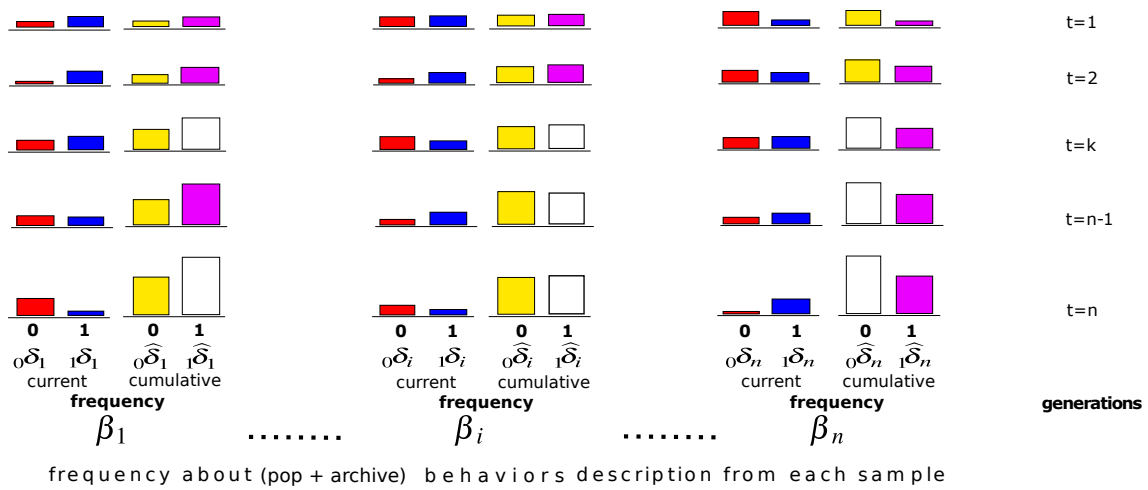


Figure 5.7 – Representation of the PNS novelty measure, where each column represents one feature βi of the AD vector and each row is a different generation. In each column, two graphics are presented, on the left is the frequency of individuals with either a 1 or 0 for that particular feature in the current population, and on the right the cumulative frequency over all generations. Behavior features with a high frequency of 1s correspond to fitness cases that are easy to classify, and vice versa.

for the 1s, and for the 0s it is given by

$$P_i(\beta^t_{j,i} = 0) = 1 - P_i(\beta^t_{j,i} = 1). \tag{5.8}$$

Then, the probabilistic novelty of a new behavior β^t_j that appears at generation t can be computed by

$$\phi_{PNS}{}^t_j = \prod_{i=1}^{n} \frac{1}{P_i(\beta^t_{j,i}) + \epsilon}\,, \tag{5.9}$$

where ϵ is a small real value to avoid numerical errors caused by divisions by zero. One way to set it is to use the inverse of the population size.

The general idea of the PNS measure for a behavior descriptor β with a binary string representation, is depicted in Figure 5.7.

5.6 Experimental Evaluation

In this section we present an experimental comparison of all the novelty-based variants discussed thus far, compared against a standard OS. The algorithms are compared on real-world classification datasets taken from several public datasets summarized in Table 5.1. The first eight datasets in Table 5.1 are used to pose binary classification problems, when datasets with 3 or more datasets are divided into independent binary classification problems using different combinations of classes (e.g., C1C2, C1C3, etc.) and solved with SRS-GP ZHANG et SMART [2006]. The last three datasets are solved with M3GP MUÑOZ et collab. [2015] as multiclass classification task. Datasets $1-5$ are balanced regarding the number of instances per class, while datasets 6 to 11 pose imbalanced problems.

In all, 4 different novelty-based algorithms are compared with OS, these are: NS, PNS, MCNS$_{bsf}$ and MCPNS$_{bsf}$, the latter of which is a hybrid that combines PNS with the MCNS$_{bsf}$. For the binary classification problems all algorithms use the parameters given in Table 5.2. Similarly, to compare with M3GP the parameters are slightly modified following MUÑOZ et collab. [2015], where: population size is 500, initialization uses the full method, crossover and mutation probabilities are 0.5, the function set is $F = \{+, -, \times, \div\}$, maximum depth is 17 levels and selection uses a lexicographic tournament of size 4 SILVA et ALMEIDA [2003].

Table 5.1 – Real-world datasets for binary and multiclass classification problems, taken from the UC Irvine Machine Learning Repository ♢, the U.S. geological survey (USGS) earth resources observation systems (EROS) data center ⊗ and from the KEEL dataset repository ⊙.

Type	No.	Dataset Name	Short	Classes	Attributes	Instances
Binary	1	♢Pima Indians Diabetes	Diab	2	8	536
	2	♢Iris	Iris	3	4	150
	3	♢Parkinsons	Parkin	2	22	96
	4	♢Teaching Assistant Evaluation	TAE	3	5	147
	5	♢Wine	Wine	3	13	144
	6	♢Cardiotocography	Cardio	3	21	2126
	7	♢Indian Liver Patient	Liver	2	10	579
	8	♢Fertility	Fertil	2	9	100
Multiclass	9	⊗Satellite dataset	IM-3	3	6	322
	10	⊙Segment	SEG	7	19	2310
	11	⊙Movement-Libras	M-L	15	90	360

Table 5.2 – Parameters for the GP systems.

Parameter	**Description**
Population size	50 individuals.
Generations	100 generations.
Initialization	*Ramped Half-and-Half*, with 6 levels of maximum depth.
Operator probabilities	Crossover $p_c = 0.8$, Mutation $p_\mu = 0.2$.
Function set	$\{+, -, \times, \div, \lvert\cdot\rvert, x^2, \sqrt{x}, log, sin, cos, if\}$.
Terminal set	$\{x_1, ..., x_i, ..., x_p\}$, where x_i is a dimension of the data patterns $\mathbf{x} \in \mathbb{R}^n$.
Hard maximum depth	20 levels.
Selection	Tournament of size 4.

This is a selection method similar to a "tournament", however if two individuals are equally fit, then the smallest one (the tree with less nodes) is chosen SILVA et ALMEIDA [2003]. All algorithms are executed 30 times and performance is analyzed based on the median value over all runs. In each run 70% of the data is used for training and the rest for testing, the data partition is randomly selected for each run. The objective function is given by the classification error, which is used by all NS variants to choose the best solution at the end of the run and by OS to guide the search and to choose the best solution.

For NS and MCNS$_{bsf}$, the behaviors from both the current and previous generations are taken into account to compute novelty according with Equation 5.1. Moreover, ρ_{th} is set to 50% of the largest possible distance as well as k-neighbors is set to 50% of the population size, and the archive is a FIFO queue with a size three times that of the population.

For the MCNS$_{bsf}$ algorithm, the minimal criterion for each individual is that its fitness must be within a certain percentage of the best solution found so far. Six versions of MCNS$_{bsf}$ are tested, from 5% to 30% (MCNS5 - MCNS30) of the fitness from the best-so-far solution, in increments of 5%. In this way, any solution that is more than x% worse than the best solution has its novelty value set to 0. Finally, all algorithms were coded using Matlab and the GPLAB toolbox SILVA et ALMEIDA [2003].

The algorithms are compared based on training error, test error (error on test data) and the mean size of all individuals in the population (hereafter referred to as mean population

Table 5.3 – Binary classification performance for all MCNS$_{best}$ variants. Table shows the median classification error on the test data for the best solution found, where bold indicates the best performance.

Dataset	MCNS5	MCNS10	MCNS15	MCNS20	MCNS25	MCNS30
Diab C1C2	**0.306**	0.325	0.309	0.328	0.338	0.322
Iris C2C3	0.100	0.100	**0.067**	0.100	0.100	0.100
Parkin C1C2	0.250	**0.214**	**0.214**	**0.214**	0.214	0.250
TAE C1C2	0.429	0.429	**0.393**	**0.393**	**0.393**	0.429
TAE C1C3	0.411	0.429	**0.321**	0.357	0.357	0.357
TAE C2C3	0.464	0.429	0.411	**0.375**	0.429	0.429
Wine C1C2	0.143	0.143	**0.125**	0.143	0.161	**0.125**
Wine C1C3	0.036	0.054	**0.000**	**0.000**	**0.000**	**0.000**
Wine C2C3	0.125	0.143	**0.036**	0.107	0.071	0.089
Cardio C1C2	0.126	**0.109**	0.119	0.110	0.113	0.111
Liver C1C2	0.298	0.309	**0.286**	0.298	0.289	0.292
Fertil C1C2	**0.138**	**0.138**	**0.138**	0.155	0.172	**0.138**

size). To verify statistical significance, the Friedman test with Bonferroni-Dunn correction of the p-values is performed between the control algorithm (OS) and each NS variant as suggested in DERRAC et collab. [2011]. In Tables 5.4 and 5.6 an asterisk indicates that a statical difference has been detected between the novelty-based search and OS at the $\alpha = 0.05$ significance level. The p-values of the statistical tests are given in Tables 5.5 and 5.7.

5.6.1 Results: Binary Classification

First, we analyze the performance of different versions of the MCNS$_{bsf}$ algorithm. Table 5.3 shows the median test error for all classification problems; each problem is denoted by an abbreviation (Diab, Parkin, etc.) followed by the classes used to pose the binary problem (C1, C2, C3). In general, most variants are quite similar, with MCNS15 achieving the overall best results. Therefore, for the sake of clarity and conciseness only MCNS15 is included in the subsequent comparisons with other methods and will simply be referred to as MCNS.

Figure 5.8 presents convergence plots of the median training error of the best solution. In general, OS converges quicker than most NS variants, with PNS showing the most similar convergence to OS. For some problems (plots i-l), OS shows slower convergence but PNS converge faster. Indeed, plots (a)-(h) correspond with well balanced problems. On the other hand, the problems of plots (i)-(l) present a larger class imbalance, suggesting that convergence of some NS algorithms is faster on unbalanced problems. However, NS shows the slowest convergence rates, in many cases not reaching the best training error found by other algorithms.

Figure 5.9 presents plots of how the mean size of the population evolves. It is clear that OS pushes GP toward larger program sizes on these problems, a trend that is particularly evident in plots (a)-(j). In general most NS variants produce smaller GP trees with some noteworthy tendencies. First, NS evolves smaller trees than OS in all cases except for the Fertility C1C2 problem and in most cases the difference is quite large. Second, MCNS also evolves very small trees, in some cases ((b),(c),(g),(h),(i)) the average program size is smaller than those generated by NS. This is a particularly encouraging result given the quality of the solutions found by MCNS. Third, PNS in some cases ((a),(c),(d),(e),(f),(g),(h)) also evolves smaller trees than OS. Finally, OS evolves smaller trees than most methods on two problems, Liver C1C2 and Fertility C1C2. It seems that this result may be correlated with the slow convergence of the search on these problems, possibly produced by the high class imbalance in both datasets.

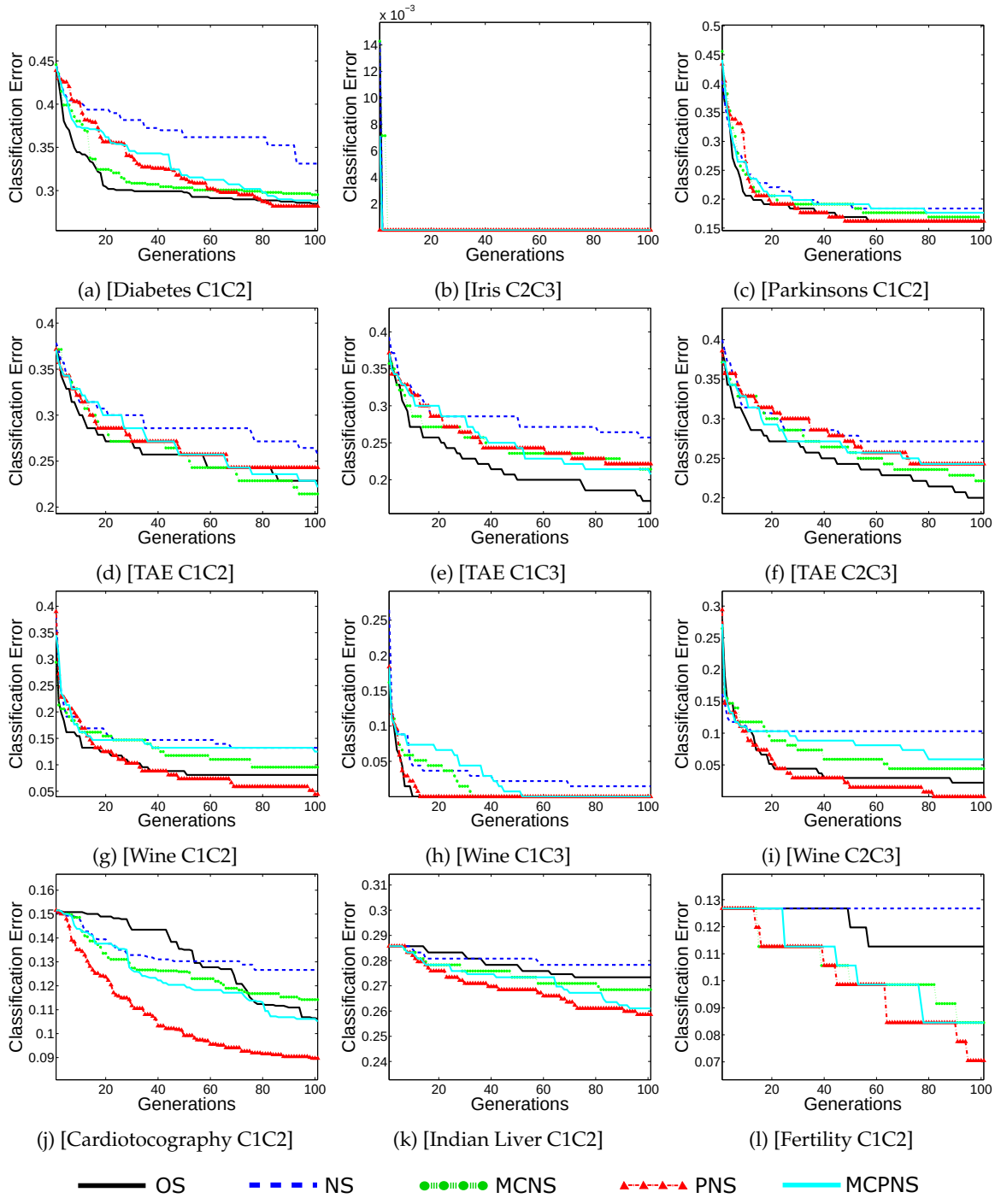


Figure 5.8 – Convergence of the classification error on the training data for the best solution found, showing the median value over all runs.

Numerical comparisons based on the test error and average program size are presented in Table 5.4. Moreover, the p-values of the statistical tests, comparing each method with the control method OS, are given in Table 5.5. To illustrate the performance differences among the methods, Figure 5.10 shows a box plot comparison of the test error from the best solution found in each run.

In general, all NS algorithms are very competitive relative to OS. In fact, all methods show no statistical difference based on test error, with only four exceptions: NS is worse on Wine C2C3; PNS is better on Cardio C1C2; PNS is worse on Liver C1C2; and MCPNS is worse on Fertility C1C2. Even in the cases where the differences are statistically signifi-

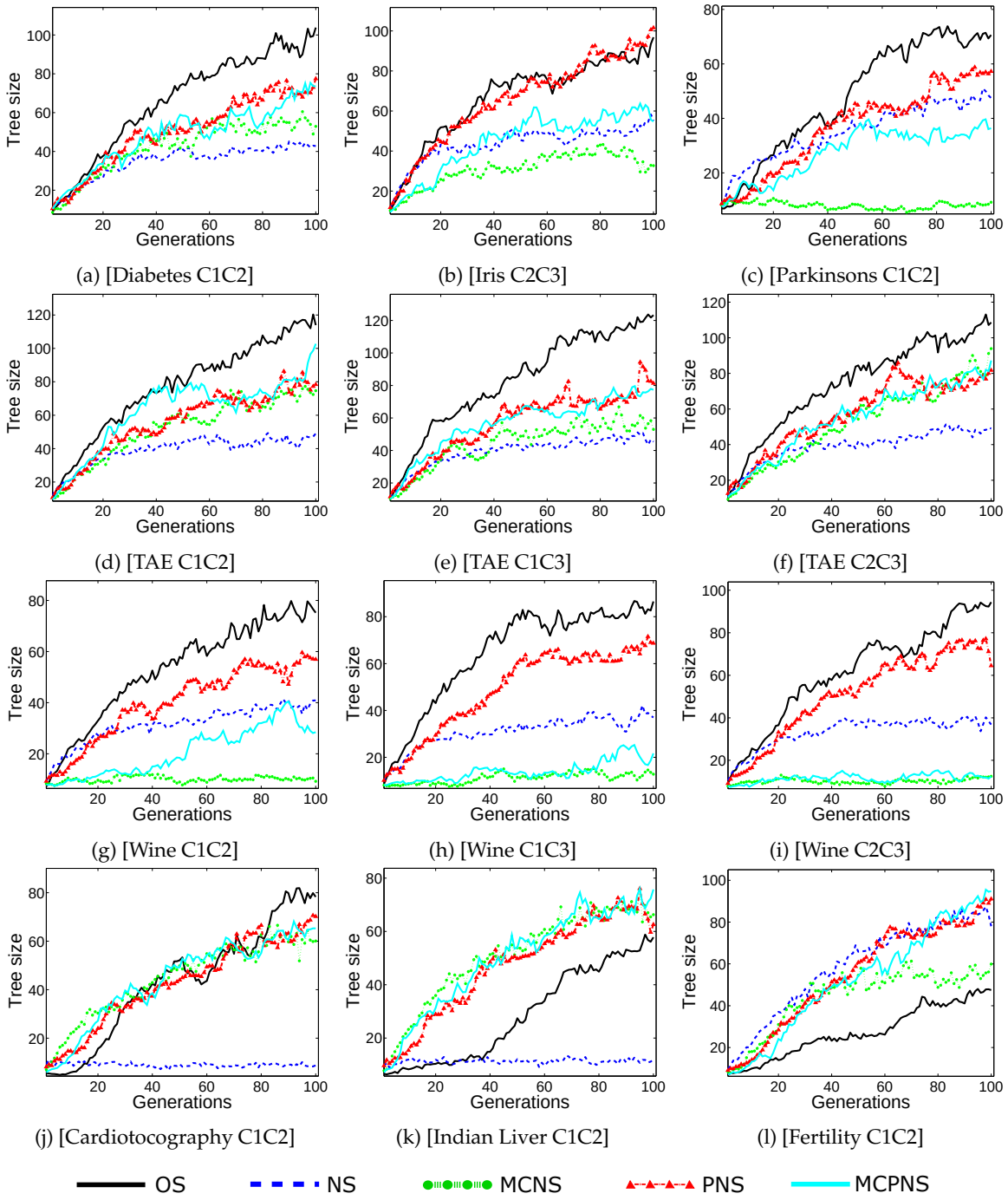


Figure 5.9 – Evolution of the average size of the population at each generation, showing the median value over all runs.

cant, the relative difference is rather low. However, when we consider the average program size it is clear that all NS variants evolve much smaller populations. In particular NS and MCNS show an intrinsic bloat-control property in this domain. There is one exception, in Fertility C1C2 NS evolves larger populations, which might be related to the class imbalance issue. Regarding PNS and MCPNS, both also control code growth on several problems with respect to OS, but the difference is less compared to NS and MCNS.

Finally, it is instructive to comment on the results for the most imbalanced problems, Liver C1C2 and Fertility C1C2. If we inspect the convergence plots in Figure 5.8, several trends appear. In both cases (plots (k) and (l)) the search performed by OS and NS seem

Table 5.4 – Binary classification performance, showing the median classification error on the test data (Test) for the best solution found, and the median of the average program size in the last generation (A-size). Statistically significant with respect to the control method with a p-value less than 0.05 is marked with an asterisk (*). Bold indicates the best performance.

Dataset	**Measure**	**OS**	**NS**	**MCNS**	**PNS**	**MCPNS**
Diabet	Test	0.331	0.341	**0.309**	0.347	0.334
C1C2	A-size	103.84	**42.77**∗	52.92∗	77.47	73.54
Iris	Test	**0.067**	**0.067**	**0.067**	**0.067**	0.100
C2C3	A-size	96.81	55.15∗	**32.69**∗	101.56	54.70
Parkin	Test	0.250	0.250	**0.214**	0.250	0.232
C1C2	A-size	70.61	47.48∗	**9.20**∗	57.26	36.40∗
TAE	Test	0.357	0.446	0.393	0.375	**0.321**
C1C2	A-size	113.91	**49.04**∗	74.66∗	78.39∗	102.74
TAE	Test	0.357	0.393	**0.321**	**0.321**	0.393
C1C3	A-size	123.30	**45.36**∗	52.93∗	81.36∗	77.32∗
TAE	Test	0.411	0.393	0.411	**0.375**	0.429
C2C3	A-size	108.64	**49.03**∗	93.94∗	81.73∗	87.27
Wine	Test	**0.107**	0.143	0.125	**0.107**	0.161
C1C2	A-size	75.20	40.93∗	**9.25**∗	56.97	28.45∗
Wine	Test	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
C1C3	A-size	86.51	37.03∗	**12.79**∗	68.67	21.71∗
Wine	Test	0.071	0.143∗	**0.036**	**0.036**	0.071
C2C3	A-size	94.43	36.83∗	12.46∗	64.51	**12.44**∗
Cardio	Test	0.117	0.128	0.119	**0.098**∗	0.112
C1C2	A-size	78.27	**8.79**∗	60.10	70.17	65.27
Liver	Test	**0.283**	0.289	0.286	0.306∗	0.295
C1C2	A-size	57.81	**11.91**∗	66.31	62.59	75.77
Fertil	Test	**0.103**	**0.103**	0.138	0.138	0.138∗
C1C2	A-size	**47.50**	78.01∗	59.91	91.07∗	94.75∗

to be stagnated, with only minimal improvements across generations. It is evident that these algorithms have converged to an almost trivial solution for imbalanced problems; i.e, assign to all (or a large majority) of the samples the class label of the majority class. In these cases, such trivial solutions can be generated randomly and will have the same objective score removing thus the selective pressure from the search and making OS function as a random search. This is consistent with the size of the evolved programs generated by OS on these problems, which are among the smallest of all the algorithms as shown in Figure 5.9. Conversely, the convergence plots for PNS, MCNS and MCPNS are substantially different, displaying a clear tendency towards incremental improvement during the training phase.

5.6.2 Discussion

Two aspects will be discussed; classification error (Test) and the average size (A-size) of the individuals in the population. First, Table 5.4 presents the average classification error on the test data which are consistent with those reported in NAREDO et collab. [2013], showing

Table 5.5 – Resulting p-values of the Friedman's test with Bonferroni-Dunn correction, for the binary classification problems using OS as the control method. The null hypothesis is rejected with a p-value less than 0.05, marked with an asterisk (*).

Dataset	Measure	NS	MCNS	PNS	MCPNS
Diabet	Test	2.86	0.27	2.86	3.41
C1C2	A-size	0.00*	0.01*	0.58	0.27
Iris	Test	0.182	0.057	0.629	1.269
C1C3	A-size	0.00	0.00	0.58	1.09
Parkin	Test	3.39	0.11	2.78	1.41
C1C2	A-size	0.04*	0.00*	1.09	0.00*
TAE	Test	0.07	0.71	0.24	3.39
C1C2	A-size	0.00*	0.00*	0.04*	1.09
TAE	Test	0.09	1.34	1.41	4.00
C1C3	A-size	0.00*	0.00*	0.00*	0.01*
TAE	Test	3.39	1.80	0.41	3.41
C2C3	A-size	0.00*	0.04*	0.01*	0.11
Wine	Test	2.26	2.78	4.00	3.39
C1C2	A-size	0.00*	0.00*	1.09	0.00*
Wine	Test	0.63	4.00	3.13	0.36
C1C3	A-size	0.00*	0.00*	0.11	0.00*
Wine	Test	0.01*	0.47	0.13	1.73
C2C3	A-size	0.00*	0.00*	0.58	0.00*
Cardio	Test	0.78	1.41	0.02*	0.33
C1C2	A-size	0.00*	0.58	1.09	1.86
Liver	Test	1.41	1.03	0.03*	0.05
C1C2	A-size	0.00*	1.09	1.09	0.58
Fertil	Test	0.79	0.16	0.29	0.01*
C1C2	A-size	0.00*	0.58	0.00*	0.01*

that NS performs well and achieves basically the same results compared to OS. However, there is a trend, NS seems to perform relatively better on the more difficult problems and worse on the easier ones. Basically, the explorative search performed by NS is fully exploited when random initial solution perform badly, in these conditions the search for novelty can lead towards better solutions. Conversely, for easy problems random solutions can perform quite well, thus the search for novelty can lead the search towards solutions with undesirable performance. A common criticism of Novelty Search is that it is efectively random or exhaustive search because it tries solutions in an unordered manner until a correct one is found VELEZ et CLUNE [2014]. Novelty Search is not a random or exhaustive search process, but instead is method to accumulate information through the time about the environment. Examples comparing NS against random search can be found on URBANO et collab. [2014]; VELEZ et CLUNE [2014].

The results are encouraging, especially if we consider the evolution of average program size which is related to bloat SILVA et COSTA [2009]. Figure 5.9 shows how the average size of all individuals in the population evolves across generations. First, consider OS that shows a typical GP behavior, with a clear tendency of code growth and consequently more bloat

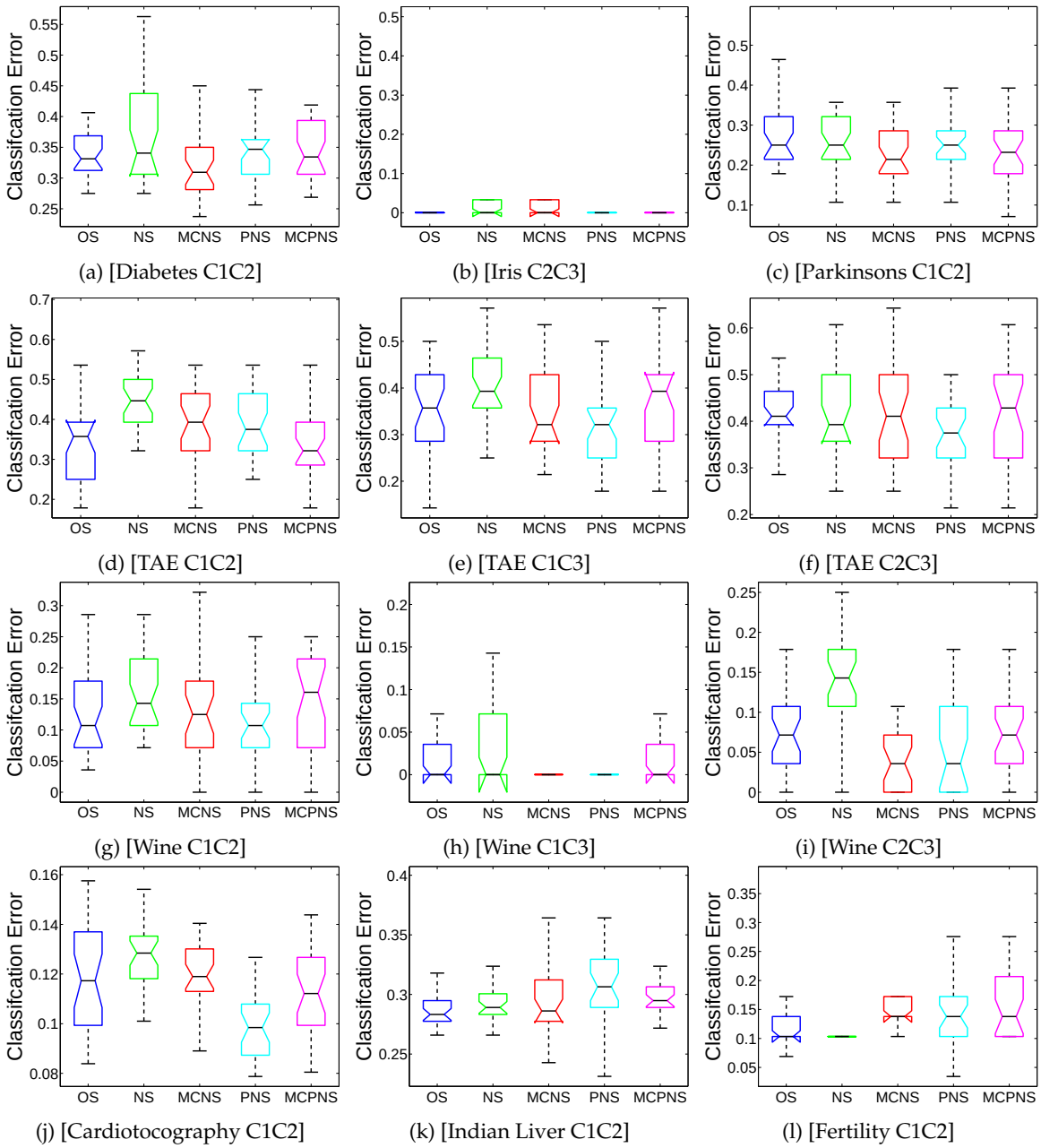


Figure 5.10 – Classification error on the test data for the best solution found, showing box plots of the median value over all runs.

specially on difficult problems. In the case of NS, it shows a control of the code growth quite effectively, exhibiting the same average program size on all problems.

Based on these results, we can revisit the fitness-causes-bloat theory of Langdon and Poli LANGDON et POLI [1997]. It basically states that the search for better fitness (given by the objective function) will bias the search towards larger trees, simply because there are more large programs than small ones. Silva and Costa SILVA et COSTA [2009] state it clearly:

> ... one cannot help but notice the one thing that all the [bloat] theories have in common, the one thing that if removed would cause bloat to disappear, ironically the one thing that cannot be removed without rendering the whole process useless: the search for fitness.

The above results correlate nicely with the fitness-causes-bloat theory, bloat was avoided by abandoning an explicit objective function.

Table 5.6 – Multiclass classification performance, showing the median classification error on the test data (Test) for the best solution found, and the median of the average program size in the last generation (A-size). Statistically significant with respect to the control method with a p-value less than 0.05 is marked with an asterisk (*). Bold indicates the best performance.

Dataset	Measure	OS	NS	MCNS	PNS	MCPNS
IM-3	Test	**0.046**	0.052	0.062	0.052	**0.046**
	A-size	66.22	71.08	55.05	55.42	**53.84**
SEG	Test	0.044	0.043	0.042	0.043	**0.041**
	A-size	111.10	143.61	**104.57**	138.48	123.01
M-L	Test	0.429	0.414	**0.394**	**0.394**	0.444
	A-size	13.05	12.77∗	12.82∗	**12.63**∗	12.69∗

Table 5.7 – Resulting p-values of the Friedman's test with Bonferroni-Dunn correction, for the multiclass classification problems using OS as the control method. The null hypothesis is rejected with a p-value less than 0.05, marked with an asterisk (*).

Dataset	Measure	NS	MCNS	PNS	MCPNS
IM-3	Test	1.73	0.57	2.78	2.12
	A-size	1.09	2.86	0.58	1.86
SEG	Test	2.86	1.41	0.28	1.79
	A-size	0.11	2.86	1.86	2.86
M-L	Test	1.79	0.37	1.86	3.36
	A-size	0.00∗	0.04∗	0.04∗	0.00∗

5.6.3 Results: Multiclass Classification

In these tests we use three problems (IM-3, SEG and M-L) with a different number of classes (3, 7 and 15). Moreover, the SEG problem has 2,310 instances which leads to a very large behavior descriptor; i.e., using 70% of the data for training we obtain a descriptor length of 1,617 bits, which gives a very large behavioral space. Assessing the performance of the NS variants on this problem is of particular interest, since previous works have shown that the performance of NS degrades when behavioral space is very large KISTEMAKER et WHITESON [2011].

The numerical comparison of the algorithms is given in Tables 5.6 and 5.7, the former shows the median test error and median population size, while the latter shows the corresponding p-values of the statistical tests. Similar to the binary case, all NS variants achieve basically the same performance as OS, with slight improvements on some problems (particularly M-L) but not statistically significant. Indeed the similarity in terms of performance is even more evident when we analyze the convergence plots for each problem shown in Figure 5.11, which shows how the classification error evolves for the best solution found on the training and testing sets. On the other hand, code growth is not controlled like in the previous tests. In only one problem (M-L) the NS variants produces statistically significant differences in terms of average program size.

5.6.4 Results: Analysis

The above results show that NS can be used to solve binary and multiclass classification problems, without a performance drop-off relative to standard OS. This was not expected,

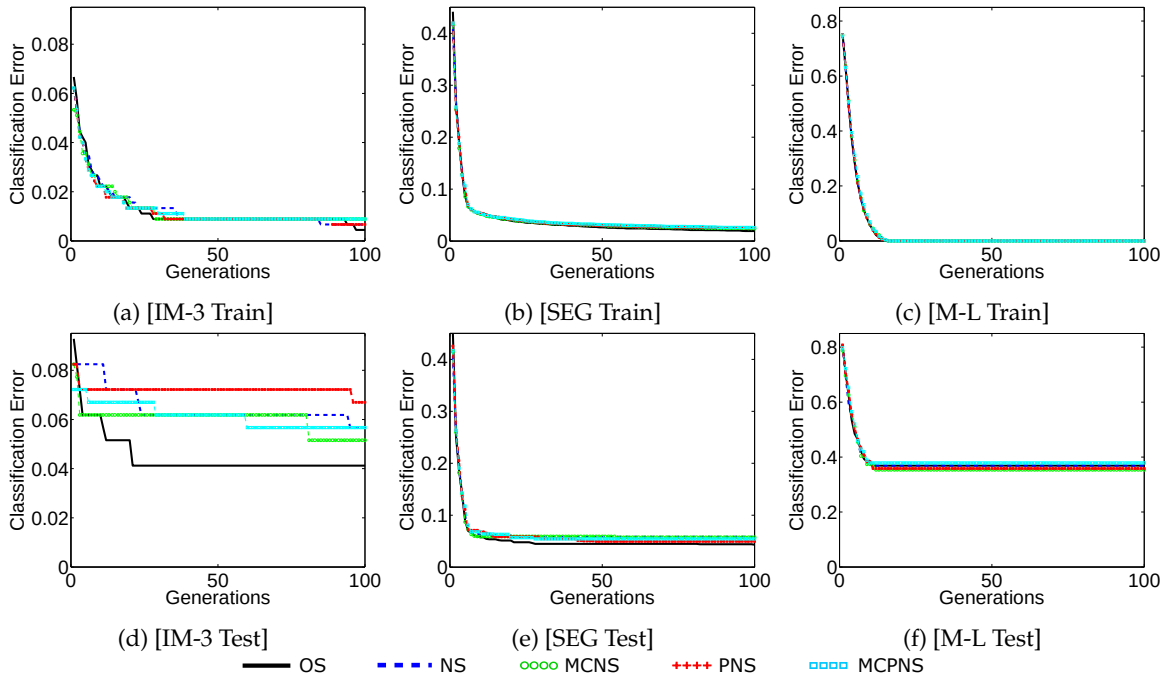


Figure 5.11 – Convergence of training and testing error for multiclass problems.

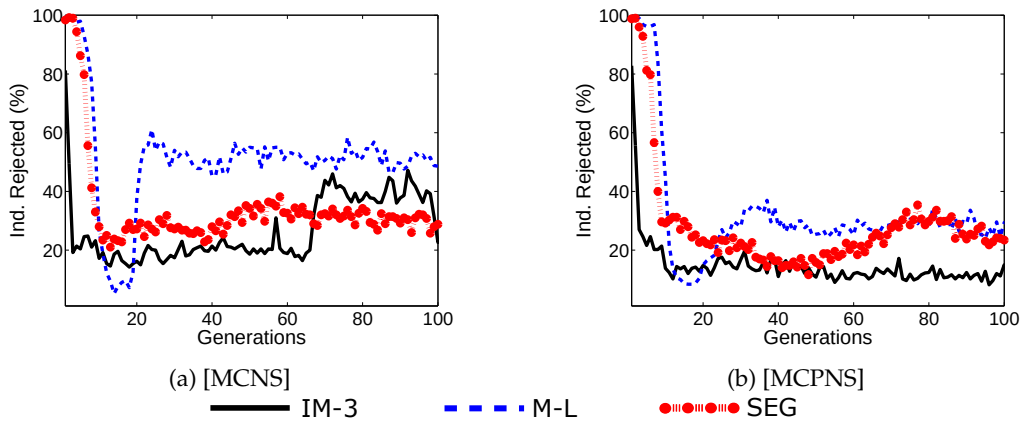
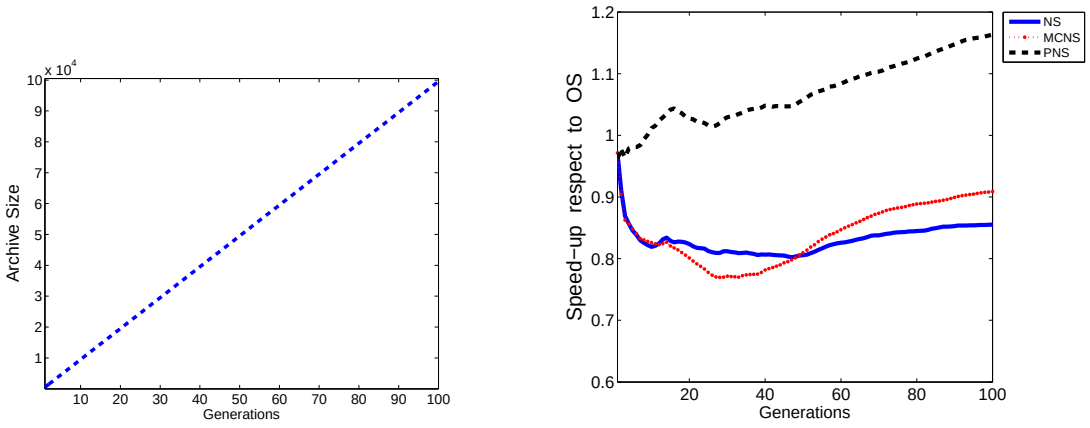


Figure 5.12 – Plot showing the percentage of rejected individuals, with respect to the population size, that did not satisfy the MC.

given that the search process omits the use of a standard objective function. Moreover, on some problems, mainly in binary classification tasks, the NS variants provide an intrinsic bloat control property.

Despite the similar performance achieved by the compared algorithms, this section provides a deeper analysis of the experimental results. Regarding the MC variants, namely MCNS and MCPNS, we show the impact of the penalty assigned in Equation 5.2 when the MC is not satisfied. Figure 5.12 plots the percentage of individuals that did not satisfy the MC at each generation, for all three multiclass problems. For both algorithms, we can see very similar patterns on all problems. At the beginning of the run, most individuals do not satisfy the MC, the number of rejected individuals then quickly declines and then more or less stabilizes after about 20 to 40 generations.

Another important aspect is to consider the computational effort of each NS method, rel-



(a) [Archive Size] (b) [Relative Speed-up]

Figure 5.13 – Analysis of NS variants on the IM-3 problem: (a) Archive size for NSn; (b) Relative speed-up for NS, PNS and MCNS.

ative to standard OS. As stated previously, PNS is posed as an approximation of the original NS algorithm working under a naive Bayesian assumption. PNS characterizes the complete distribution of behaviors but uses a possibly unsatisfiable assumption, while NS with a FIFO archive computes a more realistic measure of sparseness but sacrifices historical information to keep computational overhead low. One possible alternative would be to run NS without an archive limit, such that all individuals generated by the search are included into the archive. However, computing the NS in this scenario would surely have a detrimental effect on computational efficiency. Figure 5.13(a) shows how the size of the population archive grows when all individuals are stored during NS runs on the IM-3 problem, we refer to this variant as NSn. Then, Figure 5.13(b) plots the median speed-up of each method relative to OS ($Time_{OS}/Time_{NS}$) on the same problem, based on all runs. Values above unity represent a proportional reduction in total CPU time and values below unity represent an increase in total CPU time. In this plot we compare NS, PNS and MCNS. Results show the advantage of using PNS, with CPU run times showing a slight speed-up relative to OS of about 10%. On the other hand, NS and MCNS clearly pay the price of the more expensive sparseness estimation method. It is correct to assume that NSn would fair even worse, since it uses a much larger archive.

Finally, to compare the selective pressure applied by each algorithm, Figures 5.14 and 5.15 compare the relative ranking of each individual in the population for problems IM-3 and SEG; the plots for problem M-L are omitted because they are very similar to those of IM-3. The first row of plots in each figure (indices a, b and c) shows the percentage of individuals in the population that are ranked as the best solution of each generation (the top-ranked solutions). In Figures 5.14(a) and 5.15(a) the runs were guided by OS, in Figures 5.14(b) and 5.15(b) by NS, and in Figures 5.14(c) and 5.15(c) selective pressure was applied by PNS. Conversely, the second row of plots in these figures (indices d, e and f) shows the average ranking of the best individuals at each generation based on the other two fitness measures.

Figure 5.14(a) shows that OS promotes exploration in the first 25 generations, where the percentage of individuals tied for first place (top-ranked) is relatively small. Afterwards OS tends to converge, since the percentage of individuals tied for first place (top-ranked) grows quickly rising to about 20% of the total population. Notice that this 25 generation threshold coincides with the moment at which training error converges on this problem, as shown in Figure 5.11(a). Figure 5.14(d) takes these top-ranked individuals and computes new ranks for them based on the NS and PNS novelty measures, the plot then shows their average rank based on these methods. For NS, the plot shows that the ranking provided by the sparse-

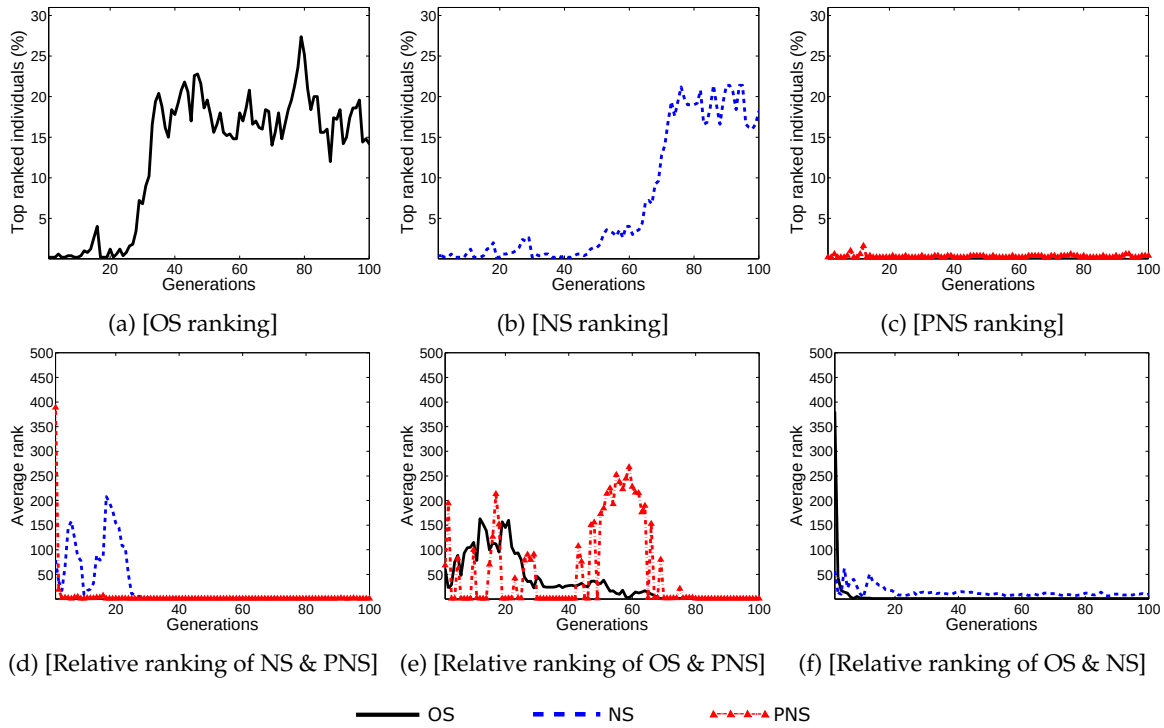


Figure 5.14 – Convergence of the best individuals for the **IM-3** problem in one run, analysing first the ranking given by the current method used in the search every generation, and then the relative ranking given by the other methods not used in the search. Figures at the top (a-c), show the convergence of the top ranked (best) individuals respect to the whole population for each method (OS, NS, and PNS). Figures at the bottom (d-e), show the relative ranking given by the other two methods (not used in the search) respect to the best solutions found by the current method (showed at the top) at every generation.

ness measure disagrees substantially with OS in the initial generations, reaching a peek at generation 20. Afterwards, the ranking of individuals by NS is basically equivalent with that of OS, this corresponds with the similar convergence plots of both algorithms. PNS, on the other hand, differs with OS only at the very beginning of the run, afterwards the ranking is the same across all generations showing in fact that PNS slow down the convergence process. Figure 5.14(b) and Figure 5.14(e) show a similar comparison, but in this case NS provides the selective pressure. Notice that the percentage of top-ranked individuals is very similar to OS, but in this case we can see larger disagreement in terms of ranking by the other methods. In particular, we can see that OS shows large disagreement at the beginning of the runs (up to generation 25), afterwards the differences start to reduce and are equal after generation 70. On the other hand, PNS ranking is more erratic, we can see that in some generations the ranking of the best individuals is in complete agreement with NS, while in other moments the average difference can become quite high (around generation 60). Finally, when PNS applies the selective pressure, Figure 5.14(c) and Figure 5.14(f), we can see a different pattern. First, the percentage of top-ranked individuals is always small, suggesting that PNS does not converge to many similar behaviors. Second, OS and NS mostly agree with PNS ranking after the first initial generations, but surprisingly NS shows a larger ranking difference than OS.

Figure 5.15 presents a similar analysis on the SEG problem with different results. The percentage of top-ranked individuals oscillates with OS, while NS and PNS exhibit similar patterns, with a small number of individuals achieving the top-rank every generation. When selective pressure is applied by OS, the relative ranking of NS and PNS is quite similar, with large differences at the beginning of the run and then converging to similar rankings

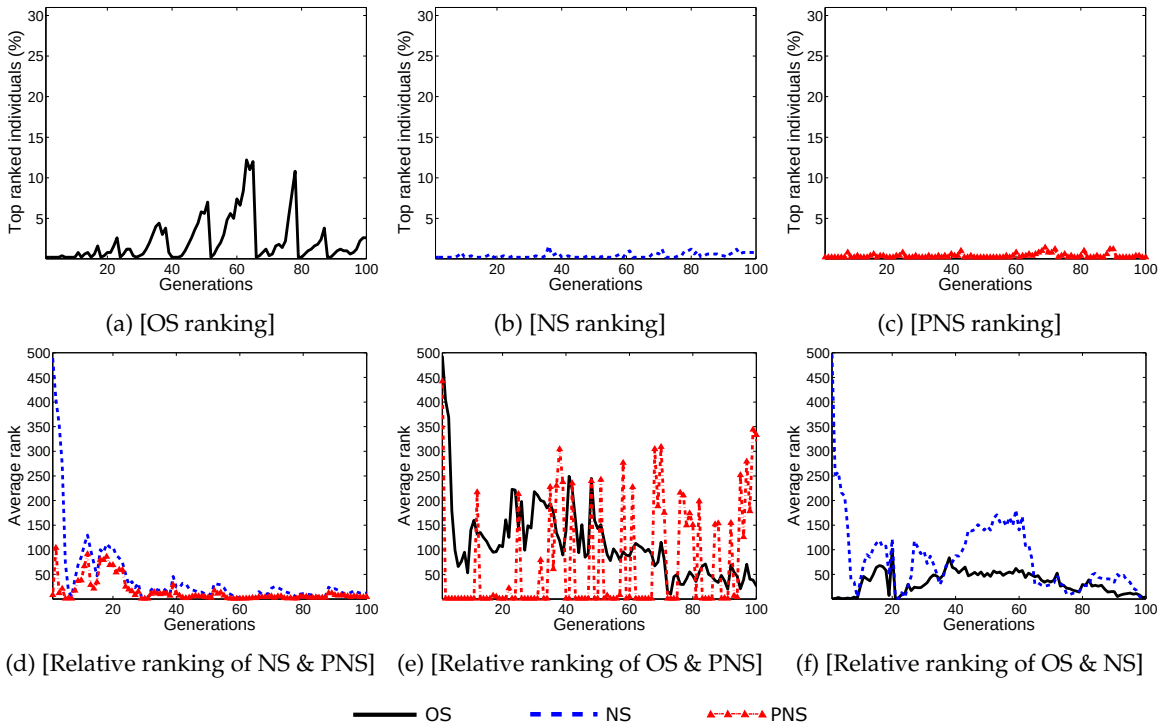


Figure 5.15 – Convergence of the best individuals for the **SEG** problem in one run, analysing first the ranking given by the current method used in the search every generation, and then the relative ranking given by the other methods not used in the search. Figures at the top (a-c), show the convergence of the top ranked (best) individuals respect to the whole population for each method (OS, NS, and PNS). Figures at the bottom (d-e), show the relative ranking given by the other two methods (not used in the search) respect to the best solutions found by the current method (showed at the top) at every generation.

at about 40 generations, see Figures 5.15(d). Conversely, when NS and PNS apply selective pressure we can observe a larger disagreement by the other two methods, shown in Figures 5.15(e) and 5.15(f). In particular, when NS applies selective pressure we can see that OS disagrees with the rankings throughout most of the run, with the differences progressively declining. On the other hand, the disagreement between PNS and NS seems more erratic, with total agreement in some generations and large disagreements in others, as seen in Figure 5.15(e). Figure 5.15(f) shows the relative differences in ranking when PNS applies selective pressure. In this case, both OS and NS disagree with PNS throughout most of the run, but both methods converge to similar rankings at the end of the search.

In summary, these plots provide useful insights regarding the different selective pressure applied by each method. The results confirm that the naive Bayesian assumption made by PNS does in fact lead to differences in search dynamics relative to NS, this might partially explain the differences in average solution size by both methods in binary classification tasks. However, these plots also show that while PNS and NS are ranking individuals differently than OS during some portions of the run, after a certain number of generations these differences start to decline. This is a plausible explanation for the similar performance achieved by all methods in terms of classification error.

5.7 Summary and Conclusions

This work presents the first application of the NS approach to supervised classification with GP, presenting several contributions. First, the concept of behavioral space is framed as a conceptual middle-ground between the well-known concept of objective space and the re-

cently popular semantic space in GP and can be extended to include both of them. Second, a domain-specific descriptor has been proposed and tested on supervised classification tasks considering real-world data for binary and multiclass problems. The proposed descriptor is a binary vector, where each element corresponds with each fitness case in the training set, taking a value of 1 when that fitness case is correctly classified and a 0 value otherwise. Third, two extensions to the basic NS approach have been developed, PNS and MCNS$_{bsf}$, as well as a hybrid method MCPNS. PNS provides a probabilistic framework to measure a solution's novelty, eliminating all of the underlying NS parameters while reducing the computational overhead that the original NS algorithm suffers from. On the other hand, the proposed MCNS$_{bsf}$ extends the minimal criteria approach by combining the objective function with the sparseness measure, constraining the NS algorithm by specifying a minimal solution quality, a dynamic criterion that is proportional to the quality of the best solution found so far.

Experimental results are evaluated based on two measures, solution quality and average size of all solutions in the population. In terms of performance, results show that all NS variants achieve competitive results relative to the standard OS approach in GP. These results show that the general open-ended approach towards evolution followed by NS can compete with objective driven search in traditional machine learning domains. On the other hand, in terms of solutions size and the bloat phenomenon, the NS approach can lead the search towards maintaining smaller program trees, particularly in the simpler binary tasks. In particular, NS and MCNS show substantial reductions in program size relative to OS.

Finally, a promising aspect of the present work is that several future lines of research can be explored, in no particular order we consider the following. Firstly, there seems to be a possible link between the PNS algorithm and two similar methods in evolutionary computation, estimation of distribution algorithms (EDAs) LARRAÑAGA et LOZANO [2001], the frequency fitness assignment (FFA) method WEISE et collab. [2014]. While EDAs use a distribution over genotype space to generate new individuals, PNS uses a distribution in behavioral space to measure the novelty of each solution. FFA favors solutions with unique objective scores, instead of uniqueness in behavioral space as done in PNS. Nonetheless, many of the theoretical and practical insights derived from EDA and FFA research might be brought to bear during further development of the PNS approach. Moreover, further comparisons with recent diversity preservation techniques might also be of interest NGUYEN et collab. [2012]. Secondly, we might extend the proposed PNS variants in other ways, such as testing PNS with real-valued behavior descriptors or applying PNS within semantic space, similar to the approach suggested in CASTELLI et collab. [2014]. Fourthly, the effect that NS has on bloat should be studied further, it is clear that standard NS and MCNS$_{bsf}$ provide the best bloat control but it is unclear why this effect was not observed on the multiclass problems. Finally, the proposed algorithms should be evaluated in other machine learning problems, such as unsupervised clustering NAREDO et TRUJILLO [2013] and symbolic regression MARTÍNEZ et collab. [2013].

Acknowledgements: This research was partially supported by CONACYT Basic Science Research Project No. 178323, DGEST (México) Research Projects No.5149.13-P and TIJ-ING-2012-110, as well as by FP7-Marie Curie-IRSES 2013 European Commission program with project ACoBSEC with contract No. 612689.

References

BANZHAF, W. 2014a, «Genetic programming and emergence», *Genetic Programming and Evolvable Machines*, vol. 15, nº 1, p. 63–73. 106

BANZHAF, W. 2014b, «Response to comments on "genetic programming and emergence"», *Genetic Programming and Evolvable Machines*, vol. 15, nº 1, p. 103–108. 106

BEADLE, L. et C. G. JOHNSON. 2008, «Semantically driven crossover in genetic programming», dans *Proceedings of the Tenth Conference on Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, CEC'08, IEEE Press, p. 111–116. 109, 115

BEADLE, L. et C. G. JOHNSON. 2009, «Semantically driven mutation in genetic programming», dans *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, CEC'09, IEEE Press, p. 1336–1342. 115

BRAMEIER, M. F. et W. BANZHAF. 2010, *Linear Genetic Programming*, 1re éd., Springer Publishing Company, Incorporated, ISBN 1441940480, 9781441940483. 108

BREIMAN, L. 2001, «Random forests», *Machine learning*, p. 5–32. 116

BROOKS, R. A. 1999, *Cambrian intelligence: the early history of the new AI*, MIT Press, Cambridge, MA, USA. 111

BRYLL, R., R. GUTIERREZ-OSUNA et F. QUEK. 2003, «Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets», *Pattern Recognition*, vol. 36, nº 6, p. 1291 – 1302, ISSN 0031-3203. 116

CASTELLI, M., L. TRUJILLO, L. VANNESCHI et A. POPOVIČ. 2015, «Prediction of energy performance of residential buildings: A genetic programming approach», *Energy and Buildings*, vol. 102, p. 67 – 74. 109, 116

CASTELLI, M., L. VANNESCHI et S. SILVA. 2014, «Semantic search-based genetic programming and the effect of intron deletion», *IEEE Transactions on Cybernetics*, vol. 44, nº 1, p. 103–113. 131

CUCCU, G., F. J. GOMEZ et T. GLASMACHERS. 2011, «Novelty-based restarts for evolution strategies.», dans *IEEE Congress on Evolutionary Computation*, IEEE, ISBN 978-1-4244-7834-7, p. 158–163. 114

DAWKINS, R. 1996, *Climbing Mount Improbable*, W.W. Norton & Company. 106

DERRAC, J., S. GARCÍA, D. MOLINA et F. HERRERA. 2011, «A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.», *Swarm and Evolutionary Computation*, vol. 1, nº 1, p. 3–18. 120

DOUCETTE, J. et M. HEYWOOD. 2010, «Novelty-based fitness: An evaluation under the santa fe trail», dans *EuroGP*, *Lecture Notes in Computer Science*, vol. 6021, Springer, p. 50–61. 114

EIBEN, A. et J. SMITH. 2007, *Introduction to Evolutionary Computing*, Natural Computing, Springer-Verlag, Berlin. 106

GOMES, J., P. URBANO et A. CHRISTENSEN. 2012, «Progressive minimal criteria novelty search», dans *Advances in Artificial Intelligence IBERAMIA 2012*, *Lecture Notes in Computer Science*, vol. 7637, édité par J. Pavón, N. D. Duque-Méndez et R. Fuentes-FernÃ¡ndez, Springer Berlin Heidelberg, ISBN 978-3-642-34653-8, p. 281–290. 114

GOMES, J. C., P. URBANO et A. L. CHRISTENSEN. 2013, «Evolution of swarm robotics systems with novelty search», *CoRR*, vol. abs/1304.3362. 106

HO, T. K. 1998, «The random subspace method for constructing decision forests», *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, nº 8, p. 832–844, ISSN 0162-8828. 116

INGALALLI, V., S. SILVA, M. CASTELLI et L. VANNESCHI. 2014, «A multi-dimensional genetic programming approach for multi-class classification problems», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 8599, édité par M. Nicolau, K. Krawiec, M. Heywood, M. Castelli, P. Garcia-Sanchez, J. J. Merelo, V. Rivas Santos et K. Sim, Springer Berlin Heidelberg, ISBN 978-3-662-44302-6, p. 48–60. 114

KISTEMAKER, S. et S. WHITESON. 2011, «Critical factors in the performance of novelty search», dans *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, ACM, p. 965–972. 107, 114, 126

KOWALIW, T., A. DORIN et J. MCCORMACK. 2012, «Promoting creative design in interactive evolutionary computation», *Evolutionary Computation, IEEE Transactions on*, vol. 16, nº 4, p. 523 –536. 106

KOZA, J. 2010, «Human-competitive results produced by genetic programming», *Genetic Programming and Evolvable Machines*, vol. 11, nº 3, p. 251–284. 106

KOZA, J. R. 1992, *Genetic programming: on the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, USA, ISBN 0-262-11170-5. 107, 108

KRAWIEC, K. et T. PAWLAK. 2013, «Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators», *Genetic Programming and Evolvable Machines*, vol. 14, nº 1, p. 31–63. 115

LANGDON, W. B. et R. POLI. 1997, «Fitness causes bloat», dans *Proceedings of the Second Online World Conference on Soft Computing in Engineering Design and Manufacturing*, Springer-Verlag, p. 13–22. 125

LARRAÑAGA, P. et J. A. LOZANO. 2001, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, Norwell, MA, USA, ISBN 0792374665. 131

LEHMAN, J. et K. O. STANLEY. 2008, «Exploiting open-endedness to solve problems through the search for novelty», dans *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, édité par S. Bullock, J. Noble, R. Watson et M. A. Bedau, MIT Press, Cambridge, MA, p. 329–336. xiv, 106, 111, 112, 113, 114, 116, 117

LEHMAN, J. et K. O. STANLEY. 2010a, «Efficiently evolving programs through the search for novelty», dans *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, ACM, p. 837–844. 106, 114, 116, 117

LEHMAN, J. et K. O. STANLEY. 2010b, «Revising the evolutionary computation abstraction: Minimal criteria novelty search», dans *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ACM, p. 103–110. 106, 107, 113, 114, 116, 117

LEHMAN, J. et K. O. STANLEY. 2011a, «Abandoning objectives: Evolution through the search for novelty alone», *Evol. Comput.*, vol. 19, nº 2, p. 189–223. 106, 114, 116, 117

LEHMAN, J. et K. O. STANLEY. 2011b, «Evolving a diversity of virtual creatures through novelty search and local competition», dans *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, ACM, New York, NY, USA, ISBN 978-1-4503-0557-0, p. 211–218. 106

LEVITIS, D. A., W. Z. LIDICKER JR et G. FREUND. 2009, «Behavioural biologists do not agree on what constitutes behaviour», *Animal Behaviour*, vol. 78, nº 1, p. 103–110. 111

LUKE, S. 2013, *Essentials of Metaheuristics*, 2e éd., Lulu. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/. 106

MARTÍNEZ, Y., E. NAREDO, L. TRUJILLO et E. G. LÓPEZ. 2013, «Searching for novel regression functions», dans *IEEE Congress on Evolutionary Computation*, p. 16–23. 107, 131

MCDERMOTT, J., E. GALVÁN-LOPÉZ et M. O'NEILL. 2011, «A fine-grained view of phenotypes and locality in genetic programming», dans *Genetic Programming Theory and Practice IX*, édité par R. Riolo, E. Vladislavleva et J. H. Moore, Genetic and Evolutionary Computation, Springer New York, ISBN 978-1-4614-1769-9, p. 57–76. 108, 111

MORAGLIO, A., K. KRAWIEC et C. G. JOHNSON. 2012, «Geometric semantic genetic programming», dans *Proceedings of the 12th international conference on Parallel Problem Solving from Nature - Volume Part I*, PPSN'12, Springer-Verlag, Berlin, Heidelberg, p. 21–31. 109, 115

MOURET, J.-B. 2011, «Novelty-based multiobjectivization», dans *New Horizons in Evolutionary Robotics*, *Studies in Computational Intelligence*, vol. 341, édité par S. Doncieux, N. Bredèche et J.-B. Mouret, Springer Berlin Heidelberg, ISBN 978-3-642-18271-6, p. 139–154. 114

MUÑOZ, L., S. SILVA et L. TRUJILLO. 2015, «M3gp multiclass classification with gp», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 9025, édité par P. Machado, M. I. Heywood, J. McDermott, M. Castelli, P. García-Sánchez, P. Burelli, S. Risi et K. Sim, Springer International Publishing, p. 78–91. 114, 116, 118

NAREDO, E. et L. TRUJILLO. 2013, «Searching for novel clustering programs», dans *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, ACM, New York, NY, USA, ISBN 978-1-4503-1963-8, p. 1093–1100. 107, 131

NAREDO, E., L. TRUJILLO et Y. MARTÍNEZ. 2013, «Searching for novel classifiers», dans *Proceedings from the 16th European Conference on Genetic Programming, EuroGP 2013, LNCS*, vol. 7831, Springer-Verlag, p. 145–156. 107, 123

NELSON, A. L., G. J. BARLOW et L. DOITSIDIS. 2009, «Fitness functions in evolutionary robotics: A survey and analysis», *Robot. Auton. Syst.*, vol. 57, nº 4, p. 345–370. 111

NGUYEN, Q., X. NGUYEN, M. O'NEILL et A. AGAPITOS. 2012, «An investigation of fitness sharing with semantic and syntactic distance metrics», dans *Proceedings of the 15th European Conference on Genetic Programming, EuroGP'12*, Springer Berlin Heidelberg, p. 109–120. 131

OFRIA, C. et C. O. WILKE. 2004, «Avida: a software platform for research in computational evolutionary biology», *Artif. Life*, vol. 10, nº 2, p. 191–229. 106

PETER, J. M. 2000, *Cartesian Genetic Programming*, 1re éd., Natural Computing Series, Springer-Verlag Berlin Heidelberg, ISBN 978-3-642-17309-7, XXII, 346 p.. 108

POLI, R., W. B. LANGDON et N. F. MCPHEE. 2008, *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd. 108

SILVA, S. et J. ALMEIDA. 2003, «Gplab–a genetic programming toolbox for matlab», dans *Proceedings of the Nordic MATLAB conference*, édité par L. Gregersen, p. 273–278. 118, 119

SILVA, S. et E. COSTA. 2009, «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories», *Genetic Programming and Evolvable Machines*, vol. 10, nº 2, p. 141–179. 124, 125

SPECTOR, L. et A. ROBINSON. 2002, «Genetic programming and autoconstructive evolution with the push programming language», dans *Genetic Programming and Evolvable Machines*, p. 7–40. 108

SQUILLERO, G. 2005, «Microgp - an evolutionary assembly program generator», *Genetic Programming and Evolvable Machines*, vol. 6, nº 3, p. 247–263, ISSN 1389-2576. 108

STANLEY, K. O. et J. LEHMAN. 2015, *Why Greatness Cannot Be Planned: The Myth of the Objective*, 2015e éd., Springer, ISBN 978-3-319-15523-4, doi:10.1007/978-3-319-15524-1. 112

TRUJILLO, L., E. NAREDO et Y. MARTÍNEZ. 2013a, «Preliminary study of bloat in genetic programming with behavior-based search», dans *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, *Advances in Intelligent Systems and Computing*, vol. 227, Springer International Publishing, ISBN 978-3-319-01127-1, p. 293–305. 107

TRUJILLO, L., G. OLAGUE, E. LUTTON et F. F. DE VEGA. 2008, «Discovering several robot behaviors through speciation», dans *Proceedings of the 2008 conference on Applications of evolutionary computing*, Evo'08, Springer-Verlag, p. 164–174. 111

TRUJILLO, L., L. SPECTOR, E. NAREDO et Y. MARTÍNEZ. 2013b, «A behavior-based analysis of modal problems», dans *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation Companion*, GECCO Companion '13, p. 1047–1054. 107

URBANO, P. et G. LOUKAS. 2013, «Improving grammatical evolution in santa fe trail using novelty search», dans *Advances in Artificial Life, ECAL*, p. 917–924. 106

URBANO, P., E. NAREDO et L. TRUJILLO. 2014, «Generalization in maze navigation using grammatical evolution and novelty search», dans *Theory and Practice of Natural Computing*, *Lecture Notes in Computer Science*, vol. 8890, Springer International Publishing, p. 35–46. 106, 114, 124

UY, N. Q., N. X. HOAI, M. O'NEILL, R. I. MCKAY et E. GALVÁN-LÓPEZ. 2011, «Semantically-based crossover in genetic programming: application to real-valued symbolic regression», *Genetic Programming and Evolvable Machines*, vol. 12, nº 2, p. 91–119. 109, 115

VELEZ, R. et J. CLUNE. 2014, «Novelty search creates robots with general skills for exploration», dans *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, ACM, New York, NY, USA, ISBN 978-1-4503-2662-9, p. 737–744. 124

WEISE, T., M. WAN, P. WANG, K. TANG, A. DEVERT et X. YAO. 2014, «Frequency fitness assignment», *Evolutionary Computation, IEEE Transactions on*, vol. 18, nº 2, p. 226–243, ISSN 1089-778X. 131

WHITLEY, L. D. 1991, «Fundamental principles of deception in genetic search», dans *Foundations of Genetic Algorithms*, Morgan Kaufmann, p. 221–241. 106

WOOLLEY, B. G. et K. O. STANLEY. 2012, «Exploring promising stepping stones by combining novelty search with interactive evolution», *CoRR*, vol. abs/1207.6682. 106

ZHANG, M. et W. SMART. 2006, «Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification», *Pattern Recogn. Lett.*, vol. 27, nº 11, p. 1266–1274. 111, 114, 116, 118

Chapter 6

Evaluating the Effects of Local Search in Genetic Programming

This chapter is related to the PhD thesis of Emigdio Z. Flores (ITT Tijuana) and the European project ACOBSEC, and has been published and presented at the conference EVOLVE 2014, Jul 2014, BEIJING, China. Work carried out with Emigdio Z. Flores, Leonardo Trujillo and Oliver Schütze.

Contents

Abstract

Genetic programming (GP) is an evolutionary computation paradigm for the automatic induction of syntactic expressions. In general, GP performs an evolutionary search within the space of possible program syntaxes, for the expression that best solves a given problem. The most common application domain for GP is symbolic regression, where the goal is to find the syntactic expression that best fits a given set of training data. However, canonical GP only employs a syntactic search, thus it is intrinsically unable to efficiently adjust the (implicit) parameters of a particular expression. This work studies a Lamarckian memetic GP, that incorporates a local search (LS) strategy to refine GP individuals expressed as syntax trees. In particular, a simple parametrization for GP trees is proposed, and different heuristic methods are tested to determine which individuals should be subject to a LS, tested over several benchmark and real-world problems. The experimental results provide necessary insights in this insufficiently studied aspect of GP, suggesting promising directions for future work aimed at developing new memetic GP systems.

6.1 Introduction

It is widely understood that effectively balancing exploration and exploitation is one of the key issues underlying any successful search process. Indeed, a useful taxonomy of search algorithms is based on the idea of differentiating between search methods that rely on exploratory search operators, and search methods that can guarantee (local) optimality by exploiting the local structure of a fitness landscape. Local search (LS) algorithms can guarantee convergence to local optima if their underlying assumptions are satisfied and are computationally efficient. However, the success of a LS strongly depends on the initial point, particularly in highly irregular, multimodal or discontinuous fitness landscapes GILL et collab. [1981]. On the other hand, global search algorithms include a variety of deterministic and stochastic strategies. Here, we focus on evolutionary algorithms (EAs), metaheuristic strategies based on an abstract model of Neo-Darwinian evolution EIBEN et SMITH [2003]; LUKE [2013]. EAs are particularly useful when a good initial solution is difficult to propose, or when LS strategies tend to converge on undesirable local optima. Moreover, EAs are particularly well suited when a single monolithic solution is not sufficient, and instead a set of different solutions is required COELLO et collab. [2006]; DUNN et collab. [2006]. In general, EAs cannot guarantee convergence towards local optima in most realistic scenarios. Moreover, they tend to be computationally costly, and in many cases can be highly inefficient. Nevertheless, EAs have produced extremely competitive solutions in difficult domains and problem instances KOZA [2010].

Considering the strengths and weaknesses of both global and local methods, it is intuitive to conclude that the best strategy should be a hybrid approach, commonly referred to as memetic search. These methods have been extensively studied over recent years, combining EAs with a variety of local searchers CHEN et collab. [2011].

While all EAs are based on the same general principles DE JONG [2006], there is a large variety among current methods, each with their respective strengths and weaknesses. This paper focuses on an EA paradigm that presents unique properties among other EAs, genetic programming (GP) KOZA [1992]; POLI et collab. [2008]. The goal of GP is to automatically evolve specialized syntactic expressions that best solve a given tasks, which can be interpreted as mathematical functions or computer programs. In particular, this paper studies how the standard syntactic search can be improved by including a LS method. While this paper is not the first to develop a memetic GP system, it does present the first systematic evaluation of what could be the best strategies to accomplish this, by considering several variants and evaluating them on current benchmark problems for symbolic regression.

The remainder of this chapter proceeds as follows. First, an overview of GP is provided in Section 6.2. Then, Section 6.3 reviews previous work on memetic optimization with EAs. Afterwards, Section 6.4 describes our proposed memetic algorithms to perform real-valued parameter optimization of evolved syntactic expressions, considering several basic variants. The experimental setup is presented in Section 6.5 and results are discussed in Section 6.6. Finally, a summary of the paper and concluding remarks are outlined in Section 6.7.

6.2 Genetic Programming

GP can be understood as a generalization of the basic genetic algorithm, its main features can be summarized as follows POLI et collab. [2008]. First, GP was originally proposed as an EA that evolves simple programs, functions, operators, or in general symbolic expressions that perform some form of computation. GP is basically used to evolve solutions to different types of design problems, with examples as varied as quantum algorithms SPECTOR [2006], computer vision operators OLAGUE et TRUJILLO [2011] and satellite antennas HORNBY et collab. [2011]. Second, solutions are expressed as variable length structures, such as linked lists, parse trees or graphs POLI et collab. [2008]. These structures encode

the syntax of each individual program. Therefore, in a canonical GP algorithm, search operators, such as crossover and mutation, perform syntactic variations on the evolving population. Third, by considering each individual in a GP run as a program, the evolutionary process is basically attempting to write the best program syntax that solves a given problem. Therefore, a finite set of basic symbols needs to be defined, which is called the primitive set $\mathbb{P}$. Within the primitive set there is a subset of basic operations or functions of different arity, called the function set F, and a subset of input variables, constants or zero arity functions called the terminal set T, such that $\mathbb{P} = F \cup T$.

Given the variety of possible GP configurations and applications, this work focuses on the problem of symbolic regression using a tree representation. In symbolic regression, the goal is to search for the symbolic expression $K^O : \mathbb{R}^p \rightarrow \mathbb{R}$ that best fits a particular training set $\mathbb{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n input/output pairs with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. In such problems, for instance, the function set can be defined as $F = \{+, -, /, *\}$ and the terminal set can be composed by each of the features in the input data, such that $T = \{x_i\}$ with $i = 1, \ldots, p$, but other terminal elements can be included such as integer or real-valued constants.

The general symbolic regression problem can then be defined as

$$(K^O, \boldsymbol{\theta^O}) \leftarrow \underset{K \in \mathbb{G}; \boldsymbol{\theta} \in \mathbb{R}^m}{arg\,min}\ f(K(\mathbf{x}_i, \boldsymbol{\theta}), y_i)\ with\ i = 1, \ldots, p\,, \tag{6.1}$$

where $\mathbb{G}$ is the solution or syntactic space defined by $\mathbb{P}$, f is the fitness function based on the difference between a program's output $K(\mathbf{x}_i, \boldsymbol{\theta})$ and the expected output y_i, such as the mean square error, and $\boldsymbol{\theta}$ is a particular parametrization of the symbolic expression K, assuming m real-valued parameters. This dual problem, of simultaneously optimizing syntax (structure) as well as its parametrization is also discussed in EMMERICH et collab. [2001]; LOHMANN [1991]. The authors differentiate between two possible approaches towards solving such a task. The first group is *hierarchical structure evolution* (HSE), when $\boldsymbol{\theta}$ has a strong influence on fitness, and thus a local search is required at each generation of the global (syntactic) search, configured as a nested process. The second group is called *simultaneous structure evolution* (SSE), when $\boldsymbol{\theta}$ has a marginal effect on fitness, in such cases a single evolutionary loop could optimize both syntax and parameters simultaneously. However, such categorizations are highly abstract, and a particular implementation could easily be classified into both groups. Nevertheless, it is reasonable to state that the standard GP approach falls in the SSE group.

In standard GP, however, parameter optimization is usually not performed explicitly, since GP search operators only affect syntax. Therefore, the parameters are only implicitly considered. For instance, a GP individual might have the following syntax $K(x) = x + sin(x)$; in this case, we might propose the following parametrization: $\boldsymbol{\theta} = (\alpha_1, \alpha_2, \alpha_3)$, with $K(x) = \alpha_1 x + \alpha_2 sin(\alpha_3 x)$. In a traditional GP search, these parameters are all set to 1, which does not necessarily lead to the best possible performance for this particular syntax. Indeed, if the optimal solution is $K^O(x) = 3.3x + 1.003sin(0.0001x)$, then individual K might be easily lost during the selection or survival processes [1].

This seems to be a glaring weakness in the standard GP approach. While the syntactic-based search in GP has solved a variety of difficult problems, it is nonetheless very inefficient, leading to important practical limitations, such as search stagnation, bloat and uninterpretable solutions SILVA et COSTA [2009]. In other words, GP performs a highly explorative search, since the search operators can produce large fitness changes with only modest syntactic modifications or vice-versa. Therefore, the inclusion of a LS procedure could enhance the performance of the search. In this paper, the goal is to study what are the effects of including a local optimization strategy within a GP run, configured as an HSE system.

[1] Some GP systems do include numerical terminals or random ephemeral constants, while these constants can act as coefficients in an evolved expressions, they are not subject to optimization after they are introduced into the syntax of a particular individual.

6.3 Previous Work

As stated above, many works have studied how to combine an EA with a local optimizer (also referred to as a refinement process). In general, such approaches are considered to be a simple type of memetic search CHEN et collab. [2011]. The basic idea is straightforward: include within the optimization process an additional search operator that takes an individual (or several) as an initial point and searches for the local optima around it. Such a strategy can help guarantee that the local region around each individual is fully exploited. However, there can be some negative consequences to such an approach. The most evident is computational overhead, while the cost of a single LS might be negligible, performing it on every individual might become inefficient. Second, LS can produce overfitted solutions, stagnating the search on local optima. These issues aside, hybrid techniques have produced impressive results in a variety of scenarios CHEN et collab. [2011].

A useful taxonomy of this type of memetic algorithms can be derived based on how inheritance is carried out during the evolutionary process CHEN et collab. [2011]. Suppose that $\mathbf{h}, \mathbf{h}^o \in \mathbb{G}$, where $\mathbf{h}$ is an individual solution and $\mathbf{h}^o$ is the solution generated after a LS is applied on $\mathbf{h}$. Obviously, for a minimization problem, $f(\mathbf{h}^o) \leq f(\mathbf{h})$, with f the objective function. Thus, $\mathbf{h} \neq \mathbf{h}^o$, unless $\mathbf{h}$ was in fact a local optima. Then, a memetic algorithm could proceed in two distinct ways with respect to inheritance. In a *Lamarckian* algorithm, the traits acquired during the local search, captured in $\mathbf{h}^o$, replace those of the original individual $\mathbf{h}$; i.e., the inheritance of acquired characteristics $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}^o, f(\mathbf{h}^o))$. On the other hand, in a *Baldwinian* algorithm, the local optimization process only modifies the fitness of an individual; $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}, f(\mathbf{h}^o))$; i,e., ontogenic evolution.

An extensive survey of these methods is presented in CHEN et collab. [2011] by Chen et al.. A noteworthy aspect of this survey is an almost complete lack of papers that deal with GP. Of the more than two hundred papers covered in CHEN et collab. [2011], only a couple deal with memetic GP. This illustrates how the GP community has not addressed the topic adequately. Nonetheless, some works have been developed over recent years. For instance, WANG et collab. [2011] presents a memetic GP approach to evolve decision trees. The authors report good results using domain-specific LS heuristics. In ESKRIDGE et HOUGEN [2004] the authors present a memetic crossover for GP, instead of a local search strategy, crossover is based on a more general notion of memetic search.

Indeed, the complete optimization problem defined in Section 6.2 has not received much attention. In TOPCHY et PUNCH [2001], gradient descent is used to optimize numerical constants within a GP tree, achieving good results on five symbolic regression problems. However, the work only optimizes the value of numerical terminal elements (leaves), it does not consider parameters within internal nodes. Additionally, the paper presents results from a small sample size of runs, and only considers training fitness, a highly deceptive measure of learning. Similarly, in ZHANG et SMART [2004] and GRAFF et collab. [2013] a LS algorithm is used to optimize the value of constant terminal elements. In ZHANG et SMART [2004] gradient descent is used and tested on classification problems, while GRAFF et collab. [2013] uses Resilient Backpropagation (RPROP) and evaluates the proposal on a real-world problem, in both cases leading towards improved results. While these works show promise, they include several design choices that are not analyzed or justified in detail. For instance, ZHANG et SMART [2004] applies gradient descent on every individual of the evolving population, an obvious computational bottleneck, while GRAFF et collab. [2013] only applies RPROP on the best individual from each generation. A similar strategy is included in the HeuristicLab optimization environment, where local optimization can be performed on the final solution found by GP WAGNER et KRONBERGER [2012]. In these cases, it is not evident which proposed strategy can offer the best results in new scenarios. Moreover, GRAFF et collab. [2013]; ZHANG et SMART [2004] only evaluate their approaches on specific problem instances.

Closer to the problem discussed here, SMART et ZHANG [2004] includes weight parameters for each function node, what the authors call inclusion factors; these weights modulate

the importance that each node has within the tree. Indeed, the authors identify what we are here referring to as implicit program parameters, and optimize these values by applying gradient descent on all trees. The authors also propose a series of new search operators that explicitly contemplate the parametrization of each GP tree. However, only a limited experimental validation is performed on specialized classification problems, with mixed results. The performance of the memetic GP is indeed better, but not substantially in some tests. Additionally, the improvement in performance is misleading, since the GP systems are executed a fixed number of generations, but the added computational search performed by gradient descent leads to an unfair comparison based on total search effort. Moreover, it seems that the proposed parameter aware search operators have a negative effect on the search. The GP system is evaluated using an uncommonly small function set (only 2 functions), an unrealistic configuration.

Finally, recent works have decided to completely change the basic GP framework in order to account for the lack of an explicit parametrization of syntactic expressions. The fast function extraction (FFX) algorithm McConaghy [2011], for instance, poses the symbolic regression problem as that of finding a linear combination of a subset of candidate basis functions. Thus, FFX builds linear in parameter models, and optimizes using a modified version of the elastic net regression technique, eliminating the evolutionary process altogether. Another recent example is the prioritized grammar enumeration (PGE) technique Worm et Chiu [2013], that employs dynamic programming and eliminates the basic search operators of traditional GP. Parameter values of the symbolic expressions produced by PGE are optimized using non-linear optimization with the Levenberg-Marquardt algorithm.

6.4 Integrating Local Optimization Strategies within GP

Intuitively, through tree parametrization and local optimization, a GP search should converge faster towards high quality solutions. As stated before, the proposal is to develop an HSE-GP, by parameterizing GP trees and including a Lamarckian memetic strategy. Therefore, the parametrization of GP trees must be defined; then, a particular LS method must be chosen. Finally, a decision strategy must be suggested to determine on which individuals should a LS be applied. The proposals for each of these issues are presented below.

6.4.1 Parametrization of GP trees

For this study, we propose a simple and naive approach to add parameters within GP trees. For each function in the function set $g_k \in F$, we add a unique weight coefficient $\theta_k \in \mathbb{R}$, such that each function is now defined by

$$g'_k = \theta_k g_k \tag{6.2}$$

where g'_k is the new parameterized function, $k \in \{1, \ldots, r\}$ and $r = |F|$. Note that each θ_k is linked to a single g_k, such that the parameter vector of tree K_i is given by $\boldsymbol{\theta} \in \mathbb{R}^m$ with m the number of different functions included in K_i such that $m \leq r$. Notice that if a tree contains several instances of a particular function, all instances of this function share the same coefficient. Indeed, this severely constraints the optimization process, particularly for large trees that can include many instances of the same function. To be clear, it is not argued that such a parametrization should be considered as the best possible alternative. Nonetheless, it does have one important consequence, it bounds the size of the search space for each LS process, something that could become overwhelming if the GP trees grew to large, which tends to happen frequently due to bloat Silva et Costa [2009].

In all trees, every θ_k is initialized to unity, which would be their implicit value in a standard GP. Since the proposed algorithm is a Lamarckian memetic search, the standard GP

search operators (subtree mutation and subtree crossover) still only operate at the syntax level, exchanging g'_k nodes without affecting their respective θ_k.

6.4.2 Local Search mechanism

Potentially, any tree can be of linear or non-linear form; however, for convenience we treat every tree as non-linear expression. This is a multidimensional non-linear optimization problem, which can be solved using least squares.

The above problem, is formally expressed by the following cost function

$$\min_{\boldsymbol{\theta}} \|K(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{y}\|_2^2 = \min_{\boldsymbol{\theta}} \sum_i (K(\mathbf{x_i}, \boldsymbol{\theta}) - \mathbf{y_i})^2 \,, \tag{6.3}$$

where $\mathbf{x}$ are the input data points, $\mathbf{y}$ are the output data points, i is the index for the regression instances, and K is the non-linear function. The problem posed in (6.3) can be solved using different methods YUAN [1996] LAWSON et HANSON [1995]. In this case, we chose to use a well known technique with good convergence properties and good scalability in complexity for high dimensional problems, called trust region optimization SORENSEN [1982].

The trust region optimization method tries to minimize a smooth non-linear function subject to bounds on the variables, given by

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^m} \|K(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{y}\|_2^2, \; l_i \leq \theta_i \leq u_i \; \forall i \in \{1..m\} \,, \tag{6.4}$$

where $l_i \in \{\mathbb{R} \cup \{-\infty\}\}$, $u_i \in \{\mathbb{R} \cup \{\infty\}\}$, and $K(x, .) : \mathbb{R}^m \to \mathbb{R}$. Conceptually, a trust region approach replaces a m-dimensional unconstrained optimization problem by a m-dimensional constrained problem. This results is an approximate solution, since it does not need to be solved with high accuracy. One of the appealing points of these methods is their strong convergence properties COLEMAN et LI [1994]. The idea behind a trust region method is simple. The increment $s_k = x_{k+1} - x_k$ is an approximate solution to a quadratic subproblem with a bound on the step size

$$\min_{s \in \mathbb{R}^m} \left\{ \psi_k(s) \stackrel{def}{=} g_k^T s + \frac{1}{2} s^T B_k s : \|\overline{D}_k\| \leq \Delta_k \right\} \,, \tag{6.5}$$

where $g_k \stackrel{def}{=} \nabla f(x_k)$, B_k is a symmetric aproximation to the Hessian matrix $\nabla^2 f(x_k)$, $\overline{D}_k$ is a scaling matrix, and Δ_k is a positive scalar representing the trusted region size. Solving (6.5) efficiently is not a trivial task, see MORÉ et SORENSEN [1983]; SHULTZ et collab. [1982]; SORENSEN [1982]; STEIHAUG [1983]. Here, the method proposed in COLEMAN et LI [1993] is used, which does not require the solution of a general quadratic programming subproblem at each iteration.

6.4.3 Integrating LS into GP

The final issue that must be considered is to determine on which individuals is the LS applied, and at what moment during the evolutionary process. For the latter point, LS is applied after a complete evolutionary loop is completed; i.e., LS is applied after the survival criterion is applied and before the following generation begins. This means that the local optimization of individuals from generation t impacts the search at generation $t+1$. For the former point, several different approaches are evaluated.

First, three naive approaches are considered, two of which have been used in previous memetic GP systems. Probably the simplest is to apply a LS on all of the individuals, this method is referred to as LSALL-GP. This approach will inherently introduce a large computational overhead. Another approach is to be more selective, and only apply a LS on the best individual from each generation, this method is referred to as LSBEG-GP. Conversely, LS can also be applied on the worst individual from each generation, this variant is called

Table 6.1 – GP Parameters for the different methods.

Parameter	LSBEG-GP	LSWEG-GP	LSRP-GP	LSBS-GP	LSWS-GP	LSALL-GP
θ	vector of ones					
F_{BEST}	yes	no	no	no	no	no
F_{WORST}	no	yes	no	no	no	no
p_{LS}	1	1	0.1	0.5	0.5	1
PER_{LS}	0	0	100%	10%	10%	100%

LSWEG-GP. This variant might be useful for extreme cases of individuals that present the correct structure, but a highly suboptimal parametrization.

The fourth variant is referred to LS Random Population or LSRP-GP, where every individual of the population is a viable candidate for a LS optimization with a probability p_{LS}, basically implemented as a mutation operator. Here, it is assumed that a low probability is desirable, to minimize the added computational cost. The fifth variant is called LS Best Subset or LSBS-GP, which is the same as LSRP-GP, except that the individuals that are valid candidates for LS are those in the best PER_{LS} percentile of the population, a second algorithm parameter. Finally, the sixth variant is called LS Worst Subset or LSWS-GP, takes the same approach as LSRP-GP, but candidate individuals for a LS are those in the worst PER_{LS} percentile. All of the methods and their parameters are summarized in Table 6.1.

6.5 Experimental Setup

The algorithms that are evaluated are LSBEG-GP, LSWEG-GP, LSRP-GP, LSBS-GP, LSWS-GP and LSALL-GP; also, a standard GP is included as a control method. All the algorithms were implemented in Matlab using the GPLAB[2] Toolbox SILVA et ALMEIDA [2003] modifying its core functionality to integrate the LS procedure. The set of experiments cover a series of well known benchmark symbolic regression problems. Five synthetic problems were used: Keijzer-6 KEIJZER [2003], Korns-12 KORNS [2011], Vladislavleva-4 VLADISLAVLEVA et collab. [2009], Nguyen-7 UY et collab. [2011], Pagie-1 PAGIE et HOGEWEG [1998]; and one real-world problem. All of them are suggestions made in WHITE et collab. [2013], a recent survey on GP benchmarking.

Two performance measures are used to evaluate the different algorithms. First, fitness evaluation over the test set for the best solution found thus far. Second, the average population size, a relevant measure regarding the bloat phenomenon and solution complexity TRUJILLO et collab. [2013]. The evolution of these measures is analysed with respect to the total number of fitness function evaluations instead of generations, to account for the LS iterations. The LS performs a maximum of 400 iterations. However, we do not consider any additional computational effort due to the LS, which underestimates the computational cost of performing LS. The stopping criteria is the number of function evaluations. The GP configuration parameters for all variants are shown in Table 6.2. Some of these parameters vary depending on the problem, as suggested in MCDERMOTT et collab. [2012]. For the case of Tower problem, total samples were divided in a ratio of 30/70% for the training and testing sets. As stated before, Table 6.1 summarizes the parameter values for the memetic GP variants.

[2] http://gplab.sourceforge.net/

Table 6.2 – General GP parameters

Parameter	Value
Runs	30
Population	500
Function evaluations	2'000,000
Training set	as indicated in McDermott et collab. [2012]
Testing set	as indicated in McDermott et collab. [2012]
Crossover operator	Standard subtree crossover, 0.9 probability
Mutation operator	Mutation probability per node 0.05
Tree initialization	Ramped Half-and-Half, maximum depth 6
Function set	as indicated in McDermott et collab. [2012]
Terminal set	Input variables, constants as indicated in McDermott et collab. [2012]
Selection for reproduction	Tournament selection of size 3
Elitism	Best individual survives
Maximum tree depth	17

6.6 Results and Summary

Figures 6.1-6.5 summarize the results of the all tested techniques, showing the median values computed over 30 independent runs. The first problem is Keijzer-6 shown in Figure 6.1, considered a simple or easy problem. The difficulty is raised with the testing data, since the solution must extrapolate outside the training domain. In general, most methods exhibit similar performance, with some notable exceptions. First, LSWEG-GP obviously achieves the worst test performance, while LSBS-GP and LSALL-GP exhibit the best. In particular, LSBS-GP and LSALL-GP converge quickly to very good performance; in fact LSBS-GP could have been halted very early in the run. However, after a million function evaluations LSALL-GP, shows a noticeable improvement, achieving the best results. In terms of size, both of these methods also exhibit the best results, with LSALL-GP producing the smaller trees.

Figure 6.2(left) presents the results for the Korns-12 problem, considered difficult in GP literature since standard GP usually cannot find the true expression Korns [2011]. The problem includes redundant input data that does not influence its output. The idea is to test GP algorithms on their ability to avoid overfitting. With respect to test error, Figure 6.2(a) shows that most algorithms perform quite similarly, all of them converging to their best median values quite early in the runs. What is noticeable is the performance of LSBS-GP, with the worst test performance, but also the best training performance, shown in Figure 6.2(c), indicating a slight overfitting. Regarding the evolution of size, shown in Figure 6.2(e), all methods exhibit similar sizes, but code growth is noticeably slower for LSALL-GP.

For the Vladislavleva-4 problem shown in Figure 6.2(right), considered a difficult test case for GP Vladislavleva et collab. [2009], overfitting is also noticeable. In Figure 6.2(b) we can see that test fitness varies greatly across the runs for most methods, particularly for LSBS-GP. Conversely in Figure 6.2(d) we can observe a smooth monotonic convergence for the training fitness. The case of LSBS-GP is noticeable, particularly since it is clear that the method achieves its best test fitness early in the run, and then starts to overfit. The only method that did not overfit was LSALL-GP, exhibiting the best performance over the test data. Finally, with respect to program size (Figure 6.2(f)), once again all methods show similar trends, with LSALL-GP producing substantially smaller trees.

Problem Nguyen-7 and Pagie-1 exhibit similar outcome in terms of fitness and program size. In both, among all methods LSBS-GP exhibits a faster convergence to a better solution quality computed over test data and also achieves the best results along with LSALL-GP. Once again LSALL-GP produces the smallest trees, with LSBS-GP the second best, however in these problems the difference between both of these methods is smaller than in the other cases. All other variants show similar performance based on both measures.

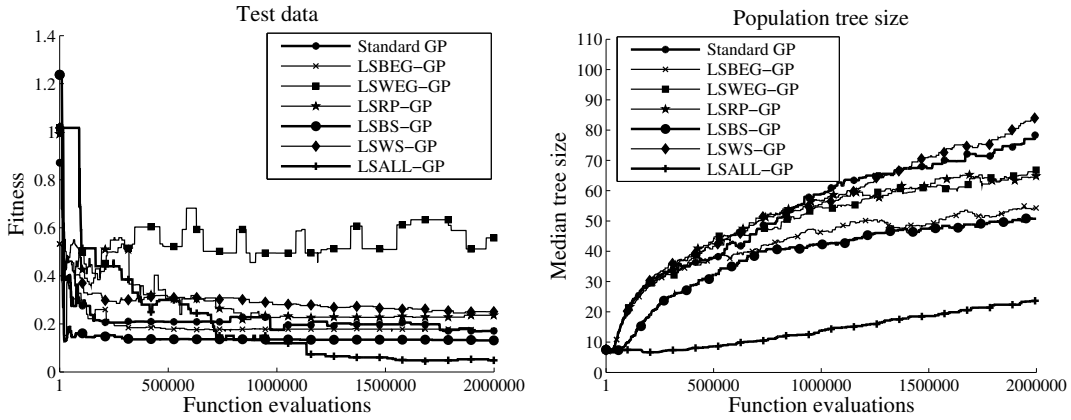


Figure 6.1 – Results for problem Keijzer-6 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

Finally, the Tower problem is an industrial real-world problem on modeling gas chromatography measurements of the composition of a distillation tower. This problem contains 5000 records and 23 potential input variables. The measurements (5000 for each variable) are not treated as time series, but simply used as samples for a regression model. In this case, LSALL-GP again achieves the most interesting results, both in regards to test-fitness and solution size. From the remaining methods, LSBS-GP shows the best performance, but still noticeably worse that LSALL-GP.

A final summary of the performance of each method is presented in Table 6.3, that shows three different snapshots of the median performance of each algorithm, after 250,000, 1 million and 2 million function evaluations. In general, LSALL-GP exhibits the best performance after all of the allowed function evaluations. However, if we only consider 250,000 function evaluations, then LSBS-GP exhibits the best performance on three of the six problems, and also the second best performance on two others.

From the remaining methods, those that show the worst performance are LSWEG-GP and LSWS-GP, that apply LS to the worst solutions in the population, what definitely seems to be a bad strategy. Moreover, while LSBS-GP shows good performance, the deterministic LSBEG-GP that applies LS to the best solution is notably inferior to the best methods, a noteworthy results since several previously published works have employed this memetic strategy.

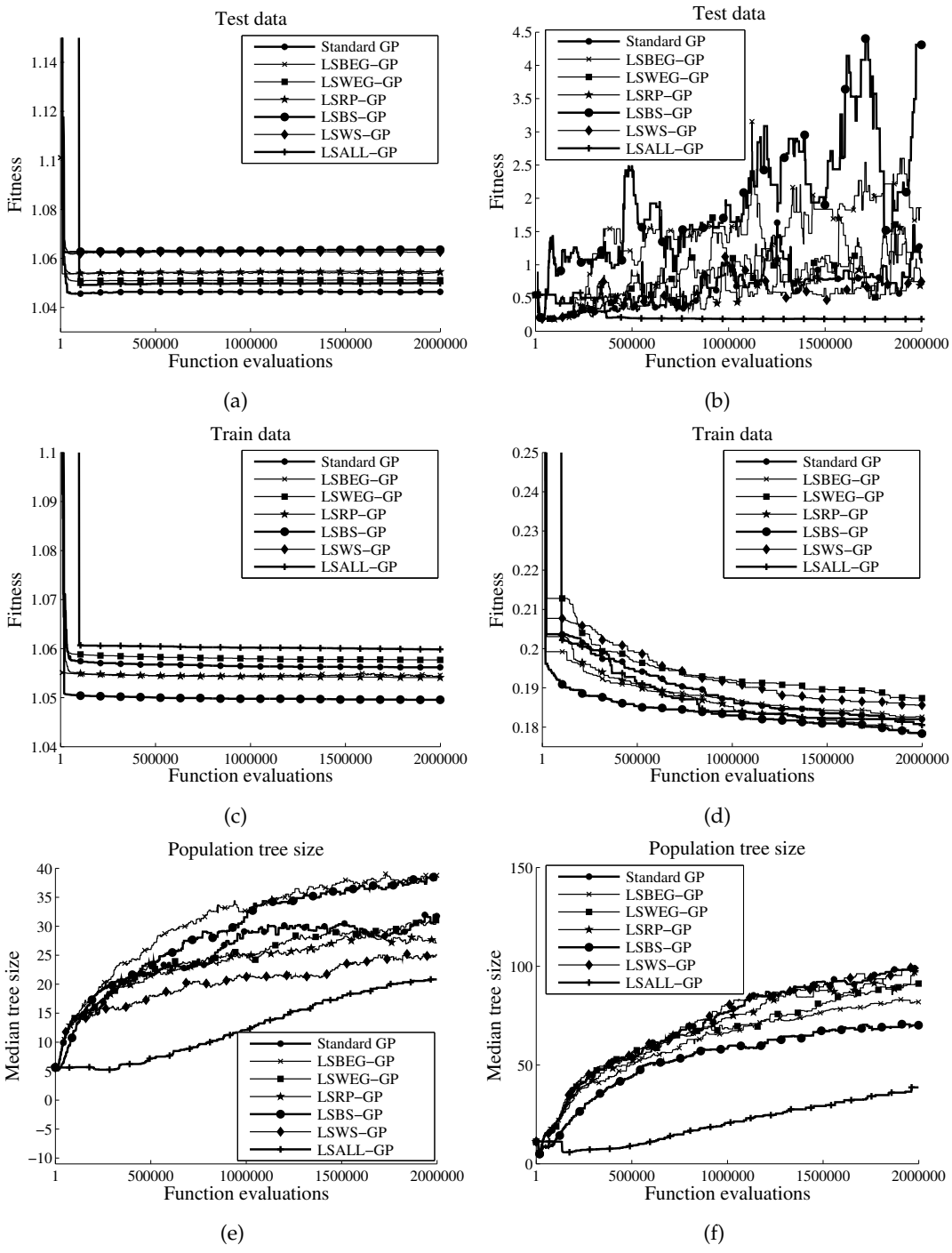


Figure 6.2 – Results for problems Korns-12 (a,c,e) and Vladislavleva-4 (b,d,f): (a,b) Fitness over test data; (c,d) Fitness over training data; and (e,f) Average program size. All plots show median values over 30 independent runs.

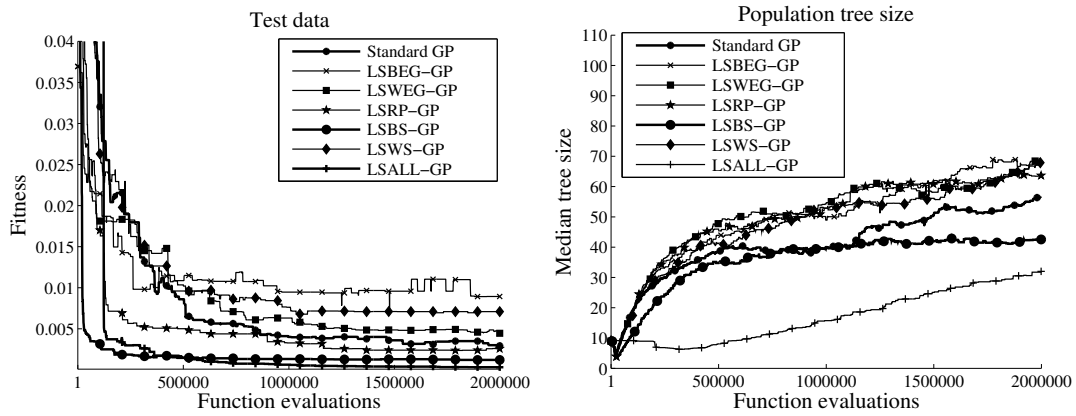


Figure 6.3 – Results for problem Nguyen-7 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

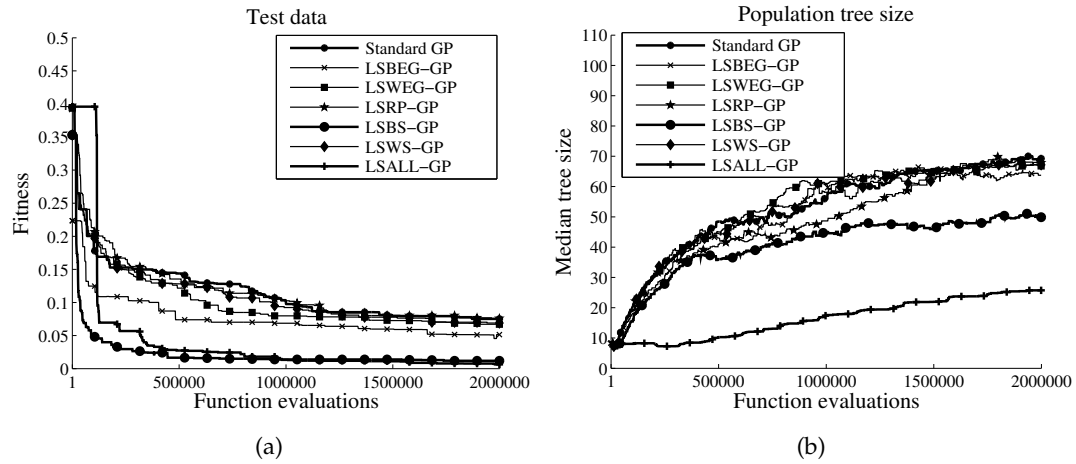


Figure 6.4 – Results for problem Pagie-1 plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

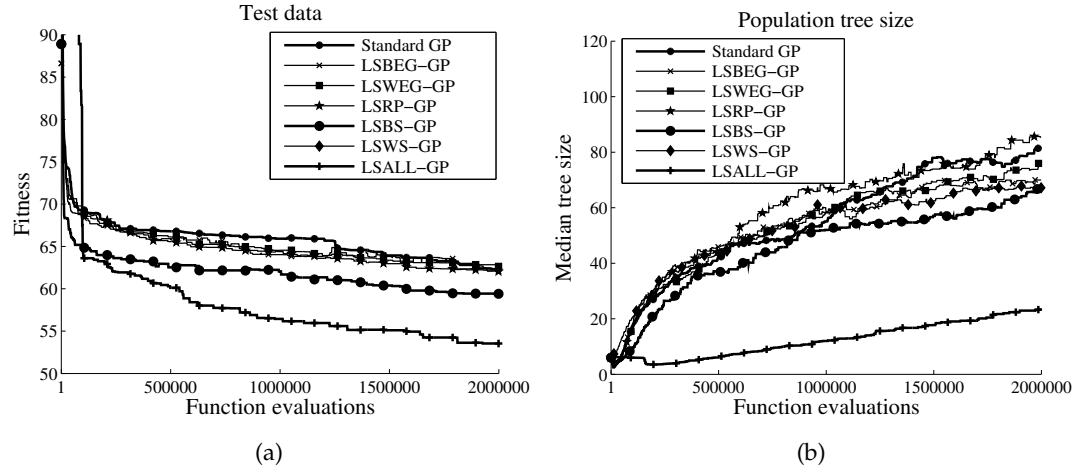


Figure 6.5 – Results for Tower problem plotted with respect to total function evaluations: (a) Fitness over test data; and (b) Average program size. Both plots show median values over 30 independent runs.

Table 6.3 – Summary of median fitness computed over the test set of each problem. The *Sample* column indicates the number of function evaluations performed; bold indicates the best results.

Problem	Sample	S-GP	LSBEG-GP	LSWEG-GP	LSRP-GP	LSBS-GP	LSWS-GP	LSALL-GP
Keijzer-6	250,000	0.2	0.2	0.5	0.54	**0.15**	0.29	0.46
	1000000	0.17	0.17	0.49	0.22	0.13	0.28	**0.11**
	2000000	0.17	0.17	0.5	0.23	0.13	0.25	**0.04**
Korns-12	250,000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
	1000000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
	2000000	**1.04**	1.05	1.05	1.05	1.06	1.06	**1.04**
Vladislavleva-4	250,000	**0.28**	0.47	0.31	0.35	1.03	0.34	0.5
	1000000	1.03	1.33	0.98	0.6	1.89	0.99	**0.18**
	2000000	1.01	1.85	1.18	0.68	4.3	0.74	**0.18**
Nguyen-7	250,000	0.02	0.01	0.02	0.005	**0.002**	0.02	0.003
	1000000	0.004	0.009	0.006	0.003	0.001	0.007	**0.0006**
	2000000	0.002	0.008	0.004	0.002	0.001	0.007	**0.0003**
Pagie-1	250,000	0.15	0.1	0.15	0.16	**0.02**	0.14	0.05
	1000000	0.1	0.06	0.07	0.1	0.01	0.09	**0.01**
	2000000	0.07	0.05	0.06	0.07	0.01	0.06	**0.006**
Tower	250,000	67.5	66.9	67	67.4	63.2	66.8	**61.98**
	1000000	65.92	64	64.54	64.39	62	64.52	**56.43**
	2000000	62.31	62.08	62.64	62.03	59.41	62.42	**53.53**

6.7 Conclusions

This work studies the problem of integrating a local optimization process into a GP, using a Lamarckian HSE memetic approach. A comparative study is performed, evaluating different ways in which to incorporate a LS during the basic evolutionary process of GP, evaluating performance on symbolic regression problems. A simple tree parametrization is proposed, bounding the size of parameter space for each individual tree, even if bloat occurs during the run. The local optimization is done by a trust region technique that determines the optimal coefficients posing a basic non-linear curve fitting problem. As stated before, it is not clear what might be the best strategy to incorporate LS into GP, so different stochastic and deterministic variants are extensively evaluated over a set of widely used benchmark problems. In particular, each method uses different heuristic decisions to determine which individuals in the GP run should be subject to a local optimization process; experimental results suggest the following.

In general, it seems that a memetic GP almost always outperforms a standard GP, in terms of both solution quality and solution size. Moreover, among the different methods that were tested, several insights can be gathered. First, it does not appear to be beneficial to use a LS as another mutation strategy, such that all individuals might be candidates for a LS, given the average performance of LSRP-GP. Second, it does not seem useful to apply LS to the worst individuals in the population, as seen by the performance of LSWEG-GP and LSWS-GP. Third, many works have used the simple deterministic heuristic of applying LS to the best solution found, either at the end of the run or at each generation. However, this to does not seem to be an adequate strategy, given the performance of LSBEG-GP. Finally, of all the methods, the best performance was achieved when LS is applied to all of the solutions (LSALL-GP) or to random individuals chosen from the top percentile (w.r.t. fitness) of the population (LSBS-GP).

For all problems, the best performance was achieved by LSALL-GP. However, if only a small amount of computational effort is feasible, then LSBS-GP seems to be the best option, given its fast convergence. Moreover, the difference in computational effort between both methods is understated in the results presented here, since only the total iterations in the LS are considered, omitting the total effort devoted towards computing approximate derivatives or matrix inversions. Nevertheless, LSALL-GP also shows a substantial ability to curtail bloating during the search, this was indeed expected. Since the search process in LSALL-GP focuses primarily on parameter optimization, limiting the total of syntactic search performed by the GP crossover and mutation operators; i.e., the total number of generations is quite low for LSALL-GP, basically eliminating the possibility of bloating.

Future work will be centered on exploring and evaluating other parametrization schemes. Moreover, a method must be implemented that determines if a tree requires linear or non-linear parameter optimization, in order to simplify the process whenever possible and to use the parameter values as decision criteria to prune unnecessary subtrees. Finally, the GP search operators could be enhanced to explicitly take into account the parameter values of a tree.

Acknowledgments

Funding provided by CONACYT (Mexico) Basic Science Research Project No. 178323, DGEST (Mexico) Research Projects No.5149.13-P and TIJ-ING-2012-110, and FP7-PEOPLE-2013-IRSES project ACOBSEC financed by the European Commission with contract No. 612689.

References

CHEN, X., Y.-S. ONG, M.-H. LIM et K. C. TAN. 2011, «A multi-facet survey on memetic computation», *Trans. Evol. Comp*, vol. 15, nº 5, p. 591–607. 138, 140

COELLO, C. A. C., G. B. LAMONT et D. A. V. VELDHUIZEN. 2006, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA. 138

COLEMAN, T. F. et Y. LI. 1993, «An interior trust region approach for nonlinear minimization subject to bounds», cahier de recherche, Ithaca, NY, USA. 142

COLEMAN, T. F. et Y. LI. 1994, «On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds», *Mathematical Programming*, vol. 67, nº 1, p. 189–224, ISSN 1436-4646. 142

DE JONG, K. 2006, *Evolutionary Computation: A Unified Approach*, Bradford Book, Mit Press. 138

DUNN, E., G. OLAGUE et E. LUTTON. 2006, «Parisian camera placement for vision metrology», *Pattern Recogn. Lett.*, vol. 27, nº 11, p. 1209–1219. 138

EIBEN, A. E. et J. E. SMITH. 2003, *Introduction to Evolutionary Computing*, SpringerVerlag. 138

EMMERICH, M., M. GRÖTZNER et M. SCHÜTZ. 2001, «Design of graph-based evolutionary algorithms: A case study for chemical process networks», *Evol. Comput.*, vol. 9, nº 3, p. 329–354. 139

ESKRIDGE, B. et D. HOUGEN. 2004, «Imitating success: A memetic crossover operator for genetic programming», dans *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, IEEE Press, Portland, Oregon, p. 809–815. 140

GILL, P. E., W. MURRAY et M. H. WRIGHT. 1981, *Practical optimization*, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London. 138

GRAFF, M., R. PEA et A. MEDINA. 2013, «Wind speed forecasting using genetic programming», dans *IEEE Congress on Evolutionary Computation*, IEEE, p. 408–415. 140

HORNBY, G. S., J. D. LOHN et D. S. LINDEN. 2011, «Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission», *Evol. Comput.*, vol. 19, nº 1, p. 1–23. 138

KEIJZER, M. 2003, «Improving symbolic regression with interval arithmetic and linear scaling», dans *Proceedings of the 6th European Conference on Genetic Programming*, EuroGP'03, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-00971-X, p. 70–82. URL http://dl.acm.org/citation.cfm?id=1762668.1762676. 143

KORNS, M. F. 2011, «Accuracy in symbolic regression», dans *Genetic Programming Theory and Practice IX*, édité par R. Riolo, E. Vladislavleva et J. H. Moore, chap. 8, Genetic and Evolutionary Computation, Springer, Ann Arbor, USA, p. 129–151, doi:doi:10.1007/978-1-4614-1770-5_8. 143, 144

KOZA, J. 2010, «Human-competitive results produced by genetic programming», *Genetic Programming and Evolvable Machines*, vol. 11, nº 3, p. 251–284. 138

KOZA, J. R. 1992, *Genetic programming: on the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, USA, ISBN 0-262-11170-5. 138

LAWSON, C. L. et R. J. HANSON. 1995, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics. 142

LOHMANN, R. 1991, «Proceedings of parallel problem solving from nature (ppsn i) first workshop», dans *Proceedings from the 16th European Conference on Genetic Programming, EuroGP 2013, LNCS*, vol. 496, Springer-Verlag, p. 198–208. 139

LUKE, S. 2013, *Essentials of Metaheuristics*, 2e éd., Lulu. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/. 138

MCCONAGHY, T. 2011, «FFX: Fast, scalable, deterministic symbolic regression technology», dans *Genetic Programming Theory and Practice IX*, édité par R. Riolo, E. Vladislavleva et J. H. Moore, chap. 13, Genetic and Evolutionary Computation, Springer, Ann Arbor, USA, p. 235–260. 141

MCDERMOTT, J., D. R. WHITE, S. LUKE, L. MANZONI, M. CASTELLI, L. VANNESCHI, W. JASKOWSKI, K. KRAWIEC, R. HARPER, K. DE JONG et U.-M. O'REILLY. 2012, «Genetic programming needs better benchmarks», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, ACM, New York, NY, USA, p. 791–798. 143, 144

MORÉ, J. J. et D. C. SORENSEN. 1983, «Computing a trust region step», *SIAM J. Scientific and Statistical Computing*, vol. 4, p. 553–572. 142

OLAGUE, G. et L. TRUJILLO. 2011, «Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming», *Image Vision Comput.*, vol. 29, no 7, p. 484–498. 138

PAGIE, L. et P. HOGEWEG. 1998, «Evolutionary consequences of coevolving targets», *Evolutionary Computation*, vol. 5, p. 401–418. 143

POLI, R., W. B. LANGDON et N. F. MCPHEE. 2008, *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd. 138

SHULTZ, G., R. SCHNABEL, R. BYRD et C. U. A. B. D. O. C. SCIENCE. 1982, *A Family of Trust Region Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties*, Defense Technical Information Center. URL http://books.google.com.mx/books?id=5bI7OAAACAAJ. 142

SILVA, S. et J. ALMEIDA. 2003, «Gplab–a genetic programming toolbox for matlab», dans *Proceedings of the Nordic MATLAB conference*, édité par L. Gregersen, p. 273–278. 143

SILVA, S. et E. COSTA. 2009, «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories», *Genetic Programming and Evolvable Machines*, vol. 10, no 2, p. 141–179. 139, 141

SMART, W. et M. ZHANG. 2004, «Continuously evolving programs in genetic programming using gradient descent», dans *Proceedings of The Second Asian-Pacific Workshop on Genetic Programming*, édité par R. I. Mckay et S.-B. Cho, Cairns, Australia, p. 16pp. 140

SORENSEN, D. 1982, *Newton's Method with a Model Trust Region Modification*, Defense Technical Information Center. 142

SPECTOR, L. 2006, *Automatic Quantum Computer Programming: A Genetic Programming Approach (Genetic Programming)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA. 138

STEIHAUG, T. 1983, «The Conjugate Gradient Method and Trust Regions in Large Scale Optimization», *SIAM Journal on Numerical Analysis*, vol. 20, no 3, p. 626–637. 142

TOPCHY, A. et W. F. PUNCH. 2001, «Faster genetic programming based on local gradient search of numeric leaf values», dans *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, édité par L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon et E. Burke, Morgan Kaufmann, p. 155–162. 140

TRUJILLO, L., E. NAREDO et Y. MARTÍNEZ. 2013, «Preliminary study of bloat in genetic programming with behavior-based search», dans *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, *Advances in Intelligent Systems and Computing*, vol. 227, Springer International Publishing, p. 293–305. 143

UY, N. Q., N. X. HOAI, M. O'NEILL, R. I. MCKAY et E. GALVÁN-LÓPEZ. 2011, «Semantically-based crossover in genetic programming: application to real-valued symbolic regression», *Genetic Programming and Evolvable Machines*, vol. 12, nº 2, p. 91–119. 143

VLADISLAVLEVA, E. J., G. F. SMITS et D. DEN HERTOG. 2009, «Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming», *Trans. Evol. Comp*, vol. 13, nº 2, doi:10.1109/TEVC.2008.926486, p. 333–349, ISSN 1089-778X. URL http://dx.doi.org/10.1109/TEVC.2008.926486. 143, 144

WAGNER, S. et G. KRONBERGER. 2012, «Algorithm and experiment design with heuristic lab: An open source optimization environment for research and education», dans *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion*, GECCO Companion '12, ACM, New York, NY, USA, p. 1287–1316. 140

WANG, P., K. TANG, E. P. K. TSANG et X. YAO. 2011, «A memetic genetic programming with decision tree-based local search for classification problems», dans *IEEE Congress on Evolutionary Computation*, IEEE, p. 917–924. 140

WHITE, D. R., J. MCDERMOTT, M. CASTELLI, L. MANZONI, B. GOLDMAN, G. KRONBERGER, W. JA'SKOWSKI, U.-M. O'REILLY et S. LUKE. 2013, «Better gp benchmarks: community survey results and proposals», *Genetic Programming and Evolvable Machines*, vol. 14, nº 1, doi:10.1007/s10710-012-9177-2, p. 3–29. URL http://link.springer.com/article/10.1007/s10710-012-9177-2. 143

WORM, T. et K. CHIU. 2013, «Prioritized grammar enumeration: Symbolic regression by dynamic programming», dans *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, GECCO '13, ACM, New York, NY, USA, p. 1021–1028. 141

YUAN, J. Y. 1996, «Numerical methods for generalized least squares problems», *Journal of Computational and Applied Mathematics*, vol. 66, nº 12, p. 571 – 584. 142

ZHANG, M. et W. SMART. 2004, «Genetic programming with gradient descent search for multiclass object classification», dans *Genetic Programming 7th European Conference, EuroGP 2004, Proceedings*, *LNCS*, vol. 3003, édité par M. Keijzer, U.-M. O'Reilly, S. M. Lucas, E. Costa et T. Soule, Springer-Verlag, Coimbra, Portugal, p. 399–408. 140

Chapter 7

A Local Search Approach to Genetic Programming for Binary Classification

This chapter is related to the PhD thesis of Emigdio Z. Flores (ITT Tijuana) and the European project ACOBSEC, and has been published at the conference GECCO, Jul 2015, Madrid, Spain. Work carried out with Emigdio Z. Flores, Leonardo Trujillo and Oliver Schütze.

Contents

Abstract

In the domain of evolutionary computation, genetic programming (GP) excels as an algorithm for the automatic induction of symbolic expressions. In standard GP, a search is performed over a syntax space defined by the algorithm primitives, looking for the best expressions that minimizes a computed cost function based on a training set. However, standard GP lacks a numerical optimization method to fine tune the implicit parameters of each candidate solution, instead performing more exploratory search operators at the syntax level. This work proposes a memetic GP, tailored for binary classification problems, extending previous work on symbolic regression. In particular, each node in a GP is weighted by a real-valued parameter, which is then numerically optimized using a continuous transfer function and the trust-region algorithm used as a local search method. Experimental results show that potential classifiers produced by GP are improved by the local searcher, and hence the overall search is improved achieving substantial performance gains. The insight found by this research gives us a more complete view of the local search effects over GP, since we present results that are competitive with state-of-the-art methods on well-known benchmarks.

7.1 Introduction

The general goal of the Genetic programming (GP) paradigm is to evolve specialized syntactic expressions that can solve a user defined problem, mostly used in supervised learning tasks. In particular, GP is widely used to generate mathematical functions, or operators, that solve symbolic regression and classification problems, which can be stated as follows. The goal is to search for the symbolic expression $K^O : \mathbb{R}^p \to \mathbb{R}$ that best fits a particular training set $\mathbb{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n input/output pairs with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, states as

$$(K^O, \boldsymbol{\theta}^O) \leftarrow \underset{K \in \mathbb{G}; \boldsymbol{\theta} \in \mathbb{R}^m}{arg\,min} \; f(K(\mathbf{x}_i, \boldsymbol{\theta}), y_i) \; with \; i = 1, \ldots, p \,, \tag{7.1}$$

where $\mathbb{G}$ is the solution or syntactic space defined by the primitive set $\mathbb{P}$ of functions and terminals, f is the fitness function which is based on the difference between a program's output $K(\mathbf{x}_i, \boldsymbol{\theta})$ and the desired output y_i, and $\boldsymbol{\theta}$ is a particular parametrization of the symbolic expression K, assuming m real-valued parameters. This dual problem, of simultaneously optimizing syntax (structure) as well as its parametrization is also discussed in EMMERICH et collab. [2001]; LOHMANN [1991]. The authors differentiate between two possible approaches towards solving such a task. The first group is *hierarchical structure evolution* (HSE), when $\boldsymbol{\theta}$ has a strong influence on fitness, and thus a Local Search (LS) is required at each generation of the global (syntactic) search, configured as a nested process. The second group is called *simultaneous structure evolution* (SSE), when $\boldsymbol{\theta}$ has a marginal effect on fitness, in such cases a single evolutionary loop could simultaneously optimize both syntax and parameters. These are very abstract categories, but it is reasonable to state that the standard GP approach falls in the SSE group. Here, standard GP refers to a GP system that uses a tree representation, subtree crossover and mutation, and tournament selection, as proposed by John Koza KOZA [1992].

Many researchers include constants or random numerical values as terminal elements within the primitive set, to allow the evolutionary process to explicitly include numerical values within the evolved expressions. However, mutation events are mostly rare compared to the use of subtree crossover, thus in GP parameter optimization is usually not an integral part of the search process. On the contrary, GP performs a highly explorative search, since the search operators can produce large fitness changes with only modest syntactic modifications or vice-versa. The inclusion of a LS method aimed at optimizing the (sometimes implicit) numerical parameters of the evolved expression could enhance search

performance. Therefore, this work proposes a methodology to parameterize GP trees: to employ a LS method in order to numerically tune them, and to implement this process in binary classification problems. Here, a domain specific fitness function is used, based on the Receiver Operating Characteristic (ROC) curve of each GP individual. Experimental results are encouraging, performance is significantly improved relative to a standard GP search, based both on fitness and solution size. Moreover, by using widely used real-world benchmarks, it is clear that our proposal compares favorably with other proposals from the GP and machine learning literature.

The remainder of this work is organized as follows. Section 7.2 reviews related work and highlights our main contributions. Then, Section 7.3 describes our approach and Section 7.4 presents the experimental work. Finally, concluding comments are given in Section 7.5.

7.2 Previous work

Many works have studied how to combine an EA with a local optimizer, usually referred to as memetic algorithms CHEN et collab. [2011]. The basic idea is to include within the optimization process an additional operator that takes an individual as an initial point and searches for its optimal neighbor. Such a strategy can help guarantee that the local region around each individual is fully exploited. However, these algorithms can have a high computational overhead or produce overfitted solutions. These issues aside, memetic algorithms have produced impressive results in a variety of scenarios CHEN et collab. [2011].

A useful taxonomy of this type of memetic algorithms can be derived based on how inheritance is carried out during the evolutionary process CHEN et collab. [2011]. Suppose that $(\mathbf{h}, \mathbf{h}^o) \in \mathbb{G}^2$, where $\mathbf{h}$ is an individual solution and $\mathbf{h}^o$ is the solution generated after a LS is applied on $\mathbf{h}$. Obviously, for a minimization problem, $f(\mathbf{h}^o) \leq f(\mathbf{h})$, where f is the objective function. Thus, $\mathbf{h} \neq \mathbf{h}^o$, unless $\mathbf{h}$ was in fact a local optima. Then, a memetic algorithm could proceed in two distinct ways with respect to inheritance. In a *Lamarckian* algorithm, the traits acquired during the local search, captured in $\mathbf{h}^o$, replace those of the original individual $\mathbf{h}$; i.e., the inheritance of acquired characteristics $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}^o, f(\mathbf{h}^o))$. On the other hand, in a *Baldwinian* algorithm, the local optimization process only modifies the fitness of an individual; $(\mathbf{h}, f(\mathbf{h})) \rightarrow (\mathbf{h}, f(\mathbf{h}^o))$; i,e., ontogenic evolution.

When applying a LS strategy to tree-based GP, there are basically two broad approaches to follow. (1) To apply a LS on the syntax of a tree or (2) to apply it numerically on the parameters of the tree, as described in Equation 7.1. Regarding the latter, two recent works are noteworthy. In AZAD et RYAN [2014], the authors apply a greedy search on a randomly chosen GP node, attempting to determine the best function to use in that node among all the functions in the primitive set, constrained only by the arity of the initial function. Moreover, to reduce computational overhead the authors apply a heuristic decision rule to decide which trees are subject to local optimization, in particular they prefer smaller trees in the population. This heuristic has the positive effect that it biases the search towards smaller trees, so it is adopted in our current proposal. In WANG et collab. [2014], the authors propose a multiobjective GP algorithm to evolve decision tree classifiers, and use two specialized mutation operators that limit their resulting phenotypic variations within a local neighborhood of the parent classifier.

Regarding the optimization of numerical parameters within the tree, the following works are palpable. In TOPCHY et PUNCH [2001], gradient descent is used to optimize numerical constants within a GP tree, achieving good results on symbolic regression problems. However, the work only optimizes the value of the terminal elements (tree leaves), and it does not consider parameters within internal nodes. Additionally, the paper only considers training fitness, a highly deceptive measure of learning. Similarly, in ZHANG et SMART [2004] and GRAFF et PE [2013] a LS algorithm is used to optimize the value of constant terminal elements. In ZHANG et SMART [2004] gradient descent is used and tested on classification

problems, while in GRAFF et PE [2013] uses Resilient Backpropagation (RPROP) and evaluates the proposal on a real-world problem. In ZHANG et SMART [2004], the authors apply gradient descent on every individual of the evolving population, an obvious computational bottleneck, while in GRAFF et PE [2013] only applies RPROP on the best individual from each generation. However, it is not evident which strategy can offer the best results in new scenarios, particularly since both GRAFF et PE [2013]; ZHANG et SMART [2004] evaluate their approaches on specific problem instances.

Other recent works have completely changed the basic GP framework to include an explicit parametrization of syntactic expressions. The fast function extraction (FFX) algorithm MCCONAGHY [2011], for instance, poses the symbolic regression problem as that of finding a linear combination of a subset of candidate basis functions. Thus, FFX builds linear in parameter models, and optimizes using a modified version of the elastic net regression technique, eliminating the evolutionary process altogether. In the prioritized grammar enumeration (PGE) technique WORM et CHIU [2013], dynamic programming replaces the the basic search operators of traditional GP, and numerical parameters are optimized using the non-linear Levenberg-Marquardt algorithm.

In the chapter 6 a very simple parametrization approach for GP trees is proposed, by constraining the number of internal parameters of each tree regardless of its size. That work evaluates several strategies to determine which individuals are subject to the LS process, concluding that it is often best to apply LS on the entire population or a subset of the best individuals. The LS used is called Trust Region optimization SORENSEN [1982], and results showed substantial improvements on several symbolic regression problems relative to a standard GP. A similar approach was developed by KOMMENDA et collab. [2013], with some noteworthy differences; parameters are only used to replace constants terminals, and each tree is enhanced by adding a linear upper tree, that effectively adds a weight coefficient and a bias to the entire tree. Then, the Levenberg-Marquardt optimizer is used to find the optimal values for these parameters.

Here, we propose a Lamarckian memetic algorithm, where the main goal is to combine the explorative capabilities of GP without changing its canonical implementation, and incorporate stronger exploitative features by means of numerical optimization.

7.3 Integrating the Local Search mechanism within GP

In supervised learning, classification paradigm has several approaches. In this work we focus exclusively on binary classification. For this problematic, we begin defining it as the task of classifying the elements of a given set into two groups on the basis of classification rule. We assign a distinct label to each of these groups (Class 1 and Class 2).

For binary classification, GP is used as a supervised learning problem, where a training set x of n-dimensional patterns with a known classification, are used to derive a mapping function $g(x) : I\!R^n \rightarrow \{0, 1\}$, with a threshold δ as classification decision. Afterwards, for convenience, we transform the classification problem into a regression one by inserting a continuous function in order to get gradient information, and to be able to use a numerical optimizer.

7.3.1 GP Tree Parameterization

First, we follow the strategy proposed in KOMMENDA et collab. [2013], where a small linear subtree is added on top of the root node of the original tree $K(x)$, such the new extended tree $K'(x)$ is given by

$$K'(x) = \theta_2 + \theta_1(K(x)) \,, \tag{7.2}$$

where θ_1 and θ_2 are the first two parameters from $\boldsymbol{\theta}$, the tree parametrization, as shown in Figure 7.1. This subtree adds an additional strength to the original tree, providing shifting

and scaling properties, so the optimizer later on can have more freedom in its capabilities to find a better solution. This approach is closely related to the linear scaling technique KEIJZER [2003], allowing the syntactic search operators to focus on evolving the desired shape of the evolved function without worrying about optimizing the scale or bias of the output. In the proposed memetic setting, the scale and bias parameters are optimized by the LS process.

We now propose a tree parameterization that follows the next scheme. For each node n_k in a given tree we add an exclusive weight coefficient $\theta_k \in \mathbb{R}$, such that each node is now defined by

$$n'_k = \theta_k n_k \ , \tag{7.3}$$

where n'_k is the new modified node, $k \in \{1, ..., r\}$, $r = |Q|$ and Q is the tree representation. Notice that each node has an unique parameter that can be modified to help meet the overall optimization criteria of the non-linear expression.

At the beginning of the GP run, each parameter is initialized by $\theta_i = 1$, where $|\theta| = r$. During the GP syntax search, subtrees belonging to different individuals are swapped, added or removed (following the standard crossover/mutation rules) together with its corresponding parameters, without affecting their values. This follows a memetic search process with Lamarckian inheritance.

7.3.2 A Continuous Transfer Function

In a binary classification problem, the desired output is either 0 or 1, a discrete output. Indeed, this discontinuity will have an homologous discontinuity in parameter space, and thus the LS algorithm can easily fail. To provide a continuous output, and provide gradient information to the LS method, a transfer function is added as a root node. Here, a sigmoid function is used, producing the following non-linear tree

$$sig(K'(\boldsymbol{x}, \boldsymbol{\theta})) = \frac{1}{1 + e^{\rho(K'(\boldsymbol{x},\boldsymbol{\theta})-\delta)}} \ , \tag{7.4}$$

where K' is the program output evaluated over the input vector $\boldsymbol{x}$, $\boldsymbol{\theta}$ is the parameter vector corresponding to the program K', ρ is the sigmoid slope parameter and δ is the reference point (classification threshold) where the function has a value of 0.5. The sigmoid function is normalized to the range [0,1]: therefore, if $sig(K'(\boldsymbol{x}, \boldsymbol{\theta})) \leq 0.5$ then the class label is 0, and 1 otherwise. The proposed tree extensions and parameterizations are depicted in Figure 7.1.

Figure 7.1 visualizes a tree transformation, by inserting parameters, linear subtree and sigmoid at the root node.

The following subsection explains how the optimization is performed in order to achieve an improved quality solution in terms of classification.

7.3.3 The Local Search Mechanism

In this work we treat each tree as a non-linear expression. We also assume a non-discontinuity for the evaluated tree over the interval constrained by the input training data.
The problem to be solved is an optimization one, where we want to best fit a non-linear function, constrained by parameters, to an output vector containing either zero or ones values (belonging to each class respectively). It can be formally defined as

$$\min_{\boldsymbol{\theta}} \|sig(K'(\boldsymbol{x}, \boldsymbol{\theta})) - \boldsymbol{y}\|_2^2 = \min_{\boldsymbol{\theta}} \sum_i (sig(K'(x_i, \boldsymbol{\theta})) - y_i)^2 \ , \tag{7.5}$$

where $\boldsymbol{x}$ is the input data vector, $\boldsymbol{y}$ is the class label data vector (either 0 or 1), sig is the sigmoid calculated over the program K', i is the index for the problem instances and $\boldsymbol{\theta}$ is the parameter vector.
The problem presented in (7.5) can be solved using different techniques YUAN [1996]LAWSON et HANSON [1995]. In this work, we prefer to use a well known algorithm, the Trust

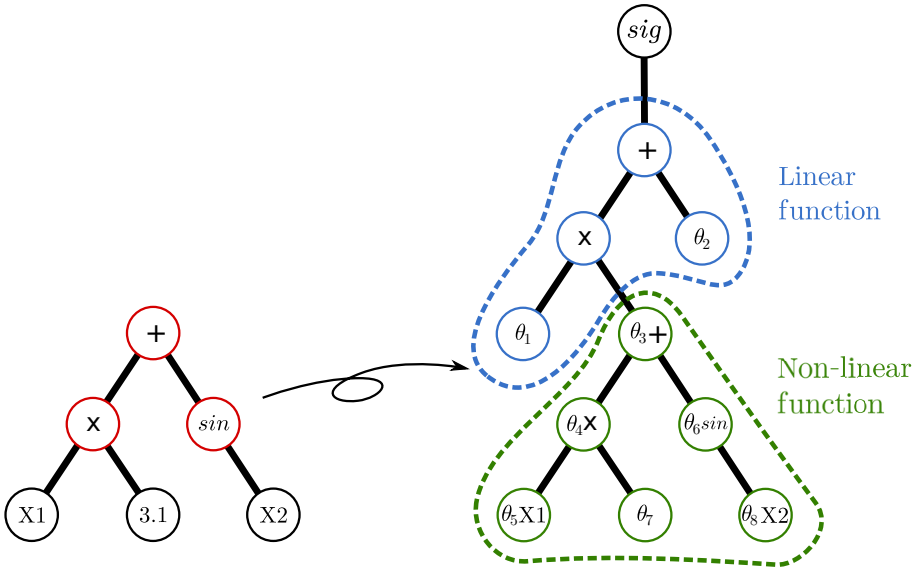


Figure 7.1 – Example of tree transformation, product of parameterization and subtree addition.

Region optimizer, belonging to the family of non-linear Gauss-Newton methods SORENSEN [1982]. One example of this family is the recent approach to the Levenberg-Marquardt algorithm, which follows the Trust Region concept.

In the Trust Region algorithm, an iterative process is performed with the following goal

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^m} \|sig(K'(\boldsymbol{x},\boldsymbol{\theta})) - \boldsymbol{y}\|_2^2,\ l_i \leq \theta_i \leq u_i\ \forall i \in \{1..m\}\ , \tag{7.6}$$

where $l_i \in \{\mathbb{R} \cup \{-\infty\}\}$, $u_i \in \{\mathbb{R} \cup \{\infty\}\}$, and $K'(x,.) : \mathbb{R}^m \rightarrow \mathbb{R}$. Conceptually, a Trust Region approach replaces a m-dimensional unconstrained optimization problem by a m-dimensional constrained problem. This results in an approximate solution, since it does not need to be solved with high accuracy. One of the appealing points of these methods is their strong convergence properties COLEMAN et LI [1992]. The idea behind a Trust Region method is simple. The increment $s_k = x_{k+1} - x_k$ is an approximate solution to a quadratic subproblem with a bound on the step size

$$\min_{s\in\mathbb{R}^m} \left\{ \psi_j(s) \stackrel{def}{=} g_j^T s + \frac{1}{2} s^T B_j s : \|\overline{D}_j\| \leq \Delta_j \right\}\ , \tag{7.7}$$

where $g_j \stackrel{def}{=} \nabla f(x_j)$, B_j is a symmetric approximation to the Hessian matrix $\nabla^2 f(x_j)$, $\overline{D}_j$ is a scaling matrix, and Δ_j is a positive scalar representing the trusted region size. Solving (7.7) efficiently is not a trivial task, see [MORÉ et SORENSEN, 1983; SORENSEN, 1982]. Figure 7.2 shows an illustration of the trust region method.

Here, the method proposed in COLEMAN et LI [1993] is used, which does not require the solution of a general quadratic programming subproblem at each iteration.
Figure 7.3 shows the effect of performing LS over a specific individual.

7.3.4 The Fitness Function

In the case of binary classification, a simple method is to consider only accuracy in the training stage EGGERMONT et collab. [1999]; WINKLER et collab. [2007]. In this work we use the following fitness measure $fitness = 1 - Acc$ where Acc (accuracy; computed as the ratio of correctly classified instances among total number of cases) is calculated based on the best threshold found using the ROC curve. The ROC curve, which is commonly used by the machine learning community, can provide a robust measure of classifier performance

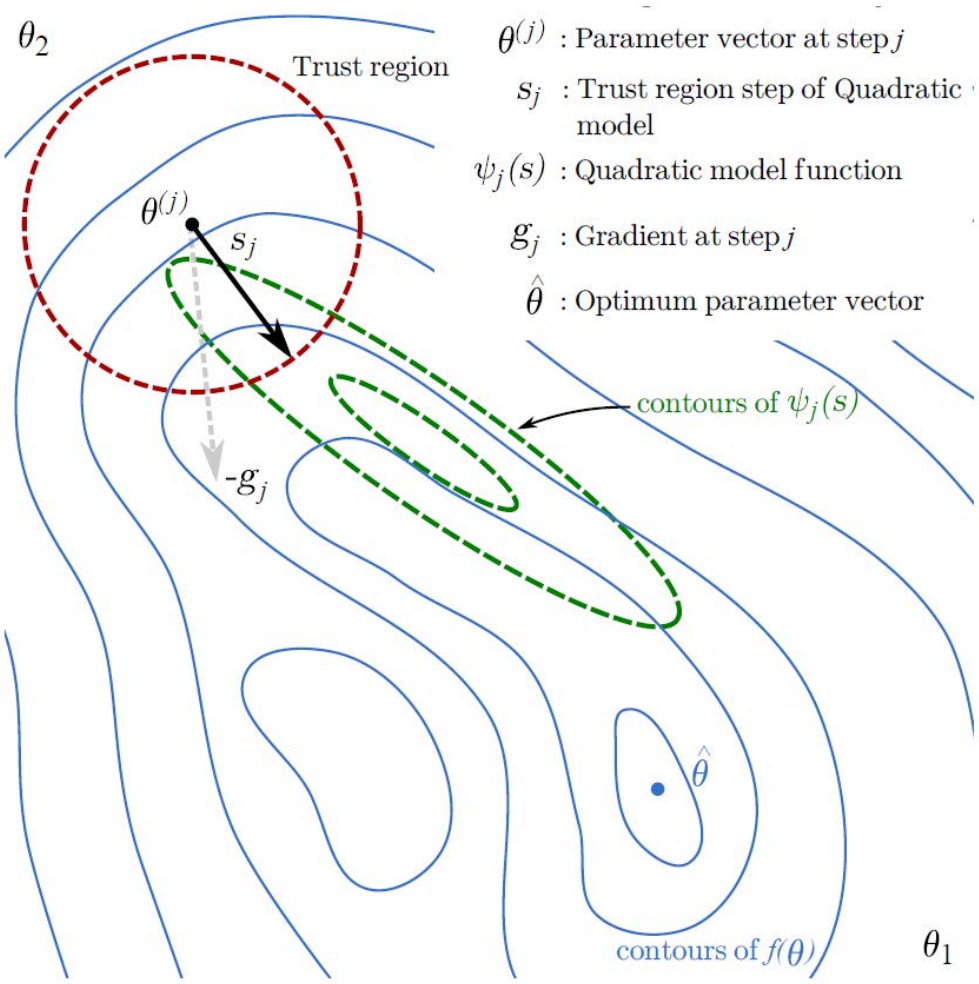


Figure 7.2 – Visualization of the trust region algorithm showing the landscape of $f(\theta)$ as the objective function, equivalent to Equation 7.6, at iteration j.

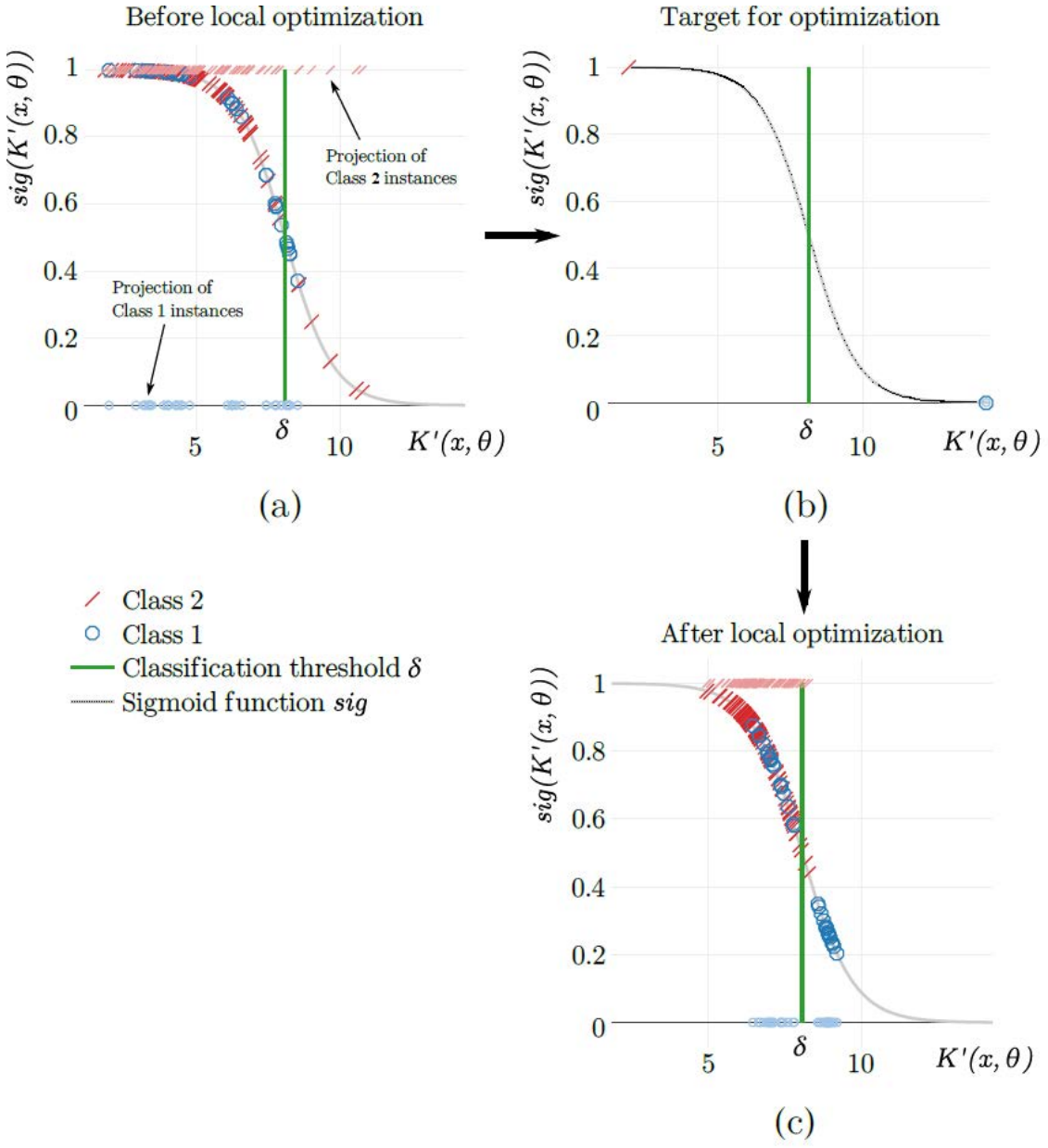


Figure 7.3 – Example of the optimization effect over an actual individual for the Parkinsons problem LITTLE et collab. [2007]. Even though the solution was clearly a bad classifier, after optimization accuracy improved. Projection of both classes are shown just for clearer visualization on the position of instances from classification threshold.

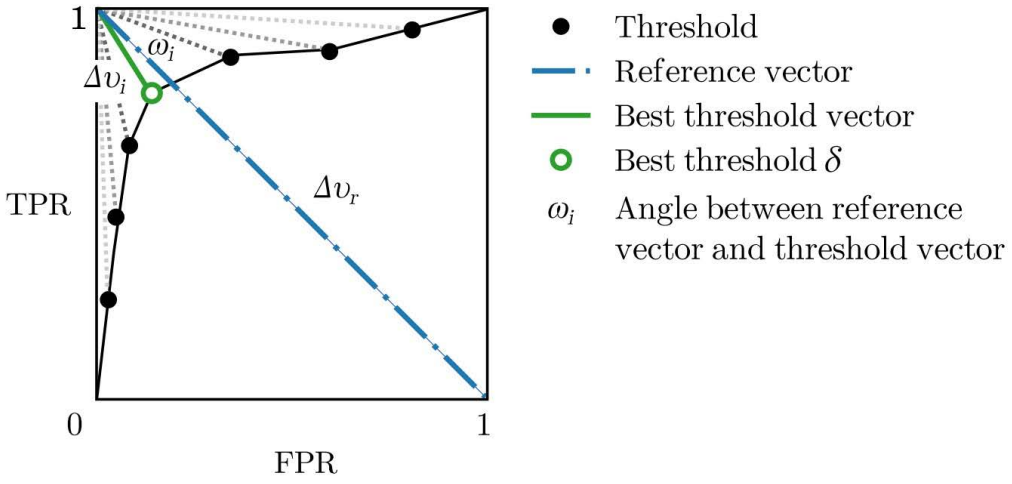


Figure 7.4 – Example of a ROC curve and its best threshold. Note that each tree will generate a different ROC curve, thus presenting a different classification threshold.

even with an unbalanced dataset. Each operating point on a ROC curve represents the classification accuracy at a single threshold δ_i. The ROC curve is constructed by changing the classification threshold and evaluating its True Positives Rate (TPR) and False Positives Rate (FPR) values. These are calculated with TPR=TP/P and FPR=FP/N, respectively, where TP is the True Positive value, FP is the False Positive value, P is the total of positive instances and N the negative ones.

In the implementation presented here, we used 10 uniformly spaced threshold values within $[min(K'(\boldsymbol{x}, \boldsymbol{\theta})), max(K'(\boldsymbol{x}, \boldsymbol{\theta}))]$, which is the range given by minimum and maximum output of the extended tree, over all fitness cases. This methodology is similar as the one presented in BHOWAN et collab. [2012]. In the ROC curve, the top-left area represents the most accurate classification results. Ideally, a perfect classifier should have a TPR value of one with zero FPR. Based on this, we propose a simple method to choose the best threshold picked up from the ROC curve. A vector with 45 degree slope is traced with coordinates [0,1]-[1,0], in ROC space, which represent our reference vector v_r. We then build a numerical array $\overline{\omega}$ containing the angle differences between each threshold vector v_i, with coordinates [0,1]- $[FPR_i, TPR_i]$ and the reference vector, as seen in Figure 7.4. This is calculated by

$$\Delta v_r = v_r(y) - v_r(x), \Delta v_i = v_i(y) - v_i(x) \tag{7.8}$$

and

$$\omega_i = acos\left(\frac{(\Delta v_r \cdot \Delta v_i)|\Delta v_r|}{|\Delta v_i|}\right), \tag{7.9}$$

where v_i is the vector corresponding to each threshold δ_i and w_i is the difference angle expressed in radians. Consequently, the array $\overline{\omega} = [\omega_1, ..., \omega_k]$ is obtained. The threshold corresponding to the minimum value in this array will represent our chosen threshold for the particular classifier, as seen in Figure 7.4.

Once the best threshold has been chosen, True Positives, False Positives, False Negatives (FN) and True Negatives (TN) are calculated. Over this, overall accuracy can be determined following the next formula

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.10}$$

This is a typical classification measure used in several works EGGERMONT et collab. [1999]; WINKLER et collab. [2007]. The difference in this work is how the best threshold is chosen in order to calculate its accuracy.

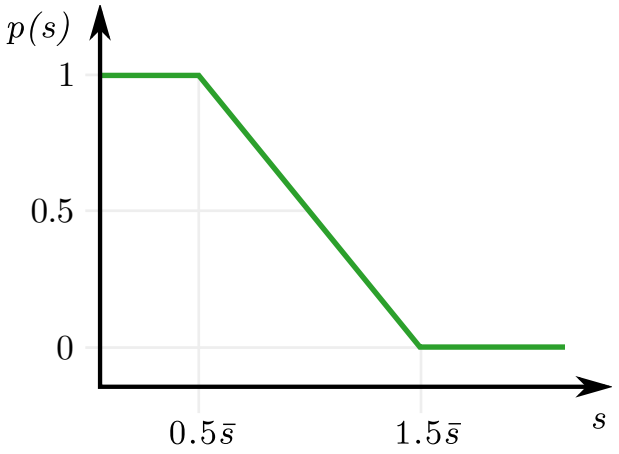


Figure 7.5 – Transfer function for $p(s)$ corresponding to the heuristic method based on individuals tree sizes, selected for performing LS.

As said before, the final fitness measure is calculated as $fitness = 1 - Acc$ which leads to a problem of minimization for GP.

7.3.5 Integrating LS into GP

Finally, we need to determine which individuals should be improved by the LS method. For instance, some researchers apply the LS to the best solution in the population, but TRUJILLO et collab. [2014] showed that better results are obtained when it is applied to all the population or a subset of the best solutions in each generation. However, in our case, the number of parameters grows proportionally with the size of the trees, providing incentive to limit the amount of times the LS is applied, especially for larger trees. Therefore, in this work we use the heuristic rule propose by Muhammad and Ryan in AZAD et RYAN [2014], which they tested with a syntactic LS process. In AZAD et RYAN [2014] the LS method is applied stochastically for each tree, based on a probability $p(s)$ determined by the tree size s (number of nodes) and the average size of the population $\overline{s}$, based on

$$p(s) = \begin{cases} y = c - \frac{s}{\overline{s}} & \text{if } 0 \leq y \leq 1 \\ 1 & \text{if } y > 1 \\ 0 & \text{otherwise .} \end{cases} \tag{7.11}$$

Based on Equation (7.11), shown in Figure 7.5, smaller trees are more likely to be optimized by the LS then larger trees; in particular, by setting $c = 1.5$ trees that are 50% larger than the population average are not subject to the LS process. This heuristic could provide two positive results. First, the LS method is mostly applied on small trees that have a relatively small number of parameters, simplifying the LS problem. Second, by focusing on improving smaller trees, this increases their chances for survival, increasing the chances of finding solutions that are not hampered by the bloat problem. Hereafter, we call this method Local Search Heuristic by Size (LSHS).

7.4 Experimentation

7.4.1 Experimental Setup

The algorithm LSHS was evaluated and compared with a canonical GP implementation as control method. The GP settings were the same between both variants, including the fitness function used for the training data. A random subset from original data was taken as test dataset and evaluated for the best individual each generation.

The LSHS algorithm was implemented over GPLAB[1] Matlab toolbox SILVA et ALMEIDA [2003]. The set of experiments covered a series of benchmark binary classification problems. These were taken from the UCI repository BACHE et LICHMAN [2013] and are considered real problems with distinct complexity. Table 7.1 shows a summary of some characteristics of each evaluated problem.

[1] http://gplab.sourceforge.net/

Table 7.1 – Binary classification problems summary used to evaluate proposed algorithm in this work.

Name	# of instances	# of features	Brief description
Parkinsons LITTLE et collab. [2007]	197	23	Biomedical voice measurements related to Parkinson's disease
Diabetes BACHE et LICHMAN [2013]	768	8	Diabetes present on Pima Indians patients
Wine BACHE et LICHMAN [2013]	178	13	Chemical analysis of wines
Sonar BACHE et LICHMAN [2013]	208	60	Sonar signals off metal cylinder at various angles and conditions
Wholesale BACHE et LICHMAN [2013]	440	8	Clients information of wholesale distributor
Banknote BACHE et LICHMAN [2013]	1372	5	Image information from genuine & forged banknote-like specimens
LSVT TSANAS et collab. [2014]	126	309	Signal data from speech rehabilitation treatment on Parkinson patients

Table 7.2 – GP parameters

Parameter	Value
Runs	20
Population	200
Function evaluations	1500000
Training set	70% of complete data
Testing set	30% of complete data
Crossover operator	Standard subtree crossover, 0.8 prob.
Mutation operator	Mutation probability per node 0.15
Tree initialization	Ramped Half-and-Half, max. depth 6
Function set	+,-,×, sin,cos,log,sqrt,tan,tanh
Terminal set	Input features, constants
Selection for reproduction	Tournament selection of size 7
Elitism	Best individual survives
Maximum tree depth	15

A couple of quality measurements were used to compare the results of the algorithm. One of these is the accuracy calculated over the test data. The other measure is the average population size, needed to assess the performance in terms of bloat SILVA et COSTA [2009] and consumed resources. The evolution of these measure is analyzed with respect to the total number of fitness function evaluations instead of generations, to account for the LS iterations. The evaluations performed by the iterative Trust Region algorithm were also taken in account to better justify the fair comparison between standard GP and LSHS. In this work, the Trust Region algorithm performs a maximum of 500 iterations. The stopping criteria is a predefined number of function evaluations. The GP configuration used in this research is listed in Table 7.2. The ratio for the training/testing data set was 0.7.

7.4.2 Results

Figure 7.6 summarizes the results of the tested algorithm over several classification problems. The figure shows convergence plots for training and testing fitness with respect to the number of fitness function evaluations. In each of the evaluated problems, the accuracy result of LSHS testing data is either similar or better compared to standard GP. At least in four of them the improvement in the method is significantly better.

Figures 7.6(a-c) shows the results for the problems Parkinsons, Diabetes and Wine. In all of them we can see a superior solution quality evolved by the LSHS method. Notice that for the Wine problem, considered an easy classification problem, standard GP error already reaches almost zero for testing data, yet the LSHS algorithm presents a perfect classification accuracy.

Figures 7.6(d-g) presents the performance results for Sonar, Wholesale, Banknote and LSVT problems. For the Sonar problem the accuracy reached in the testing data is closely similar to the one obtained by standard GP. In Banknote problem, the LSHS improvement is significant, where the classification error almost manages to reach zero.

Figure 7.7 discloses descriptive statistics from test fitness, at the end of each run, using notched box plots. The Wilcoxon rank sum test was calculated for the fitness test data, shown in first column of Table 7.3. The average population size is also presented in Figure Figure 7.7(b), and its corresponding rank sum test is given in the second column of Table 7.3. In both cases, the null hypothesis is that the underlying distributions in each pair of experiments have equal medians. The statistical tests show that the differences in performance exhibited by the standard GP and LSHS-GP are significant on four of the seven problems, particularly Parkinsons, Diabetes, Wholesale and Banknote, with $\alpha = 0.05$. Regarding size,

statistical difference was detected on five problems, namely Parkinsons, Diabetes, Wine, Wholesale and Banknote. It is interesting to note, that on all four problems where performance improved, size was also reduced. Here we can state that for these problems, the increased average size in standard GP does indeed represent bloat, since better solutions can be found while maintaining smaller populations. The Wine problem seems relatively easy for GP, the median performance of both algorithms is practically a perfect accuracy, but still GP evolves unnecessarily large populations, since the memetic GP with LSHS can produce equivalent performance with smaller populations in terms of average program size. The two outliers seem to be the Sonar and LSVT problems, two difficult problems on which no improvements in terms of size or fitness are obtained. It could be, that this is related to the large number of problem features in both cases, Sonar has 60 features and LSVT has 309, however confirming this hypothesis is left as future research. Table 7.4 presents an informal comparison with some recently published results for the seven problems tested here, using average classification accuracy. Thus providing a quick survey to contextualize the results presented above. The comparison is encouraging, the proposed LSHS method shows relatively strong performance on all problems, except on Parkinsons and LSVT. However, the best results on Parkinsons [MA et collab., 2014] uses a domain specific preprocessing stage, while other methods report similar results with our work. Similarly, [TSANAS et collab., 2014] uses a state-of-the-art feature selection algorithm to reduce the dimensions of the problem given to the classifier, important for this dataset since it contains 309 features, while GP takes the entire feature vectors as input, a clear disadvantage.

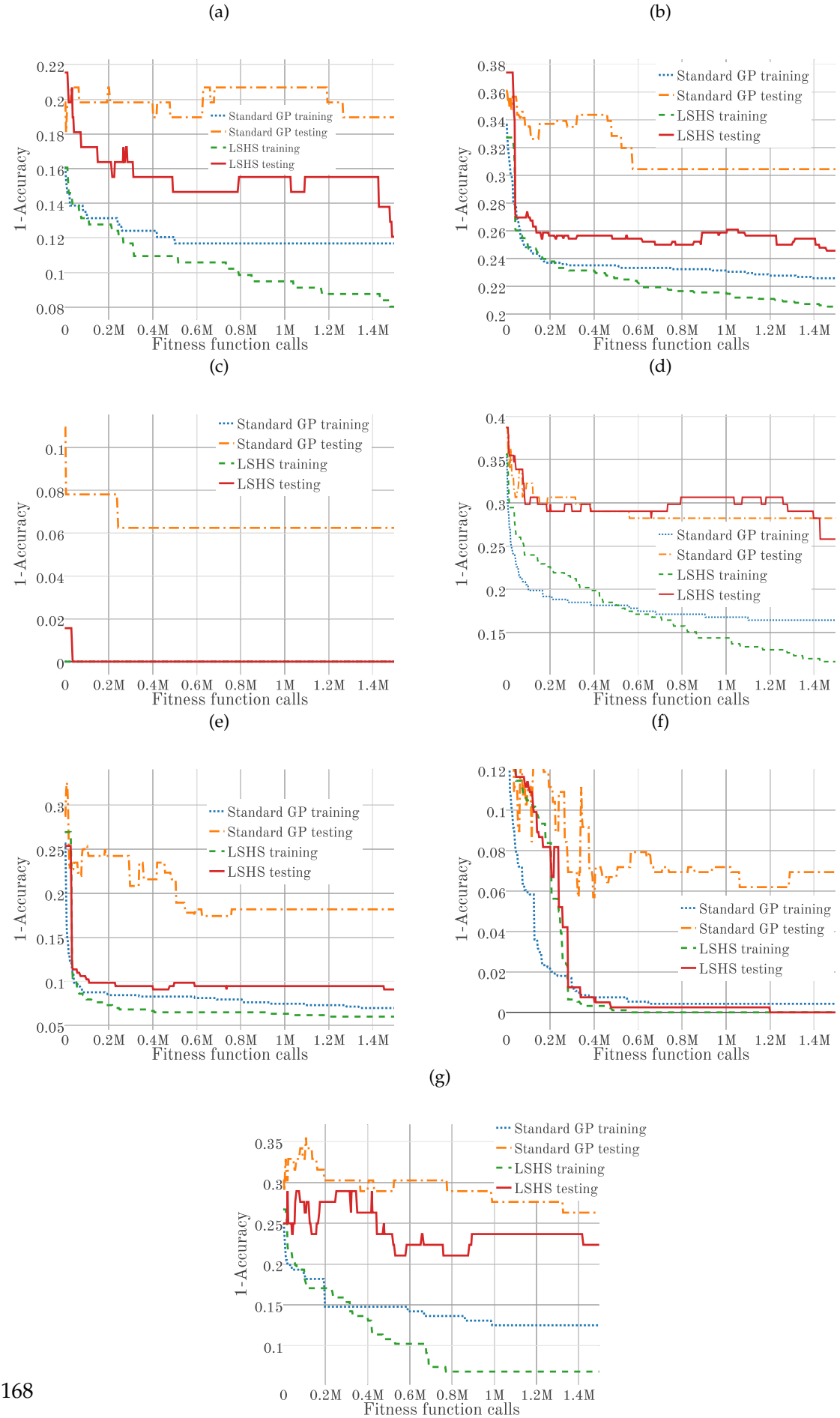
Table 7.3 – Wilcoxon rank sum test, with $\alpha = 0.05$. Bold values are less than α.

	Fitness	Size
Problem	p-value	
Parkinsons	**0.0027**	**0.0000**
Diabetes	**0.0000**	**0.0000**
Wine	0.0630	**0.0001**
Sonar	0.0859	0.1212
Wholesale	**0.0001**	**0.0017**
Banknote	**0.0000**	**0.0000**
LSVT	0.2978	0.7043

Table 7.4 – Classification accuracy comparison among several state-of-art methods, including from the GP community.

Problem	**Study**	**Accuracy** (%)	**Brief description of the study**
Parkinsons	Ozcift and Gulten OZCIFT et GULTEN [2011]	87.7 (5-fold CV)	Dirichlet process mixtures
	Ozcift and Gulten OZCIFT et GULTEN [2011]	87.1 (10-fold CV)	CFS-RF
	Ma et al. MA et collab. [2014]	**99.49** (10-fold CV)	SCFW-KELM with feature preprocessing
	Dufourq and Pillay DUFOURQ et PILLAY [2013]	86.7 (testing)	GP based (arithmetic, logical and decision trees)
	Koshiyama et al. KOSHIYAMA et collab. [2013]	89.33 (testing)	GP based (GPF-CLASS)
	Wang et al. WANG et collab. [2014]	86.96	GP based (MOGP maximizing ROCCH)
	This work	88.9 (testing)	GP + LS (LSHS)
Diabetes	Eggermont et al. EGGERMONT et collab. [2004]	74.1 (10-fold CV)	GP based (decision trees)
	This work	**75.5** (testing)	GP + LS (LSHS)
Wine	Tan and Dowe TAN et DOWE [2004]	93.2 (90/10 training/testing)	Minimum message length Decision Trees
	This work	**100** (testing)	GP + LS (LSHS)
Sonar	Tan and Dowe TAN et DOWE [2004]	76 (90/10 training/testing)	Minimum message length Decision Trees
	Koshiyama et al. KOSHIYAMA et collab. [2013]	**77** (testing)	GP based (GPF-CLASS)
	This work	74.2 (testing)	GP + LS (LSHS)
Wholesale	Jayadeva JAYADEVA [2015]	**92.72** (5-fold CV)	Minimal Complex Machine
	This work	91 (testing)	GP + LS (LSHS)
Banknote	Ghazvini et al. GHAZVINI et collab. [2014]	95.99 (10-fold CV)	MLP
	This work	**100** (testing)	GP + LS (LSHS)
LSVT	Tsanas et al. TSANAS et collab. [2014]	**90**	SVM with feature subset
	This work	78 (testing)	GP + LS (LSHS)

Figure 7.6 – Results of fitness performance of problems Parkinsons (a), Diabetes (b), Wine (c), Sonar (d), Wholesale (e), Banknote (f) and LSVT (g). Fitness performance. Plots show the median over 20 independent runs.



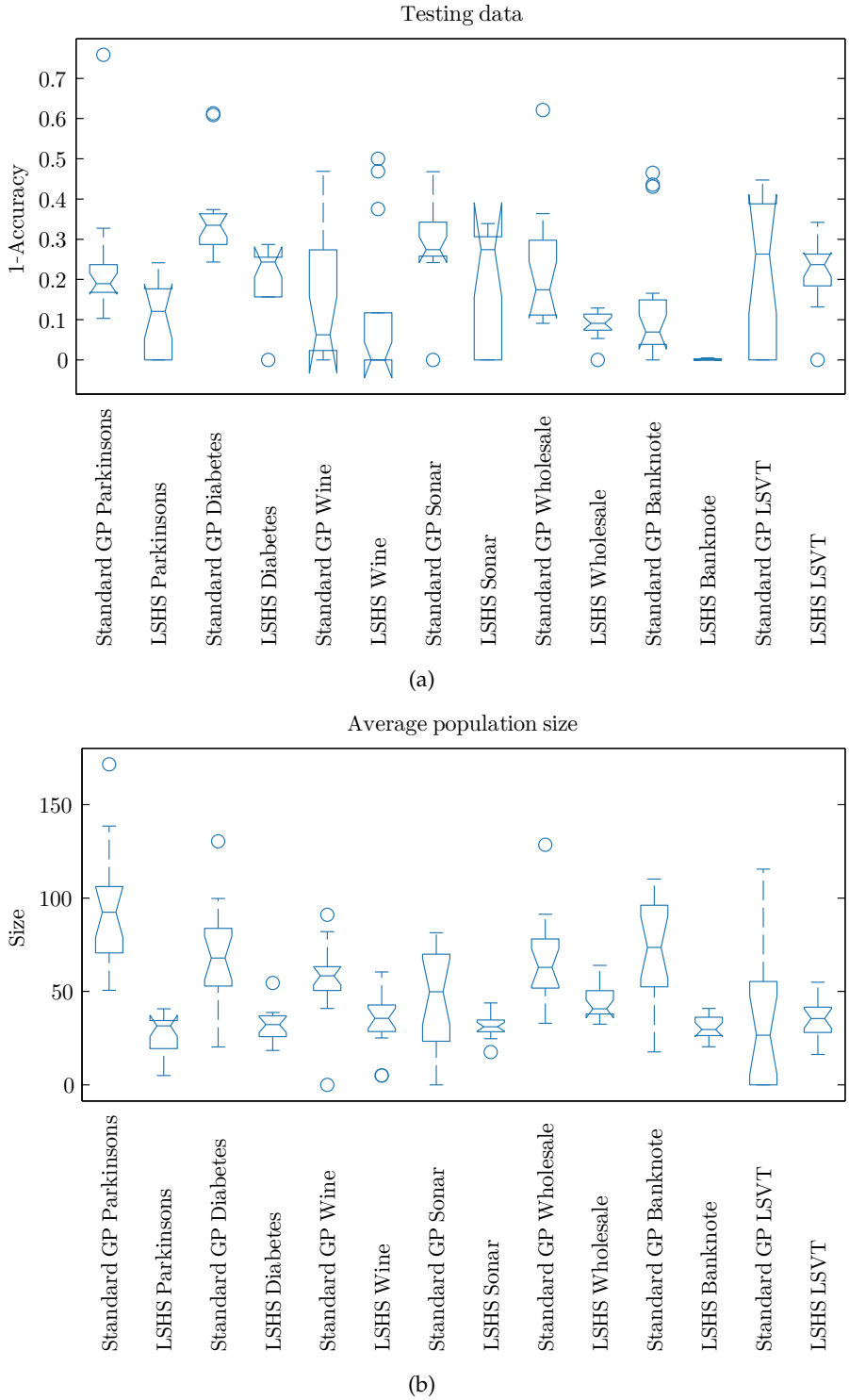


Figure 7.7 – Notched box plots from 20 runs at the end of each run. (a) present the fitness test data and (b) the average population size.

7.5 Conclusion

In this study, a methodology to incorporate a LS method into GP was proposed, that tackles the challenges imposed by a supervised classification task. A tree parameterization is introduced where a candidate solution in GP is extended by a simple mechanism where a weight is added to each node. Using a well known and robust non-linear optimizer the solution is improved targeting a better classification rate. A heuristic is used to determine which solutions are improved by the LS method, preferring smaller than average trees within the population. The presented results illustrate the benefits of using a LS method during the GP search. Indeed, in most problems the improvements in terms of classification error was significant, and a survey of recently published results confirm the quality of the evolved solutions. Moreover, the evolution of program growth also provides interesting insights. In particular, GP+LS with LSHS was able to produce high quality solutions, in most cases significantly better than standard GP, while maintaining smaller individuals within the population in terms of average size. This suggests that code growth is not necessary to obtain improved performance if the search space is properly exploited, which the LS method allows us to do. Moreover, it is important to notice that the proposed LSHS-GP system should not increase the complexity, or more precisely reduce the interpretability of the evolved solutions relative to standard GP. In fact, it might help produce simpler trees (results at least show that they are smaller), since the LS process allows simple numerical tuning of the evolved programs to improve fitness; something that otherwise might require complex syntactical changes. This contrast nicely with other recent approaches based on geometric semantic operators [MORAGLIO et collab., 2012], that improve GP performance at the cost of lowering interpretability. Future work will consider algorithmic improvements and rigorous comparisons with other methods. For instance, evaluating other transfer functions, utilizing gradient-free LS methods such as evolutionary strategies, and extending the method to multi-class problems, as well as comparing our proposal with a wider variety of classification methods.

Acknowledgments

Funding for this work was provided by CONACYT Basic Science Research Project No. 178323, DGEST (Mexico) Research Project 5414.14-P, and FP7-PEOPLE-2013-IRSES project ACOBSEC financed by the European Commission with contract No. 612689.

References

AZAD, R. et C. RYAN. 2014, «A Simple Approach to Lifetime Learning in Genetic Programming-Based Symbolic Regression», *Evolutionary computation*, vol. 22, nº 2, doi: 10.1162/EVCO, p. 287–317. URL http://www.mitpressjournals.org/doi/abs/10.1162/EVCO_a_00111. 155, 162

BACHE, K. et M. LICHMAN. 2013, «UCI machine learning repository», URL http://archive.ics.uci.edu/ml. 163, 164

BHOWAN, U., M. JOHNSTON et M. ZHANG. 2012, «Developing new fitness functions in genetic programming for classification with unbalanced data.», *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 42, nº 2, doi:10.1109/TSMCB.2011.2167144, p. 406–21, ISSN 1941-0492. URL http://www.ncbi.nlm.nih.gov/pubmed/21954215. 161

CHEN, X., Y.-S. ONG, M.-H. LIM et K. C. TAN. 2011, «A multi-facet survey on memetic computation», *Trans. Evol. Comp*, vol. 15, nº 5, p. 591–607. 155

COLEMAN, T. F. et Y. LI. 1992, «On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds», . 158

COLEMAN, T. F. et Y. LI. 1993, «An interior trust region approach for nonlinear minimization subject to bounds», cahier de recherche, Ithaca, NY, USA. 158

DUFOURQ, E. et N. PILLAY. 2013, «A Comparison of Genetic Programming Representations for Binary Data Classification», dans *Third World Congress on Information and Communication Technologies*, ISBN 9781479932306, p. 134–140. URL http://www.titan.cs.unp.ac.za/~nelishiap/uploads/3.pdf. 167

EGGERMONT, J., A. EIBEN et J. VAN HEMERT. 1999, «Adapting the fitness function in GP for data mining», *Genetic Programming*, , nº 2, p. 193–202. URL http://link.springer.com/chapter/10.1007/3-540-48885-5_16. 158, 161

EGGERMONT, J., J. N. KOK et W. A. KOSTERS. 2004, «Genetic Programming for Data Classification: Partitioning the Search Space», *SAC '04*, doi:http://doi.acm.org/10.1145/967900.968104, p. 1001–1005. URL http://portal.acm.org/citation.cfm?id=968104#. 167

EMMERICH, M., M. GRÖTZNER et M. SCHÜTZ. 2001, «Design of graph-based evolutionary algorithms: A case study for chemical process networks», *Evol. Comput.*, vol. 9, nº 3, p. 329–354. 154

GHAZVINI, A., J. AWWALU et A. A. BAKAR. 2014, «Comparative Analysis of Algorithms in Supervised Classification : A Case study of Bank Notes Dataset», *Computer Trends and Technology*, vol. 17, nº 1, p. 39–43. 167

GRAFF, M. et R. PE. 2013, «Wind Speed Forecasting using Genetic Programming», *Evolutionary Computation*, p. 408–415. 155, 156

JAYADEVA. 2015, «Learning a hyperplane classifier by minimizing an exact bound on the VC dimension», *Neurocomputing*, vol. 149, doi:10.1016/j.neucom.2014.07.062, p. 683–689. 167

KEIJZER, M. 2003, «Improving symbolic regression with interval arithmetic and linear scaling», EuroGP'03, Springer-Verlag, Berlin, Heidelberg, p. 70–82. 157

KOMMENDA, M., G. KRONBERGER, S. WINKLER, M. AFFENZELLER et S. WAGNER. 2013, «Effects of constant optimization by nonlinear least squares minimization in symbolic regression», *GECCO '13 Companion*, doi:10.1145/2464576.2482691, p. 1121. URL http://dl.acm.org/citation.cfm?doid=2464576.2482691. 156

KOSHIYAMA, A., T. ESCOVEDO, D. DIAS, M. VELLASCO et R. TANSCHEIT. 2013, «GPFCLASS: A Genetic Fuzzy model for classification», *2013 IEEE Congress on Evolutionary Computation*, doi:10.1109/CEC.2013.6557971, p. 3275–3282. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6557971. 167

KOZA, J. R. 1992, *Genetic programming: on the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, USA, ISBN 0-262-11170-5. 154

LAWSON, C. L. et R. J. HANSON. 1995, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics. 157

LITTLE, M., P. MCSHARRY, S. ROBERTS, D. COSTELLO et I. MOROZ. 2007, «Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection», *BioMedical Engineering OnLine*, vol. 6, nº 1, doi:10.1186/1475-925X-6-23, 23. URL http://dx.doi.org/10.1186/1475-925X-6-23. xv, 160, 164

LOHMANN, R. 1991, «Proceedings of parallel problem solving from nature (ppsn i) first workshop», dans *EuroGP 2013*, *LNCS*, vol. 496, Springer-Verlag, p. 198–208. 154

MA, C., J. OUYANG, H.-L. CHEN et X.-H. ZHAO. 2014, «An Efficient Diagnosis System for Parkinson's Disease using Kernel-based Extre me Learning Machine with Subtractive Clustering Features Weighting Approach», vol. 2014. 166, 167

MCCONAGHY, T. 2011, «FFX: Fast, Scalable, Deterministic Symbolic Regression Technology», dans *Genetic Programming Theory and Practice IX*, chap. 13, p. 235–260, doi:10.1007/978-1-4614-1770-5_13. 156

MORAGLIO, A., K. KRAWIEC et C. G. JOHNSON. 2012, «Geometric semantic genetic programming», dans *Proceedings of the 12th international conference on Parallel Problem Solving from Nature - Volume Part I*, PPSN'12, Springer-Verlag, Berlin, Heidelberg, p. 21–31. 170

MORÉ, J. J. et D. C. SORENSEN. 1983, «Computing a trust region step», *SIAM J. Scientific and Statistical Computing*, vol. 4, p. 553–572. 158

OZCIFT, A. et A. GULTEN. 2011, «Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms», *Computer Methods and Programs in Biomedicine*, vol. 104, nº 3, doi:10.1016/j.cmpb.2011.03.018, p. 443–451, ISSN 01692607. URL http://dx.doi.org/10.1016/j.cmpb.2011.03.018. 167

SILVA, S. et J. ALMEIDA. 2003, «Gplab–a genetic programming toolbox for matlab», dans *Proceedings of the Nordic MATLAB conference*, édité par L. Gregersen, p. 273–278. 163

SILVA, S. et E. COSTA. 2009, «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories», *Genetic Programming and Evolvable Machines*, vol. 10, doi:10.1007/s10710-008-9075-9, p. 141–179, ISSN 13892576. 165

SORENSEN, D. 1982, *Newton's Method with a Model Trust Region Modification*, Defense Technical Information Center. 156, 158

TAN, P. J. et D. L. DOWE. 2004, «MML Inference of Oblique Decision Trees», *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, p. 1082–1088, ISSN 03029743. 167

TOPCHY, A. et W. F. PUNCH. 2001, «Faster Genetic Programming based on Local Gradient Search of Numeric Leaf Values», *GECCO'01*, , nº 1997, p. 155–162. URL http://www.cs.bham.ac.uk/~wbl/biblio/gecco2001/d01.pdf. 155

TRUJILLO, L., L. MUNOZ, E. NAREDO et Y. MARTINEZ. 2014, «Neat, there's no bloat», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 8599, Springer Berlin Heidelberg, ISBN 978-3-662-44302-6, p. 174–185, doi:10.1007/978-3-662-44303-3_15. URL http://dx.doi.org/10.1007/978-3-662-44303-3_15. 162

TSANAS, A., M. LITTLE, C. FOX et L. RAMIG. 2014, «Objective automatic assessment of rehabilitative speech treatment in parkinson's disease», *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, nº 1, doi:10.1109/TNSRE.2013.2293575, p. 181–190, ISSN 1534-4320. 164, 166, 167

WANG, P., K. TANG, T. WEISE, E. TSANG et X. YAO. 2014, «Multiobjective genetic programming for maximizing ROC performance», *Neurocomputing*, vol. 125, doi:10.1016/j.neucom.2012.06.054, p. 102–118, ISSN 09252312. URL http://www.sciencedirect.com/science/article/pii/S0925231213001938. 155, 167

WINKLER, S., M. AFFENZELLER et S. WAGNER. 2007, «Advanced Genetic Programming Based Machine Learning», *Journal of Mathematical Modelling and Algorithms*, vol. 6, nº 3, doi:10.1007/s10852-007-9065-6, p. 455–480, ISSN 1570-1166. URL http://link.springer.com/10.1007/s10852-007-9065-6. 158, 161

WORM, T. et K. CHIU. 2013, «Prioritized grammar enumeration: Symbolic regression by dynamic programming», *GECCO' 13*, p. 1021–1028. URL http://dl.acm.org/citation.cfm?id=2463486. 156

YUAN, J. Y. 1996, «Numerical methods for generalized least squares problems», *Journal of Computational and Applied Mathematics*, vol. 66, nº 12, p. 571 – 584. 157

ZHANG, M. et W. SMART. 2004, «Genetic programming with gradient descent search for multiclass object classification», *Genetic Programming*. URL http://link.springer.com/chapter/10.1007/978-3-540-24650-3_38. 155, 156

Chapter 8

RANSAC-GP: Dealing with Outliers in Symbolic Regression with Genetic Programming

This chapter is related to the PhD thesis of Uriel Lopez Islas (ITT Tijuana) and has been presented at EUROGP 2017 and published in the LNCS Volume 10196, Springer, 2017. Work carried out with Uriel Lopez, Leonardo Trujillo, Yuliana Martinez, Enrique Naredo and Sara Silva.

Contents

Abstract

Genetic programming (GP) has been shown to be a powerful tool for automatic modeling and program induction. It is often used to solve difficult symbolic regression tasks, with many examples in real-world domains. However, the robustness of GP-based approaches has not been substantially studied. In particular, the present work deals with the issue of outliers, data in the training set that represent severe errors in the measuring process. In general, a datum is considered an outlier when it sharply deviates from the true behavior of the system of interest. GP practitioners know that such data points usually bias the search and produce inaccurate models. Therefore, this work presents a hybrid methodology based on the RAndom SAmpling Consensus (RANSAC) algorithm and GP, which we call RANSAC-GP. RANSAC is an approach to deal with outliers in parameter estimation problems, widely used in computer vision and related fields. On the other hand, this work presents the first application of RANSAC to symbolic regression with GP, with impressive results. The proposed algorithm is able to deal with extreme amounts of contamination in the training set, evolving highly accurate models even when the amount of outliers reaches 90%.

8.1 Introduction

One of the most common application domains of genetic programming (GP) is to solve regression problems (or real-valued learning problems), with an approach referred to as symbolic regression. Unlike other regression approaches, the search/learning process is not focused on determining the best fit parameters for a pre-specified model. The problem is stated more generally, such that GP searches for both the structure (symbolic expression) and the optimal parametrization of the model that best describes a set of learning or training data. Indeed, GP has produced a variety of successful results in this domain.

However, one problem that has not received an adequate amount of attention is the impact that outliers (gross errors) in the training set can have on the quality of the solution found. Furthermore, almost no research work has been devoted to developing GP-based symbolic regression that is robust to the presence of outliers in the training data. To the best of the authors knowledge, KOTANCHEK et collab. [2010] is the only work that deals with the problem of outlier detection in a GP-based system, but does not propose a general approach for robust symbolic regression in such scenarios.

While robust regression, as is reviewed next, has been the focus of large amounts of work in standard regression literature, symbolic regression has not followed suit so far. In general, most symbolic regression research works under the assumption (even if not explicitly stated) that the input data is "clean" (without outliers), giving complete confidence on the error estimates of the evolved solutions with respect to the training data. This assumption is realistic, even in real-world scenarios. If a small amount of outliers are present in the data, then preprocessing or filtering approaches might be able to remove outliers from the training set before running the symbolic regression system. However, this assumption will fail when the contamination is severe, above what is commonly referred to as the breakdown point for a robust regression method, when 50% or more of the data is in fact outliers. Other issues that might limit the usefulness of pre-processing methods is when the data sampling is sparse and non-uniform, making it difficult to apply filters that require accurate estimates of local signal statistics.

Therefore, this work presents an initial study to fill this research gap, with the following main contributions. First, we explore the effect that outlier contamination has on the performance of symbolic regression models evolved with GP. In particular, we evaluate standard GP using a typical error measure, as well as robust error estimates that are widely used in linear regression tasks when outliers are present in the data. Moreover, we evaluate a recent set of fitness case sampling methods, that do not use all of the training data instances at each fitness evaluation. In all cases, we conclusively show that these approaches fail when the amount of outliers is large, particularly above the breakdown point. Second, we propose a hybrid approach for robust symbolic regression modeling with GP, based on the RANdom SAmpling Consensus (RANSAC) algorithm FISCHLER et BOLLES [1981], TARSHA-KURDI et collab. [2007]. RANSAC has been shown to be a very robust approach for parameter estimation TORR et ZISSERMAN [2000], particularly popular in the computer vision community, used to determine the epi-polar geometry for stereo reconstruction LACEY et collab. [2000], ZULIANI [2009]. However, despite its success the application of RANSAC for symbolic regression with GP has not been studied before. Third, the results presented in this chapter are extremely encouraging, with our proposed hybrid RANSAC-GP algorithm identifying highly accurate models (with a testing error $\epsilon = 0.01$) even when data contamination greatly exceeds the breakdown point, with as much as 90% contamination of the training data with outliers.

The remainder of this chapter proceeds as follows. Section 8.2 presents a quick overview of related background. An introduction to robust linear regression is presented in Section 8.3, while also discussing fitness case sampling methods that are hypothesized to be useful in dealing with outlier contamination in the training set. Then, the RANSAC algorithm and the proposed RANSAC-GP hybrid are presented in Section 8.4. Section 8.5 presents our

experimental work and summarizes our main results. Finally, concluding remarks are given in Section 8.6.

8.2 Background

Let us begin by framing the basic regression task, where given a training dataset $\mathbb{T} = \{(\mathbf{x}_i, y_i); i = 1, ..., n\}$, the goal is to derive a model that predicts y_i based on $\mathbf{x}_i$, where $\mathbf{x}_i \in \Re^p$ and $y_i \in \Re$. In GP literature we can refer to each input/output pair $(\mathbf{x}_i, y_i)$ as a fitness case. For linear regression, the model is expressed as

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad i = 1, .., n \tag{8.1}$$

where the model parameters $\beta = (\beta_1, ..., \beta_p) \in \Re^p$, are estimated by $\widehat{\beta}_1, ..., \widehat{\beta}_p$ using the least squares method ROUSSEEUW [1984], which can be expressed as

$$(\widehat{\beta}_1, ..., \widehat{\beta}_p) \leftarrow \underset{\beta \in \Re^p}{arg\ min} \sum_{i=1}^{n} r_i^2 \ , \tag{8.2}$$

to find the best fit parameters of the linear model, where r_i denotes the residuals $r_i(\widehat{\beta}_1, ..., \widehat{\beta}_p) = y_i - (\widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_p x_{ip})$ and the errors ε_i have an expected value of zero ALFONS et collab. [2013]; if the summation in Equation 8.2 is divided by n, the error measure that must be minimized is the mean squared error (MSE).

Conversely, the symbolic regression problem solved with standard GP[1] can be expressed as

$$K^o \leftarrow \underset{K \in \mathbb{G}}{arg\ min}\ f(K(x_i), y_i)\ with\ i = 1, \ldots, n \ , \tag{8.3}$$

where the goal is to find the best model K^o that minimizes the error computed by the fitness function f between the expected output y_i and the estimate given by each model (program) $K(\mathbf{x}_i)$, with $\mathbb{G}$ representing the space of all possible models. In practice, f is usually expressed by an error measure such as the one of Equation 8.2 (MSE), or other error measures such as the mean absolute error (MAE) or the root mean squared error (RMSE).

8.2.1 Outliers

As stated above, GP has been shown to be very competitive in symbolic regression tasks, with many real-world examples and even commercial GP-based software tools. However, the effect that outliers have on GP performance has not been studied in depth. First, lets define outliers as follows:

Definition 1 *An outlier is a measurement of a system that is anomalous with respect to the behavior of the system.*

Unlike other definitions given in the literature PEARSON [2005], we do not focus on the dataset that results after a measuring session of a system of interest, and instead focus on the behavior of the actual system under observation. We do so because it is entirely possible that a given dataset might be severely corrupted with more than 50% of the data representing outliers. One may ask, if the outliers are a majority in a dataset, then are they truly outliers? That is why it is important to distinguish between a given measurement (observation) and the true value of a variable of interest. It is therefore reasonable for a dataset to be contaminated by a majority of outliers, which can be produced by several factors including measurement errors, equipment malfunction, human errors or missing data points. All

[1] All results presented in this work are based on a standard GP implementation, using a tree representation, subtree crossover and subtree mutation.

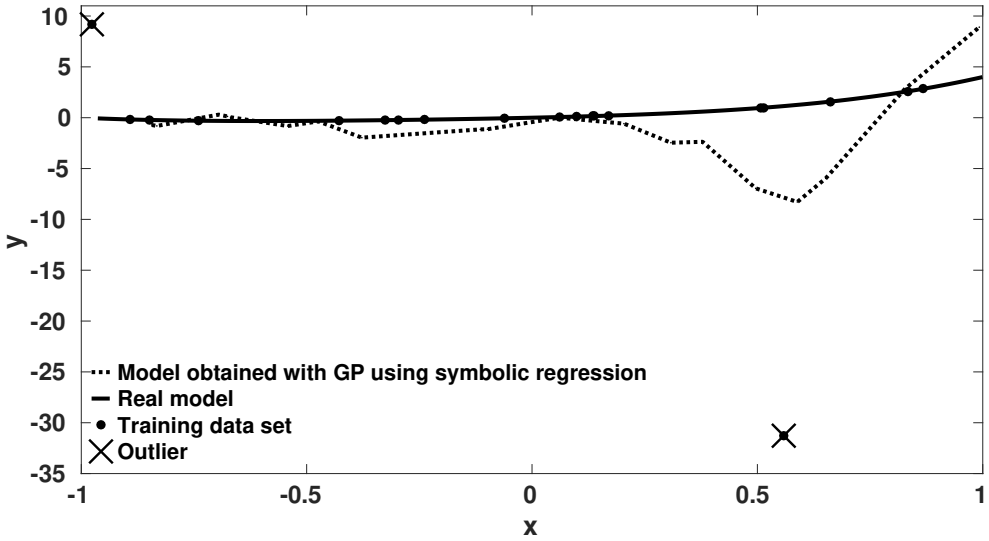


Figure 8.1 – Comparison of the model found by GP using symbolic regression (shown in dashed line) using a training set $\mathbb{T}$ (shown in dots) with two outliers (crosses), compared against the real model (shown in a solid line).

these are common problems found in real-world settings, that are often left unaccounted for in most GP-based symbolic regression research.

In particular, for simplicity we will focus on unidimensional problems, such that $\mathbf{x}_i \in \Re$. In such cases, we can say that a fitness case (x_i, y_i) is an outlier if

$$| y_i - y_o | > t\zeta \tag{8.4}$$

where y_i is the value to be characterized, y_o is a reference value, ζ is a measure of data variation, and t is a user defined threshold that controls to what extent the deviation of a particular fitness case can be regarded as "anomalous" and therefore be considered to be an outlier PEARSON [2005]. For instance, Equation 8.4 can be used to identify and remove outliers from a dataset using the Hampel identifier PEARSON [2005], where a moving window W centered on x_i is used to compute y_o and ζ. In particular, y_o is set to the median of all y_j in window W and ζ is given by $1.4826\times$MAD (Mean Absolute Deviation) within W. In the case of outlier removal, when y_i is determined to be an outlier it is replaced by y_o such that the new fitness case is now (x_i, y_o). Filters such as the Hampel identifier can be used to preprocess a dataset before applying linear or symbolic regression, however there are several drawbacks. First, if the percentage of outliers is above 50% then the newly inserted value y_o should not be used. Moreover, it is difficult to set the size of the moving window W when the training set does not provide a uniform sampling of the independent variable. Finally, a bigger issue lies in the fact that such methods are not easily extended to multidimensional spaces.

Let us provide a quick example of the effect that even a small number of outliers can have on a simple symbolic regression problem, as depicted in Figure 8.1. In this example, the training set is contaminated with 2 outliers, representing only 10% of the entire training set. Even with this small amount of outliers, we can see how the GP search is biased by the outliers and is unable to find the real underlying model of the data.

8.3 Robust regression

In robust linear regression GILONI et PADBERG [2002], NUNKESSER et MORELL [2010]; PEARSON [2005]; ROUSSEEUW [1984] the most common approach to deal with outliers in the training data is to substitute the objective function of the least squares problem. In particular,

measures such as the MSE or RMSE are very sensitive to outliers in the data. The sensitivity of a least squares method is measured by the breakdown point PEARSON [2005], which is reached when a certain percentage of outliers is present in the training set which produces an arbitrarily large bias in the final model. For instance, for standard linear least squares regression the breakdown point is 0%, a single outlier can induce arbitrarily large bias, which is also evidenced in GP symbolic regression as shown in Figure 8.1. In fact, the highest breakdown point for a robust least squares method is 50% ALFONS et collab. [2013], GILONI et PADBERG [2002], NUNKESSER et MORELL [2010]; PEARSON [2005]. Therefore, in this work we will experimentally test two popular robust estimators (with a 50% breakdown point) on symbolic regression with GP; these measures are described next, based on NUNKESSER et MORELL [2010].

8.3.0.0.1 Least Median Squares. The first robust measure is the Least Median Squares (LMS), given by

$$(\widehat{\beta}_1, ..., \widehat{\beta}_p) \leftarrow \underset{\beta_1,...,\beta_p}{arg\ min}\ med\ \{r_1^2, ..., r_n^2\} \tag{8.5}$$

where med represents the median, such that the summation average of the MSE method is substituted by the median of the residuals. One attractive feature of this method is that it is relatively simple and efficient to implement, only requiring a sorting of the residuals.

8.3.0.0.2 Least Trimmed Squares. The second robust approach is the Least Trimmed Squares (LTS) minimization problem, given by

$$(\widehat{\beta}_1, ..., \widehat{\beta}_p) \leftarrow \underset{\beta_1,...,\beta_p}{arg\ min} \sum_{i=1}^{hp} r_i^2 \tag{8.6}$$

where the squared residuals r_i^2 are ordered from smallest to largest. The method is called "trimmed" because it will search for the best combination (subsample), from among $\binom{n}{hp}$, of the complete training set with the smallest summation of least squares errors GILONI et PADBERG [2002], where hp is the trimmed proportion of the training set. This method achieves a breakdown point of 50% when $hp = \frac{n}{2} + 1$.

8.3.0.0.3 Fitness case sampling methods. These methods can be roughly defined as those that use only a portion of the total fitness cases when computing the fitness of an individual, which can be done in different ways, including using a subset of fitness cases at every other generation GONÇALVES et SILVA [2013], or by using a single fitness case when selecting individuals for reproduction LA CAVA et collab. [2016], SPECTOR [2012]. While these methods have been derived for different purposes, they have been extensively evaluated recently in MARTINEZ et collab. [2016]; MARTÍNEZ et collab. [2014], showing that they can improve performance relative to a standard GP. Moreover, based on the LTS measure, in this work we hypothesize that they might be useful in dealing with outliers, since they rely on a similar general approach, to bias the search based on a subsample of the training data. In particular, the fitness case sampling methods tested in this work are[2]:

1. Interleaved Sampling (IS): Use all the fitness cases in every odd numbered generations, and use a randomly chosen fitness case otherwise GONÇALVES et SILVA [2013].

2. Randon Interleaved Sampling (RIS): Uses all or one random fitness case based on a random decision at each generation GONÇALVES et SILVA [2013].

[2] An extensive discussion of these methods is beyond the scope of this work.

3. Keep Worst-Interleaved Sampling (KW-IS):Use all the fitness cases in odd numbered generations and a subset of the most difficult ones on the rest MARTÍNEZ et collab. [2014].

4. Keep Best-Interleaved Sampling (KB-IS): Simliar to KW-IS, but instead focusing on the easiest fitness cases every other generation.

5. Lexicase Selection (Lex): Randomly order the fitness cases for each parent selection event, and then sequentially discard individuals based on each fitness case until a single individual is left SPECTOR [2012].

6. ϵ-Lexicase (ϵ-Lex): Similar to Lex, but uses a threshold to compare individuals for real-valued symbolic regression LA CAVA et collab. [2016].

8.4 Proposed RANSAC-GP

8.4.0.0.1 Random Sample Consensus. RANSAC is a random sampling algorithm that is used to solve problems where data contamination is expected to exceed the 50% breakdown point of standard robust regression methods. Originally proposed in FISCHLER et BOLLES [1981] by Fischler and Bolles, it has become a standard technique in modern computer vision systems ZULIANI [2009], widely used to solve complex regression problems. However, RANSAC was originally intended, and is currently used, for parameter estimation. It is of note that, to the authors knowledge, RANSAC is not widely used in areas outside computer vision, and in particular it has not been applied on symbolic regression tasks.

While achieving strong results in this difficult problem formulation (regression fitting with more than 50% of outliers), it is in fact a very simple and intuitive algorithm, with four user defined parameters. RANSAC assumes that while the training set can be heavily contaminated by outliers, it nonetheless contains sufficient inlier's so as to reconstruct the underlying model of the "true" data. To do so, it iteratively and randomly samples the training set, and uses each sample to build a model. The size of this sample set m is the first RANSAC parameter, and the set is called the Minimal Sample Set ($MSS \subset \mathbb{T}$). It then evaluates the model with all remaining data in the training set ($\mathbb{T} \setminus MSS$), and computes what is referred to as the consensus set (CS); i.e. the set of all data points in $\mathbb{T} \setminus MSS$ that agree or are consistent with the particular model generated with the MSS. This is done by considering the residuals r_j, and marking a data point as an inlier if the corresponding residual falls below a threshold t, the second RANSAC parameter. This is done until the size of set CS reaches the estimated total of inliers, the third user parameter v, or when a maximum number of iterations l is reached, the fourth parameter. The entire pseudo-code of RANSAC DERPANIS [2010], ZULIANI [2009] is given in Algorithm 5. We must stress that in this work we are considering the original RANSAC formulation. More recent and in some sense improved versions will be studied in future work, such as the M-estimator Sample Consensus (MSAC) or the Maximum Likelihood Estimation SAmple and Consensus (MLESAC) TORR et ZISSERMAN [2000], or more recent methods such as the Optimal RANSAC algorithm HAST et collab. [2013].

8.4.1 Proposal

The proposal in this work is very straightforward, to build non-linear symbolic regression models with GP within RANSAC. Therefore, we only need to modify step 2 in Algorithm 5, where the model K is derived using a standard GP search. Indeed, one of the most attractive aspects of RANSAC is the ease with which it can be adapted to other modeling approaches.

Algorithm 5 RANSAC pseudo-code.

1. Take a random MSS_j of size m from the training set $\mathbb{T}$
2. Build a model K_j using the data in MSS_j.
3. Compute the residuals r_j for all the data points in $\mathbb{T} \setminus MSS_j$.
4. Build the consensus set CS_j with all the data points in $\mathbb{T} \setminus MSS_j$ for which $r_j < t$
5. If $|CS_j| \geq v$ then return K_j as the final model.
6. Repeat |1 trough 4| until a maximum number of iterations, otherwise return K_j with maximum $|CS_j|$.

8.5 Experiments and Results

The goal of the experimental work is twofold. First, we want to experimentally evaluate standard GP using the MSE as a fitness function, as well as the reviewed robust estimators (LMS and LTS) and all the the fitness case sampling methods (IS, RIS, KW-IS, KB-IS, Lex and ϵ-Lex), on benchmark problems contaminated with different amounts of outliers. We consider a wide range of contaminations, from 10% to 90% (in 10% increments), using relatively simple benchmark problems, to fully illustrate how even a small amount of outliers in the training set can bias the resulting model[3]. The common parameters of the GP system for all these experiments are summarized in Table 8.1.

Table 8.1 – GP parameters used for the benchmark symbolic regression problems.

Parameter	Description
Population size	100 Individuals
Generations	200 Generations
Initialization	*Ramped Half-and-Half*, with maximum depth level 6
Operator probabilities	Crossover $p_c = 0.9$, Mutation $p_\mu = 0.1$
Function set	$(+, -, \times, \div, sin, cos)$
Terminal set	$x, randint(-1, 1)$
Maximum tree depth	17 levels
Selection	Tournament size 3 (Except in Lex and ϵ-Lex)
Elitism	Best individual always survives

The second goal of our work is to show how the proposed RANSAC-GP method can easily handle large amounts of outliers in the training set. In particular, we will present a detailed analysis of the models found for the most difficult scenarios, when the contamination is at 90%. All of our experiments and algorithms were coded using the Distributed Evolutionary Algorithms in Python library (DEAP) FORTIN et collab. [2012], a Python library for evolutionary computation. However, we begin this section by explaining how the

[3]For all practitioners, it is intuitively evident that 10% of outliers can have drastic effects in the modeling process.

Table 8.2 – Benchmark problems used in this work, where U[a,b,c] denotes c uniform random samples drawn from a to b, that specifies how the training and testing sets are constructed, consisting solely of inliers.

Objective function	Test Set	Function Set
$x^4 + x^3 + x^2 + x$	U[-1, 1, 20]	U[-1, 1, 20]
$x^5 - 2x^3 + x$	U[-1, 1, 20]	U[-1, 1, 20]
$x^3 + x^2 + x$	U[-1, 1, 20]	U[-1, 1, 20]
$x^5 + x^4 + x^3 + x^2 + x$	U[-1, 1, 20]	U[-1, 1, 20]
$x^6 + x^5 + x^4 + x^3 + x^2 + x$	U[-1, 1, 20]	U[-1, 1, 20]

training data is constructed and artificially contaminated by outliers.

8.5.0.0.1 Benchmark Problems. For the experimental work, five benchmark problems are used from McDermott et collab. [2012], described in Table 8.2. These problems are chosen for the following reasons. First, preliminary runs of our GP algorithm were able to consistently find optimal solutions with nearly perfect training and testing errors on all of them. This requirement was considered to be important for this work, to properly evaluate the effect that outlier contamination has on the performance of the symbolic regression task. If the modeling fails (large testing or generalization error), we want to be certain that this is due to the presence of outliers in the training set and not due to the underlying difficulty of the problem for the chosen GP algorithm. Second, since these are unidimensional problems, it is straightforward to properly contaminate the training set with outliers, as will be described next.

8.5.0.0.2 Training set contamination with outliers. The proposed approach to contaminate the data is to use the inverse of the Hample identifier defined in Equation 8.4. In particular, to turn a particular fitness case (x_i, y_i) into an outlier, we must first solve Equation 8.4 for y_i, such that

$$\begin{aligned} & y_i > y^o + t\zeta \\ \text{or} \quad & y_i < y^o - t\zeta. \end{aligned} \tag{8.7}$$

In particular, we randomly choose a percentage of fitness cases (for a different contamination percentage) and then randomly add or substract ζ from the ground truth y_i. The value of t is set randomly within the range $[10, 100]$ to guarantee a large amount of deviance from the original data, and ζ was computed by the median of all y_i within the domain of each symbolic regression benchmark.

8.5.1 Results

The results are presented in two parts. First, we consider the testing performance given by the median MSE over 30 runs of the standard GP using the fitness functions MSE, robust fitness functions (LMS and LTS), and all of the sampling methods. The results are presented as plots of the median testing error relative to the amount of outlier contamination on each benchmark. Second, the results of the proposed RANSAC-GP are presented, focusing on the most extreme cases of outlier contamination.

8.5.1.0.1 Robust fitnesss measures and sampling methods. The results are summarized in Figure 8.2, where MSE corresponds to standard GP with the MSE fitness function. The median performance (vertical axis) is plotted in a logarithmic scale, since the testing error of most methods reaches quite large values. In fact, most methods perform very poorly even

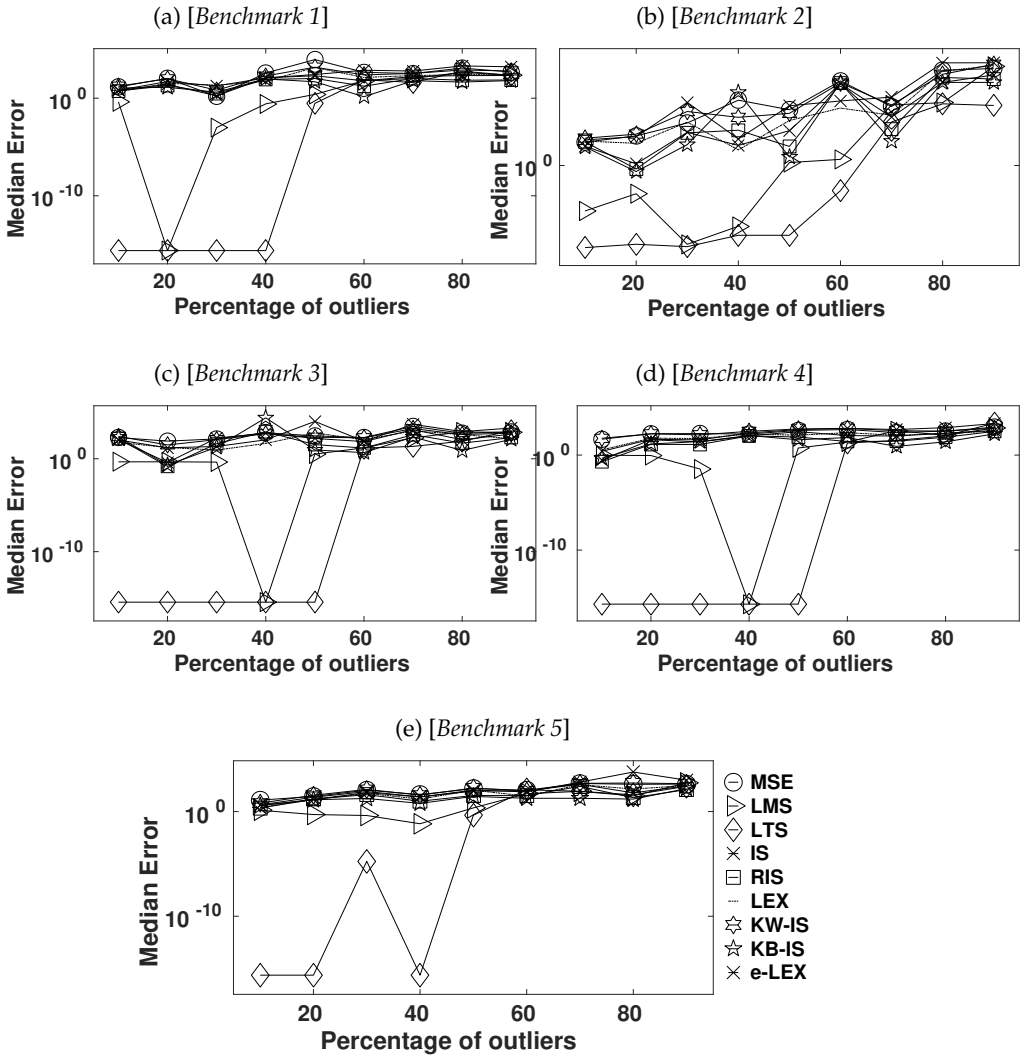


Figure 8.2 – Comparison of the median error of 30 runs for the five benchmarks at different levels of contamination for all methods.

for the smallest amount of outliers considered here (10%). It is clear that standard MSE and all the fitness case sampling methods are strongly biased by the presence of outliers. There are some good news though, when the contamination is equal or under 50% the robust fitness measures perform quite well. In particular, LTS shows the best performance of all the methods, with LMS also achieving strong results. This behavior, however, does not hold above the breakdown point of 50%, above which all methods perform poorly.

Finally, it is important to consider that the LTS measure is computationally much more complex than the one used by LMS, since it needs to cycle across all possible combinations of the training set. Therefore, as a first conclusion, it is reasonable to state that when faced with a real-world symbolic regression problem that might be contaminated by outliers, it is preferable to use LTS only when the size of the training set is relatively small, otherwise LMS should be preferred. This can improve the chances of achieving an accurate and general model, but only when the contamination is expected to be below 50%.

8.5.1.0.2 RANSAC-GP. Here we present the results for the proposed RANSAC-GP algorithm, with some important considerations discussed first. Regarding the size m of the MSS, it is assumed that the MSS should contain at least as much data as required by the modeling process to find an accurate model. For instance, for a linear model in a one dimensional problem two points would be sufficient. However, this is not defined for the studied bench-

Table 8.3 – RANSAC-GP parameters used for the symbolic regression problems.

Parameter	Description	Value
v	Size of the consensus set CS	Total inliers for each level of contamination
t	Threshold to consider a r_j of a fitness training case an inlier	$t = 0.01$
m	Minimal sample set	$m = 50\%$ of inliers

Table 8.4 – Iterations RANSAC-GP required to find a CS per level of contamination.

	% Outliers								
Problems	10%	20%	30%	40%	50%	60%	70%	80%	90%
Benchmark 1	1	1	1	2	1	2	138	579	795
Benchmark 2	1	1	2	2	6	72	494	456	892
Benchmark 3	2	1	4	1	1	13	75	107	690
Benchmark 4	1	2	2	3	8	16	563	530	907
Benchmark 5	2	2	2	1	4	93	237	249	7066

marks. Moreover, we increase the training set size 10-fold (to 200), and contaminate this new extended set using the procedure described in Section 8.5.0.0.2. Therefore in our work, m is equal to 50% of inliers at each level of contamination.

However, given the size of the MSS, for extreme contamination levels it will be extremely difficult for the random sampling to find a MSS consisting entirely of inliers. This is problematic, since we know that GP struggles with even a small amount of outliers (see the results for 10% contamination discussed above). Therefore, the modeling process performed within RANSAC-GP uses a standard GP search but the fitness function is substituted by the LMS measure. This simplifies the problem quite a bit, since we know that good performance is achieved by LMS when the contamination is below 50%. In this way the MSS does not need to contain only inliers, it is expected to perform accurately even when the MSS is contaminated by at most 50% of outliers. Moreover, the threshold for a fitness case to be included in the CS is set quite tightly to $t = 0.01$. In practice, such levels of accuracy might not be necessary but our intention is to test the algorithm in a difficult scenario. The configuration and parameters of RANSAC-GP are summarized in Table 8.3, and the rest of the parameters for the embedded GP search are the same as in Table 8.1 with the exception of the number of generations which is now set to 300 and the function set only includes arithmetic operators. Finally, for these experiments only a single execution of RANSAC-GP is reported for each level of data contamination, for the following reasons. First, we already know that GP is quite accurate and stable across multiple runs on widely used benchmarks, including when LMS is used as the fitness function (as shown above). Second, for higher levels of contamination the number of required iterations (with each iteration performing a single GP run on the MSS) can grow quite large, making experimental tests somewhat prohibitive. Nonetheless, even these runs provide a reasonable approximation of how the algorithm will perform on a real-world setting.

The stopping criterion is also based on the size of CS with v being equal or greater than the total inliers, assuming that the MSS may contain up to 50% of outliers, forcing the algorithm to find the most accurate model possible. Table 8.4 summarizes the number of iterations required by RANSAC-GP to find a model based on the parametrization given in Table 8.3. Notice that when the contamination is below 50%, RANSAC-GP finds an accurate model in eight or fewer iterations, a nice result that will surely encourage this approach in practice. As the number of outliers increases, it is evident that the number of required

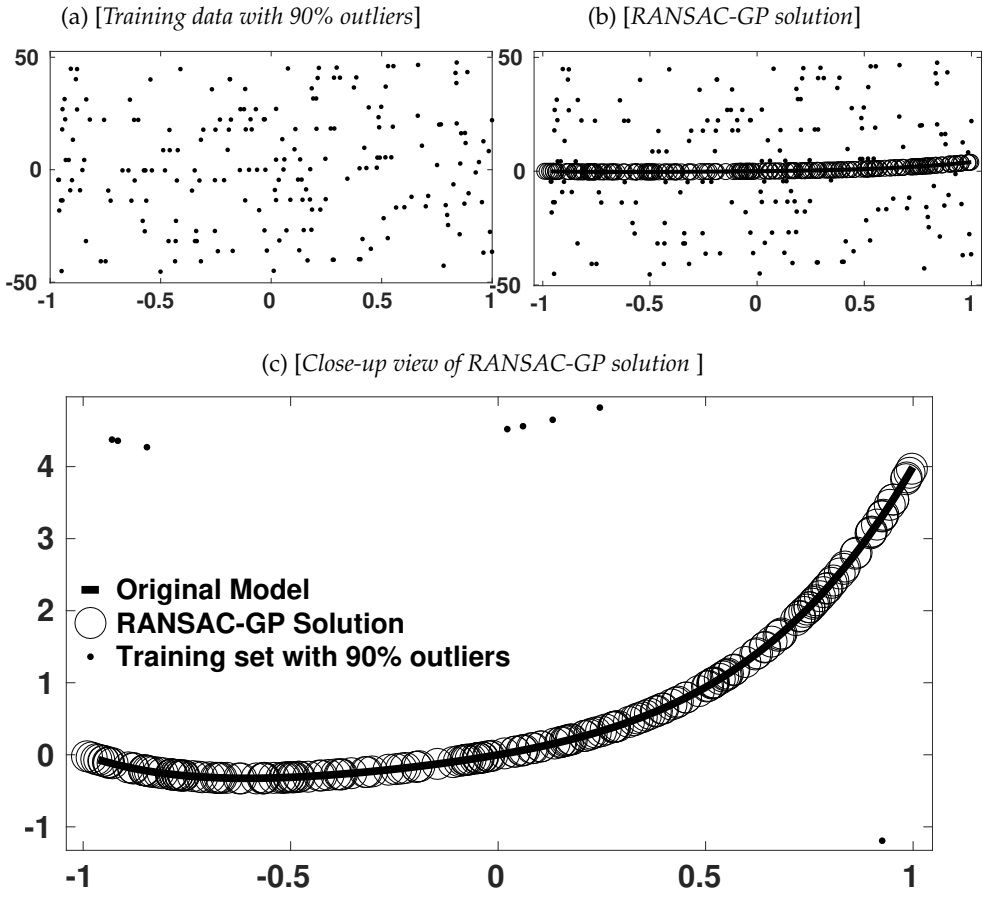
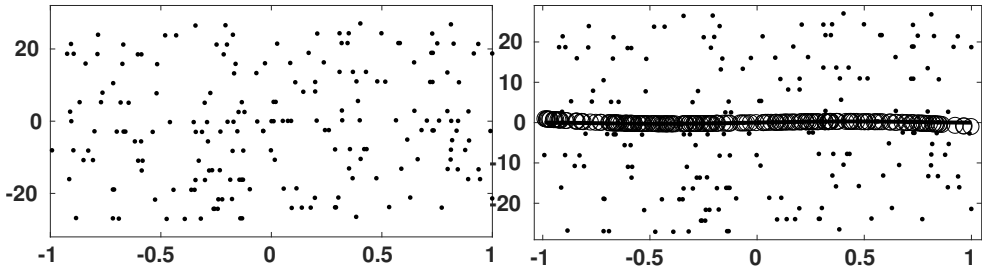


Figure 8.3 – Solution found by RANSAC-GP with 90% outliers for benchmark 1.

iterations (samples) also increases sharply, requiring several hundreds of iterations and in a single extreme case over $7,000$. For some this might seem like an excessive price to pay for such scenarios, but in practice there are several important aspects to consider. First, a parallel implementation of RANSAC-GP can be easily derived, since each MSS is taken independently, such that several iterations can be performed in parallel. Moreover, when considering a highly contaminated training set the threshold t for acceptance into the consensus set might reasonably be set much larger than in our experiments, allowing the algorithm to more quickly find a useful model.

Let us now turn to the modeling results on the most extreme case, where 90% of the training data are outliers, presented in Figures 8.3 to 8.7. Each figure shows three plots. First, the contaminated training set, which by visual inspection clearly shows a very difficult regression problem. Second, the training set with the original model highlighted, as well as the RANSAC-GP solution. Finally, a zoomed view of the RANSAC-GP solution and the original ground truth model. In almost all problems the quality of the found model is clear, with almost perfect accuracy. Performance is slightly worse on the second benchmark, and on the fifth to a lesser extent. Nonetheless, it is important to remember the extreme nature of the training data, where all GP practitioners and researchers will surely conclude that no other GP-based approach would have been able to detect the original model with the level of accuracy reported here.

(a) [*Training data with 90% outliers*] (b) [*RANSAC-GP solution*]



(c) [*Close-up view of RANSAC-GP solution*]

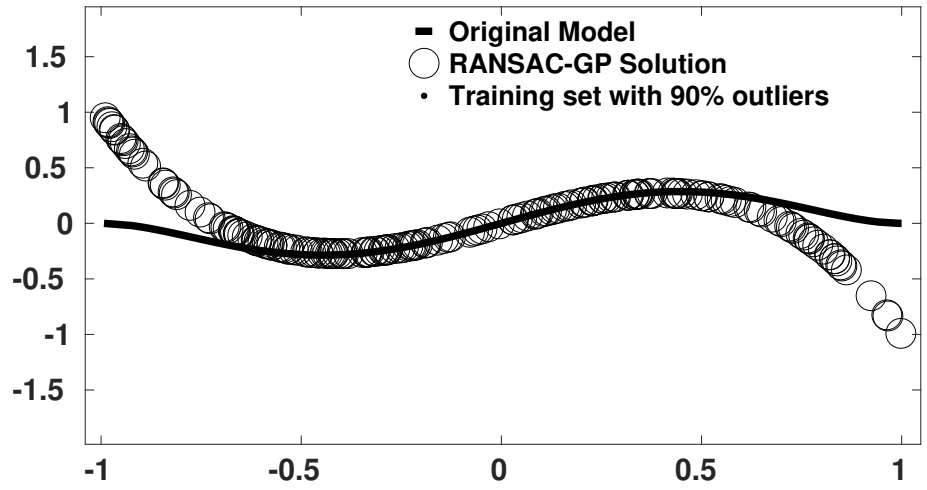
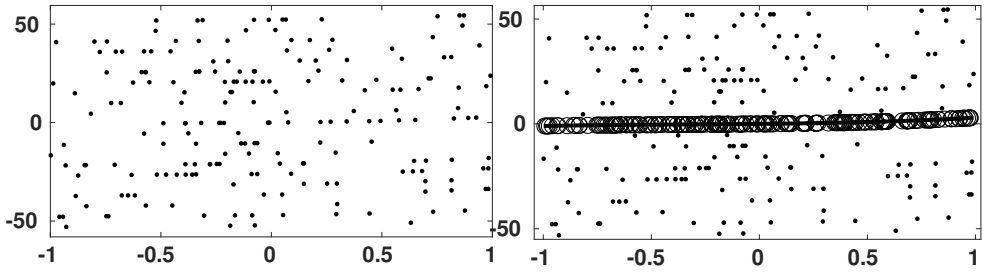


Figure 8.4 – Solution found by RANSAC-GP with 90% outliers for benchmark 2.

(a) [*Training data with 90% outliers*] (b) [*RANSAC-GP solution*]



(c) [*Close-up view of RANSAC-GP solution*]

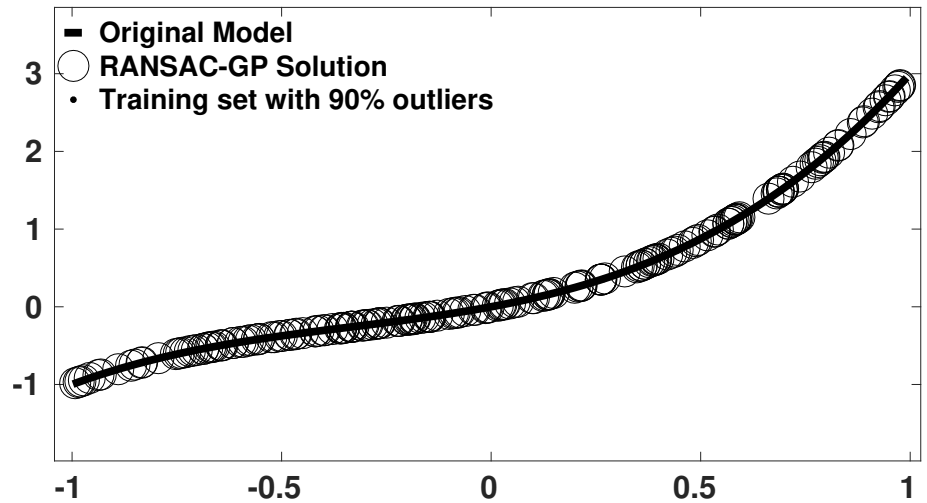
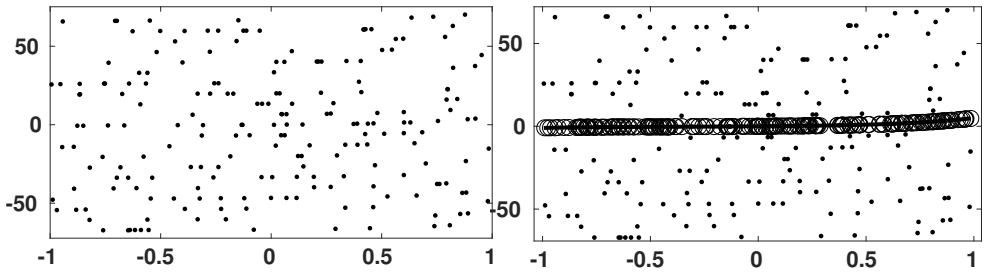


Figure 8.5 – Solution found by RANSAC-GP with 90% outliers for benchmark 3.

(a) [*Training data with 90% outliers*] (b) [*RANSAC-GP solution*]



(c) [*Close-up view of RANSAC-GP solution*]

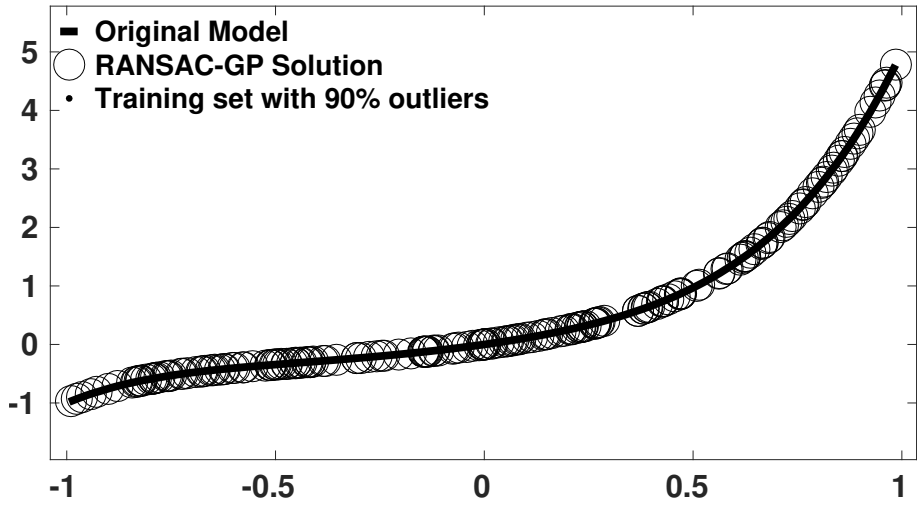
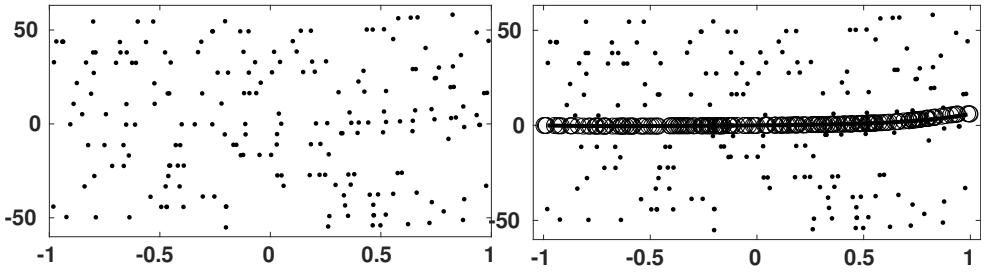


Figure 8.6 – Solution found by RANSAC-GP with 90% outliers for benchmark 4.

(a) [*Training data with 90% outliers*] (b) [*RANSAC-GP solution*]



(c) [*Close-up view of RANSAC-GP solution*]

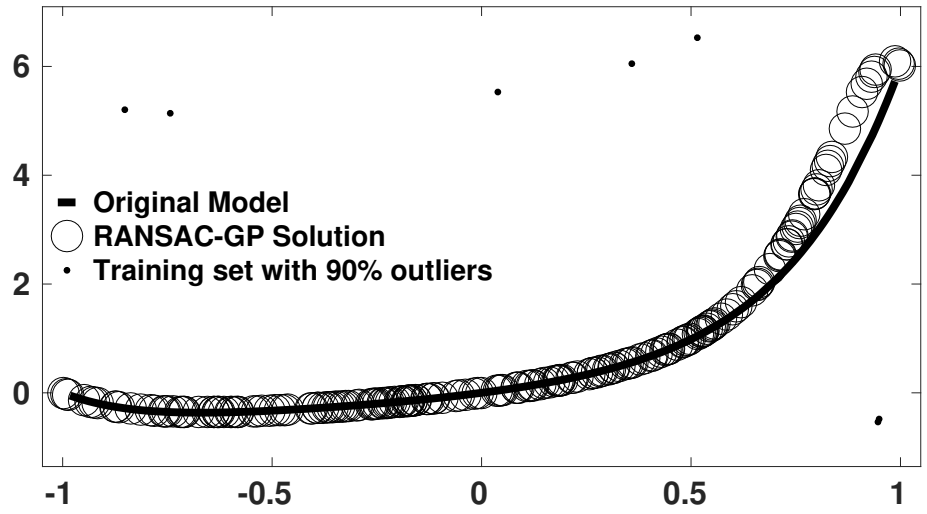


Figure 8.7 – Solution found by RANSAC-GP with 90% outliers for benchmark 5.

8.6 Conclusion and future work

This chapter presents RANSAC-GP, a hybrid approach for robust symbolic regression in the presence of severe contamination of the training set with outliers. Indeed, results are both impressive and encouraging, showing that the proposed method was able to find high quality solutions even when data contamination reached 90%, well beyond the breakdown point of robust regression techniques (such as LMS or LTS). The approach will be of great practical use in applied domains where the observations of a variable of interest are prone to severe errors. The results presented here are unique within the GP community, and are meant to encourage future work on automatic modeling under large amounts of data contamination.

Future work on this topic will be further explored, focusing on the following. First, it is straightforward to derive a parallel implementation, in order to reduce the computational burden of the large amounts of samples that need to be processed for highly contaminated training sets. Second, the threshold t used to build the consensus set should be made to be dynamic and problem dependent. Future research will also focus on a more detailed analysis of the RANSAC-GP process, to measure the effect and importance of each algorithm parameter. Third, the general scheme can, and will, be applied to other GP variants, including bloat-free GP, Geometric Semantic GP and GP systems that utilize different program representations. Fourth, it seems reasonable that the algorithm could be tuned based on the regularity of the training set, where a large amount of outliers will tend to produce highly irregular datasets and vice versa. Finally, the approach has to be tested on more complex problems, including multidimensional problems and real-world datasets.

Acknowledgments

This research was partially supported by CONACYT Basic Science Research Project No. 178323, CONACYT Fronteras de la Ciencia 2015-2 No. 944, as well as by FP7- Marie Curie-IRSES 2013 European Commission program with project ACoBSEC with contract No. 612689.

References

ALFONS, A., C. CROUX, S. GELPER et collab.. 2013, «Sparse least trimmed squares regression for analyzing high-dimensional large data sets», *The Annals of Applied Statistics*, vol. 7, nº 1, p. 226–248. 177, 179

DERPANIS, K. G. 2010, «Overview of the ransac algorithm», *Image Rochester NY*, vol. 4, nº 1, p. 2–3. 180

FISCHLER, M. A. et R. C. BOLLES. 1981, «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography», *Communications of the ACM*, vol. 24, nº 6, p. 381–395. 176, 180

FORTIN, F.-A., F.-M. DE RAINVILLE, M.-A. GARDNER, M. PARIZEAU et C. GAGNÉ. 2012, «DEAP: Evolutionary algorithms made easy», *Journal of Machine Learning Research*, vol. 13, p. 2171–2175. 181

GILONI, A. et M. PADBERG. 2002, «Least trimmed squares regression, least median squares regression, and mathematical programming», *Mathematical and Computer Modelling*, vol. 35, nº 9, p. 1043–1060. 178, 179

GONÇALVES, I. et S. SILVA. 2013, «Balancing learning and overfitting in genetic programming with interleaved sampling of training data», dans *Genetic Programming*, *LNCS*, vol. 7831, édité par K. Krawiec et collab., Springer Berlin Heidelberg, ISBN 978-3-642-37206-3, p. 73–84. 179

HAST, A., J. NYSJÖ et A. MARCHETTI. 2013, «Optimal ransac-towards a repeatable algorithm for finding the optimal set», *Journal of WSCG*, vol. 21, nº 1, p. 21–30. 180

KOTANCHEK, M. E., E. Y. VLADISLAVLEVA et G. F. SMITS. 2010, «Symbolic regression via genetic programming as a discovery engine: Insights on outliers and prototypes», dans *Genetic Programming Theory and Practice VII*, Springer, p. 55–72. 176

LA CAVA, W., L. SPECTOR et K. DANAI. 2016, «Epsilon-lexicase selection for regression», dans *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, ACM, New York, NY, USA, ISBN 978-1-4503-4206-3, p. 741–748, doi:10.1145/2908812.2908898. 179, 180

LACEY, A., N. PINITKARN et N. A. THACKER. 2000, «An evaluation of the performance of ransac algorithms for stereo camera calibrarion.», dans *BMVC*, p. 1–10. 176

MARTINEZ, Y., E. NAREDO, L. TRUJILLO, P. LEGRAND et U. LOPEZ. 2016, «A comparison of fitness-case sampling methods for genetic programming.», Accepted to appear in the Journal of Experimental & Theoretical Artificial Intelligence. 179

MARTÍNEZ, Y., L. TRUJILLO, E. NAREDO et P. LEGRAND. 2014, «A comparison of fitness-case sampling methods for symbolic regression with genetic programming», dans *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, *Advances in Intelligent Systems and Computing*, vol. 288, édité par A.-A. Tantar et collab., Springer International Publishing, p. 201–212. 179, 180

MCDERMOTT, J., D. R. WHITE, S. LUKE, L. MANZONI, M. CASTELLI, L. VANNESCHI, W. JASKOWSKI, K. KRAWIEC, R. HARPER, K. DE JONG et U.-M. O'REILLY. 2012, «Genetic programming needs better benchmarks», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, ACM, New York, NY, USA, p. 791–798. 182

NUNKESSER, R. et O. MORELL. 2010, «An evolutionary algorithm for robust regression», *Computational Statistics & Data Analysis*, vol. 54, nº 12, p. 3242–3248. 178, 179

PEARSON, R. K. 2005, *Mining imperfect data: Dealing with contamination and incomplete records*, Siam. 177, 178, 179

ROUSSEEUW, P. J. 1984, «Least median of squares regression», *Journal of the American statistical association*, vol. 79, nº 388, p. 871–880. 177, 178

SPECTOR, L. 2012, «Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report», dans *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, ACM, ISBN 978-1-4503-1178-6, p. 401–408. 179, 180

TARSHA-KURDI, F., T. LANDES, P. GRUSSENMEYER et collab.. 2007, «Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data», dans *Proceedings of the ISPRS Workshop on Laser Scanning*, vol. 36, p. 407–412. 176

TORR, P. H. et A. ZISSERMAN. 2000, «Mlesac: A new robust estimator with application to estimating image geometry», *Computer Vision and Image Understanding*, vol. 78, nº 1, p. 138–156. 176, 180

ZULIANI, M. 2009, «Ransac for dummies», *Vision Research Lab, University of California, Santa Barbara*. 176, 180

Part II

Estimation of signal regularity

Chapter 9

Holderian regularity

This chapter includes reminders on the analysis of Hölderian regularity as well as contributions to refine the estimation methods of the Hölder exponent. The tools and some of the developments presented in this chapter will be used in the chapters 12, 13, 15 and 16.

Contents

Introduction

The regularity of a signal and its estimation constitute a a relatively recent problem in the field of signal processing. Nevertheless this is one of the fundamental characteristics of a signal even if there are an infinite number of signals with the same regularity in each point. Regularity is an information which is not enough to fully describe a signal but which however, characterizes it very precisely.

Therefore, attention should be paid to the theoretical tools describing the regularity. This will be the subject of the first part of this chapter. The Hölderian analysis, through the Hölder exponent, offers mathematical objects characterizing the regularity of a signal. In particular, Hölder's exponent corresponds well to the visual VEHEL [1998] and auditory VEHEL et DAOUDI [1996] perceptions of "regularity".

In a second step, we will present three methods to estimate the Hölder exponent. The construction of these estimators will be described in detail.

9.1 Reminders on Hölderian regularity

We will start this chapter with some reminders about the Hölderian regularity. For more complete presentations on this On this subject, refer to JAFFARD et MEYER [1996], MEYER [1997],JAFFARD [2004], TRICOT [1995] or VEHEL et SEURET [2004].

9.1.1 Pointwise Hölder exponent

The Hölder pointwise exponent is the most common tool used to measure the regularity of a signal at a given point. Here are some definitions concerning this exponent.

Definition 2 *Let f be a function from $\mathbb{R}$ to $\mathbb{R}$, $s > 0$, $s \in \mathbb{R}\backslash\mathbb{N}$ and $x_0 \in \mathbb{R}$. Then $f \in C^s(x_0)$ if and only if there is a real $\eta > 0$, a polynomial P of degree smaller than s and a constant c such that*

$$\forall x \in B(x_0, \eta), \quad |f(x) - P(x - x_0)| \leq c|x - x_0|^s \tag{9.1}$$

By definition, the pointwise exponent of f at x_0, noted $\alpha_p(x_0)$ is the supremum of s such as $f \in C^s(x_0)$.

An equivalent definition can be given to the without directly displaying the C^s space.

Definition 3

$$\alpha_p(x_0) = liminf_{h \to 0} \frac{\log |f(x_0 + h) - f(x_0)|}{\log |h|} \tag{9.2}$$

This definition is valid if f is not derivable in x_0, otherwise one has to remove its regular part (JAFFARD [1997]).

Geometrically, the equation 9.2 means that the graph of the f function around x_0 is included in a envelope which will be called the Hölderian envelope (see figure 9.1). For every $\epsilon > 0$, there is a neighborhood of x_0 such that the f graph in this neighborhood is all included in the space defined by the two curves that associate x respectively $f(x_0)+c|x-x_0|^{\alpha_p(x_0)-\epsilon}$ and $f(x_0)-c|x-x_0|^{\alpha_p(x_0)-\epsilon}$ and such that this property is no longer true for the space defined by curves that associate to x respectively $f(x_0) + c|x - x_0|^{\alpha_p(x_0)+\epsilon}$ and $f(x_0) - c|x - x_0|^{\alpha_p(x_0)+\epsilon}$. We see that the higher the $\alpha_p(x_0)$, the more the signal is smooth and inversely that the more $\alpha_p(x_0)$ is small the more the signal is irregular in x_0. For example, f continuous in zero implies $\alpha_p \geq 0$ and f derivable implies $\alpha_p \geq 1$.
We dispose of a coefficient giving a measure of the irregularity, always defined and calculable. In addition, its definition extends without difficulty in the higher dimensions. However, the Hölder pointwise exponent is not stable by pseudo-differential operators.

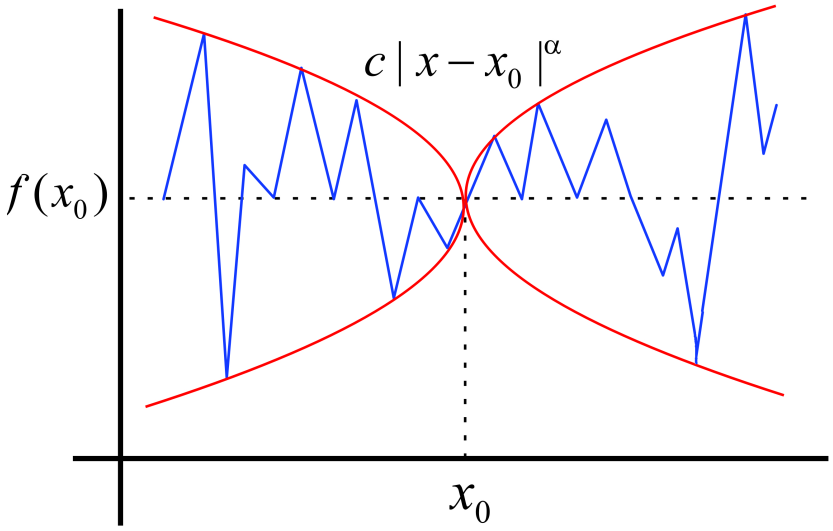


Figure 9.1 – Hölderian envelope of a signal at the point x_0.

Proposition 9.1.1
Let be α the pointwise Hölder exponent of the function f in x_0, then the pointwise Hölder exponent of f' in x_0 is less than or equal to $\alpha - 1$.

Equality is only verified in some cases, for instance:

- Let's consider $f(x) = x + |x|^{\frac{5}{2}}$
  We remove the regular part (by subtracting from f its Taylor's polynomial) which gives us $\alpha(0) = 5/2$. Now, let's consider the derivative, $f'(x) = 1 + \frac{5}{2}|x|^{\frac{3}{2}}$.
  The Hölder exponent of f' in zero is therefore $\frac{3}{2} = \frac{5}{2} - 1$.

- On the other hand, if $f(x) = |x|^{\lambda} \sin \frac{1}{|x|^{\beta}}$ then $\alpha(0) = \lambda$. When we derive f, we get:

$$f'(x) = \lambda |x|^{\lambda-1} \sin \frac{1}{|x|^{\beta}} - \beta |x|^{\lambda-\beta-1} \cos \frac{1}{|x|^{\beta}}$$

  The pointwise Hölder exponent of f' in zero is therefore

$$\lambda - \beta - 1 < \lambda - 1$$

We take advantage of these reminders on the Hölder pointwise exponent to recall the definition of space $C^{\alpha}_{log}(x_0)$ that will be used in the proposal 9.2.3 section 9.2.3.

Definition 4 JAFFARD *[2004] The function $f \in C^{\alpha}_{log}(x_0)$ if there is $c > 0$, $\delta > 0$ and a polynomial P of a degree smaller than α such that:*

$$\text{If } |x - x_0| \leq \delta, \quad |f(x) - P(x - x_0)| \leq c|x - x_0|^{\alpha} \log\left(\frac{1}{|x - x_0|}\right) \tag{9.3}$$

9.1.2 Local Hölder exponent

Sometimes it is necessary to have information on the regularity of a signal not at a point but in the neighbourhood of this point, and it is with the aim of integrating this information into a exponent that the local Hölder exponent α_l is introduced. The pointwise exponent

focused mainly on the envelope of the the function, the local exponent that we will define in this paragraph also takes into account oscillations. Several reasons motivate the use in some cases of the local exponent VEHEL et LUTTON [2001]. First, the pointwise exponent is not enough always to give a perfect description of the regularity of a function. Examples include cusp and chirp functions:

$$\text{cusp}: \ x \mapsto |x|^{\gamma} \ \text{ avec } \gamma \text{ un réel positif} \tag{9.4}$$

$$\text{chirp}: \ x \mapsto |x|^{\gamma} \sin\left(\frac{1}{|x|^{\beta}}\right) \ \text{ avec } \gamma \text{ et } \beta \text{ des réels positifs} \tag{9.5}$$

These two functions have the same pointwise exponent in zero (γ), but their behavior is very different in the neighbourhood of zero as can be seen in the figure below 9.2. The local exponent will take into account the oscillations.

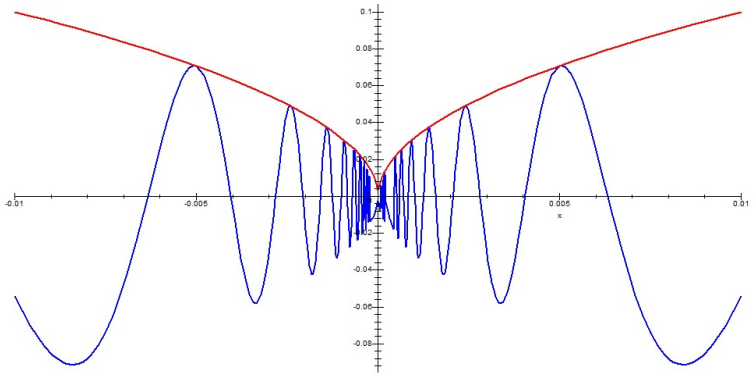


Figure 9.2 – Functions cusp (red) and chirp (blue). These two functions have an identical pointwise exponent in zero but have fundamentally different behaviours.

On the other hand, we saw that the pointwise exponent was not stable by pseudo-differential operators. The local exponent has not this bad property as we will see in the proposal 9.1.2.
Let us recall the definition of global Hölderian regularity.

Definition 5 *Let us consider $f : \Omega \to \mathbb{R}$ a function, with Ω an open interval in $\mathbb{R}$. One says that $f \in C^s(\Omega)$, with $0 < s < 1$ if there is a constant c such that for any couple x, y in Ω,*

$$|f(x) - f(y)| \leq c|x - y|^s \tag{9.6}$$

If $m < s < m + 1$ (with $m \in \mathbb{N}$), then $f \in C^s(\Omega)$ means that there is a constant c such that for any couple x, y in Ω,

$$|f^{(m)}(x) - f^{(m)}(y)| \leq c|x - y|^{s-m} \tag{9.7}$$

The function f is uniformly Hölderian if there is $\varepsilon > 0$ such as $f \in C^{\varepsilon}(\mathbb{R})$.
To define the local Hölder exponent of f in x_0, we are just going to "locate" the definition 5 around x_0.

Definition 6

$$\alpha_f(\Omega) = \sup\{s : f \in C^s(\Omega)\} \tag{9.8}$$

Lemme 9.1.1 *Let $\{\Theta_i\}_{i\in I}$ be a decreasing sequence of open intervals such as*

$$\bigcap_i \Theta_i = \{x_0\} \tag{9.9}$$

Then $\sup\{\alpha_f(\Theta_i), i \in I\}$ is independent of the chosen sequence $\{\Theta_i\}_{i\in I}$.

We can therefore define the local Hölder exponent at x_0 by using any sequence of intervals containing x_0 since it will be independent of the chosen sequence.

Definition 7 *Let f be a function defined in the neighbourhood of x_0. Let $\{I_n\}_{n\in\mathbb{N}}$ be a decreasing sequence of open intervals converging on x_0. The local Hölder exponent of the function f at x_0 is:*

$$\alpha_l(x_0) = \sup_{n\in\mathbb{N}} [\alpha_f(I_n)] = \lim_{n\to+\infty} \alpha_f(I_n) \tag{9.10}$$

Definition 8 *(Equivalent definition)*
If $\alpha_l < 1$, then α_l is the supremum of α such that the inequality below is verified:

$$\exists c \ \ et \ \ \rho_0 > 0 \ \ such\ as \ \ \forall \rho < \rho_0 \quad \sup_{x,y\in B(x_0,\rho)} \frac{|f(x) - f(y)|}{|x-y|^\alpha} \leq c \tag{9.11}$$

Proposition 9.1.2
α_l is stable by the action of operators integro-differential. More specifically, if f has as an exponent local α_l to x_0 then f' has as local exponent $\alpha_l - 1$ and $\int_0^t f(x)dx$ a for local exponent α_l+1.

Proposition 9.1.3 GUIHENEUF *et* VEHEL *[1998]*
Let $f : I \to \mathbb{R}$ be a continuous function on I an interval of $\mathbb{R}$. So $\forall x \in I$,

$$\alpha_l \leq \min\left(\alpha_p, \lim_{t\to x} inf\ (\alpha_p(t))\right) \tag{9.12}$$

Example:
Let us consider again the chirp function that associating x to $|x|^\gamma \sin\frac{1}{|x|^\beta}$ with $0 < \gamma < \min(1,\beta)$. Then:

$$\alpha_p(0) = \gamma \tag{9.13}$$

$$\alpha_l(0) = \frac{\gamma}{1+\beta} \tag{9.14}$$

9.1.3 Hölder Functions

Since the Hölder exponent is defined at any point, it is possible to associate its Hölder function to the signal. This construction is possible for both the local exponent and the pointwise exponent (SEURET et VEHEL [2002]).
This allows to define the pointwise and the local Hölder functions (see DAOUDI et collab. [1998] and GUIHENEUF et VEHEL [1998]).

Definition 9 *Let f be a continuous function. Hölder's functions respectively pointwise and local of f, α_p^f and α_l^f, are the functions that at any x associate the pointwise and local Hölder exponent of f at x.*

Theorem 9.1.1
Let us consider g a function from $\mathbb{R}$ to $\mathbb{R}^+$. The following two assertions are equivalent:

- *g is the inferior limit of a series of continuous functions.*
- *There is a continuous function f such that the pointwise Hölder function α_p^f of f verifies $\alpha_p^f(x) = g(x), \quad \forall x$.*

Theorem 9.1.2
Let us consider g a function from $\mathbb{R}$ to $\mathbb{R}^+$. The following two assertions are equivalent:

- *g is a lower semi-continuous function.*
- *There is a continuous function such that the local Hölder function α_l^f of f verifies $\alpha_l^f(x) = g(x), \quad \forall x$.*

In some cases, it is possible to build easily a function that has the desired Hölder function.

Theorem 9.1.3

Let h be a C^1 function with values in $[0.1]$. Let the generalized Weierstrass function:

$$Wg(x) = \sum_{n=1}^{\infty} \lambda^{-n.h(x)} \sin(\lambda^n x) \quad \textit{with } \lambda \geq 2 \tag{9.15}$$

Then we have $\forall x$:

$$\alpha_l^{W_g}(x) = \alpha_p^{W_g}(x) = h(x) \tag{9.16}$$

9.2 Estimation of the local regularity

After recalling various notions about the Hölder exponent and Hölderian analysis in general, we will now present classical methods for estimating regularity.

Several methods are possible to estimate the Hölder exponent. The most natural, because it follows the definition of the exponent, and which consists in studying the oscillations around the considered point, will be presented in the subsection 9.2.1.

We will continue this section by detailing two techniques of discrete wavelet based estimation (for reminders on wavelets, see DAUBECHIES [1992] and MEYER [1990]). These techniques, are based on two theorems by Stéphane Jaffard.

An intensive comparison of the three methods is presented in LEGRAND [2004] in order to highlight their qualities and limitations.

9.2.1 Oscillations

The estimation of regularity by the oscillation technique is the most natural since it consists in the direct application of the Hölder exponent definition (see definition 8). This method is now briefly presented (see TRICOT [1995]).
We remind that a function $f(t)$ is Hölderian of exponent $\alpha \in [0.1[$ at t if there is a constant c such that for any t' in a neighbourhood of t,

$$|f(t) - f(t')| \leq c|t - t'|^{\alpha} \tag{9.17}$$

In terms of oscillations, this condition can be written: A function $f(t)$ is Hölderian of exponent α at t, with $0 < \alpha < 1$ if there is a constant c such that for any τ,

$$osc_\tau(t) \leq c\tau^\alpha \tag{9.18}$$

with

$$osc_\tau(t) = \sup_{|t-t'|\leq\tau} f(t') - \inf_{|t-t'|\leq\tau} f(t') = \sup_{t',t''\in[t-\tau,t+\tau]} |f(t') - f(t'')| \tag{9.19}$$

Indeed, $osc_\tau(t) \leq c\tau^\alpha$ implies directly $|f(t) - f(t')| \leq c|t-t'|^\alpha$. Conversely, suppose that $|f(t) - f(t')| \leq c|t-t'|^\alpha$ for any t'. Let be t_1 such as $|t-t_1| \leq \tau$ and $f(t_1) = \sup_{|t-t'|\leq\tau} f(t')$. Similarly, t_2 such as $|t-t_2| \leq \tau$ and $f(t_2) = \inf_{|t-t'|\leq\tau} f(t')$. So we have:

$$osc_\tau(t) = f(t_1) - f(t_2) = f(t_1) - f(t) + f(t) - f(t_2) \leq 2c\tau^\alpha \tag{9.20}$$

The regularity estimator will be constructed at each point as the slope of the regression of the logarithm of the oscillation as a function of the size of the ball in which the oscillation is calculated. From an algorithmic point of view, we show that it is preferable not to use all ball sizes between two values $rmin$ and $rmax$: in the same way as for the discrete wavelet transform which only considers dyadic scales, we will calculate the oscillation at the t point only on intervals of the form $[t - base^r : t + base^r]$. Then we regress the logarithm of the oscillation according to r which takes all the integer values between $rmin$ and $rmax$. The input parameters are $rmin$, $rmax$ and $base$ (the base for increasing the intervals).

9.2.2 Regression of wavelet coefficients in the cone of influence (RCO)

A method of estimating from wavelet coefficients is now presented. For this method of estimating regularity, we will use a theorem by Stéphane Jaffard. This theorem shows how the regularity of a signal can be estimated at a point from the wavelet coefficients (when the wavelets used verify certain regularity conditions JAFFARD [2004]).

Being Ψ a mother wavelet of the classical form $\{\Psi_{j,k}\}_{j,k}$ that makes an orthonormal base of $\mathbf{L}^2$, wavelet coefficients of f are denoted as $c_{j,k}$ where j corresponds to the scales and k corresponds to the temporal location.

Theorem 9.2.1 *(S. Jaffard)*

Let f be a uniformly Hölderian function and let $alpha$ be the pointwise Hölder exponent of f in t_0 then there is a constant $c > 0$ such that the wavelet coefficients satisfy:

$$|c_{j,k}| \leq c2^{-j(\alpha+\frac{1}{2})}(1 + |2^j t_0 - k|)^\alpha \quad \forall j,k \in \mathbb{Z}^2 \tag{9.21}$$

Conversely ;

$$If \quad \forall j,k \in \mathbb{Z}^2 \ \ one \ \ has \ \ |c_{j,k}| \leq c2^{-j(\alpha+\frac{1}{2})}(1 + |2^j t_0 - k|)^{\alpha'} \tag{9.22}$$

for a $\alpha' < \alpha$ then, the Hölder exponent of f in t_0 is α.

The condition 9.22 was introduced by J.M. Bony and is noted $f \in C^{\alpha,-\alpha'}(t_0)$.
The interpretation of this theorem is as follows; the wavelet coefficients are dominated in absolute value by a quantity dependent on Hölder's exponent and moreover there is a part of them which are in the order of this quantity.
From this theorem, a classical estimator of regularity at one point is obtained by focusing only on the cone of influence: for the point t_0, we are only interested in the j,k indices such as $|k - 2^j t_0| < cst$. So we assume that there are coefficients in the cone of influence in the order of $2^{-j(\alpha+\frac{1}{2})}$. This simplifying assumption is verified if and only if the local exponent is equal to the pointwise exponent in t_0 VEHEL et SEURET [2004].

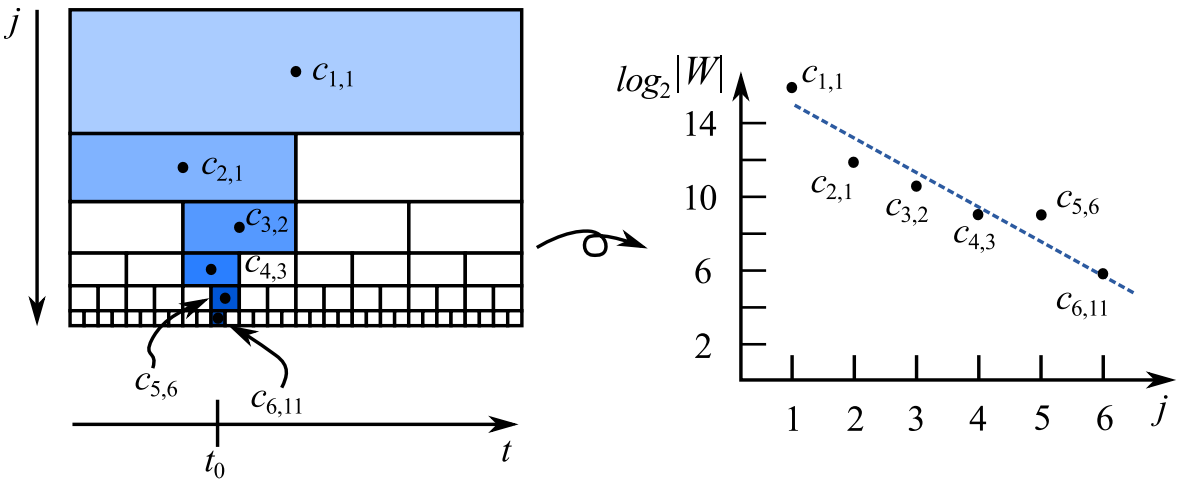
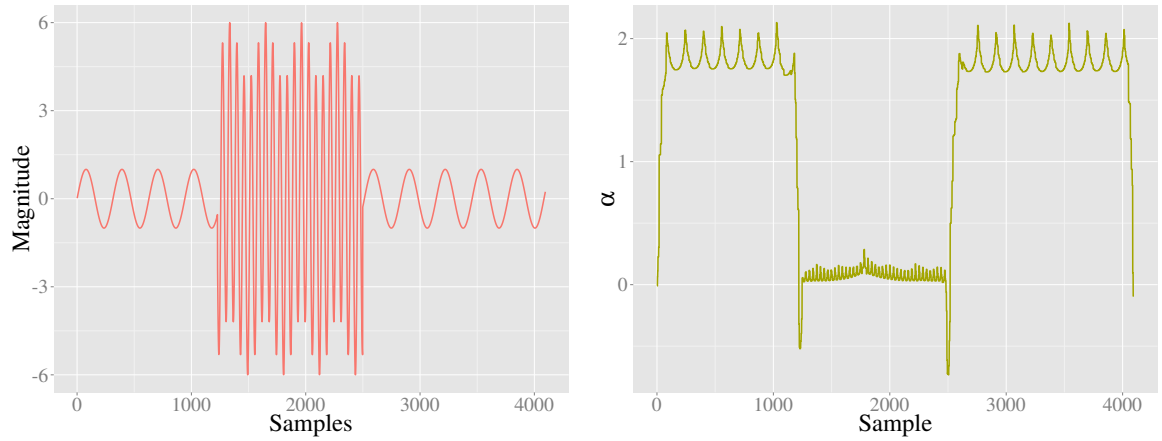


Figure 9.3 – Regression calculated over a point of the signal. Left image shows a dyadic wavelet decomposition, and the right image display the actual regression calculated over the point t_0, where each dot corresponds to each log_2 of the wavelet coefficient magnitude located above t_0.



(a) [Raw signal, superposition of two waveforms with different frequencies.]

(b) [Regularity measure.]

Figure 9.4 – (b) Hölderian regularity calculated over a sample signal (a), where α is the estimated Hölder exponent.

Under this assumption, an estimator of the exponent of Hölder is obtained simply via the p slope of the regression line of $\log_2 |C_{j,k}|$ as a function of j (see figure 9.3) : $\alpha(n, t_0) = -p - \frac{1}{2}$ with n the number of decomposition levels. Figure 9.4 presents an example of regularity computation on a time series. The input signal is composed by the superposition of two waveforms with different amplitudes, frequencies and starting times. The regularity measure characterizes the signal singularities. The amplitude given by α corresponds to the regularity of the signal around a given point. Clearly, the low frequency waveform is more regular than the high frequency one. The hard transition between both waveforms is also captured by the presence of highly irregular spikes.

Definition 10 *At each point t_0 of the signal, the regularity is estimated by:*

$$\alpha(n, t_0) = -p - \frac{1}{2} \tag{9.23}$$

with p the slope of the least square linear regression of the logarithms of the wavelet coefficients "above" this point as a function of the scales.

Proposition 9.2.1
At each point t_0 of the signal decomposed on n scales, we estimate the regularity by the following

formula:

$$\boxed{\alpha(n,t_0) = -\frac{1}{2} - K_n \sum_{j=1}^{n} s_j \log_2 |c_{j,k}|} \tag{9.24}$$

with $K_n = \frac{12}{n(n-1)(n+1)}$ *et* $s_j = j - \frac{n+1}{2}$. *The* $c_{j,k}$ *are the wavelet coefficients above* t_0.

We note k but in reality the value is $\lfloor \frac{t_0+1}{2^{n-j+1}} \rfloor$. For reasons of simplicity of writing, this misuse of notation will be used later in this document.
When considering signals of dimension greater than 1, the principle of regularity estimation is the same in all directions of the separable wavelet transform and identical to that described above.
Proof
The slope p of the regression line is given by:

$$p = \frac{cov(L_{t_0}, S)}{var(S)}$$

With $S = [1..n]$ and $L_{t_0} = [\log_2 |c_{1,k}|.. \log_2 |c_{n,k}|]$ the logarithms of the wavelet coefficients above t_0.
$E[S] = \frac{n+1}{2}$ and $var(S) = \sum_{j=1}^{n} \left[j - \frac{n+1}{2} \right]^2 = \frac{n(n^2-1)}{12}$.
Moreover $cov(L_{t_0}, S) = \langle (L_{t_0} - \overline{L}_{t_0})(S - \overline{L}_{t_0}) \rangle$ with $\overline{L}_{t_0}$ the mean of L_{t_0} and $\overline{S}$ the mean of S. Then

$$cov(L_{t_0}, S) = \sum_{j=1}^{n} \left(\log_2 |c_{j,k}| - \frac{\sum_{i=1}^{n} \log_2 |c_{i,k}|}{n} \right) \left(j - \frac{n+1}{2} \right) = \sum_{j=1}^{n} \log_2 |c_{j,k}| \left(j - \frac{n+1}{2} \right)$$

Consequently

$$p = \frac{12}{n(n^2-1)} \sum_{j=1}^{n} \log_2 |c_{j,k}| \left(j - \frac{n+1}{2} \right)$$

Since $\alpha(n,t_0) = -p - \frac{1}{2}$ then $\alpha(n,t_0) = -\frac{1}{2} - K_n \sum_{j=1}^{n} s_j \log_2 |c_{j,k}|$.

■

For the remainder of this document, when referring to this estimator, we will refer to it as a RCO type estimate.

Remark 9.2.1 *If the estimator presented above converges towards* $\alpha_p = \alpha_l$, *then he also estimates an exponent named the Weak Scaling exponent (see* MEYER *[1997]) and noted* β_w.

Definition 11 *The weak scaling exponent of the signal* X *in* t_0 *is given by:*

$$\beta_w(t_0) = \sup\{s : \exists\, l, \quad X^{(-l)} \in C^{s+n}(t_0)\} \tag{9.25}$$

where $X^{(-l)}$ *represents a primitive of order* l *of* X *and* $C^s(t_0)$ *the pointwise Hölder space in* t_0.

Proposition 9.2.2
When the pointwise exponent and the local exponent coincide, then:

$$\alpha_l = \alpha_p = \beta_w \tag{9.26}$$

It is precisely in this configuration that our estimator is expected to give good results. It should be noted that it has been assumed that there are large coefficients in the cone of influence above the considered point, therefore, the weakness of this estimator lies in the signals such that the large coefficients are not in this cone.
Note that for signals for which the regularity remains constant over time, the weak scaling exponent at each point corresponds to the Hurst exponent, whose estimation will be treated in chapter 10.

9.2.2.1 Scope of validity of the method

The estimation method described above will give good results under three conditions:

- **H1:** In the cone of influence there are coefficients in the order of $2^{-j(\alpha+\frac{1}{2})}$.
- **H2:** The regression converges.
- **H3:** The regression converges to the correct value.

We will now present a favourable case and then unfavourable cases in order to show the limitations of the method.

Favourable case

If a sufficient number of scales are available for perform the regression then we can correctly estimate the Hölder exponent in simple cases. Let's take the example of the Weierstrass function, for this illustration we choose $\alpha = 0.5$ and a point t_0 at random in the signal of 4096 points. We apply the estimator to this function (figure 9.5) and we get 0.51789.

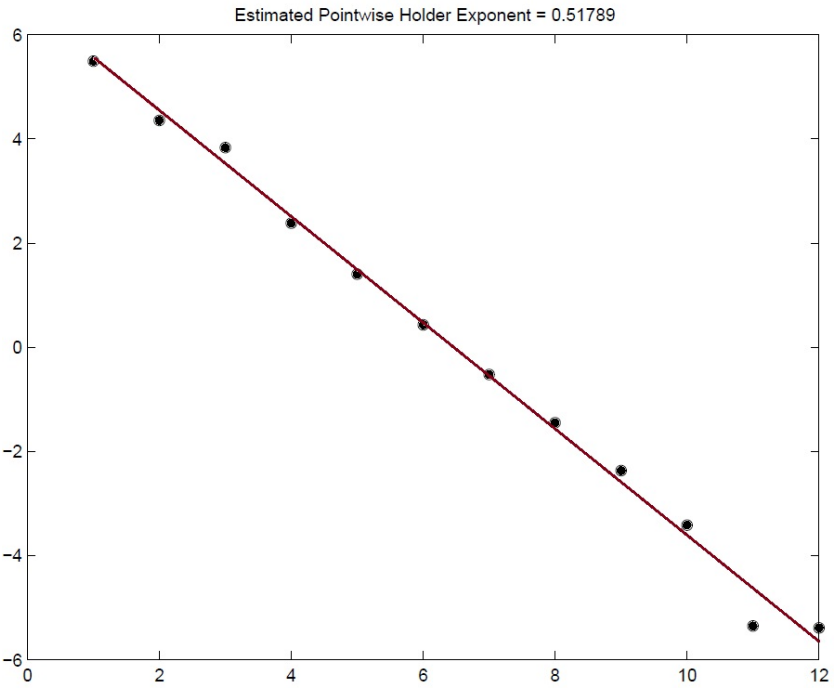


Figure 9.5 – Estimation of the Hölder exponent at a point of a Weierstrass function of 4096 points and regularity of 0.5. The estimator returns the value 0.51789.

Unfavourable case

To illustrate unfavourable cases, we use a by one the conditions that should be verified and exhibited examples for which they are not.

H1: All large coefficients are at outside the cone of influence. We can take the example of the chirp, of equation $|x|^\gamma \sin\left(\frac{1}{|x|^\beta}\right)$, represented on the figures 9.6 and 9.7 for different parameters β. The more β increases, the larger the wavelet coefficients are far from the cone of influence above zero. This is why the estimate is worse for $\beta = 2.9$. (estimate at 0.137 for 0.3) than for $\beta = 0.9$ (estimate at 0.21 for 0.3).

Note that since there are only small coefficients in the cone of influence, it was expected to estimate an exponent of Hölder higher than γ. The fact that we get lower values is due to

misalignment of logarithms wavelet coefficients as can be seen on the figures 9.6 and 9.7 below on the left. The slope of the regression line makes no sense here.

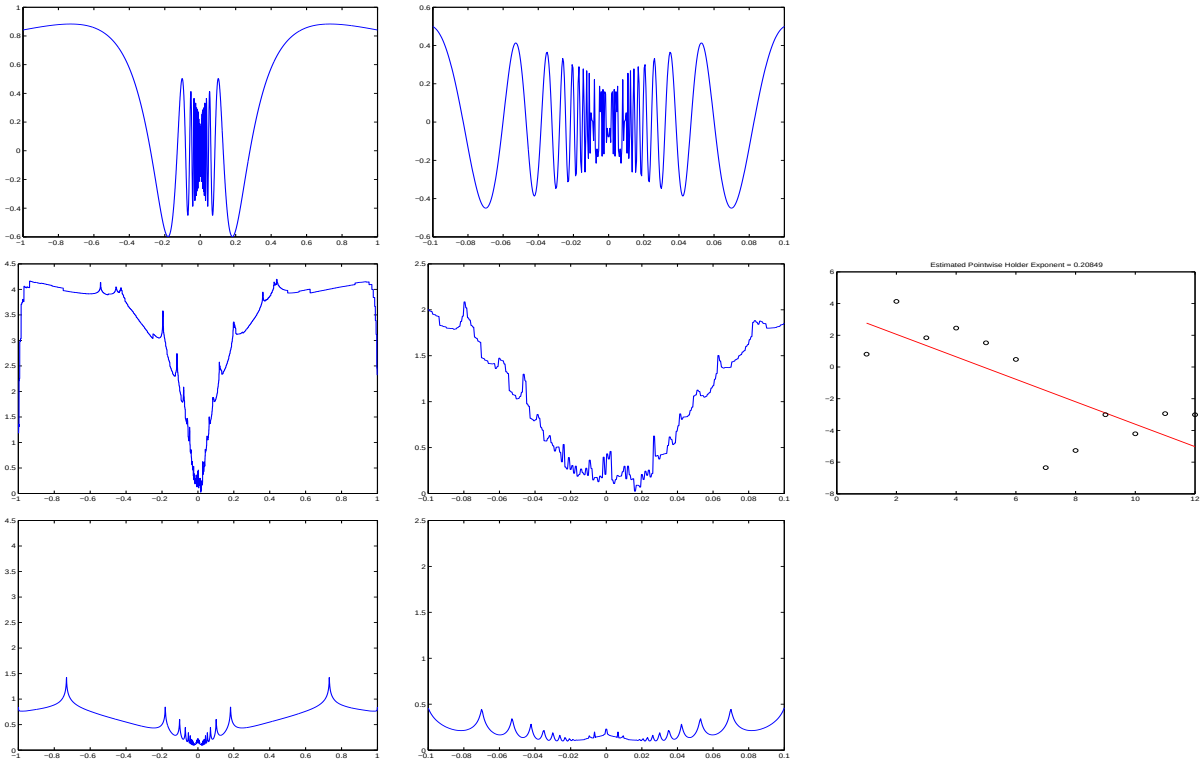


Figure 9.6 – Estimation of the regularity of a Chirp, equation $|x|^{\gamma}\sin\left(\frac{1}{|x|^{\beta}}\right)$ with here $\gamma = 0.3$ and $\beta = 0.9$ (a signal of 4096 points). Up to Left Chirp, and right a zoom around zero. Second line at On the left, estimation of the Hölder function obtained by a estimation of the exponent at each point by the RCO method. Second line in the middle, zoom around zero. Second line at right, estimation of regularity in zero, on the abscissa the scales and ordinates the logarithms based on two of the wavelet coefficients (RCO method). The Hölder exponent is estimated at 0.21 while the theoretical value is 0.3. Third line, estimation of the Hölder function obtained by an estimation of the exponent at each point by the method of oscillation. In zero, with a base of 2.1, $rmin = 1$ and rmax=12, Hölder exponent is estimated at 0.2290.

H2: The regression does not converge.
It is possible to build a signal such that the regression does not converge. For example, let's take a signal for which the wavelet coefficients verifies n_1 scales in $2^{-j(\alpha_1+\frac{1}{2})}$, $2n_1$ scales in $2^{-j(\alpha_2+\frac{1}{2})}$, $4n_1$ scales in $2^{-j(\alpha_1+\frac{1}{2})}$, etc... Under these conditions, the slope of the regression does not converge with n and the estimated exponent will oscillate between α_1 and α_2. This example is illustrated in figure 9.8.

H3: The regression converges but not towards the good value.

To present this case, we construct a signal such as its odd scales verify $|C_{j,k}| = 2^{-(\alpha_1+\frac{1}{2})}$ and the even $|C_{j,k}| = cst.2^{-j(\alpha_2+\frac{1}{2})}$. We illustrate the result obtained with $\alpha_1 = 0.2$ and $\alpha_2 = 0.7$ on the figure 9.9.

It can be seen that for these three examples, the oscillation method gives better results.

The sections 9.2.3 and 9.2.4 describe how to improve results when the conditions **H1**, **H2** and **H3** are not verified.

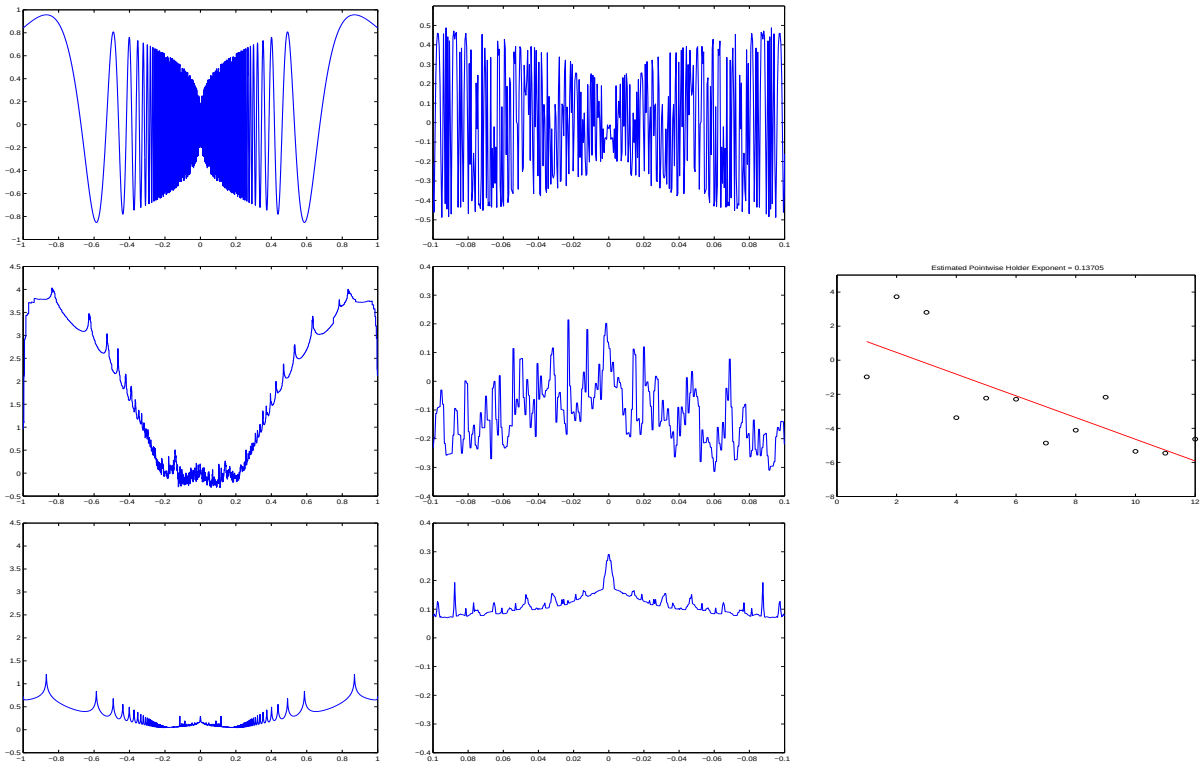


Figure 9.7 – Estimation of the regularity of a Chirp, equation $|x|^{\gamma}\sin\left(\frac{1}{|x|^{\beta}}\right)$ with here $\gamma = 0.3$ and $\beta = 2.9$ (a signal of 4096 points). Up to Left Chirp, and right a zoom around zero. Second line at On the left, estimation of the Hölder function obtained by a estimation of the exponent at each point by the RCO method. Second line in the middle, zoom around zero. Second line at right, estimation of regularity in zero, on the abscissa the scales and ordinates the logarithms based on two of the wavelet coefficients (RCO method). The Hölder exponent is estimated at 0.137 while the theoretical value is 0.3. Third line, estimation of the Hölder function obtained by an estimation of the exponent at each point by the method of oscillation. In zero, with a base of 2.1, $rmin = 1$ and rmax=12, Hölder exponent is estimated at 0.2907.

9.2.3 Regression on Wavelet Leaders

Another estimation technique is now presented. This method is similar to the previous one and based on the Wavelet leaders JAFFARD [2004]. This method theoretically makes it possible to estimate regularity even when the condition **H1** is not not verified. We will compare this refinement to the method described previously in terms of construction as well as in several digital applications. We will show that this technique improves results when conditions **H2** and **H3** are not verified.
A dyadic cube of the scale j is a cube of the form:

$$\lambda = \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right[\tag{9.27}$$

In d dimensions, this interval is extended to a cube in $\mathbb{R}^d$.
In this section, for greater readability, we note $c_\lambda = c_{j,k}$.

Definition 12 *The Wavelet Leaders are*

$$\boxed{d_\lambda = \sup_{\lambda' \subset \lambda} |c_{\lambda'}|} \tag{9.28}$$

We note $\lambda_j(t_0)$ the dyadic cube at the scale j containing the point t_0 of side 2^{-j}.

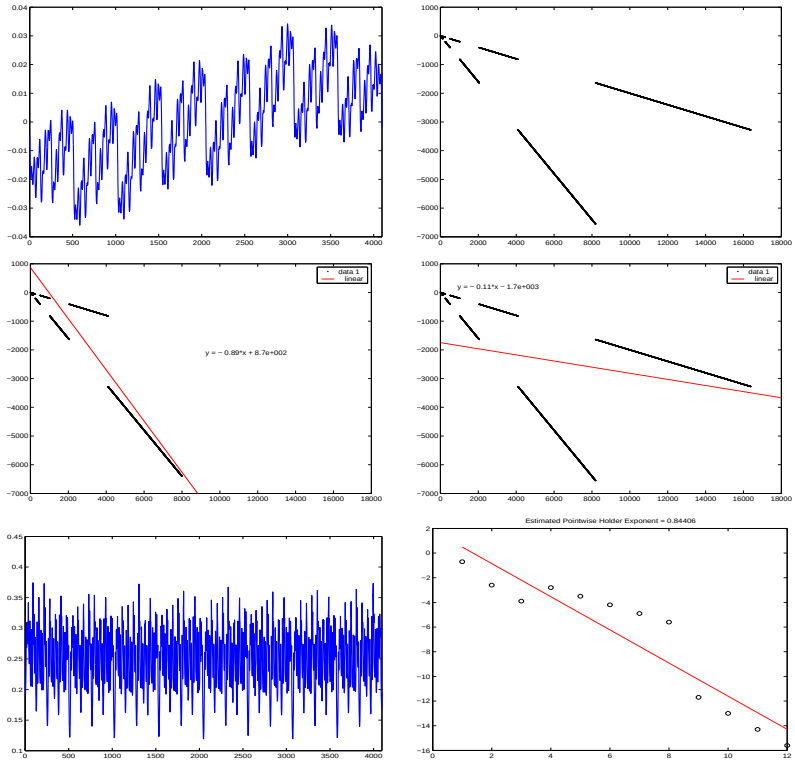


Figure 9.8 – Signal for which the regression does not converge with the number n of decomposition levels. On the left at the top, signal of 4096 points. On the right at the top, logarithm of the wavelet coefficients as a function of scale for a signal of the same type but by 16,000 points. In the middle, the estimated regularity will oscillate depending on n between the two values α_1 and α_2. Here, $\alpha_1 = 0.2$ and $\alpha_2 = 0.8$. The estimate is applied to the signal of 4096 points, which gives us with the method of oscillation an average of 0.2419 for the Hölder function (en bottom left) and for the RCO method the value 0.844 in each point (bottom right). It should be remembered that the theoretical value is 0.2.

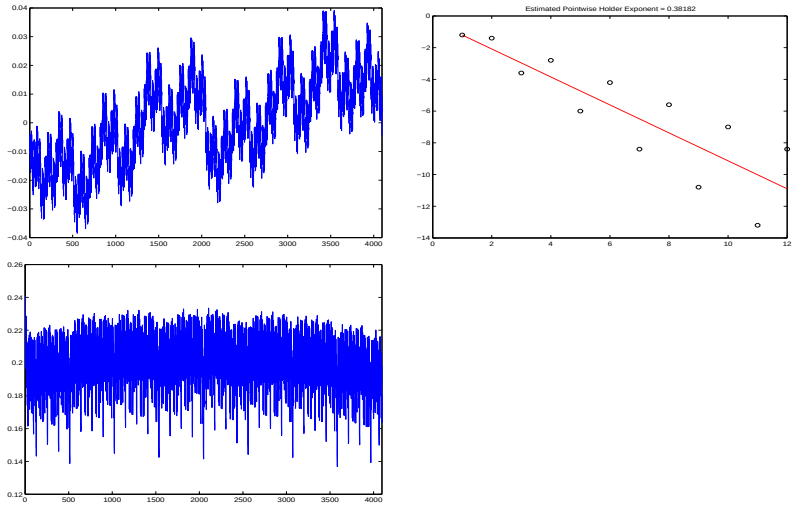


Figure 9.9 – Estimation of the Hölder exponent at a point of a signal of 4096 points and regularity 0.2 for odd scales and 0.7 for even scales. The theoretical value of the pointwise Hölder exponent is 0.2. On the left at the top, signal. On the right, the estimator performs an unfortunate average and returns the value 0.38 at each point. At the bottom, Hölder function obtained by oscillations whose average is 0.199.

Definition 13 *Two dyadic cubes λ_1 and λ_2 are said to be adjacent if they are on the same scale and if $dist(\lambda_1, \lambda_2)$ =0. It should be noted that a dyadic cube is adjacent to itself. We note $adj(\lambda_1)$ the set of dyadic cubes adjacent to λ_1.*

Let

$$d_j(t_0) = \sup_{\lambda' \in adj(\lambda_j(t_0))} |c_{\lambda'}| \tag{9.29}$$

Note that taking the adjacent cubes is a choice that determines the width of the cone of influence, in the following example, for the construction process, we will simply take the dyadic cube instead of the 3 adjacent cubes for simplicity.
The following theorem characterizes pointwise regularity by a condition on the decrease of $d_j(t_0)$ with j.

Proposition 9.2.3 *(S. Jaffard)*
If $f \in C^\alpha(t_0)$, then

$$\exists c > 0, \forall j \geq 0, \qquad d_j(t_0) \leq c 2^{-(\alpha+\frac{1}{2})j} \tag{9.30}$$

Conversely, if the equation 9.30 is verified, and if f is uniformly Hölderian, then f belongs to $C^\alpha_{\log}(t_0)$.

We now adapt the estimator of the section 9.2.2 by replacing in the regression the wavelet coefficients above t_0 by the Wavelet Leaders.

9.2.3.1 Construction:

The coefficients that will be taken into account in the regression will therefore not be the same as before. An example is given in figure 9.10. A dyadic grid is schematized. This grid contains the wavelet coefficients of the signal under consideration. The low frequency coefficient is at the top and the high frequency coefficients are just "above" the signal. Let us consider the framed point of the signal, the example described here will focus on the analysis of regularity at this point.

- Estimation with RCO method:

  The wavelet coefficients for this point are noted as A, B, C and D. Each of these coefficients gives us information on the frequency content at this point: A represents the low frequencies and D the high frequencies. The regularity estimator will be obtained by regressing the 2 logarithm of the absolute value of these coefficients according to the scale.

- Estimation with Wavelet Leader:

  To make an estimation with Wavelet Leaders, the method is different. We start from the point of the signal and look successively at the dyadic cubes concerning it. First the cube containing the D coefficient. This coefficient is the only coefficient of the dyadic cube at this level, so we keep it. We then go up a scale to now consider the dyadic cube containing C (this cube obviously contains D and another coefficient. At this level, we keep the $\max$ among the 3 coefficients of this cube. Suppose that the $\max$ is the C coefficient, so it is the one we keep for regression. We go up again in the scales to take into account the dyadic cube containing B, C and D. We're looking at the maximum $\max$ of the coefficients of this cube. Suppose that this $\max$ is represented by E, then E will "replace" B in the regression. This replacement is illustrated in figure 9.11.

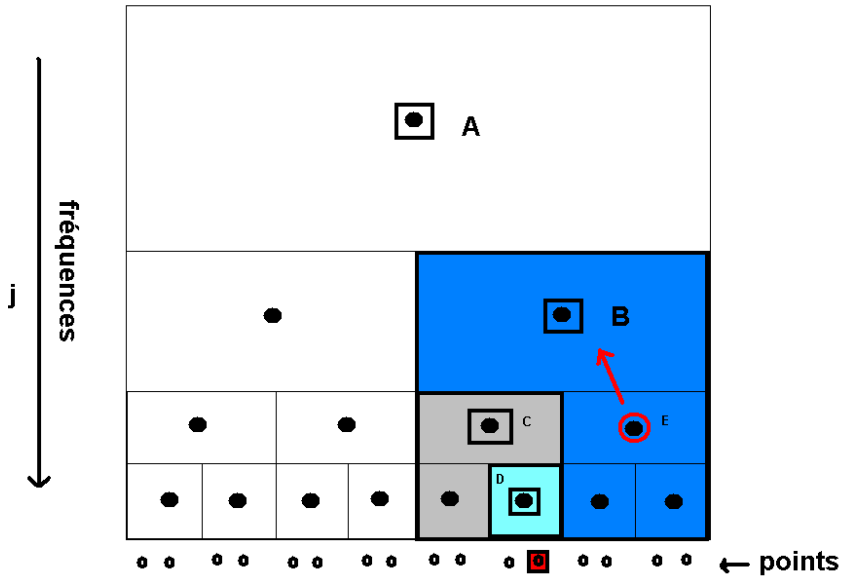


Figure 9.10 – Dyadic grid containing the wavelet coefficients of the considered signal. Illustration of the "Wavelet Leaders" technique for estimating regularity.

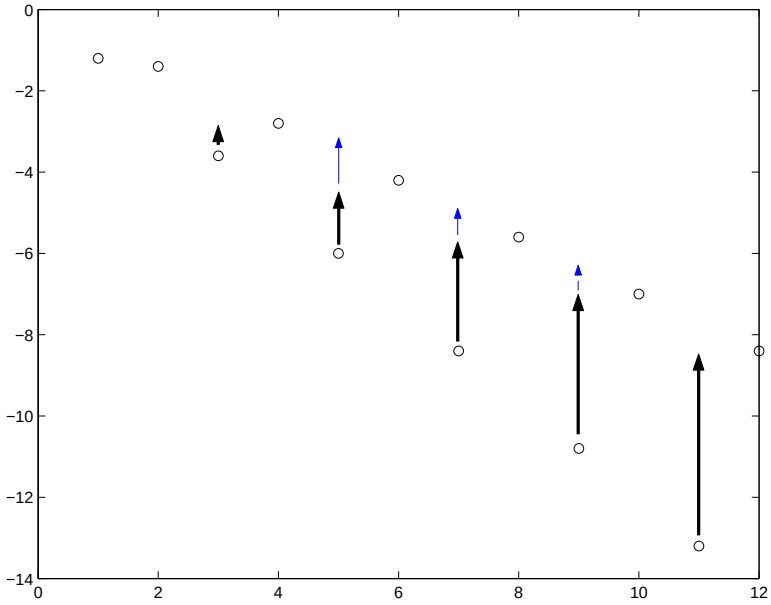


Figure 9.11 – Regression above one point of the signal. The abscissa axis carries the scales. Each circle corresponds to the log2 of a wavelet coefficient located above the considered point. Wavelet Leader's technique replaces in the regression the coefficient above the point by its "leader". It should be noted that the logarithm of the value of the coefficient rises at least as high as those of the coefficients of the highest frequencies (black arrows). The clear arrows indicate that the coefficients can go up a little higher according to the values of the other coefficients of the dyadic cube.

9.2.3.2 Modification for the Wavelets leaders

Lemme 9.2.1
At each point t_0 of the signal X decomposed into n levels, the regularity is estimated with the Wavelet

Leaders by using the following formula:

$$\boxed{\alpha_{WL}^{X}(n,t_0) = -\frac{1}{2} - K_n \sum_{j=1}^{n} s_j \log_2 \left[\max_{\lambda' \subset \lambda}(|x_{\lambda'}|)\right]} \tag{9.31}$$

Definition 14

$$\begin{array}{rccc} Soit\ \ f: & \mathrm{L}^\infty(I\!R^d) & \longrightarrow & \mathrm{L}^\infty(I\!R^d) \\ & X & \longmapsto & Y \end{array} \tag{9.32}$$

We say that f is increasing on the wavelet coefficients if for every couple of wavelet coefficients $(x_{j,k}, x_{j',k'})$ of X such that $x_{j,k} \leq x_{j',k'}$ then $y_{j,k} \leq y_{j',k'}$.

Lemme 9.2.2 *(P. Legrand) (Conservation of the "Wavelet Leaders" position)*

$$\begin{array}{rccc} Soit\ \ f: & \mathrm{L}^\infty(I\!R^d) & \longrightarrow & \mathrm{L}^\infty(I\!R^d) \\ & X & \longmapsto & Y \end{array} \tag{9.33}$$

If f is increasing on the wavelet coefficients then

$$\boxed{\forall t_0 \in I\!R\ \ et\ \ \forall j \in [1..n],\quad \underset{(j,k),y_{j,k} \in \lambda_j(t_0)}{argmax} |y_{j,k}| = \underset{(j,k),x_{j,k} \in \lambda_j(t_0)}{argmax} |x_{j,k}|} \tag{9.34}$$

Proof
Trivial. ∎

Corollaire 9.2.1 *If f is increasing on the wavelet coefficients then :*

$$\boxed{\alpha_{WL}^{f(X)}(n,t_0) = -\frac{1}{2} - K_n \sum_{j=1}^{n} s_j \log_2 \left[\max_{\lambda' \subset \lambda}(|y_{\lambda'}|)\right]} \tag{9.35}$$

9.2.3.3 Applications

The previous estimation method worked well when all three properties **H1**, **H2** and **H3** were verified. The contribution of Wavelet Leaders makes it possible to free oneself from **H1** from a theoretical point of view since the consideration of all the coefficients of the dyadic cube and the adjacent dyadic cubes allows to recover some large wavelet coefficients. This contribution is illustrated by using the example of chirp. We recall on the figures 9.12 to 9.15 the results obtained on this signal with the RCO and oscillation methods . These results are now complemented by those of obtained by the estimation by regression of Wavelet Leaders. This method improves the simple regression RCO on this signal although here we only take the dyadic cube and not the adjacent cubes. We can see the good alignments of the logarithms coefficients involved in the regression on the figures 9.13 and 9.15 indicating that the regression has a meaning here.
This refinement of the method is then applied to other signals for which the RCO estimate did not give good results. We start with a signal for which the regression does not converge. To illustrate this, we use the signal we presented earlier, which does not satisfy **H2** (figures 9.8 and 9.16). Refinement through Wavelet Leaders improves the estimation of regularity in some cases but not always.
Similarly, the Wavelet Leader method can improve the estimation when the regression does not converge to the correct value (the condition **H3** is not verified). The Wavelet Leader method improves the estimation (figures 9.9 and 9.17) since we go from a regularity estimated at 0.38 by the RCO method to a regularity estimated at 0.19 (remember that the theoretical value is 0.2 for this signal at each point).

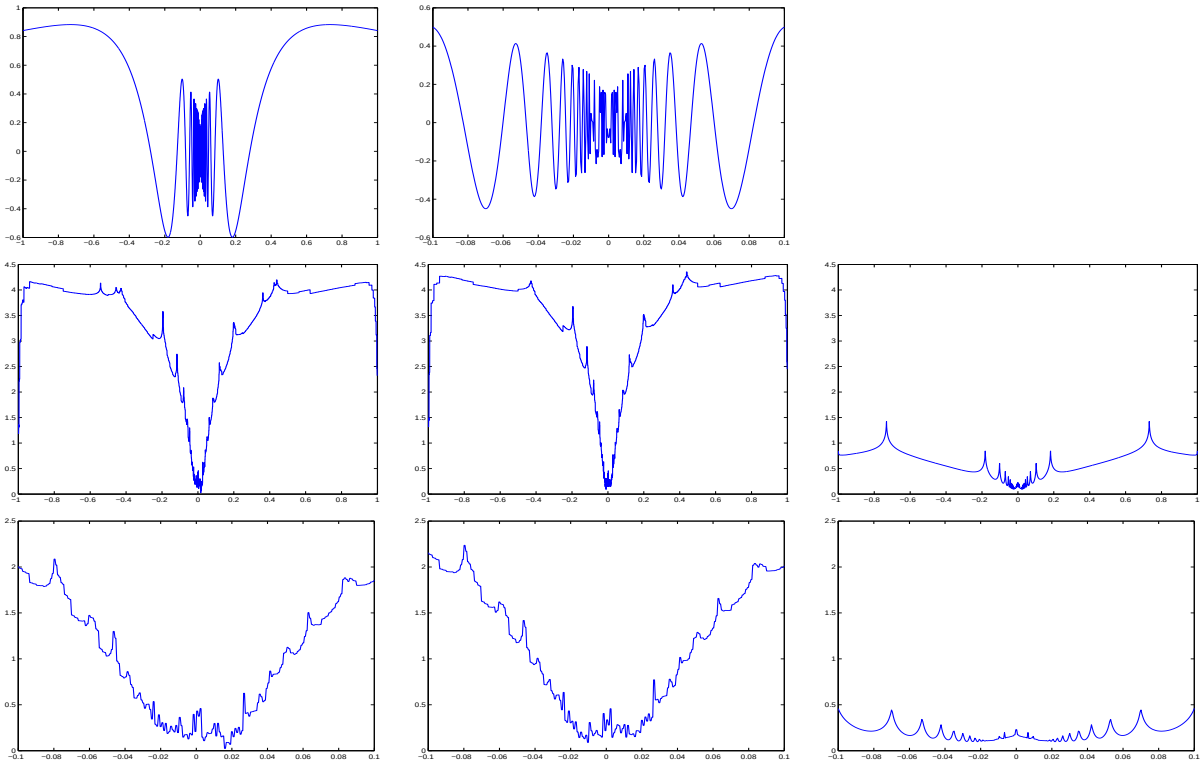


Figure 9.12 – Estimation of the regularity of a Chirp, of equation $|x|^\gamma \sin\left(\frac{1}{x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 0.9$ (a 4096 point signal). Reminder of the results obtained and addition of the estimation by Wavelet Leaders. Top left Chirp, and right a zoom around zero. Second line, estimation of the Hölder function obtained by estimating the exponent at each point respectively by the methods RCO, Wavelet leaders and oscillation. In zero, the Hölder exponent estimations are 0.21 for RCO, 0.2177 for Wavelet Leaders and 0.2290 for oscillation while the theoretical value is 0.3.

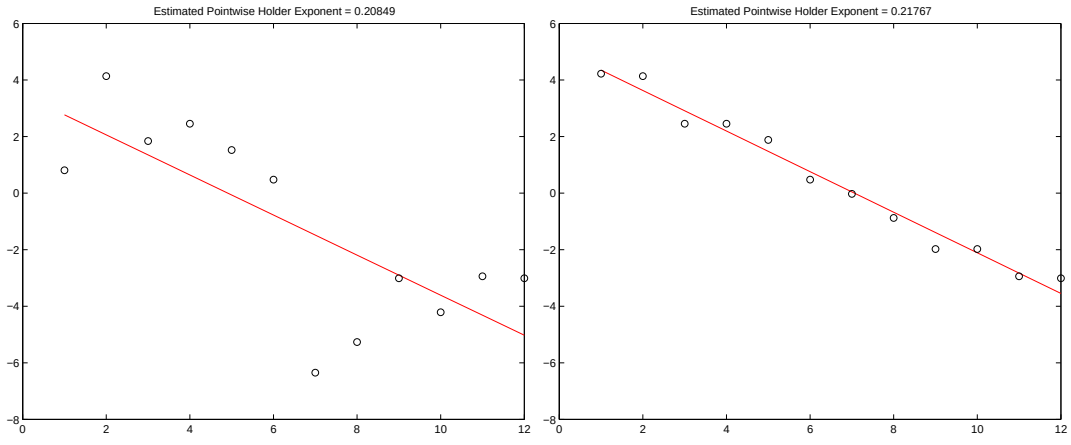


Figure 9.13 – Estimation of the regularity in zero of a chirp of equation $|x|^\gamma \sin\left(\frac{1}{|x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 0.9$. Left: Regression of the logarithms of the wavelet coefficients versus scale above zero. Wavelet Leaders regression on the right. The regression of the Wavelet Leaders makes sense because they verify a proper alignment. The estimation obtained are 0.21 for RCO and 0.2177 for Wavelet Leaders.

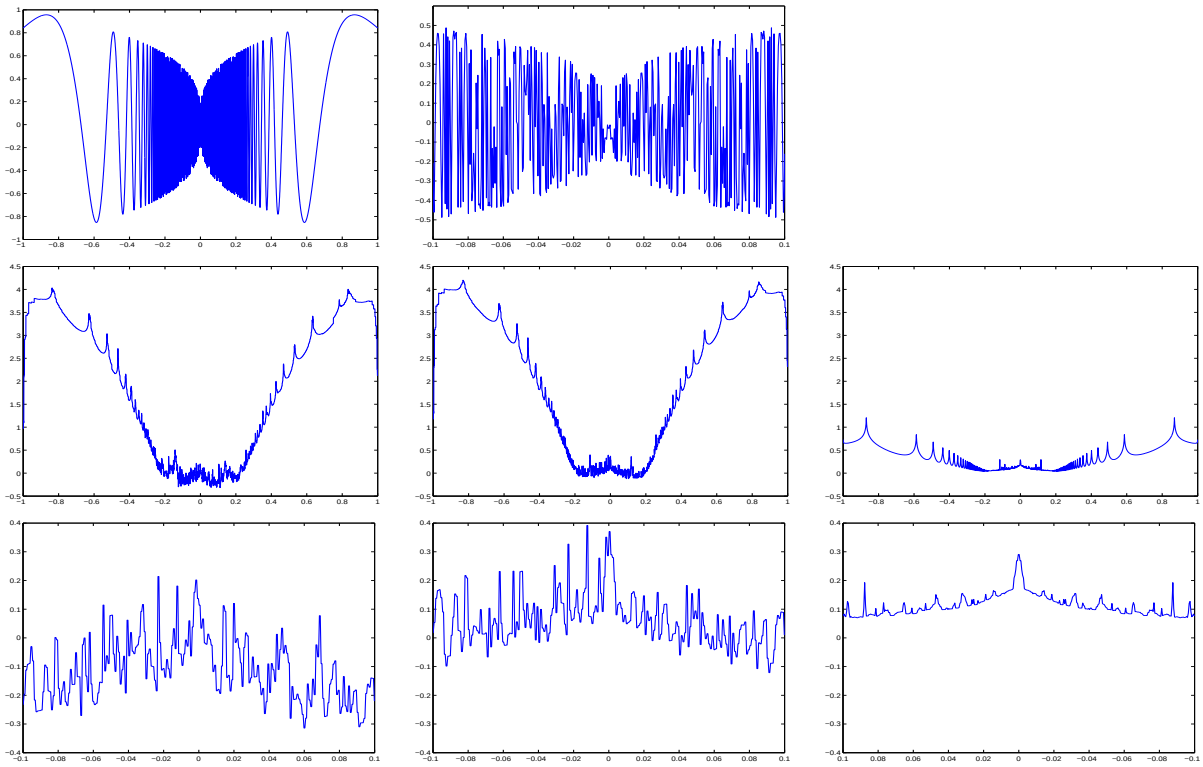


Figure 9.14 – Estimation of the regularity of a Chirp, of equation $|x|^\gamma \sin\left(\frac{1}{x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 2.9$ (a 4096 samples signal). Reminder of the results obtained and addition of the estimation by Wavelet Leaders. Top left Chirp, and right a zoom around zero. Second line, estimation of the Hölder function obtained by estimating the exponent at each point respectively by the methods RCO, Wavelet leaders and oscillation. In zero, the Hölder exponent estimations are 0.137 for RCO, 0.286 for Wavelet Leaders and 0.2907 for oscillation while the theoretical value is 0.3.

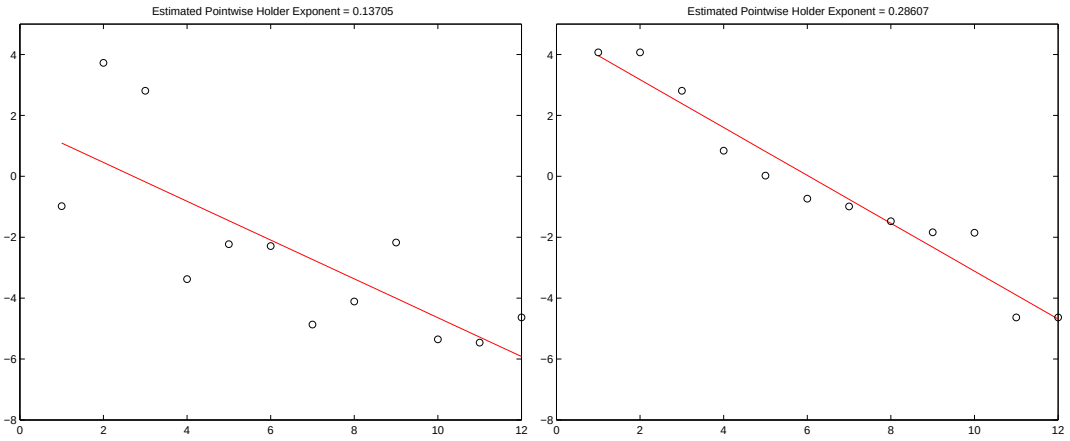


Figure 9.15 – Estimation of the regularity in zero of a chirp of equation $|x|^\gamma \sin\left(\frac{1}{|x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 2.9$. Left: Regression of the logarithms of the wavelet coefficients versus scale above zero. Wavelet Leaders regression on the right. The regression of the Wavelet Leaders makes sense because they verify a proper alignment. The estimation obtained are 0.137 for RCO and 0.286 for Wavelet Leaders.

9.2.4 Inferior and superior limit regressions

If the upper and lower limits are different, then no well-defined slope may be found in the regression. However, it is still possible to estimate the upper and lower dimensions

through a modified regression scheme, that we proceed to explain now. The use of these *liminf* and *limsup* regression methods is crucial, as it allows to attribute well-defined fractal quantities to arbitrary signals. This is of particular importance for local parameters such as the Hölder exponents. Indeed, the definition of the Hölder pointwise exponent (like that of other fractal quantities) has a lower limit rather than a limit, which allows this exponent to always be defined even if the limit does not exist. For this reason, one of our contributions is to propose a type of regression that under certain conditions allows us to reach the inferior limit or the superior limit.

This technique allows us to solve some problems encountered by our estimator on signals of the type described in the section 9.2.2.1.

9.2.4.1 Principles of the methods Inferior and Superior limit regressions

Let $(l_j)_{j\geq 1}$ be an arbitrary sequence of real numbers, and denote $u_j = \frac{l_j}{j}$. Let $a = \text{liminf}_{j\to\infty} u_j$. In our frame, think for instance of l_j as the logarithm of the number of boxes in the computation of the box dimension. Define, for all $n \geq 1$:

$$E_n^0 = \{1, \ldots, n\}$$

$$L_n^0 = \{l_1, \ldots, l_n\}$$

Let (a_n^0, b_n^0) be the parameters of the least square regression of L_n^0 with respect to E_n^0, i.e. the real numbers that minimize $\sum_{j=1}^n (l_j - aj - b)^2$ over all couples (a, b). We write :

$$(a_n^0, b_n^0) = Reg(E_n^0, L_n^0)$$

Let now:

$$E_n^1 = \{j \in E_n^0, l_j \leq a_n^0 j + b_n^0\}$$

$$L_n^1 = \{l_j, j \in E_n^1\}$$

$$(a_n^1, b_n^1) = Reg(E_n^1, L_n^1)$$

Define recursively:

$$E_n^i = \{j \in E_n^{i-1}, l_j \leq a_n^{i-1} j + b_n^{i-1}\}$$

$$L_n^i = \{l_j, j \in E_n^i\}$$

$$(a_n^i, b_n^i) = Reg(E_n^i, L_n^i)$$

for all $i = 2, \ldots N_n$, where N_n is defined as the first index such that $\#E_n^{N_n+1} < 2$.

The geometrical interpretation of the sequence (a_n^i, b_n^i) is simple: In the first step, we keep in (E_n^1, L_n^1) those points that are "below" the regression line of L_n^0 with respect to E_n^0. We then compute the regression line of L_n^1 with respect to E_n^1 to obtain (a_n^1, b_n^1), and iterate the process until at most one point remains below the regression line.

The slope of the lower limit regression is then given by $a_n^{N_n}$. The method is similar for the upper limit except that the points below the regression line are kept.

Proposition 9.2.4 *Let $(l_j)_{j\geq 1}$ be an arbitrary sequence of real numbers, and define u_j, a, a_n^i and N_n as above. Then:*

$$\lim_{n\to\infty} a_n^{N_n} = a$$

9.2.4.2 Applications

This new regression method is applied to signals that defeated the RCO type regularity estimator in the section 9.2.2.1. First of all, we take up the example of Chirp. We remind that for this signal **H1** is not verified. The lower limit regression does not necessarily improve the estimation of regularity in zero (see figures 9.18 and 9.19).

Figure 9.20 shows how this method solves the problem encountered when the condition **H2** is not verified. In the same way, this technique compensates for the difficulties of a regression that does not converge, i.e. when the condition **H3** is not verified (see figure 9.21).

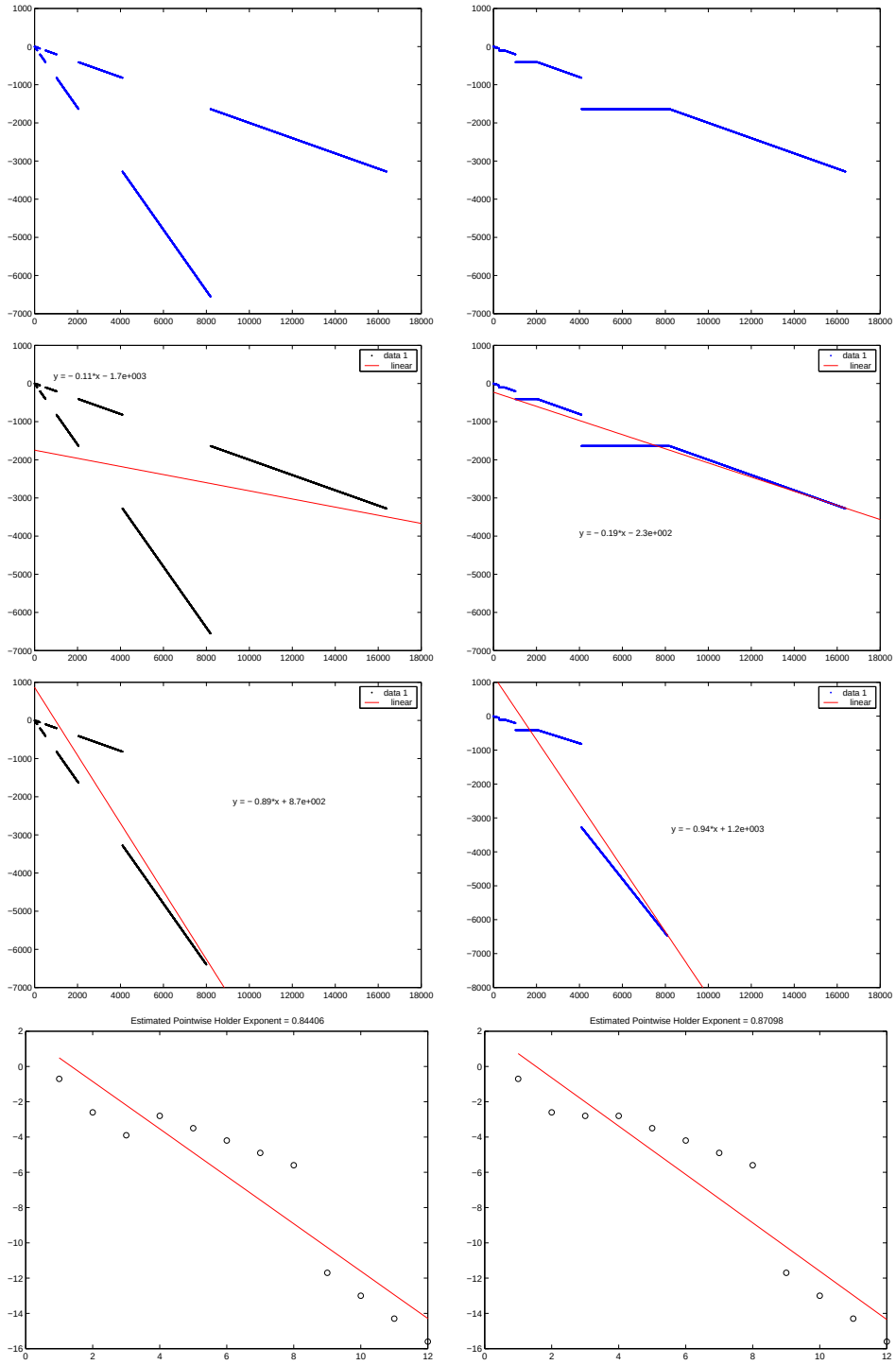


Figure 9.16 – Estimation of the regularity of a signal whose regression does not converge. Top left, logarithm of wavelet coefficients as a function of scale above a point. At the top right, with the Wavelet Leader estimation, everything will happen as if the coefficients had this shape. Left 2nd line: regression type RCO, estimation at 0.11 instead of 0.2. On the right 2nd line: regression of the Wavelet leader type, estimation at 0.19 instead of 0.2. On the left 3rd line, illustration of the non-convergence with the RCO method by removing part of the scales. On the right 3rd line, illustration of non-convergence also with the Wavelet Leaders method. We performed the regression as if we no longer had as many scales to show that the slope of the regression does not tend towards a limit value. 4th line, estimation at one point (the estimate is the same at each other point) of the 4096-point signal by the RCO and Wavelet Leaders methods. The estimated regularities are 0.844 and 0.871 instead of 0.2 respectively.

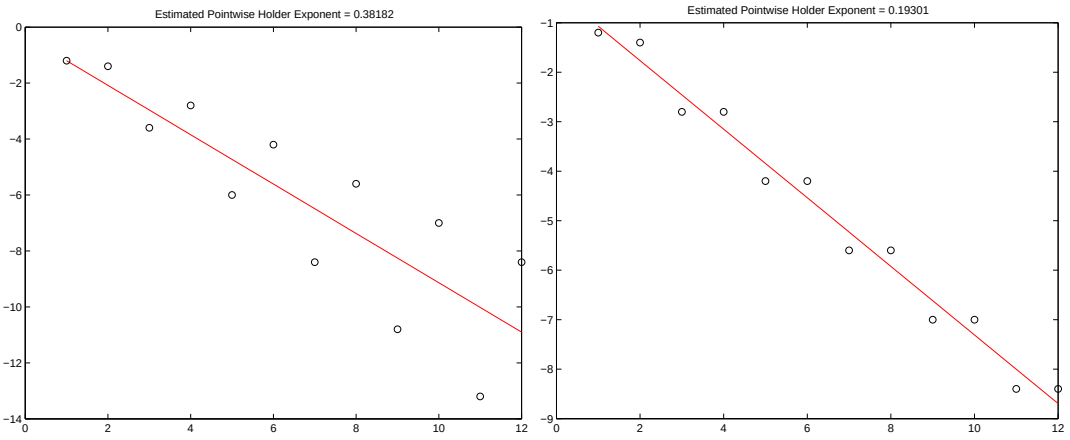


Figure 9.17 – Estimation of the regularity of a signal with "two regularities". The theoretical exponent value for this signal is 0.2. Left: regression of type RCO estimating regularity at 0.38. Right: Wavelet leader regression. As we can see, the coefficients involved in the regression are no longer the same, which gives us an estimator at 0.193 and thus improves the estimate. We remind that by oscillations, we obtain on average 0.199.

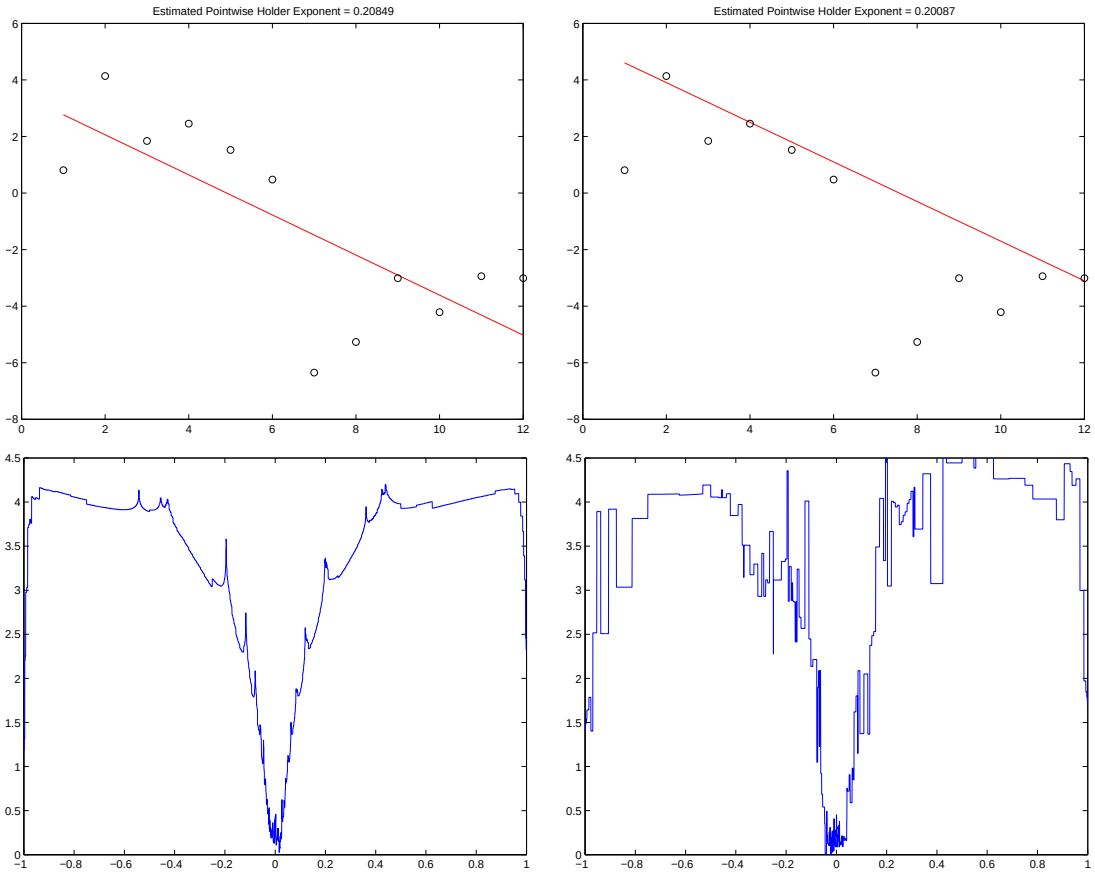


Figure 9.18 – Estimation of regularity in zero of a chirp of equation $|x|^\gamma \sin\left(\frac{1}{|x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 0.9$. Top left: estimated by the RCO method at zero point. Right: estimation by the RCO method with a lower limit type regression at the zero point. The exponent of Hölder is estimated at 0.20 while 0.21 was obtained by the linear regression at the least squares. Remember that the value is 0.3. Below, the Hölder functions obtained, by least square regression on the left and by regression of the inferior limit type on the right.

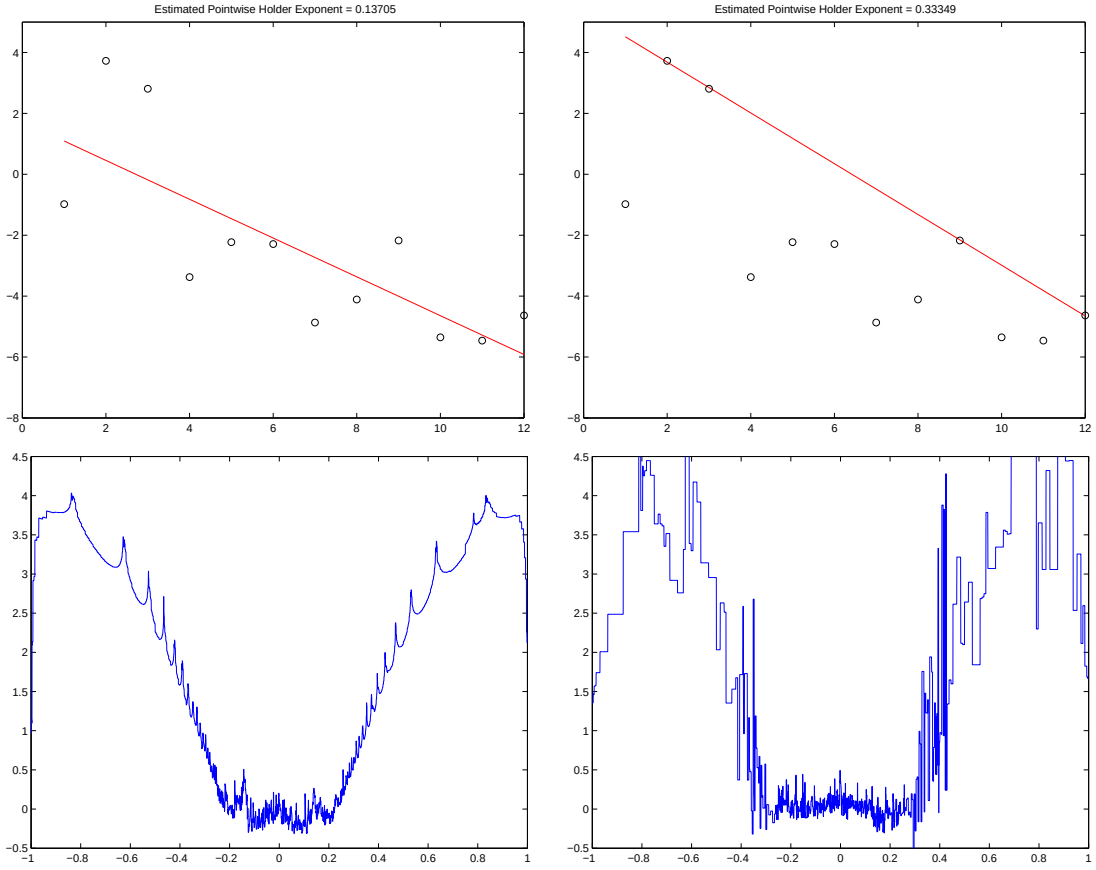


Figure 9.19 – Estimation of regularity in zero of a chirp of equation $|x|^\gamma \sin\left(\frac{1}{|x|^\beta}\right)$ with here $\gamma = 0.3$ and $\beta = 2.9$. Top left: estimated by the RCO method at zero point. Right: estimation by the RCO method with a lower limit type regression at the zero point. The exponent of Hölder is estimated at 0.33 while 0.137 was obtained by the linear regression at the least squares. Remember that the value is 0.3. Below, the Hölder functions obtained, by least square regression on the left and by regression of the inferior limit type on the right.

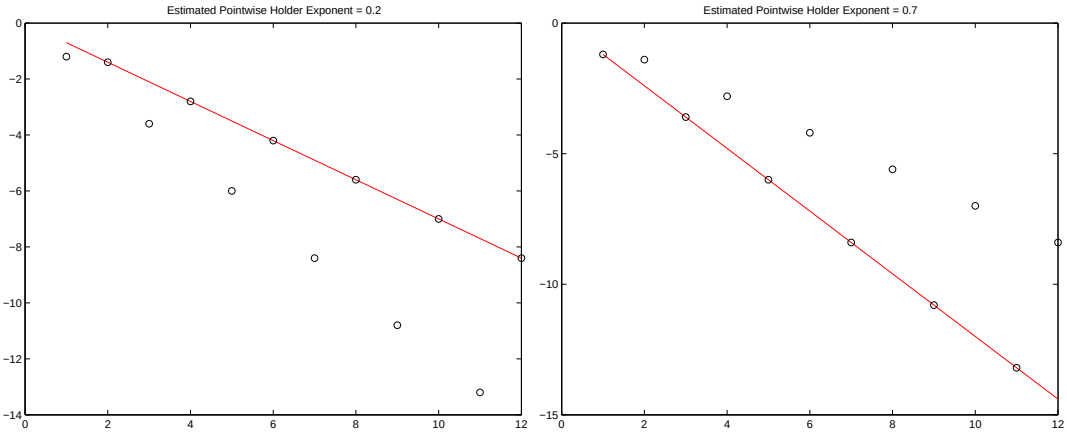


Figure 9.20 – Estimation of the regularity of a signal with "two regularities". Left: lower limit regression. Right: upper limit type regression. The Hölder exponent is estimated at 0.2 which is the theoretical value.

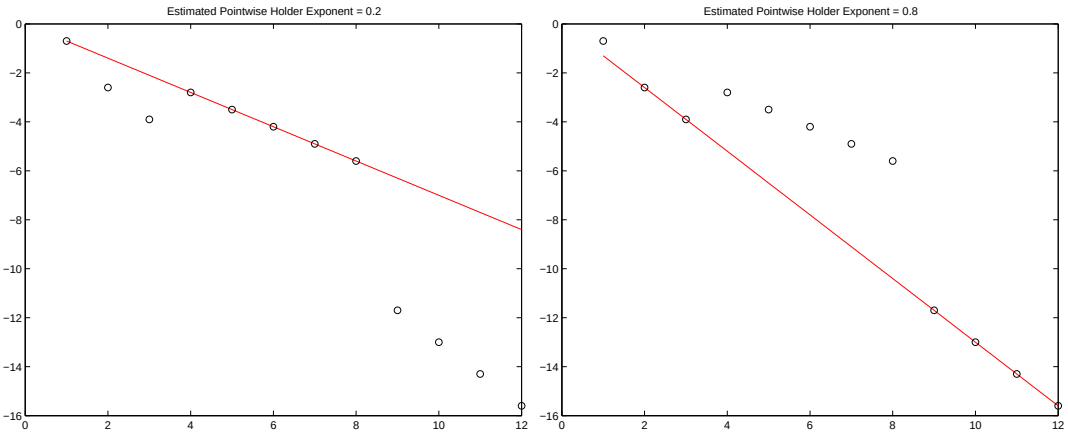


Figure 9.21 – Estimation of the regularity of a signal whose regression does not converge. Left: lower limit type regression. Right: upper limit type regression. Regardless of the number of scales available, the lower limit regression perfectly estimates the Hölder exponent at its theoretical value of 0.2.

9.3 Fraclab

All these methods and refinements have been coded and added to the **Fraclab** Toolbox VEHEL et LEGRAND [2004].

I was in charge of the development of the FracLab software (see figure 15.13) under the supervision of **Jacques Levy-Vehel**. I also participated in mathematical developments and code writing (from 2000 to 2017). FracLab is a Matlab toolbox, or stand alone (without Matlab), free, based on fractal and multifractal methods for signal processing or fractal analysis. The first version of FracLab dates back to 1998 and I built the first stand alone version in 2001. I also made its current version in 2017 under Inria license. A list of contributors is available at the following url https://project.inria.fr/fraclab/people/. Here is the list a non-exhaustive (and depreciated) list of published works using FracLab: https://project.inria.fr/fraclab/works-using-fraclab/.

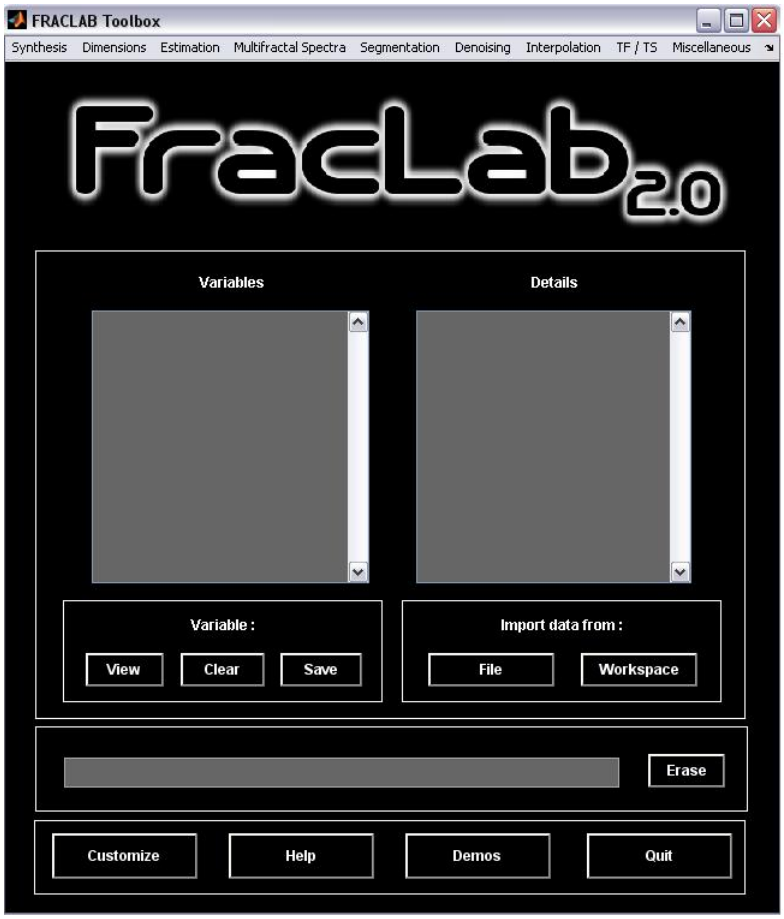


Figure 9.22 – The graphical user interface of the FracLab toolbox. https://project.inria.fr/fraclab/.

9.3.1 Motivations

Fractal and multifractal tools have found a large number of applications in recent years. They are increasingly used in areas including astronomy, medical image/signal processing, telecommunications, finance, speech processing and many more.

With the spread of fractal analysis in such diverse fields, it seems important that researchers and practitioners willing to make use of fractal tools dispose of a stable set of methods for computing e.g. fractional dimensions, correlation exponents or multifractal spectra. Such methods should both be thoroughly tested and up-to-date, so that they may serve as a common background to compare approaches and results. One aim of *FracLab* is to

fulfill such a need, by providing an *open and free software toolbox*, composed of routines that can be tested, enhanced, and may serve as benchmark in various situations.

A second aim of *FracLab* is related to an important and recent evolution in the use of fractal analysis: It has been realized that it is often beneficial to apply fractal tools to arbitrary (i.e. "non-fractal") signals. The best known example is probably fractal image compression based on IFS theory, as popularized in BARNSLEY [1988]: IFS-based compression allows to process any kind of images, without an assumption of "fractality". This is also the point of view adopted in *FracLab*: *FracLab* performs *fractal processing* of signals, rather than processing of *fractal signals*.

This approach should not be too surprising: Just as, e.g., gradient-based algorithms are often successfully applied for image segmentation even when there are no mathematical or physical reasons for the original signal to possess an ordinary derivative, a fractal analysis may yield new insights for “non-fractal” data. More generally, *FracLab* proposes to use fractal analysis in exactly the same way as other mathematical tools are used in everyday signal processing: Under certain assumptions, one may always estimate a gradient from discrete data (for instance *via* a model, or by first regularizing it), or compute its Fourier transform (for instance by extending it in a proper way outside the observation domain). In the same way, *FracLab* computes fractional dimensions or multifractal spectra (these quantities are always defined) by making adequate assumptions. These will in fact be of the same nature as in classical signal processing: We shall assume that the underlying continuous signal belongs to a parametric class, or we shall “regularize” it in some sense (in scale rather than in space). Of course, a fractal analysis will not in general give useful indications when the signal is mainly regular or smooth. It will be of interest only if there is enough singularity in the data, and if the singularity structure bears important information. There are many cases where the irregular part of the observed data contains important information that cannot be processed if only the smooth part is kept. It can even be the case that most or all of the relevant information is carried in the singular structure of the observation. Let us give an example. Radar images are difficult to process because of the presence of a specific noise, the so-called *speckle*. However, speckle is not pure noise, but rather a genuine part of the signal, caused by the interferometric nature of radar images. In this respect, it contains information which is essential about the imaged region. Although removing the speckle can be useful for purposes of e.g. segmentation, analyzing it is a necessary task for other applications, for instance classification, simply because the smoothed signal does not contain the necessary information. From a broader point of view, one may even argue that, although many image processing techniques aim at getting rid of irregularities in the data, the segmentation of simple, non noisy optical images should more logically be based on singularity analysis: One is indeed mostly interested in singularities, since edges, for instance, are basically discontinuities in the grey levels. In that respect, most classical approaches, based on smoothing, do not appear as natural as is usually assumed.

The second aim of Fraclab is then to help disseminate the use of fractal tools in the processing of irregular but arbitrary signals and images. This will allow to discover new situations where fractal analysis yields an interesting alternative to more classical signal processing tools.

9.3.2 FracLab and this manuscript

Fraclab was used to carry out the research presented in all the chapters of part III.

References

BARNSLEY, M. 1988, «Fractales Everywhere», *Academic Press, New York*. 218

DAOUDI, K., J. L. VEHEL et Y. MEYER. 1998, «Construction of functions with prescribed local regularity», *Constructive Approximation*, vol. 014(03), p. 349–385. 197

DAUBECHIES, I. 1992, *Ten Lectures on Wavelets*, vol. 61, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia. 198

GUIHENEUF, B. et J. L. VEHEL. 1998, «2-microlocal analysis and applications in signal processing», *International Wavelets Conference, Tangier*. 197

JAFFARD, S. 1997, «Multifractal formalism for functions, i and ii», *Siam J. Math. Anal.*, vol. 28(4). 194

JAFFARD, S. 2004, «Wavelet techniques in multifractal analysis», *Proceedings of Symposia in Pure Mathematics*, vol. 72. 194, 195, 199, 204

JAFFARD, S. et Y. MEYER. 1996, «Wavelet methods for pointwise regularity and local oscillations of functions», *Mem. Amer. Math. Soc.*, vol. 123(587). 194

LEGRAND, P. 2004, *Débruitage et interpolation par analyse de la régularité Hölderienne. Application à la modélisation du frottement pneumatique-chaussée.*, Theses, Ecole Centrale de Nantes (ECN) ; Université de Nantes. URL https://tel.archives-ouvertes.fr/tel-00643450. 198

MEYER, Y. 1990, *Ondelettes et Opérateurs*, Hermann, Paris. 198

MEYER, Y. 1997, «Wavelets, Vibrations and Scalings», *American Mathematical Society, CRM Monograph Series*, vol. 9. 194, 201

SEURET, S. et J. L. VEHEL. 2002, «The local hölder function of a continuous function», *Comput. Harmon. Analysis*, vol. 13, nº 3, p. 263–276. 197

TRICOT, C. 1995, *Curves and Fractal Dimension*, Springer-Verlag. 194, 198

VEHEL, J. L. 1998, «Introduction to the multifractal analysis of images», *Fractal Image Encoding and Analysis*. 194

VEHEL, J. L. et K. DAOUDI. 1996, «Generalized ifs for signal processing», *IEEE DSP workshop, Loen, Norway*. 194

VEHEL, J. L. et P. LEGRAND. 2004, «Signal and Image Processing with FracLab», *FRACTAL 2004, Complexity and Fractals in Nature, 8th International Multidisciplinary Conference, Vancouver*. 217

VEHEL, J. L. et E. LUTTON. 2001, «Evolutionary signal enhancement based on hölder regularity analysis», *EVOIASP2001, LNCS 2038*. 196

VEHEL, J. L. et S. SEURET. 2004, «The 2-microlocal Formalism», *Fractal geometry and Applications: A jubilee of Benoit Mandelbrot, Proc. Sympos. Pure Math.*, vol. 72-2, p. 153–215. 194, 199

Chapter 10

Theoretical comparison of the DFA and variants for the estimation of the Hurst exponent

This chapter is related to the PhD thesis of Bastien Berthelot (PhD student CIFRE, THALES). A part of this chapter has been submitted to the journal Digital Processing, Elsevier. Other parts of this work were presented at the conferences Gretsi 2019 and Eusipco 2019. Work carried out with Bastien Berthelot and Eric Grivel.

Contents

Abstract

The detrended fluctuation analysis (DFA) and its variants such as the detrended moving average (DMA), the adaptive fractal analysis (AFA) and the regularized DFA (RDFA) are widely used to estimate the Hurst exponent. These methods are very popular as they do not require advanced skills in the field of signal processing and statistics while providing accurate results. For the last years, a great deal of interest has been paid to compare them and to better understand their behaviors from a mathematical point of view. In this chapter, our contribution is threefold: We first propose another variant avoiding the discontinuities between consecutive local trends of the DFA by *a priori* constraining them to be continuous. Then, we show that in all these approaches, the square of the fluctuation function can be presented in a similar matrix form. Additionally, its statistical mean can be expressed from the autocorrelation function of the process and consequently from its power spectral density, without any approximation, if the process under study is assumed to be wide-sense stationary (w.s.s). Finally, using the above representation, the differences between the above-mentioned methods are highlighted in terms of band-pass filtering abilities.

10.1 Introduction

In addition to some features such as the power of the signal in certain frequency bands, the zero-crossing rate, the entropy or the multiscale entropy GAO et collab. [2015], the Hurst exponent can be used to characterize the data and to classify them. Denoted H MANDELBROT et VAN NESS [1968], it makes it possible to evaluate the long-range dependence (LRD). Thus, a process is said to have LRD if $0.5 < H < 1$. When $0 < H < 0.5$, the process is anti-persistent. There are also some particular cases: H is respectively equal to 0.5, 0 and -0.5 for a Brownian noise, a pink noise and a white noise. The reader can refer to PIPIRAS et TAQQU [2017] for the study of LRD.

The approaches estimating the Hurst coefficient can be sorted in two main families:

1. The frequency-domain estimators: they consist in studying the power spectral density (PSD) of the time series SUN [2007] and include the local Whittle method, the periodogram method, the empirical mode decomposition (EMD) RILLING et collab. [2005], the fractional Fourier transform SUN [2007], the wavelet-based method ABRY et collab. [2003] and the semi-parametric method BARDET [2000] MOULINES et collab. [2007]. Different comparative studies have also been presented. See for instance ESPOSTI et SIGNORINI [2006].

2. The time-domain estimators: they include the so-called rescaled range analysis, the aggregated variance method, the absolute-value method and the variance-of-residuals method. See TAQQU et collab. [1995] and TAQQU et TEVEROVSKY [1996] for more details.

The estimation of the Hurst coefficient has been studied by the community of mathematicians and experts in statistical signal processing. However, as it is also employed by researchers from other fields, such as physiologists, we propose to focus on methods that do not necessarily require advanced skills in signal processing. Thus, to estimate the Hurst exponent of a pure mono-fractal time series or of non-stationary time series, the fluctuation analysis (FA) PENG et collab. [1992] was first proposed early in the 90ies. After integrating the signal leading to the new sequence y_{int}, the quantity denoted as $F_{FA}(l)$ and equal to $\sqrt{< (y_{int}(i+l) - y_{int}(i))^2 >}$, where $< . >$ denotes the temporal mean, is computed for different values of the lag l. As $F_{FA}(l)$ is proportional to l^H PENG et collab. [1992], $\log(F_{FA}(l))$ is represented as a function of $\log(l)$ to estimate H in the least-square (LS) sense[1]. It should

[1]In this chapter, log denotes the logarithm to the base 10.

be noted that the link between $F_{FA}(l)$ and the normalized covariance function was studied in CARPENA et collab. [2015]. Then, since the FA was sensitive to non-stationarities, the detrended fluctuation analysis (DFA) PENG et collab. [1994] was developed and operates with the following steps: after integration, the signal is split into segments. Using a LS criterion, local trends of size N_{DFA} are deduced. The resulting piecewise linear trend is then subtracted to the whole integrated signal. The power of the residual corresponding to the square of the fluctuation function is computed for different segment lengths. Then, the log-log representation should correspond to a straight line with a slope, denoted as α and called the scaling exponent. As the first step of the DFA is an integration of the signal, α is related to the Hurst coefficient as follows: $\alpha = H + 1$. Some other comments can be done:
Firstly, H characterizes the way the autocorrelation function r_τ of the signal decreases. When $0.5 < H < 1$, the autocorrelation function exhibits persistence and decays as a function $\tau^{-\gamma}$, with $\gamma = 2 - 2H$.
Secondly, there are many other ways to obtain the global trend of a signal KIM et collab. [2009]. This is the reason why several variants of the DFA exist:

1. Instead of using a linear trend for each segment, polynomials with a degree larger than 1 can be considered. This leads to the quadratic DFA (DFA_2), the cubic DFA (DFA_3), etc. These methods are called higher-order DFA.

2. Discontinuities between local trends can be debatable. Various authors aimed at addressing this issue. Let us focus our attention on two of them: On the one hand, Tarvainen TARVAINEN et collab. [2002] suggests using a regularized LS criterion to reduce this phenomenon. The trend extraction is similar to the so-called Hodrick–Prescott filtering, widely used in econometrics HODRICK et PRESCOTT. [1997]. In the following, this method is labeled RDFA for regularized DFA. On the other hand, the authors in RILEY et collab. [2012] proposed the adaptive fractal analysis (AFA) to *a posteriori* correct the local discontinuities.

3. The detrended moving average (DMA) is based on a low-pass filtering of the whole integrated signal in order to obtain the trend. In its standard version, the filter has a finite impulse response (FIR) of length N_{DMA} ALESSIO et collab. [2002]. In economics, this way to deduce the trend is known as "moving average filtering" OSBORNE [1995]. Using either finite-impulse-response filters or infinite-impulse-response filters leads to variants known as the simple moving average, including the backward moving average and the centered moving average (CDMA), and the weighted moving average of order l, labeled as WDMA-l, as well as the weighted centered detrended moving average (WCDMA) XU et collab. [2005].

These variants are nonlinear dynamical system analysis techniques NAYAK et collab. [2018]. Unlike the other approaches based on wavelets or the local Whittle which can outperform them, the DFA and its variants have the advantage of *a priori* not requiring advanced skills in statistical signal processing because they are based on regressions and linear filtering. This is one of the main reasons of their popularity BALLJEKAR et PATIL [2012]; KANTELHARDT et collab. [2001]; MERT et AKAN [2014]; NAVARRO et collab. [2011]; PRANATA et collab. [2017]; RAVELO-GARCIA et collab. [2014]; SANYAL et collab. [2015]. This corresponds to a trade-off between performance, computational cost and simplicity of implementation and use. During the last years, the main contributions dealing with the approaches have been done on four aspects: providing extensions or generalizations of the algorithms ARIANOS et collab. [2011], addressing the cases of multifractal time series KANTELHARDT et collab. [2002], developing fast versions TSUJIMOTO et collab. [2016] TSUJIMOTO et collab. [2017], proposing mathematical analysis to better understand their behaviors KANTELHARDT et collab. [2001] HOLL et collab. [2016]; KIYONO [2015, 2017]; SHAO et collab. [2012].

In HOLL et KANTZ [2015], a relation between the fluctuation function and an estimation of the normalized autocorrelation function of the signal was given, by assuming that the process was w.s.s and ergodic and by making some approximations. The single-frequency responses of the DFA and the higher-order DFA KIYONO [2015] as well as the centered DMA KIYONO [2017] are analyzed. The authors concluded that for stochastic processes whose PSD is a function of the frequency of the form $f^{-\beta}$, using the higher-order DFA is convenient to estimate α as long as $\alpha = \frac{\beta+1}{2}$.

In this chapter, the contribution is threefold: firstly, we suggest studying another variant of the DFA. More particularly, we propose to model the global trend of the integrated signal by assuming that the consecutive local trends are continuous. The estimations of the parameters of the local trends are based on a constrained LS criterion. This method is called CDFA in the remainder of this chapter. Then, a theoretical comparison between the ways to deduce the square of the so-called fluctuation function with the DFA and its variants is proposed. Its statistical mean is then expressed from the autocorrelation function of the process when it is assumed to be w.s.s. It can be also expressed from the PSD. Therefore, the matrix formulation we propose makes it possible to analyze all the methods from a filtering point of view and to highlight the differences between them. Although N_{DFA} corresponds to the local-trend length for the DFA, the RDFA, the AFA and the CDFA and N_{DMA} corresponds to the filter order for the DMA, our purpose is to study its influence. In the following, $N_{DFA} = N_{DMA} = N$.
The remainder of the chapter is organized as follows: In section 2, the main steps of the approaches are recalled before expressing the square of the fluctuation function in a uniform matrix way. Section 3 provides a comparative analysis. In section 4, simulations are presented.

10.2 Presentation and comparison of methods based on trend extraction

After giving the general steps of the DFA and its variants and providing some notations, this section deals with a uniform way to express the trend vector with the different variants of the DFA.

10.2.1 General steps of these approaches

Let us consider M consecutive samples $\{y(m)\}_{m=1,\ldots,M}$ of the signal. The DFA and its variants are defined by the following four steps PENG et collab. [1994] ALESSIO et collab. [2002] XU et collab. [2005]:

- **Step 1**. The so-called profile

$$y_{int}(m) = \sum_{i=1}^{m} (y(i) - \mu_y)$$

  is first computed, where $\mu_y = \frac{1}{M}\sum_{m=1}^{M} y(m)$ is the mean of y.

- **Step 2**. The trend of the profile is estimated. This step will be detailed in the next subsections below.

- **Step 3**. The resulting trend is subtracted to the profile and the square root of the residual power, $F_\bullet(N)$, is computed, where $\bullet$ denotes the method.

- **Step 4**. Steps 2 and 3 are repeated for different values of N. At this stage, as $F_\bullet(N) \propto N^\alpha$ [PENG et collab., 1992], $\log(F_\bullet(N))$ is plotted as a linear function of $\log(N)$.

- **Step 5.** The final step is to search a straight line fitting the log-log representation. The quantity α is hence estimated in the LS sense as its slope.

The approaches differ in the way of deducing the trend. In the following, let us present each of them and express the trend vector in a matrix form.

10.2.2 Notations

Some notations that will be useful in what follows are listed below:

- $A(i:j,k:l)$ is the part of of the matrix A corresponding to the elements belonging to the rows i to j and to the columns k to l .
- $\mathbb{1}_{j\times k}$ and $\mathbb{0}_{j\times k}$ are matrices of size $j \times k$ filled with 1s and 0s respectively.
- $diag([.],l)$ is a matrix whose l^{th} diagonal is equal to $[.]$. Thus, $I_j = diag(\mathbb{1}_{1\times j}, 0)$ is the identity matrix of size j. $diag(\mathbb{1}_{1\times N-1}, 1)$ is the square matrix of size N whose 1^{st} sub-diagonal above the main one has its elements equal to 1.
- $J_j = I_j - \frac{1}{j}\mathbb{1}_{j\times j}$.
- T_l is a $N \times 1$ vector storing the values of the l^{th} local trend $t_l(n)$.
- Y and Y_{int} are two column vectors storing respectively the samples $\{y(n)\}_{n=1,...,M}$ and $\{y_{int}(n)\}_{n=1,...,M}$. This leads to :

$$Y_{int} = [y_{int}(1), y_{int}(2), ..., y_{int}(M)]^T = \mathrm{H}_M J_M Y \tag{10.1}$$

  with $H_M = \sum_{r=0}^{M-1} diag(\mathbb{1}_{1\times M-r}, -r)$ a lower triangular matrix filled with 1s.

- Depending on the approach used to estimate the trend of the profile, all the samples of the profile are not necessarily considered. In addition, some other transformations will be required to get the matrix form of the trend. Therefore, let us introduce the following matrix of size (j, M):

$$C_{j,k} = [\mathbb{0}_{j\times k} \quad I_j \quad \mathbb{0}_{j\times(M-(j+k))}] \tag{10.2}$$

  In this case, one can express the first LN elements of the vector Y_{int} as follows:

$$Y_{int}(1:LN) = [y_{int}(1), y_{int}(2), ..., y_{int}(LN)]^T = C_{LN,0}Y_{int} \underset{(10.1)}{=} C_{LN,0}\mathrm{H}_M J_M Y \tag{10.3}$$

- Finally, for the sake of simplicity, let us define $N' = \frac{N-1}{2}$.

10.2.3 Extraction of the trend vector

10.2.3.1 Matrix form of the vector trend with DFA

When dealing with the DFA, the profile is split into L non-overlapping segments of length N, denoted as $\{y_{int,l}(n)\}_{l=1,...,L}$ with $n \in [\![1; N]\!]$. As M is not necessarily a multiple of N, the last $M - LN$ samples of the profile are not used. In this case, the l^{th} local trend, corresponding to the trend $t_l(n)$ of the l^{th} segment $y_{int,l}(n)$, is modeled by a straight line $\forall l \in [\![1; L]\!]$ and $\forall n \in [\![1; N]\!]$:

$$t_l(n) = a_{l,1}[(l-1)N + n] + a_{l,0} \tag{10.4}$$

Then, $\forall l \in [\![1; L]\!]$, the parameter vector $\theta_l = \begin{bmatrix} a_{l,0} & a_{l,1} \end{bmatrix}^T$ is estimated in the LS sense from $\{y_{int,l}(n)\}_{n=1,...,N}$. The global trend T_{DFA} is then deduced by aggregating the local trends $\{T_l\}_{l=1,...,L}$. Using a vector form of (10.4) $\forall l \in [\![1; L]\!]$:

$$T_l = A_l \theta_l \tag{10.5}$$

where A_l is a $N \times 2$ matrix whose first column corresponds to a vector of 1s and whose second column is defined by the set of values $\{(l-1)N + n\}_{n=1,...,N}$.
By introducing the parameter vector $\Theta_{DFA} = \begin{bmatrix} \theta_1 \dots \theta_L \end{bmatrix}^T$ of size $2L \times 1$, and the $(LN \times 2L)$ matrix A_{DFA} which is block diagonal defined from the set of matrices $\{A_l\}_{l=1,...,L}$, the parameters of the local trends satisfy:

$$\underset{\Theta_{DFA}}{\arg\min} \left|\left| C_{LN,0} Y_{int} - A_{DFA} \Theta_{DFA} \right|\right|^2 \tag{10.6}$$

This leads to:

$$\hat{\Theta}_{DFA} = (A_{DFA}^T A_{DFA})^{-1} A_{DFA}^T C_{LN,0} Y_{int} \tag{10.7}$$

Then, the trend vector T_{DFA} can be deduced as follows:

$$\begin{aligned} T_{DFA} = A_{DFA} \hat{\Theta}_{DFA} &\underset{(10.7)}{=} A_{DFA} (A_{DFA}^T A_{DFA})^{-1} A_{DFA}^T C_{LN,0} Y_{int} \\ &\underset{(10.1)}{=} A_{DFA} (A_{DFA}^T A_{DFA})^{-1} A_{DFA}^T C_{LN,0} \mathrm{H}_M J_M Y \end{aligned} \tag{10.8}$$

10.2.3.2 Matrix form of the vector trend with DMA

When dealing with the DMA, known as "simple moving average" Xu et collab. [2005], the profile is low-pass filtered. Indeed, the impulse response of the filter is given by $h_{DMA}(n) = \frac{1}{N}$ for $n = 0, ..., N-1$. Due to its symmetry, it leads to a linear-phase filter with a constant group delay equal to $\frac{N-1}{2} \frac{1}{f_s}$, where f_s denotes the sampling frequency. As the trend has to be subtracted to the integrated signal, N is chosen odd in the following. Moreover, the frequency response satisfies:

$$H(f) = \begin{cases} \frac{1}{N} \frac{sin(\frac{\pi N f}{f_s})}{sin(\frac{\pi f}{f_s})} e^{-j \frac{\pi (N-1) f}{f_s}} & \text{if } f \neq 0 \\ 1 \text{ otherwise} \end{cases}$$

Note that $|H(f)| = 0$ when $\frac{\pi N f}{f_s} = k\pi$, or equivalently when $f = \frac{k f_s}{N}$ for $k = 1, ..., N-1$. This amounts to saying that the FIR filter is defined by zeros which are equal to $e^{j \frac{2\pi k}{N}}$ with $k = 1, ..., N-1$. When N increases, the width $\frac{2 f_s}{N}$ of the main lobe decreases. This corresponds to a low-pass filtering more and more selective when N increases.

The M samples of the profile are filtered. Instead of using a convolution at each time step, let us express the vector storing the filter output samples. This can be done by premultiplying Y_{int} by a filtering matrix M_{filt} defined as:

$$M_{filt} = \frac{1}{N} \sum_{r=0}^{N-1} diag(\mathbb{1}_{1 \times M-r}, -r) \tag{10.9}$$

In addition, the group delay corresponding to N' samples and induced by the filter has to be compensated. This can be done by introducing another pre-multiplication by the following $M \times M$ matrix:

$$M_{comp} = diag\Big(\mathbb{1}_{1 \times (M-N')}, N'\Big) \tag{10.10}$$

The resulting trend vector is equal to $M_{comp} M_{filt} Y_{int}$.

However, the last N' elements of this vector are equal to 0. In addition, due to the transient behavior of the filtering which corresponds to the first $N-1$ samples and the delay compensation introduced above, the first N' elements of the current trend vector should not be taken into account. For the above reasons, only a vector of size $M - N + 1$ should be considered. This amounts to adding another pre-multiplication by the matrix $C_{M-N+1,N'}$. Therefore, the trend vector satisfies:

$$T_{DMA} = C_{M-N+1,N'} M_{comp} M_{filt} Y_{int} \underset{(10.1)}{=} C_{M-N+1,N'} M_{comp} M_{filt} \mathrm{H}_M J_M Y \tag{10.11}$$

10.2.3.3 Matrix form of the vector trend with the RDFA

In the standard DFA PENG et collab. [1994], given the way the local trends are defined, if $x_{l-1}(N+1)$ and $x_l(0)$ were also considered and defined according to (10.4) $\forall l \in [1; L-1]$, discontinuities between the local trends would necessarily appear:

$$x_{l-1}(N+1) \neq x_l(1) \text{ or } x_{l-1}(N) \neq x_l(0) \tag{10.12}$$

In the frequency domain, they would lead to resonances in the spectrum at normalized frequencies multiple to $\frac{1}{N}$. The spectrum of the whole profile trend would exhibit frequency features that are not due to the profile itself but to the way the local trends are obtained.
To address the above problem, the regularized DFA (RDFA) TARVAINEN et collab. [2002] or equivalently the Hodrick–Prescott Filtering was proposed. It consists in deducing the trend $t(n)$, which concatenates the local trends by minimizing the following criterion:

$$\frac{1}{2}\sum_n (y_{int}(n) - t(n))^2 + \lambda \sum_n (t(n-1) - 2x(n) + t(n+1))^2 \tag{10.13}$$

where λ is a regularization parameter that penalizes the energy of the residual by the energy of the second-order difference of the time series. This makes it possible to induce smoothness. In KIM et collab. [2009], it is said that:

$$\frac{||y_{int} - t||^2}{||y_{int|}|^2} \leq \frac{32\lambda}{1 + 32\lambda} \tag{10.14}$$

In TARVAINEN et collab. [2002], Tarvainen *et al.* took into account the Tikhonov regularization. It consists in minimizing the following criterion:

$$\underset{\Theta_{DFA}}{\arg\min}\{||L_1(C_{LN,0}Y_{int} - A_{DFA}\Theta_{DFA})||^2 + \lambda^2 ||L_2(\Theta_{DFA} - \Theta^*_{DFA})||^2\} \tag{10.15}$$

where Θ^*_{DFA} is the initial guess for the solution, $L_1^T L_1$ is a positive definite matrix, L_2 can be expressed from the discrete approximation D_d of the d^{th} derivative operator. One has for instance:

$$D_2 = \begin{bmatrix} 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 & 1 \end{bmatrix} \tag{10.16}$$

The above criterion (10.15) can be rewritten as follows:

$$\underset{\Theta_{DFA}}{\arg\min} \left\{ \left|\left| \begin{bmatrix} L_1 A_{DFA} \\ \lambda L_2 \end{bmatrix} \Theta_{DFA} - \begin{bmatrix} L_1 C_{LN,0} Y_{int} \\ \lambda L_2 \Theta^*_{DFA} \end{bmatrix} \right|\right|^2 \right\} \tag{10.17}$$

Introducing $A_{DFA}^{ext} = \begin{bmatrix} A_{DFA} \\ I_{2L} \end{bmatrix}$, $Y_{int}^{ext} = \begin{bmatrix} C_{LN,0}Y_{int} \\ \Theta_{DFA}^{*} \end{bmatrix}$ and $L^{ext} = \begin{bmatrix} L_1 & 0_{LN\times 2L} \\ 0_{2L\times LN} & \lambda L_2 \end{bmatrix}$, this amounts to considering the following criterion:

$$\underset{\Theta_{DFA}}{\arg\min} \left\{ ||L^{ext} A_{DFA}^{ext} \Theta_{DFA} - L^{ext} Y_{int}^{ext}||^2 \right\} \tag{10.18}$$

for which the solution, labeled as RDFA for "regularized" DFA, is given by:

$$\begin{aligned} \hat{\Theta}_{RDFA} &= \left({A_{DFA}^{ext}}^T L^{ext^T} L^{ext} A_{DFA}^{ext} \right)^{-1} {A_{DFA}^{ext}}^T L^{ext^T} L^{ext} Y_{int}^{ext} \\ &= \left(A_{DFA}^T L_1^T L_1 A_{DFA} + \lambda^2 L_2^T L_2 \right)^{-1} \times \left(A_{DFA}^T L_1^T L_1 C Y_{int} + \lambda^2 L_2^T L_2 \Theta_{DFA}^{*} \right) \end{aligned} \tag{10.19}$$

In the work developed by Tarvainen TARVAINEN et collab. [2002], $L_1 = I_{LN}$, $L_2 = D_2 A_{DFA}$ with D_2 a matrix of size $LN \times LN$ and $\Theta_{DFA}^{*} = 0_{2L\times 1}$. This hence leads to the trend vector defined by:

$$\begin{aligned} T_{RDFA} &= A_{DFA}\hat{\Theta}_{RDFA} \\ &\underset{(10.19)}{=} A_{DFA} \left(A_{DFA}^T A_{DFA} + \lambda^2 A_{DFA}^T D_2^T D_2 A_{DFA} \right)^{-1} A_{DFA}^T C_{LN,0} Y_{int} \\ &= A_{DFA} \left(A_{DFA}^T \left(I_{NL} + \lambda^2 D_2^T D_2 \right) A_{DFA} \right)^{-1} A_{DFA}^T C_{LN,0} Y_{int} \\ &\underset{(10.1)}{=} A_{DFA} \left(A_{DFA}^T \left(I_{NL} + \lambda^2 D_2^T D_2 \right) A_{DFA} \right)^{-1} A_{DFA}^T C_{LN,0} \mathrm{H}_M J_M Y \end{aligned} \tag{10.20}$$

10.2.3.4 Matrix form of the trend vector with the AFA

The adaptive fractal analysis (AFA) is another variant of the DFA which aims at reducing the discontinuities between local trends. It operates with the following steps RILEY et collab. [2012]: the integrated signal is split into segments with an overlap equal to $\frac{N+1}{2}$ samples and N odd. Then, a local trend modeled by a k^{th}-degree polynomial is estimated. Usually k is set at 1. To avoid jumps or discontinuities around the boundaries of consecutive segments, the global trend is deduced by *a posteriori* patching together local polynomials fitted to the time series. In the following we suggest introducing the AFA differently. Indeed, as depicted in Fig. 10.1, the global trend can be deduced as follows: a first global trend is computed by using a variant of the DFA where a local trend starts when the previous local trend stops. A second global trend is computed similarly but on a shifted version of the profile. The final trend is deduced as a weighted sum of the two above-mentioned global trends. Before giving the matrix form of the AFA, let us first analyze how the matrix form of the trend deduced with the DFA is modified when two local trends overlap with one sample. This analysis will be useful to deduce the matrix form of the trend obtained with the AFA.

10.2.3.4.1 Matrix form of the DFA when two local trends overlap with one sample. In (10.4), the vector T_l storing the samples of the l^{th} local trend are expressed from the samples of the profile $\{y_{int}(l-1)N + n\}_{n=1,\dots,N}$. In the following, let us assume that there is an overlap of one sample between two consecutive local trends.
In this case, by introducing the floor function $\lfloor . \rfloor$, the number of local trends is modified as follows:

$$L' = \left\lfloor \frac{M-N}{N-1} \right\rfloor + 1 \tag{10.21}$$

Consequently, the number of samples in the profile that can be considered becomes equal to $N + (L'-1)(N-1)$. Then, a new vector of data is created, in which the $l(N-1)+N^{th}$ sample of the profile is reproduced, with $l = 0, ..., L'-1$:

$$\begin{aligned} &Y_{int,ext} = \\ &\left[y_{int}(1) \dots y_{int}(N)\ y_{int}(N) \dots y_{int}(2N-1)\ y_{int}(2N-1) \dots y_{int}((L'-1)(N-1)+N) \right]^T \end{aligned} \tag{10.22}$$

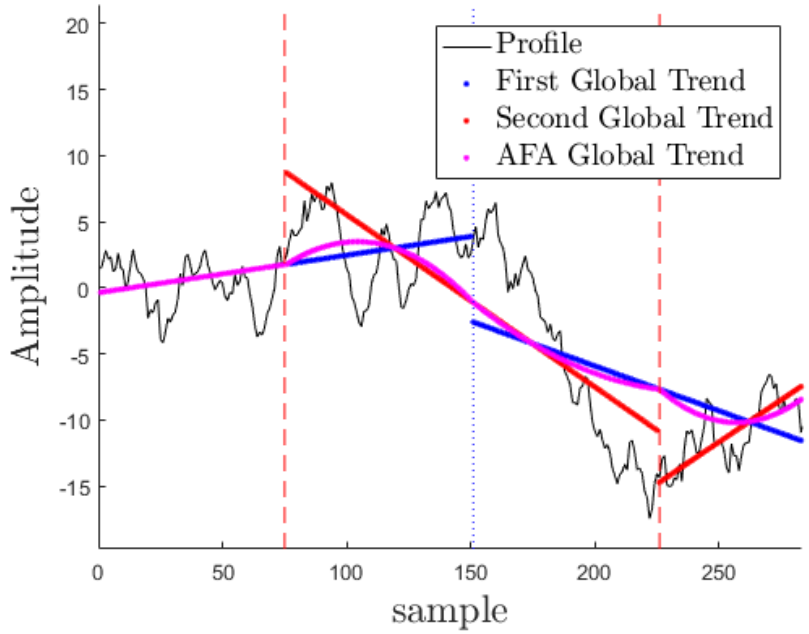


Figure 10.1 – Global trends in the AFA approach

To express the latter from Y_{int} and Y, similarly to (10.2) for the DFA, let us define $\overline{C}_{j,k}(N)$ of size (j, M) as follows:

$$\overline{C}_{j,k}(N) = [\mathbb{0}_{j\times k} \quad \overline{I}_j(N) \quad \mathbb{0}_{j\times M-(k+N+(\lfloor \frac{j}{N} \rfloor -1)(N-1)}] \tag{10.23}$$

where $\overline{I}_j(N)$ has $N + (\left\lfloor \frac{j}{N} \right\rfloor - 1)(N-1)$ columns and

$$\overline{I}_j((l-1)N + (1:N), (l-1)(N-1) + (1:N)) = I_N$$

for $l = 1, \ldots, \left\lfloor \frac{j}{N} \right\rfloor$.
Then, one has :

$$Y_{int,ext} = \overline{C}_{L'N,0}(N)Y_{int} \underset{(10.1)}{=} \overline{C}_{L'N,0}(N)H_M J_M Y \tag{10.24}$$

Then, similarly to (10.8), the trend vector can be deduced as follows:

$$\begin{aligned} T_{DFA1,ext} &= \overline{A}_{DFA1}(\overline{A}_{DFA1}^T \overline{A}_{DFA1})^{-1}\overline{A}_{DFA1}^T Y_{int,ext} \\ &\underset{(10.24)}{=} \overline{A}_{DFA1}(\overline{A}_{DFA1}^T \overline{A}_{DFA1})^{-1}\overline{A}_{DFA1}^T \overline{C}_{L'N,0}(N)H_M J_M Y \end{aligned} \tag{10.25}$$

where the matrix $\overline{A}_{DFA1}$ of size $(L'N \times 2L')$ is a block diagonal matrix defined from the set of matrices $\{\overline{A}_l\}_{l=1,\ldots,L'}$ which are matrices of size $N \times 2$ whose first column corresponds to a vector of 1s and whose second column is defined by the set of values $\{(l-1)(N-1) + n\}_{n=1,\ldots,N}$.
Since redundant samples appear in the resulting vector trend $T_{DFA1,ext}$, they have to be removed. For this purpose, let us introduce the matrix $\underline{C}_j(N)$ of size $(\left\lfloor \frac{j}{N} \right\rfloor -1)(N-1)+N\times j$, with

$$\underline{C}_j((l-1)(N-1) + (1:N-1), (l-1)N + (1:N-1) = I_{N-1} \tag{10.26}$$

for $l = 1, \ldots, \left\lfloor \frac{j}{N} \right\rfloor$. In this case, the trend T_{DFA1} is equal to:

$$\begin{aligned} T_{DFA1} &= \underline{C}_{L'N}(N)T_{DFA1,ext} \\ &\underset{(10.25)}{=} \underline{C}_{L'N}(N)\overline{A}_{DFA1}(\overline{A}_{DFA1}T\overline{A}_{DFA1})^{-1}\overline{A}_{DFA1}^T C_{L'N,0}(N)H_M J_M Y \end{aligned} \tag{10.27}$$

10.2.3.4.2 Deducing the expression of the vector trend with the AFA. As mentioned above, the trend estimated with the AFA approach can be seen as a weighed sum of two trends. The first one is the trend of the profile starting at its first sample whereas the second is the trend of the profile when the latter starts at the N^{th} sample. Therefore, the trend vector T_{AFA} of size $(L'-1)(N-1)+N\times 1$ can be seen as the weighted sum of two trend vectors:

$$T_{AFA} = W_1 T_{DFA_1} + W_2 T_{DFA_2} \tag{10.28}$$

where T_{DFA_1} is given in (10.27), T_{DFA2} follows the same steps as T_{DFA1}, but it is computed from a the profile starting at the N^{th} sample. Therefore, the matrix $\overline{A}_{DFA2}$ is computed the same way as $\overline{A}_{DFA1}$, but the number of local trends is now equal to $L'-1$. In addition, taking into account the definition of the weights given in RILEY et collab. [2012], W_1 is a matrix of size $((L'-1)(N-1)+N\times(L'-1)(N-1)+N)$ filled with 0s, and whose main diagonal is defined for $l=1,\ldots,L'$ by:

$$\begin{cases} diag(W_1(1:N')) = \mathbb{1}_{1\times N'} \\ diag(W_1(L'-1)(N-1)+N-N') : L'-1)(N-1)+N) = \mathbb{1}_{1\times N'} \\ diag(W_1((l-1)N+(N'+2-l) : lN+(N'+1-l)) = w \end{cases} \tag{10.29}$$

with

$$w = \left[1, \frac{N'-1}{N'}, \ldots, -\frac{1}{N'}, 0, \frac{1}{N'}, \ldots, \frac{N'-1}{N'}, 1\right] \tag{10.30}$$

The matrix W_2 is the computed the same way, except that:

$$diag(W_2) = \mathbb{1}_{(L'-1)(N-1)+N} - diag(W_1) \tag{10.31}$$

As T_{DFA2} and T_{DFA1} do not have the same length, the first N' columns and the last N' column of W_2 are removed.
As a consequence, the trend vector T_{AFA} can be defined as:

$$\begin{aligned} T_{AFA} = \Big(& W_1 \underline{C}_{L'N}(N)\overline{A}_{DFA1}(\overline{A}_{DFA1}^T\overline{A}_{DFA1})^{-1}\overline{A}_{DFA1}^T\overline{C}_{L'N,0}(N) \\ & + W_2 \underline{C}_{(L'-1)N}(N)\overline{A}_{DFA2}(\overline{A}_{DFA2}^T\overline{A}_{DFA2})^{-1}\overline{A}_{DFA2}^T\overline{C}_{(L'-1)N,0}(N)\Big) H_M J_M Y \end{aligned} \tag{10.32}$$

10.2.3.5 Matrix form of the vector trend with the CDFA

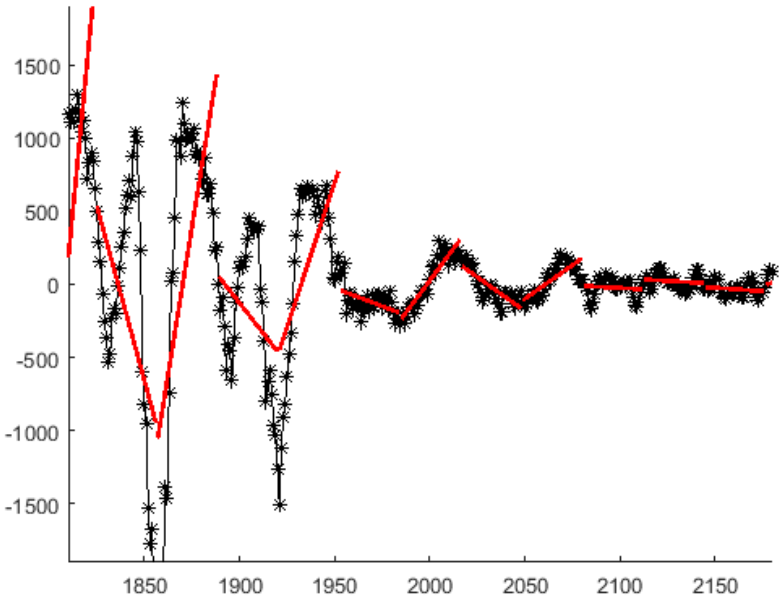
Instead of *a posteriori* correcting the discontinuities in the global trend of the data by using combinations of the consecutive local trends, we propose to model the global trend of the profile by assuming that the consecutive local trends are continuous (see Fig. 10.2). The estimations of the trend parameters are then based on a constrained LS criterion. In the following, let us detail the proposed variant.

10.2.3.5.1 Minimization approach for the CDFA. For the L segments under study, our purpose is to ensure continuity between the consecutive local trends $\forall l \in [1; L-1]$. Therefore, there are two possibilities that can be considered; either, one has:

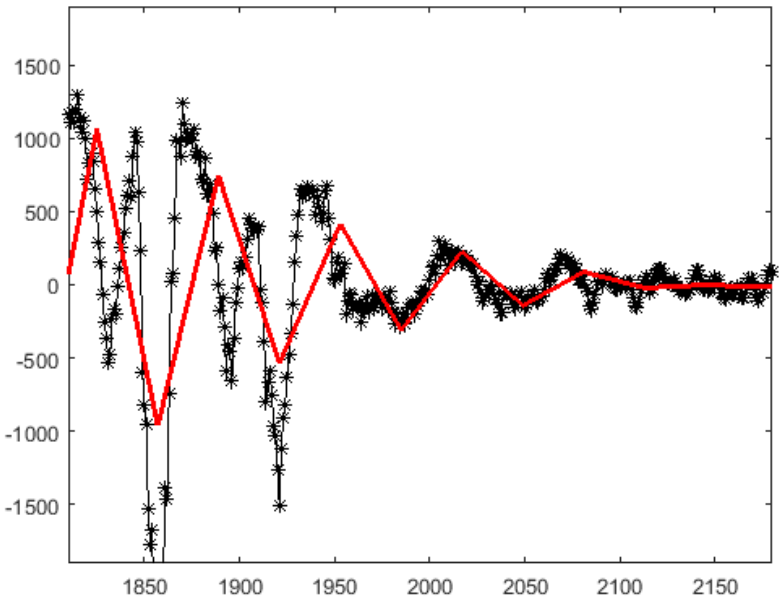
$$x_{l+1}(1) = x_l(N+1) \tag{10.33}$$

or

$$x_{l+1}(0) = x_l(N) \tag{10.34}$$



(a) Global trend with DFA PENG et collab. [1994]



(b) Global trend with our variant, the CDFA

Figure 10.2 – Global trends with DFA and CDFA

Given (10.4), defining the constraints (10.33) or (10.34) amounts to having $\forall l \in [1; L-1]$:

$$a_{l+1,0} = \beta(l)(a_{l,1} - a_{l+1,1}) + a_{l,0} \tag{10.35}$$

with $\beta(l) = lN + 1$ (resp. lN) if the first constraint (10.33) (resp. the second constraint (10.34)) is taken into account. Note that for both constraints, $\beta(l) - \beta(l-1) = N$. This remark will be useful for the parameter estimation step. Instead of using (10.35), one can consider $\forall l \in [1; L-1]$:

$$a_{l+1,0} = a_{1,0} + \sum_{j=1}^{l} \beta(j)(a_{j,1} - a_{j+1,1}) \tag{10.36}$$

The joint estimations of the $2L$ parameters $\{a_{l,1}\}_{l=1,...,L}$ and $\{a_{l,0}\}_{l=1,...,L}$ consists in minimizing the following criterion:

$$\begin{aligned} J(a_{1,1},..,a_{L,1},a_{1,0},..,a_{L,0}) &= \frac{1}{LN}\sum_{m=1}^{LN}(y_{int}(m)-t(m))^2 \\ &= \frac{1}{LN}\sum_{l=1}^{L}\sum_{n=1}^{N}(y_{int}((l-1)N+n)-t_l(n))^2 \end{aligned}$$

under $L-1$ constraints defined by (10.36). This amounts to minimizing with respect to $L+1$ variates this new criterion:

$$\begin{aligned} &J(a_{1,1},..,a_{L,1},a_{1,0}) = \sum_{n=1}^{N}(y_{int}(n)-a_{1,1}n-a_{1,0})^2 \\ &+\sum_{l=2}^{L}\sum_{n=1}^{N}[y_{int}((l-1)N+n)-a_{l,1}[(l-1)N+n]-a_{1,0}-\sum_{j=1}^{l-1}\beta(j)(a_{j,1}-a_{j+1,1})]^2 \end{aligned} \tag{10.37}$$

10.2.3.5.2 Matrix form of the CDFA approach. Let us introduce the $(L+1)\times 1$ column parameter vector $\Theta_{CDFA}=[a_{1,1},..,a_{L,1},a_{1,0}]^T$ and the $LN\times(L+1)$ matrix A_{CDFA} defined as follows:

$$A_{CDFA}(1:N,1:L+1)=\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 2 & & \vdots & & 1 \\ \vdots & & \vdots & & \vdots \\ N & & & & 1 \end{bmatrix} \tag{10.38}$$

and $\forall l\in[2;L-1]$:

$$\begin{aligned} &A_{CDFA}((l-1)\times N+1:lN,1:L+1)= \\ &\begin{bmatrix} \beta(1) & N & \cdots & N & lN+1-\beta(l) & 0 & \cdots & 0 & 1 \\ \beta(1) & N & \cdots & N & lN+2-\beta(l) & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \beta(1) & \underbrace{N & \cdots & N}_{l-2} & (l+1)N-\beta(l) & \underbrace{0 & \cdots & 0}_{L-l} & 1 \end{bmatrix} \end{aligned} \tag{10.39}$$

The criterion introduced in (10.37) can be defined as follows:

$$\begin{aligned} J(a_{1,1},..,a_{L,1},a_{1,0}) &= \left|\left|C_{LN,0}Y_{int}-A_{CDFA}\Theta_{CDFA}\right|\right|^2 \\ &= [C_{LN,0}Y_{int}-A_{CDFA}\Theta_{CDFA}]^T[C_{LN,0}Y_{int}-A_{CDFA}\Theta_{CDFA}] \end{aligned} \tag{10.40}$$

Therefore, the estimate $\hat{\Theta}_{CDFA}$ satisfies:

$$\hat{\Theta}_{CDFA}=[A_{CDFA}^TA_{CDFA}]^{-1}A_{CDFA}^TC_{LN,0}Y_{int} \tag{10.41}$$

The trend vector $T_{CDFA}=A_{CDFA}\hat{\Theta}_{CDFA}$ can be expressed this way:

$$\begin{aligned} T_{CDFA} &\underset{(10.41)}{=} A_{CDFA}[A_{CDFA}^TA_{CDFA}]^{-1}A_{CDFA}^TC_{LN,0}Y_{int} \\ &\underset{(10.1)}{=} A_{CDFA}[A_{CDFA}^TA_{CDFA}]^{-1}A_{CDFA}^TC_{LN,0}\mathrm{H}_MJ_MY \end{aligned} \tag{10.42}$$

10.3 Comparative analysis

10.3.1 Towards a uniform expression of the residual power

In the above section, for any method, the vector trend $T_\bullet$, with $\bullet = _{DFA}$, $_{DMA}$, $_{RDFA}$, $_{AFA}$ or $_{CDFA}$, has been expressed as a function of the signal vector Y. The next step is to deduce the expression of the residual vector $R_\bullet$. To take into account the fact that the trend vector is not necessarily of the same size, the matrix $\underline{C}_\bullet$ is introduced. Thus, one has:

$$R_\bullet = \underline{C}_\bullet \mathrm{Y} - T_\bullet = B_\bullet Y \tag{10.43}$$

where $\underline{C}_\bullet = C_{LN,0}\mathrm{H}_M$ for the DFA, the CDFA and the RDFA, $\underline{C}_{DMA} = C_{M-N+1,N'}\mathrm{H}_M$ and $\underline{C}_{AFA} = C_{(L'-1)(N-1)+N,0}\mathrm{H}_M$.
In table 1, the expressions of the matrices $B_\bullet$ are summarized.

Table 10.1 – Summary of the expressions and sizes of the matrix $B_\bullet$

Methods	Definition of the matrix $B_\bullet$	Size $S_\bullet$ of the trend vector
DFA	$(I_{LN} - A_{DFA}(A_{DFA}^T A_{DFA})^{-1}A_{DFA}^T)C_{LN,0}\mathrm{H}_M J_M$	LN
DMA	$C_{M-N+1,N'}\,(I_M - M_{comp}M_{filt})\,\mathrm{H}_M J_M$	$M-N+1$
RDFA	$(I_{LN} - A_{DFA}(A_{DFA}^T(I_{NL} + \lambda^2 D_2^T D_2)A_{DFA})^{-1}A_{DFA}^T)C_{LN,0}\mathrm{H}_M J_M$	LN
AFA	$\Big(C_{(L'-1)(N-1)+N,0} - (W_1\underline{C}_{L'N}(N)\overline{A}_{DFA1}(\overline{A}_{DFA1}^T\overline{A}_{DFA1})^{-1}\overline{A}_{DFA1}^T + W_2\underline{C}_{(L'-1)N}(N)\overline{A}_{DFA2}(\overline{A}_{DFA2}^T\overline{A}_{DFA2})^{-1}\overline{A}_{DFA2}^T\overline{C}_{(L'-1)N,0}(N))\Big)H_M J_M$	$(L'-1)(N-1)+N$
CDFA	$(I_{LN} - A_{CDFA}(A_{CDFA}^T A_{CDFA})^{-1}A_{CDFA}^T)C_{LN,0}\mathrm{H}_M J_M$	LN

Then, given $S_\bullet$ the size of the trend vector and introducing $\Gamma_\bullet = \frac{1}{S_\bullet}B_\bullet^T B_\bullet$, the power of the residual $F_\bullet^2(N)$, also called the square of the fluctuation function, can be deduced as follows:

$$F_\bullet^2(N) = Tr\left(\Gamma_\bullet Y Y^T\right) \tag{10.44}$$

In the following, this formalism will be useful to first express the power of the residual from the autocorrelation function of the process under study and consequently from its PSD. Then, we will compare all the methods.

10.3.2 Link between the power of the residual and the PSD of the process

By taking advantage of the symmetry of $\Gamma_\bullet$, (10.44) becomes:

$$F_\bullet^2(N) = \sum_{k=1}^{S_\bullet}\Gamma_\bullet(k,k)y^2(k) + \sum_{r=1}^{S_\bullet-1}\sum_{k=1}^{S_\bullet-r}[\Gamma_\bullet(k,k+r) + \Gamma_\bullet(k+r,k)]y(k)y(k+r) \tag{10.45}$$

By assuming that y is w.s.s and taking the statistical mean of (10.45), one has:

$$E[F_\bullet^2(N)] = \sum_{r=-S_\bullet+1}^{S_\bullet-1} Tr(\Gamma_\bullet, r)R_{y,y}(r) \tag{10.46}$$

where $R_{y,y}(r)$ is the autocorrelation function of the process y and $Tr(\Gamma_\bullet, r)$ denotes the r^{th} diagonal of the matrix $\Gamma_\bullet$. As the correlation function for real signals is symmetric and by denoting $g_{\Gamma_\bullet}(r) = Tr(\Gamma_\bullet, r)$, the above equation can be expressed as result of a convolution:

$$E[F_\bullet^2(N)] = g_{\Gamma_\bullet} * R_{y,y}(\tau)|_{\tau=0} \tag{10.47}$$

Given the Wiener-Khintchine theorem and using the inverse Fourier transform (TF^{-1}), $E[F_\bullet^2(N)]$ can be expressed from the PSD of y, denoted as $S_{yy}(f)$:

$$\begin{aligned} E[F_\bullet^2(N)] &= TF^{-1}\Big((\sum_{r=-S_\bullet+1}^{S_\bullet-1} Tr(\Gamma_\bullet, r)e^{-j2\pi f_n r})S_{yy}(f)\Big)\Big|_{\tau=0} \\ &= TF^{-1}\Big(\Psi_\bullet(f)S_{yy}(f)\Big)\Big|_{\tau=0} \end{aligned} \tag{10.48}$$

In (10.48), $\Psi_\bullet(f) = \sum_{r=-S_\bullet+1}^{S_\bullet-1} Tr(\Gamma_\bullet, r)e^{-j2\pi f_n r}$ corresponds to the Fourier transform of the sequence $\{Tr(\Gamma_\bullet, r)\}_{r=-S_\bullet+1,\ldots,S_\bullet-1}$. Let us look at the properties of the latter: first of all, as it is real and even, $\Psi_\bullet(f)$ is necessarily real and even. Moreover, as $\Gamma_\bullet$ is a Gramian matrix since it is the product between $\frac{1}{\sqrt{S_\bullet}}B_\bullet$ and its transpose, the element $\Gamma_\bullet(i,j)$ located at the i^{th} row and the j^{th} column of $\Gamma_\bullet$ corresponds to the scalar product between the i^{th} and the j^{th} rows of $\frac{1}{\sqrt{S_\bullet}}B_\bullet$. Therefore, taking advantage of the properties of the scalar product, one has:

$$|\Gamma_\bullet(i,j)| \leq |\Gamma_\bullet(i,i)| \tag{10.49}$$

As a corollary, using the inequality (10.49), one has:

$$\begin{aligned} |Tr(\Gamma_\bullet, r)| &\leq \sum_{k=1}^{S_\bullet-r} |\Gamma_\bullet(k,k+r)| \leq \sum_{k=1}^{S_\bullet-r} \Gamma_\bullet(k,k) \\ &\leq \sum_{k=1}^{S_\bullet-1} \Gamma_\bullet(k,k) = Tr(\Gamma_\bullet, 0) = Tr(\Gamma_\bullet) \end{aligned}$$

In the above, note that $Tr(\Gamma_\bullet)$ corresponds to the square of the Froebenius norm of the matrix $\Gamma_\bullet$. It is necessarily positive and the maximum value of the sequence of the traces. As a consequence, the sequence can be seen as the convolution of a vector with its flipped version and its Fourier transform $\Psi_\bullet(f)$ is necessarily positive.
Therefore $\Psi_\bullet(f)S_{yy}(f)$ can be seen as the PSD of the signal y filtered by a filter whose transfer function $H_{filter,\bullet}(z)$ satisfies: $\Psi_\bullet(f) = |H_{filter,\bullet}(z)|^2_{z=exp(j\theta)}$, with $\theta = 2\pi f/f_s$ the normalized angular frequency. Consequently, we can conclude that $E[F_\bullet^2(N)]$ corresponds to the autocorrelation function of the filter output calculated for the lag equal to 0, *i.e.* the power of the filter output.

10.4 Simulation results

In this section, let us compare the performances of the DFA, the AFA, the CDFA, the RDFA and the DMA. The usual way would be to evaluate the performance of each approach by estimating the Hurst exponent of synthetic pure mono-fractal signals. This work provides another way of comparison, based on the filtering interpretation we introduced in the above section.

10.4.1 Comparative study based on the filtering interpretation

In this subsection, let us first compare the DFA, AFA, CDFA and DMA, before addressing the case of the RDFA. Taking advantage of section 10.3, let us compare the properties of $\Psi_\bullet(f)$ with $\bullet = _{DFA,AFA,CDFA,DMA}$.
As $\Psi_\bullet(f)$ *a priori* depends on N, let us study the influence of N for a given signal of length M. Using (10.48) and the expressions of $\Psi_\bullet(f)$ summarized in 10.3.1 as well as Fig. 10.3, the following comments can be made:

1. When $N = 3$, the filters associated with the methods DFA, DMA and AFA are high-pass whereas they become band-pass for larger values of N. When using the CDFA, the filter is always band-pass. The null frequency is always rejected, which is consistent with the purpose of detrending. According to the simulations we carried out, the orders of magnitude of $\Psi_{DFA}(0)$, $\Psi_{AFA}(0)$, $\Psi_{CDFA}(0)$ are equal to 10^{-16} whereas the one of $\Psi_{DMA}(0)$ is equal to 10^{-17}.

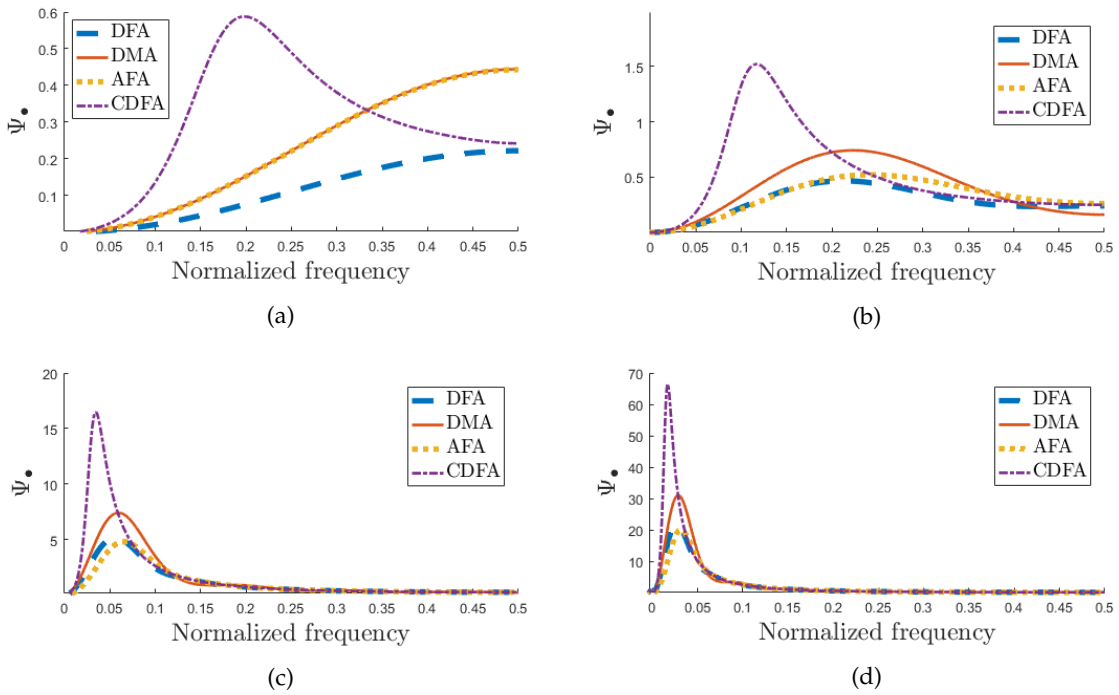


Figure 10.3 – Comparison between frequency responses of $\Psi_\bullet(f)$. (a) $N = 3$, (b) $N = 5$, (c) $N = 17$ (d) $N = 35$.

2. In the following, let $bw_\bullet$ be the -3 dB bandwidth[2] of the filter associated to $\Psi_\bullet(f)$. Fig. 10.4a and 10.4b respectively show the evolution of the resonance frequency and the bandwidth as a function of $\log(N)$. For every method, when looking at the right-hand side of Fig. 10.4b, the larger N, the smaller $bw_\bullet$ and the spikier the resonances of the frequency responses. The latter also move to low frequencies when N increases according to Fig. 10.4a.

3. For each value of N, the CDFA provides the spikiest and lowest resonance among all the studied approaches. See Fig. 10.3 and 10.4.

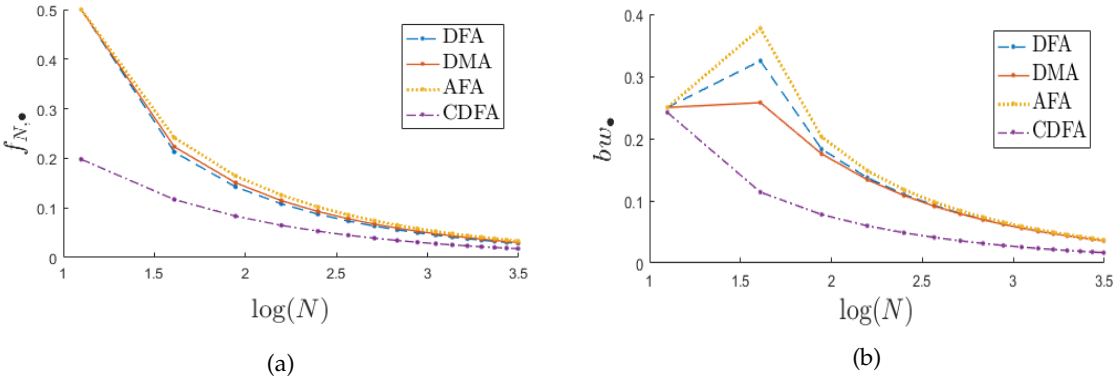


Figure 10.4 – Evolution as a function of $\log(N)$ of: (a) the resonance frequency of $\Psi_\bullet(f)$; (b) the frequency bandwidth (-3 dB) of $\Psi_\bullet(f)$.

4. As $\Psi_\bullet(f) = |H_{filter,\bullet}(z)|^2_{z=exp(j\theta)}$, measuring the difference between two frequency responses can be of interest. This can be done in many ways. In Fig. 10.5, the log

[2] It corresponds to the frequencies for which $10 \log \frac{\Psi_\bullet(f)}{\Psi_\bullet(f_{N,\bullet})} > -3$

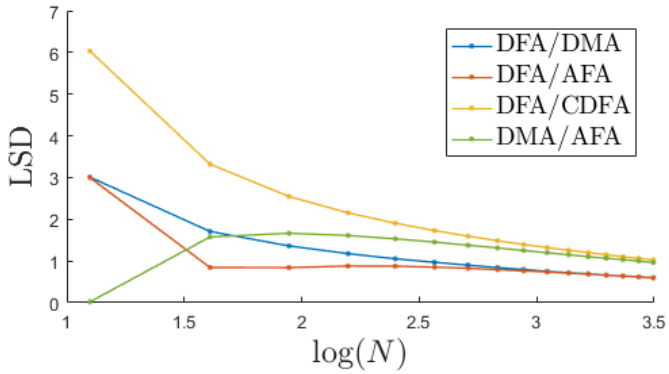


Figure 10.5 – Evolution as a function of $\log(N)$ of the log spectral distance (LSD) RABINER et JUANG [1993] between $\Psi_{DFA}(f)$ and $\Psi_\bullet(f)$, with $\bullet = DMA, AFA, CDFA$ or AFA.

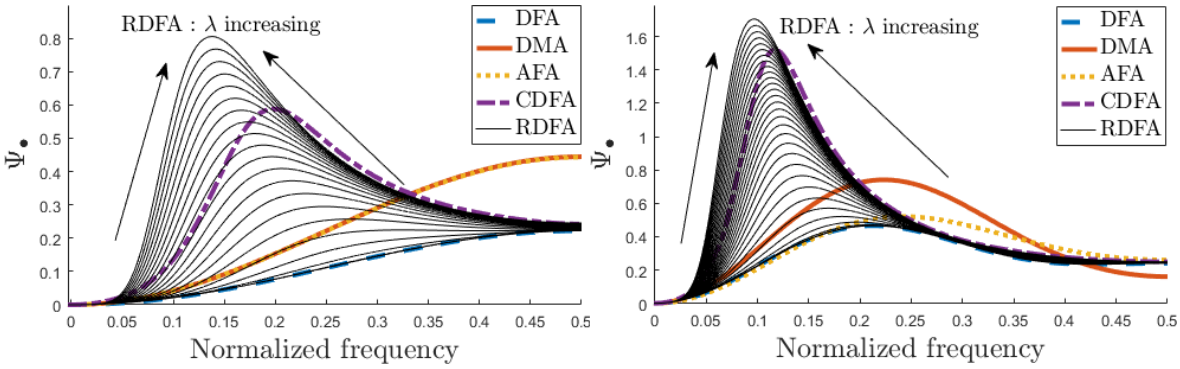
spectral distances (LSD) RABINER et JUANG [1993] between the frequency responses of the various approaches have been computed for different values of N. Whatever the two compared methods, the LSD tends to decrease when N increases. There is a value N_{min} from which the LSD between two filter responses remain more or less unchanged. From our simulations, this value of LSD has been chosen equal to 2. This induces that the difference between the fluctuation functions for all methods are negligible when $N \geq N_{min}$. Therefore, if the values of N are chosen so that $N \geq N_{min}$, the estimation of α should be more or less the same. According to our analysis, $N_{min} = 35$ (*i.e.* $\log(N_{min}) = 3.5$).

5. Our last comment is that the AFA approach is similar to the DMA for $N = 3$, but tends to act like the DFA when N increases.

Analysis of the influence of the regularization parameter in the RDFA approach: in the classical use of the RDFA approach, the regularization parameter λ is *a priori* defined by the practitioner. Let us study its influence on $\Psi_{RDFA}(f)$. The larger this parameter, the more the filter acts as a band-pass filter, and the spikier its resonance. See the three examples provided in Fig. 10.6a and 10.6b where the cases $N = 3$ and $N = 5$ are pictured.

To point out the differences between the RDFA and the other approaches, we consider $\lambda_{N,\bullet}$ defined as the value which minimize the LSD between $\Psi_{RDFA}(f)$ and $\Psi_\bullet(f)$. Fig. 10.7 shows the evolution of $\lambda_{N,\bullet}$ as a function of N for the DFA, CDFA, DMA and AFA. As expected, $\lambda_{N,DFA} = 0, \forall N$. Indeed, When λ is set at 0, this leads to the DFA. In addition, $\lambda_{N,AFA} = 0, \forall N$ except for $N = 3$, where $\lambda_{3,AFA} = \lambda_{3,DMA}$. In the case of the CDFA and DMA, the evolution of $\lambda_{N,\bullet}$ can be approximated by a linear function. The largest slope is obtained for the CDFA. To complete this result, Fig. 10.8 presents the value of the log spectral distance between $\Psi_\bullet(f)$ and $\Psi_{RDFA}(f)$ for $\lambda = \lambda_{N,\bullet}$. For the DFA, the LSD is necessarily equal to 0. For the DMA and the CDFA, the LSD decreases as $\log(N)$ increases.

To complete this section, let us show the differences between our work and the approaches presented in KIYONO [2015] and HOLL et KANTZ [2015]. In HOLL et KANTZ [2015], $F^2(N)$ is approximated by a weighted sum of the estimates of the correlation function $\hat{R}_{y,y}(r)$. Each term $\hat{R}_{y,y}(r)$ is weighted by $L_r(N) = \frac{1}{3N^2}(-r^3+3r^2N+(-3N^2+1)r+N^3-N)$ for $r \neq 0$ and $L_0(N) = \frac{N^2-1}{6N}$ whereas in our work and according to (10.46), $R_{y,y}(r)$ is weighted by $Tr(\Gamma_{DFA}, r)$. Therefore, by taking the Fourier transform of the set of weights $\{L_r(N)\}$, the frequency response of the DFA, $\hat{\Psi}_{DFA,[38]}(f)$, which would be obtained from HOLL et KANTZ [2015] can be defined and compared with $\Psi_{DFA}(f)$. In addition, we propose to define the frequency response $\hat{\Psi}_{DFA,[34]}(f)$ that can be deduced from Kiyono's work KIYONO [2015] by using our formalism. After some mathematical developments, we can



(a) (b)

Figure 10.6 – Evolution of the magnitude of the filter induced by the RDFA (black): (a) for $N = 3$, when λ increases from 0 to 2 with a step of 0.1; (b) for $N = 5$, when λ increases from 0 to 3 with a step of 0.1.

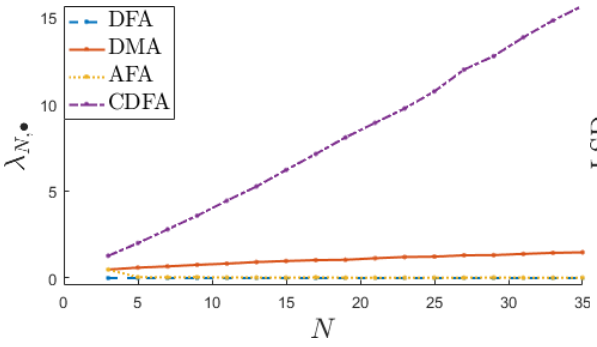


Figure 10.7 – Evolution of λ_N versus N.

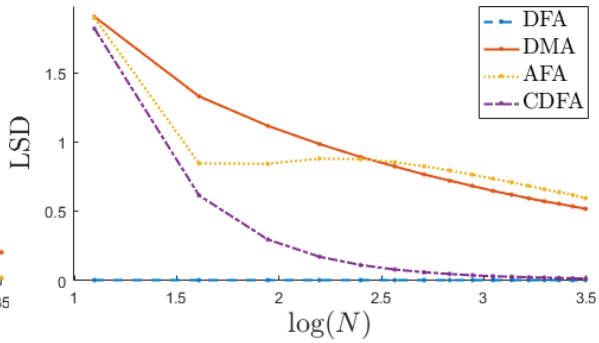


Figure 10.8 – LSDs vs. $\log(N)$ for $\lambda = \lambda_N$.

show that $\hat{\Psi}_{DFA,[\text{KIYONO, 2015}]}(\frac{k}{M}f_s)$ would be equal to:

$$\frac{2}{16\left(\frac{k}{M}\right)^6\pi^6N^4}\left(2\pi^4\left(\frac{k}{M}\right)^4N^4-4\pi^2\left(\frac{k}{M}\right)^2N^2-3+\left(3-2\pi^2\left(\frac{k}{M}\right)^2N^2\right)\cos\left(2\pi\left(\frac{k}{M}\right)N\right)+6\pi\left(\frac{k}{M}\right)N\sin\left(2\pi\left(\frac{k}{M}\right)N\right)\right)$$

for the frequency $\frac{k}{M}f_s$, $\forall k = \{1, ..., \frac{M-1}{2}\}$ with f_s the sampling frequency.

Given the comparison between the frequency responses in Fig. 10.9 for various values of N, we can conclude that Kiyono's approach KIYONO [2015] leads to frequency responses that are the closest to ours. However, for small values of N, the frequency responses deduced from HOLL et KANTZ [2015] and KIYONO [2015] are far from the ones we obtain with our approach where no approximation is made. There is a sharp difference, especially in high frequency. In the next section, we compare the behaviour of the DFA and the CDFA on mono-fractal processes.

10.4.2 Comparative study based on the estimation of the Hurst exponent of mono-fractal signals

The synthetic mono-fractal signals studied in this section consist of two types of signals. The first are white Gaussian noises known to have a prescribed value of the Hurst exponent H equal to -0.5. The second are Weierstrass functions (WEI) with prescribed values[3] of $H = 0.9$.

[3]Several simulations on processes characterized by different Hurst coefficients were conducted. However, for the sake of simplicity, only results are presented for $H = 0.9$ in the chapter.

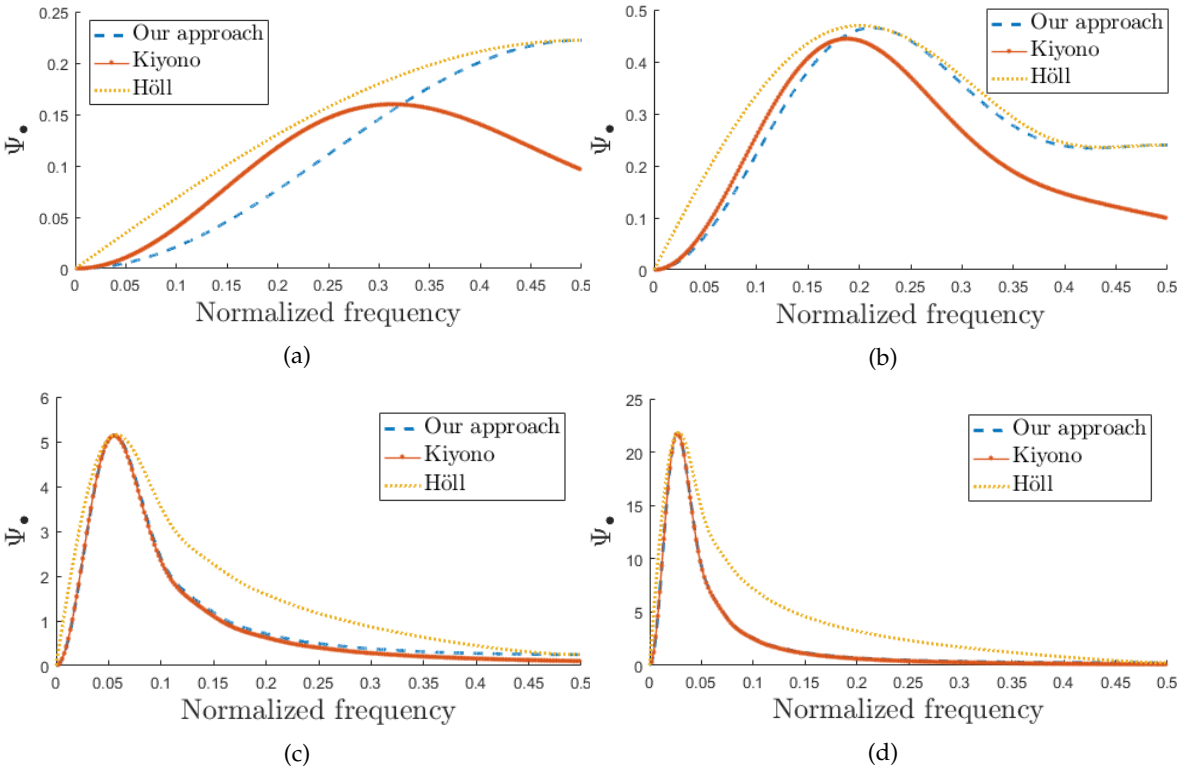


Figure 10.9 – Comparison between frequency responses. (a) $N = 3$, (b) $N = 5$, (c) $N = 17$ (d) $N = 35$.

10.4.2.1 Comparison between the DFA and the CDFA on 500 white noises

In Fig. 10.10, $\log(F(N))$ is represented as a function of $\log(N)$ for the DFA and the CDFA for one realization of a white noise. Two slopes α are computed. The first is based on the smallest values of N whereas the second is computed by using the largest. The slopes tend to be the same if large values of N ($N \geq N_{min}$) are used. It is coherent with the filtering analysis we did in the previous section, where we noticed that the LSD between $\Psi_{DFA}(f)$ and $\Psi_{CDFA}(f)$ becomes smaller and smaller as N increases.

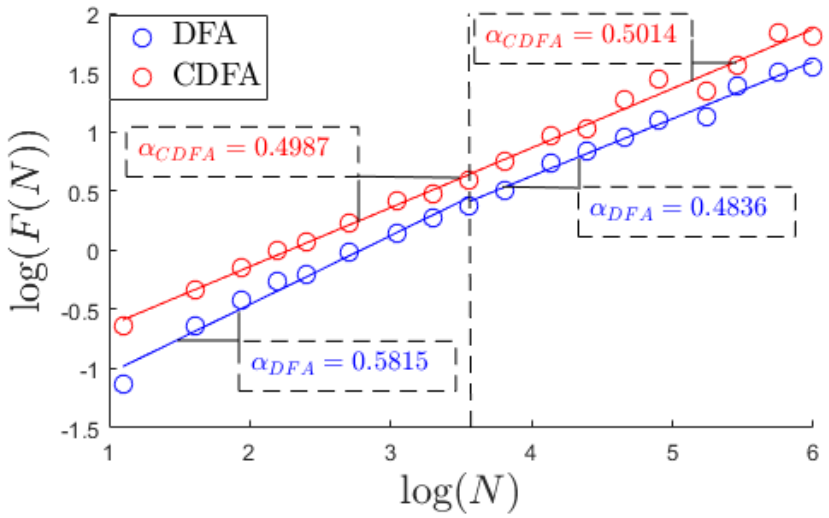


Figure 10.10 – Evolution of $\log(F(N))$ as a function of $\log(N)$ for both the DFA and the CDFA, in the case of one realization of a white noise. Theoretical expected value: $\alpha = 0.5$.

In Table 10.2, the mean and the variance obtained on 500 white noises are given. The

CDFA provides more accurate estimations of α for small values of N ($N \leq N_{min}$) and when large values of N ($N \geq N_{min}$) are considered. In addition, the difference between the estimation based on small and large values of N is smaller with the CDFA than with the DFA. Therefore, the CDFA is more reliable than the DFA[4].

	Mean	Variance	% err.
DFA	0.592	3.29×10^{-4}	18.4
CDFA	0.507	5.16×10^{-4}	1.40

	Mean	Variance	% err.
DFA	0.487	3.02×10^{-3}	2.60
CDFA	0.491	5.43×10^{-3}	1.80

Table 10.2 – Comparison of the mean and variance values of α for each approach, estimated on 500 white noises for different values of N: when $N \leq N_{min}$ (left) and $N \geq N_{min}$ (right). Theoretical expected value: $\alpha = 0.5$.

10.4.2.2 Comparative study on Weierstrass functions

Weierstrass functions (WEI) are continuous nowhere-differentiable functions HARDY [1916]. Each WEI is basically a sum of damped sines with increasing frequencies. As its Holder exponent is the same at each time instant, the value of the Hurst exponent is equal to the Holder exponent. In our experiments, 500 stochastic WEI are generated[5] with a prescribed value $H = 0.9$.

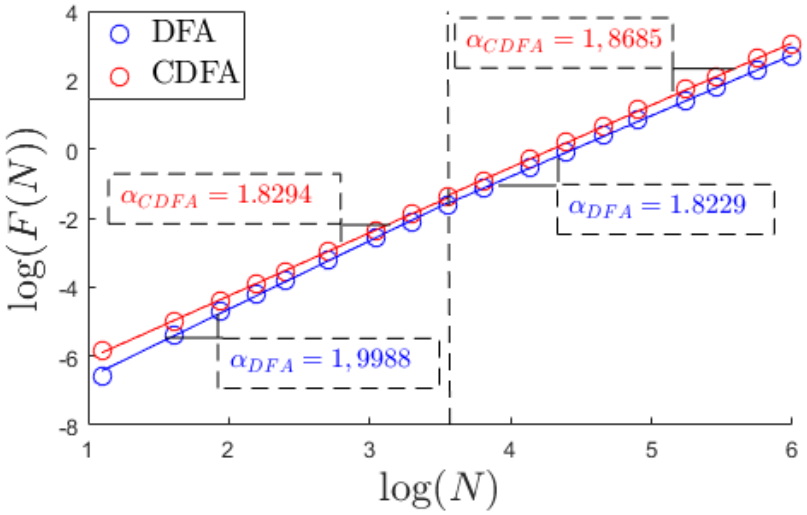


Figure 10.11 – Evolution of $\log(F(N))$ as a function of $\log(N)$ for both the DFA and the CDFA, in the case of one realization of a WEI process with $H = 0.9$. Theoretical expected value: $\alpha = 1.9$.

Fig. 10.11 shows the evolution of $\log(F(N))$ as a function of $\log(N)$ for one realization whereas Table 10.3 provides the mean values and the variances of α for the DFA and the CDFA.

The results we obtain can be explained by the following reasons: unlike a white process, the power of a WEI with $H = 0.9$ is rather located in low frequencies. This means that the values of $F(N)$ mainly depend on the properties of $\Psi_\bullet(f)$ in low frequencies. The filtering analysis we did in the previous section showed that the resonance of Ψ_{CDFA} is located in lower frequencies that the one of Ψ_{DFA}. In addition, the difference between them tends

[4] As an alternative, one could use the approach proposed in section 3.1 of KANTELHARDT et collab. [2001] correcting the values of the fluctuation function by multiplicating them with a corrective term for small values of N.

[5] This can be done by using the free Matlab Toolbox FracLab available at the following url: https://project.inria.fr/fraclab/. See also LEVY-VEHEL et LEGRAND [2004].

	Mean	Variance	% err.
DFA	1.997	4.51×10^{-3}	5.11
CDFA	1.832	1.18×10^{-2}	3.59

	Mean	Variance	% err.
DFA	1.816	1.29×10^{-2}	4.42
CDFA	1.868	2.53×10^{-2}	1.68

Table 10.3 – Mean and variance values of α for each approach, estimated on 500 WEI processes with $H = 0.9$ for different values of N: when $N \leq N_{min}$ (left) and $N \geq N_{min}$ (right). Theoretical expected value: $\alpha = 1.9$.

to be smaller when N increases. This explains that $\log(F_{CDFA}(N))$ is always larger than $\log(F_{DFA}(N))$. In addition, as the LSD between $\Psi_{DFA}(f)$ and $\Psi_{CDFA}(f)$ becomes smaller and smaller as N increases, the difference between the estimations of H tends to become smaller and smaller.

10.5 Conclusions and perspectives

In this chapter, the DFA and its variants, *i.e.* the DMA, the AFA, the RDFA and the one we propose where the trend is constrained to be continuous are compared. This is done by using a uniform way expressing the statistical mean of the square of the fluctuation function from the correlation function of the process without any approximation. Thanks to this approach, these methods can be interpreted as an *ad hoc* wavelet method. In addition, our framework makes it possible to better understand their differences in terms of behaviour, thanks to the filter-based analysis we propose. Finally, we show the differences between our analysis and previous studies done for the DFA where an expression of the square of the fluctuation function was obtained using some approximations.

References

ABRY, P., P. FLANDRIN, M. S. TAQQU et D. VEITCH. 2003, «Self-similarity and long-range dependence through the wavelet lens», *Theory and Applications of Long-range Dependence*, p. 527–556. 222

ALESSIO, E., A. CARBONE, G. CASTELLI et V. FRAPPIETRO. 2002, «Second-order moving average and scaling of stochastic time series», *The European Physical Journal B*, vol. 27, 2, p. 197–200. 223, 224

ARIANOS, S., A. CARBONE et C. TURK. 2011, «Self-similarity of higher-order moving averages», *Physical Review E*, vol. 84(4 Pt 2), p. 046 113. 223

BALLJEKAR, P. N. et H. A. PATIL. 2012, «A comparison of waveform fractal dimension techniques for voice pathology classification», *International Conference on Acoustics, Speech and Signal Processing*, p. 4461–4464. 223

BARDET, J.-M. 2000, «Testing for the presence of self-similarity of gaussian time series having stationary increments», *Journal of Time Series Analysis*, vol. 21, p. 497–515. 222

CARPENA, P., M. GOMEZ-EXTREMERA, C. CARRETERO-CAMPOS, P. BERNAOLA-GALVAN et A. V. CORONADO. 2015, «Spurious results of fluctuation analysis techniques in magnitude and sign correlations», *Entropy*, vol. 19, p. 2–61. 223

ESPOSTI, F. et M. G. SIGNORINI. 2006, «Evaluation of a blind method for the estimation of Hurst's exponent in time series», *European Signal Processing Conference*, p. 1–5. 222

GAO, J., J. HU, F. LIU et Y. CAO. 2015, «Multiscale entropy analysis of biological signals: A fundamental bi-scaling law», *Frontiers in Computational Neuroscience*, vol. 9, p. 1–64. 222

HARDY, G. 1916, «On weierstrass' non-differentiable functions», *Trans. of the American Mathematical Society*, vol. 17, (3), p. 301–325. 239

HODRICK, R. J. et E. C. PRESCOTT. 1997, «Postwar u.s. business cycles: An empirical investigation», *Journal of Money, Credit and Banking*, vol. 29, (1), p. 1–16. 223

HOLL, M. et H. KANTZ. 2015, «The relationship between the detrendend fluctuation analysis and the autocorrelation function of a signal», *The European Physical Journal B*, vol. 88, p. 327. 224, 236, 237

HOLL, M., H. KANTZ et Y. ZHOU. 2016, «Detrended fluctuation analysis and the difference between external drifts and intrinsic diffusionlike nonstationarity», *Physical Review E*, vol. 94, p. 042 201. 223

KANTELHARDT, J. W., E. KOSCIELNY-BUNDE, H. H. A. REGO, S. HAVLIN et A. BUNDE. 2001, «Detecting long-range correlations with detrended fluctuation analysis», *Physica A: Statistical Mechanics and its Applications*, vol. 295, (3-4), p. 441–454. 223, 239

KANTELHARDT, J. W., S. A. ZSCHIEGNER, E. KOSCIELNY-BUNDE, A. BUNDE, S. HAVLIN et H. E. STANLEY. 2002, «Multifractal detrended fluctuation analysis of nonstationary time series», *Physica A: Statistical Mechanics and its Applications*, vol. 316, (1-4), p. 87–114. 223

KIM, S.-J., K. KOH, S. BOYD et D. GORINEVSKY. 2009, «l_1 trend filtering», *SIAM Review*, vol. 51, (2), p. 339–360. 223, 227

KIYONO, K. 2015, «Establishing a direct connection between detrended fluctuation analysis and fourier analysis», *Physical Review E*, vol. 92, p. 042 925. 223, 224, 236, 237

KIYONO, K. 2017, «Theory and applications of detrending -operation -based fractal-scaling analysis», *International Conference on Noise and Fluctuations (ICNF)*, p. 1–4. 223, 224

LEVY-VEHEL, J. et P. LEGRAND. 2004, «Signal and image processing with fraclab», *FRACTAL04, Complexity and Fractals in Nature, International Multidisciplinary Conference*, p. 321–322. 239

MANDELBROT, B. B. et J. W. VAN NESS. 1968, «Fractional brownian motions, fractional noises and applications», *SIAM Review*, vol. 10(4), p. 422–437. 222

MERT, A. et A. AKAN. 2014, «Detrended fluctuation thresholding for empirical mode decomposition based denoising», *Digital Signal Processing*, vol. 32, p. 48–56. 223

MOULINES, E., F. ROUEFF et M. S. TAQQU. 2007, «Central limit theorem for the log-regression wavelet estimation of the memory parameter in the gaussian semi-parametric context», *Fractals*, vol. 15, p. 301–313. 222

NAVARRO, X., A. BEUCHÉE, F. PORÉE et G. CARRAULT. 2011, «Performance analysis of Hurst's exponent estimators in higly immature breathing patterns of preterm infants», *International Conference on Acoustics, Speech and Signal Processing*, p. 701–704. 223

NAYAK, S. K., A. BIT, A. DEY, B. MOHAPATRA et K. PAL. 2018, «A review on the nonlinear dynamical system analysis of electrocardiogram signal», *Journal of Healthcare Engineering*, vol. (2), p. 1–19. 223

OSBORNE, D. 1995, «Moving average detrending and the analysis of business cycles», *Oxford Bull. Econom. Statist.*, vol. 57, p. 547–558. 223

PENG, C. K., S. V. BULDYREV, A. L. GOLDBERGER, S. HAVLIN, F. SCIORTINO, M. SIMONS et H. E. STANLEY. 1992, «Long-range correlations in nucleotide sequences», *Nature*, vol. 356, p. 168–170. 222, 224

PENG, C. K., S. V. BULDYREV, S. HAVLIN, M. SIMONS, H. E. STANLEY et A. L. GOLDBERGER. 1994, «Mosaic organization of DNA nucleotides», *Physical Review E*, vol. 49, (2), p. 1685–1689. 223, 224, 227, 231

PIPIRAS, V. et M. S. TAQQU. 2017, *Long-range dependence and self-similarity*, Cambridge University Press. 222

PRANATA, A. A., G. W. ADHANE et D. S. KIM. 2017, «Detrended fluctuation analysis on ECG device for home environment», *Consumer Communications and Networking Conference*, p. 4233–4236. 223

RABINER, L. R. et B.-H. JUANG. 1993, *Fundamentals of speech recognition*, PTR Prentice Hall. xix, 236

RAVELO-GARCIA, A. G., U. CASANOVA-BLANCAS, S. MARTIN-GONZÁLEZ, E. HERNÁNDEZ-PÉREZ, I. GUERRA-MORENO, P. QUINTANA-MORALES, N. WESSEL et J. L. NAVARRO-MESA. 2014, «An approach to the enhancement of sleep apnea detection by means of detrended fluctuation analysis of RR intervals», *Computing in Cardiology*, p. 905–908. 223

RILEY, M. A., S. BONNETTE, N. KUZNETSOV, S. WALLOT et J. GAO. 2012, «A tutorial introduction to adaptive fractal analysis», *Frontiers in Physiology*, vol. 3, p. 371. 223, 228, 230

RILLING, G., P. FLANDRIN et P. GONÇALVES. 2005, «Empirical mode decomposition, fractional gaussian noise and hurst exponent estimation», *International Conference on Acoustics, Speech and Signal Processing*, p. 489–492. 222

SANYAL, S., A. BANERJEE, R. PRATIHAR, A. K. MAITY, S. DEY, V. AGRAWAL, R. SENGUPTA et D. GHOSH. 2015, «Detrended fluctuation and power spectral analysis of alpha and delta EEG brain rhythms to study music elicited emotion», *International Conference on Signal Processing, Computing and Control*, p. 206–210. 223

SHAO, Y.-H., G.-F. GU, Z.-Q. JIANG, W.-X. ZHOU et D. SORNETTE. 2012, «Comparing the performance of fa, dfa and dma using different synthetic long-range correlated time series», *Scientific Reports*, vol. 2, p. 835. 223

SUN, R. 2007, «Fractional order signal processing: techniques and applications», *Thesis of Master of science in electrical Engineering, Utah State University*. 222

TAQQU, M. S. et V. TEVEROVSKY. 1996, «On estimating the intensity of long range dependence in finite and infinite variance time series», *A practical guide to heavy tails: statistical techniques and applications*, p. 177–217. 222

TAQQU, M. S., V. TEVEROVSKY et W. WILLINGER. 1995, «Estimators for long range dependence: an empirical study», *Fractals*, vol. 3, (4), p. 785–788. 222

TARVAINEN, M. P., P. O. RANTA-AHO et P. A. KARJALAINEN. 2002, «An advanced detrending method with application to hrv analysis», *IEEE Trans. on Biomedical Engineering*, vol. 49, 2, p. 172–175. 223, 227, 228

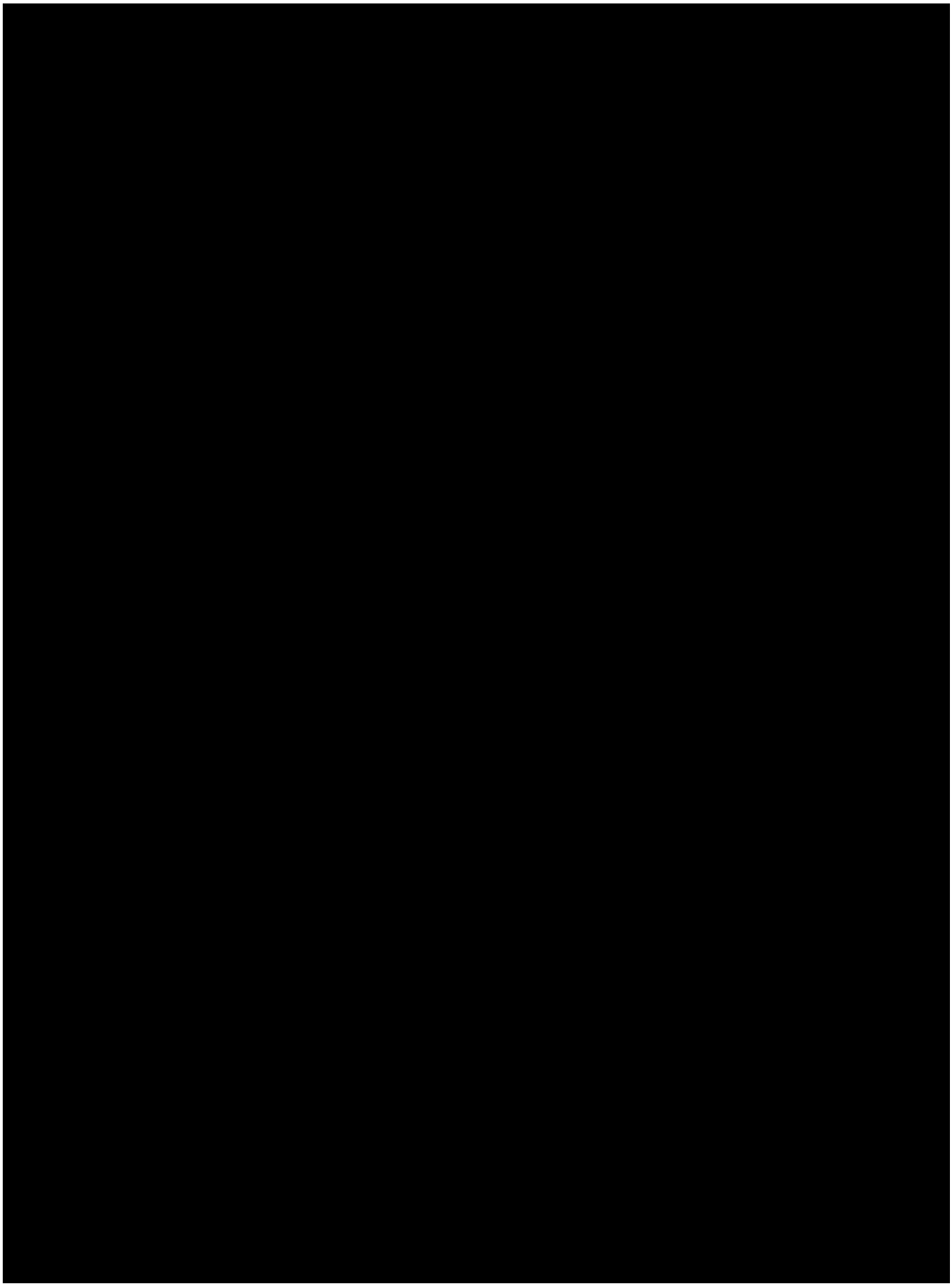
TSUJIMOTO, Y., Y. MIKI, S. SHIMATANI et K. KIYONO. 2016, «Fast algorithm for scaling analysis with higher-order detrending moving average method», *Physical Review E*, vol. 93 (5), p. 053 304. 223

TSUJIMOTO, Y., Y. MIKI, E. WATANABE, J. HAYANO, Y. YAMAMOTO, T. NOMURA et K. KIYONO. 2017, «Fast algorithm of long-range cross-correlation analysis using savitzky-golay detrending filter and its application to biosignal analysis», *International Conference on Noise and Fluctuations*, p. 1–4. 223

XU, L., P. C. IVANOV, K. HU, Z. CHEN, A. CARBONE et H. E. STANLEY1. 2005, «Quantifying signals with power-law correlations: A comparative study of detrended fluctuation analysis and detrended moving average techniques», *Physical Review E*, vol. 71, 5, p. 051 101. 223, 224, 226

Chapter 11 CONFIDENTIEL

Patents



Part III

Applications, combination of signal processing, fractal analysis and artificial evolution

Chapter 12

The Estimation of Hölderian Regularity using Genetic Programming

This chapter has been presented at the conference GECCO 2010, Jul 2010, Portland, Oregon, and received the Best Paper Award in the Track Genetic Programming. Work carried out with Leonardo Trujillo and Jacques Levy-Vehel.

Contents

Abstract

This chapter presents a Genetic Programming (GP) approach to synthesize estimators for the pointwise Hölder exponent in 2D signals. It is known that irregularities and singularities are the most salient and informative parts of a signal. Hence, explicitly measuring these variations can be important in various domains of signal processing. The pointwise Hölder exponent provides a characterization of these types of features. However, current methods for estimation cannot be considered to be optimal in any sense. Therefore, the goal of this work is to automatically synthesize operators that provide an estimation for the Hölderian regularity in a 2D signal. This goal is posed as an optimization problem in which we attempt to minimize the error between a prescribed regularity and the estimated regularity given by an image operator. The search for optimal estimators is then carried out using a GP algorithm. Experiments confirm that the GP-operators produce a good estimation of the Hölder exponent in images of multifractional Brownian motions. In fact, the evolved estimators significantly outperform a traditional method by as much as one order of magnitude. These results provide further empirical evidence that GP can solve difficult problems of applied mathematics.

12.1 Introduction

The fields of signal processing and pattern recognition are primarily concerned with analyzing and understanding the information contained within complex signals or data patterns. This paper deals with the concept of Hölderian regularity which is used to analyze prominent signal variations, and which is therefore relevant in both fields. Hölderian regularity, also known as Lipschitz regularity, characterizes the singularities contained within non-differentiable signals using local or pointwise exponents MALLAT [1999]; TRICOT [1995]. This measure can effectively describe the structure of a signal around each point, a property that has made it quite useful in various tasks, for instance see LEGRAND et LÉVY-VÉHEL [2003]; LEGRAND et VEHEL [September 14-17, 2003]; LÉVY-VÉHEL [1998]; TRUJILLO et collab. [2007]. However, one drawback of Hölderian analysis is that Hölder exponents can only be computed in a closed form for a limited number of signal types. In order to overcome this, several estimation methods have been developed and are widely used with multifractal analysis, each based on strong mathematical principles and derived using necessary assumptions regarding the underlying properties of the signal that is analyzed JAFFARD [2004]; LEGRAND [2004]. These estimators have proven to be useful tools, and an open-source toolbox exists that can be used to test these methods LÉVY-VÉHEL et LEGRAND [2004]. However, it is also important to understand that in practice these algorithms depend on the correct setting of several parameters, and other ad-hoc decisions are required to obtain a desired performance. These methods tend to be complex and relatively slow, making their use difficult in domains that require fast processing.

In the present work, the goal is to automatically synthesize operators that can estimate the Hölderian regularity using Genetic Programming (GP). In particular, we focus on estimating the pointwise Hölder exponent in 2D digital signals (images), primarily for the following reasons. First, this measure of regularity has proven to be a powerful tool for basic problems of image analysis, such as noise removal LEGRAND et VEHEL [September 14-17, 2003], interpolation LEGRAND et LÉVY-VÉHEL [2003], and edge detection LÉVY-VÉHEL [1998], to mention but a few. Moreover, it achieved highly competitive performance in the problem of local image description TRUJILLO et collab. [2007], one of the most widely used procedures in current computer vision literature MIKOLAJCZYK et SCHMID [2005]. However, despite these successful applications, the limitations of current estimation methods, described above, limit its broader use.

Given our stated goal, let us describe the manner in which we attempt to achieve it.

First, we generate several groups of synthetic images of multifractional Brownian motion. All images of the same groups share the same prescribed regularity; i.e, the images have the same regularity at each point but show different intensity patterns. The regularity is prescribed using a known function and the intensity images are then automatically generated using the methods developed in BARRIERE [2007]. Then, by using these images as reference we are able to pose an optimization problem where the goal is to find image operators that can estimate the pointwise Hölder exponent with a minimum amount of error. This problem could be solved in different ways, but in this work we have chosen to use GP, probably the most advanced form of evolutionary computation. GP has proven to be well suited for problems where a specialized mathematical expression is required but the general structure of the expression is difficult to define a priori KOZA [1992]. The experimental results show that the estimators evolved through GP provide a superior estimation when compared with a traditional method, in several cases the difference is one order of magnitude.

12.1.1 Related Work

The proposal developed in this paper is closely related to recent applications of GP in the areas of mathematics and image analysis. First, regarding the latter, we can say that our proposal is concerned with extracting a specific type of image feature that can be computed for each point, namely their Hölderian regularity. If we take this view, we can observe a close relation with the works in TRUJILLO et OLAGUE [2006, 2008]; TRUJILLO et collab. [2008], for example, where GP was used to synthesize image operators that determine the saliency of each image pixel. Another example is the work of PEREZ et OLAGUE [2009], where GP is given the task of finding operators that optimize the description of local image regions, or the work in ZHANG et ROCKETT [2005] that uses GP to detect edge points. Regarding the application of GP to mathematics, several interesting proposals have also been developed. The traditional examples are symbolic regression problems, originally descibed by Koza and further extended by works such as GUSTAFSON et collab. [2005]; KEIJZER [2004]. However, recent applications in solving differential equations BALASUBRAMANIAM et VINCENT ANTONY KUMAR [2009] and in the study of finite algebras SPECTOR et collab. [2008] have shown that GP need not be limited only to regression analysis. From these examples, we would like to restate the idea that a GP algorithm, when appropriately used, can indeed be characterized as a tool for *automatic scientific discovery* KEIJZER et BABOVIC [2002].

Given our brief introduction, we proceed to outline the remainder of this paper. Section 13.3 formally defines the concept of Hölderian regularity and describes a canonical method used to estimate the pointwise exponent. Then, in Section 13.4 we present a formal definition of our problem and give a detailed description of our GP proposal. The experimental setup is described in Section 13.5, along with a description of the results obtained. Finally, Section 13.6 contains our concluding remarks.

12.2 Hölderian Regularity

It is well understood that singular and irregular structures contain the most prominent, and most useful, information within a signal. For example, in images large discontinuities often correspond with salient image features that can be used for recognition tasks MIKOLAJCZYK et SCHMID [2005]. Hölderian regularity is a manner in which to characterize precisely these singular structures [MALLAT, 1999; TRUJILLO et collab., 2007]. The regularity of a signal at each point can be quantified by the pointwise Hölder exponent, which we define below.

Definition 1:Let $f : \mathbb{R} \to \mathbb{R}$, $s \in \mathbb{R}^{+*} \setminus \mathbb{N}$ and $x_0 \in \mathbb{R}$. $f \in C^s(x_0)$ if and only if $\exists \eta \in \mathbb{R}^{+*}$, and a polynomial P of degree $< s$ and a constant c such that

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \leq c|x - x_0|^s \,, \tag{12.1}$$

where $B(x_0, \eta)$ is the local neighborhood around x_0 with a radius η. The pointwise Hölder exponent of f at x_0 is $\alpha_p(x_0) = \sup_s \{f \in C^s(x_0)\}$.

The concept of Hölderian regularity is closely related to the Taylor series approximation of a function. However, the Hölder exponents refines this concept by also accounting for non-differentiable points [MALLAT, 1999]. Figure 12.1 shows a graphical illustration of the Hölder exponent, depicted as an envelope, for a non-differentiable signal.

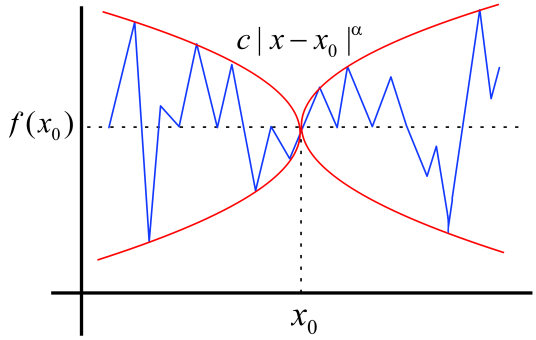


Figure 12.1 – Hölderian envelope of signal f at point x_0.

The Hölder exponent, however, can only be computed analytically for a limited number of signal types. Therefore, in order to use Hölderian regularity the exponents must be estimated, and several numerical methods have been proposed for this purpose. The most direct is the oscillations method which is directly related to the definition given above TRICOT [1995].

12.2.1 Estimation through oscillations

The Hölder exponent of function $f(t)$ at point t is the $sup(\alpha_p) \in [0, 1]$, for which a constant c exists such that $\forall\, t'$ in a neighborhood of t,

$$|f(t) - f(t')| \leq c|t - t'|^{\alpha_p} \,. \tag{12.2}$$

In terms of signal oscillations, a function $f(t)$ is Hölderian with exponent $\alpha_p \in [0, 1]$ at t if $\exists c$ $\forall \tau$ such that $osc_\tau(t) \leq c\tau^{\alpha_p}$, with

$$osc_\tau(t) = \sup_{t',t'' \in [t-\tau, t+\tau]} |f(t') - f(t'')| \,. \tag{12.3}$$

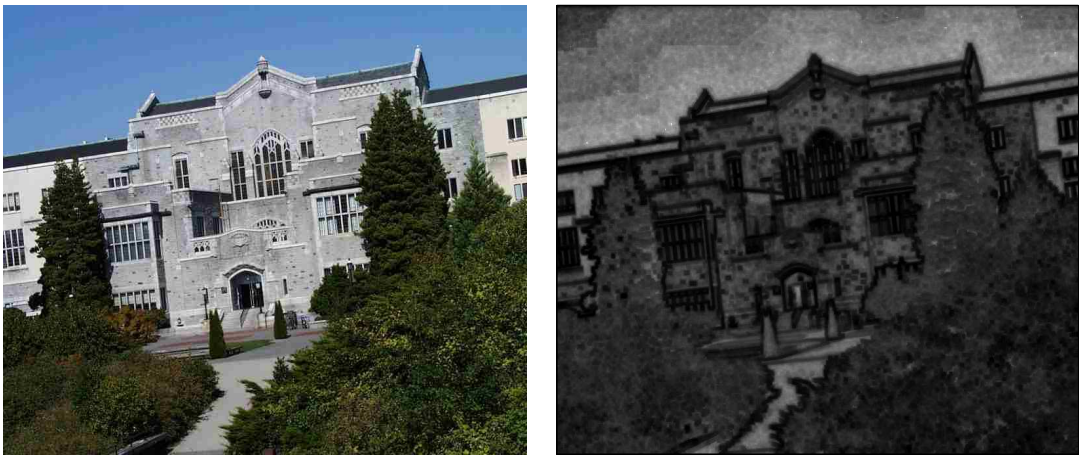
Now, if $t = x_0$ and $t' = x_0 + h$ in 13.3, we can also write that

$$\alpha_p(x_0) = \text{liminf}_{\,h \to 0} \frac{\log |f(x_0 + h) - f(x_0)|}{\log |h|} \,. \tag{12.4}$$

Therefore, the problem is that of finding an α_p that satisfies 13.3 and 13.4, and in order to simplify this process we can set $\tau = \beta^r$. Then, we can write $osc_\tau \approx c\tau_p^\alpha = \beta^{(\alpha_p r + b)}$, which is equivalent to $log_\beta(osc_\tau) \approx \alpha_p r + b$.

Therefore, an estimation of the regularity can be built at each point by computing the slope of the regression between the logarithm of the oscillations osc_τ and the logarithm of the dimension of the neighborhood at which the oscillations τ are computed. Here, we use least squares regression to compute the slope, with $\beta = 2$ and $r = 1, 2, \ldots, 7$. Also, it is preferable not to use all sizes of neighborhoods between two values τ_{min} and τ_{max}. Hence, we calculate the oscillation at point x_0 only on intervals of the form $[x_0 - \tau_r : x_0 + \tau_r]$. For a 2D signal, x_0 defines a point in 2D space and τ_r a radius around x_0, such that $d(t', t) \leq \tau_r$ and $d(t'', t) \leq \tau_r$, where $d(a, b)$ is the Euclidean distance between a and b.

Figure 13.3 shows a visual example of the type of output this algorithm produces, it presents a sample image and the corresponding Hölder exponent for each pixel. This method



(a) Original Image (b) Hölder Image

Figure 12.2 – Estimation of the Hölder exponent using oscillations.

has proven to be superior LEGRAND [2004] in some cases to the wavelet leaders method JAFFARD [2004], and useful in real-world applications LEGRAND et LÉVY-VÉHEL [2003]; LEGRAND et VEHEL [September 14-17, 2003]; LÉVY-VÉHEL [1998]; TRUJILLO et collab. [2007]. Therefore, we use the oscillations method for comparisons with our evolved estimators.

12.3 Outline of our proposal

Returning to the main goal of our work, the automatic synthesis of operators that estimate the pointwise Hölder exponent for 2D signals, we state the following formal problem.

12.3.1 Problem statement

Let I represent a digital 2D signal, or more specifically an image, and suppose that H_I is a matrix that contains the value of the pointwise Hölder exponent for every pixel in I. Then, we can pose the problem of finding an optimal operator K^o as follows,

$$K^o = \arg\min_K \{Err[K(I), H_I]\} \ , \tag{12.5}$$

where $Err[,]$ represents an error measure, which in this work is given by the root-mean-square error (RMSE)

$$Err[K(I), H_I] = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(K(x_i) - H_{x_i})^2} \ , \tag{12.6}$$

where N is the number of pixels in an image I.

Here H_I is prescribed by a function p, and for each such function it is possible to build an infinite number of images that share the same regularity (see Section 13.5). Therefore, the aim is to find the symbolic expression of an operator that minimizes the estimation error. Note that the goal is to obtain the best possible estimation, without accounting for other possible objectives such as computation time. We assume that this is the appropriate choice given the novelty of the problem, and leave a possible multi-objective formulations for future research. Nevertheless, we do account for computation time in an indirect manner, by enforcing size constraints on the search process through bloat control. In what follows, we briefly present the paradigm of GP and then describe the algorithm proposed for the stated problem.

12.3.2 Genetic Programming

Evolutionary computation has developed a rich variety of search and optimization algorithms that base their core functionality on the basic principles of Neo-Darwinian evolution DE JONG [2001]. These techniques are population-based meta-heuristics, where candidate solutions are stochastically selected and modified in order to produce new, and possibly better, solutions for a particular problem. The selection process favors individuals that exhibit the best performance and the process is carried out iteratively until a termination criterion is reached. Of current algorithms, GP is one of the most advanced forms of evolutionary search KOZA [1992]. In canonical GP each solution is represented using a tree structure, which can express a simple computer program, function, or operator. Individual trees are constructed using elements from two finite sets, internal nodes contain simple functions from a *Function* set F, and leaves contain the input variables from the *Terminal* set T. These sets define the search space for a GP algorithm, and when a depth or size limit is enforced, the space is normally very large but finite.

12.3.3 Proposed algorithm

The proposal of this work is to use standard Koza style GP to solve the optimization problem given in Eq. 12.5. In what follows, we define the fitness function and the search space for the GP algorithm.

12.3.3.1 Fitness evaluation

Fitness is defined for a maximization problem, where the fitness of an operator K is given by

$$f(K) = \frac{1}{\frac{1}{M}\sum_{j=1}^{M} Err[\widehat{K}(I_j), \widehat{H}_{I_j}] + \epsilon} , \tag{12.7}$$

where I_j is the jth image in the training set of M images, $\epsilon = 0.01$ avoids divisions by zero, and $\widehat{K}(I)$ and $\widehat{H_I}$ are normalized versions of $K(I)$ and H_I using the L2-norm. Here, fitness is assigned based on the mean RMSE computed for a set of training images. However, one constraint is added, when the standard deviation of the RMSEs is zero then the fitness is also set to zero. This is done because it would be naive to expect the same estimation for every image. The constraint removes the possibility of an operator estimating the same exponent for every pixel. For instance, if all exponents are set to zero the estimation is meaningless, but it might produce a better fitness than a random operator and could then take over the population in earlier generations. This scenario occurred often in preliminary runs of the GP-search that did not include this constraint.

12.3.3.2 Search space

The search space for a GP is established by the sets of Terminals and Functions, in our work these are

$$\begin{aligned} F = &\{+, |+|, -, |-|, |I_o|, *, \div, I_o^2, \sqrt{I_o}, log_2(I_o), k \cdot I_o\} \\ &\bigcup \{BF, G_\sigma, Avg_m, Med_m, Max_m, Min_m\} \ , \\ T = &\{I\} \ , \end{aligned} \tag{12.8}$$

where I is the input image; I_o represents I or the output from any function in F; G_σ are Gaussian smoothing filters with $\sigma \in \{1, 3, 5\}$; $Avg_m, Med_m, Max_m, Min_m$ are average, median, max and min filters with a mask size of $m \times m$ and $m \in \{3, 5, 7\}$; BF is a bilateral filter TOMASI et MANDUCHI [1998]; and a scale factor $k = 0.05$. Set F contains functions that operate in a point by point manner, such as the addition of two matrix, and image filters

that operate within a local neighborhood. The latter group allows the evolved estimators to exploit variations within a local image region, information which is essential for the oscillations method. Regarding the point to point arithmetic operations, we use the following conventions in order to avoid undefined operations: (1) we assume that $I \in \Re^+$; (2) the logarithm is protected by using $\log(0) = 0$, and $\log(-a) = \log(a)$; (3) division is protected by $\frac{a}{b} = a$ if $b = 0$; and (4) we define $\sqrt{-a} = \sqrt{a}$.

12.4 Experiments and Results

This Section describes our GP algorithm, presents the experimental results, and provides comparisons with the oscillations method.

12.4.1 Implementation

12.4.1.1 Experimental setup

The GP algorithm was set-up with the parameters presented in Table 12.1, most are based on basic GP literature KOZA [1992]. The only non-canonical aspect of our GP algorithm is the use of a bloat control method, a dynamic maximum tree depth SILVA et ALMEIDA [2003]. The algorithm was programmed using the Matlab toolbox GP-Lab [1], and estimation of the Hölder exponent was carried out using code for the oscillations method that will be provided in the next release of FracLab LÉVY-VÉHEL et LEGRAND [2004].

12.4.1.2 Training and test data

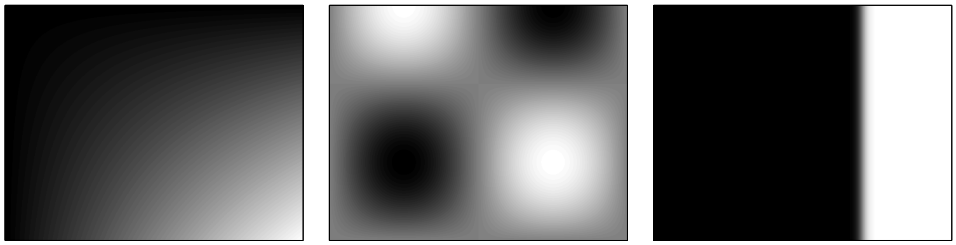
In order to compute fitness and to perform further tests, we build a set of images with prescribed regularity using 2D multifractional Brownian motions. This is a generalization of the fractional Brownian motion where the constant exponent H is replaced with a Hölder continuous function AYACHE et LÉVY-VÉHEL [2000]. This type of signal is a good model for real world data, and it can be directly generated with a prescribed regularity BARRIERE [2007]. In this work, we generate three groups of images with Fraclab, using three different functions that take as input the point coordinates (x, y) of an image and provide as output the desired regularity; these functions are: (a) a *Polynomial* $p_1(x, y) = 0.1 + 0.8xy$; (b) a *Sine* $p_2(x, y) = 0.5 + 0.2(sin(2\pi x))(cos(\frac{3}{2}\pi y))$; and (c) an *Exponential* $p_3(x, y) = 0.3 + \frac{0.3}{1+e^{-100(x-0.7)}}$. These functions provide the prescribed regularity needed to build the synthetic images used for training and testing of our evolved operators, see Figure 12.3.

Here, we generate twelve images for each of the prescribed regularity functions p_1, p_2 and p_3; all images have the same size of 512×512 pixels. Figure 12.4 shows three examples from each of these groups. What is important to see is that images with the same prescribed regularity are nevertheless quite different, and in fact an infinite number of images exist that share the same regularity! Therein lies the intrinsic difficulty of obtaining an optimal estimator for the Hölder exponent. The images we generate are divided into two sets, one for training and one for testing. The training set contains six images with polynomial regularity, and all the rest, thirty in total, are used for testing.

12.4.2 Results and comparisons

This section describes the relevant experimental results of our work. First, Figure 12.5 shows the convergence plots for each of the eleven runs of our GP. Figure 12.5a plots the best individual fitness at each iteration, and Figure 12.5b shows the average fitness of the population. Each plot is tagged using the acronym HGP (*Hölderian regularity with GP*), and the

[1] GP-Lab by Sara Silva (gplab.sourceforge.net/index.html).



(a) Polynomial p_1 (b) Sine p_2 (c) Exponential p_3

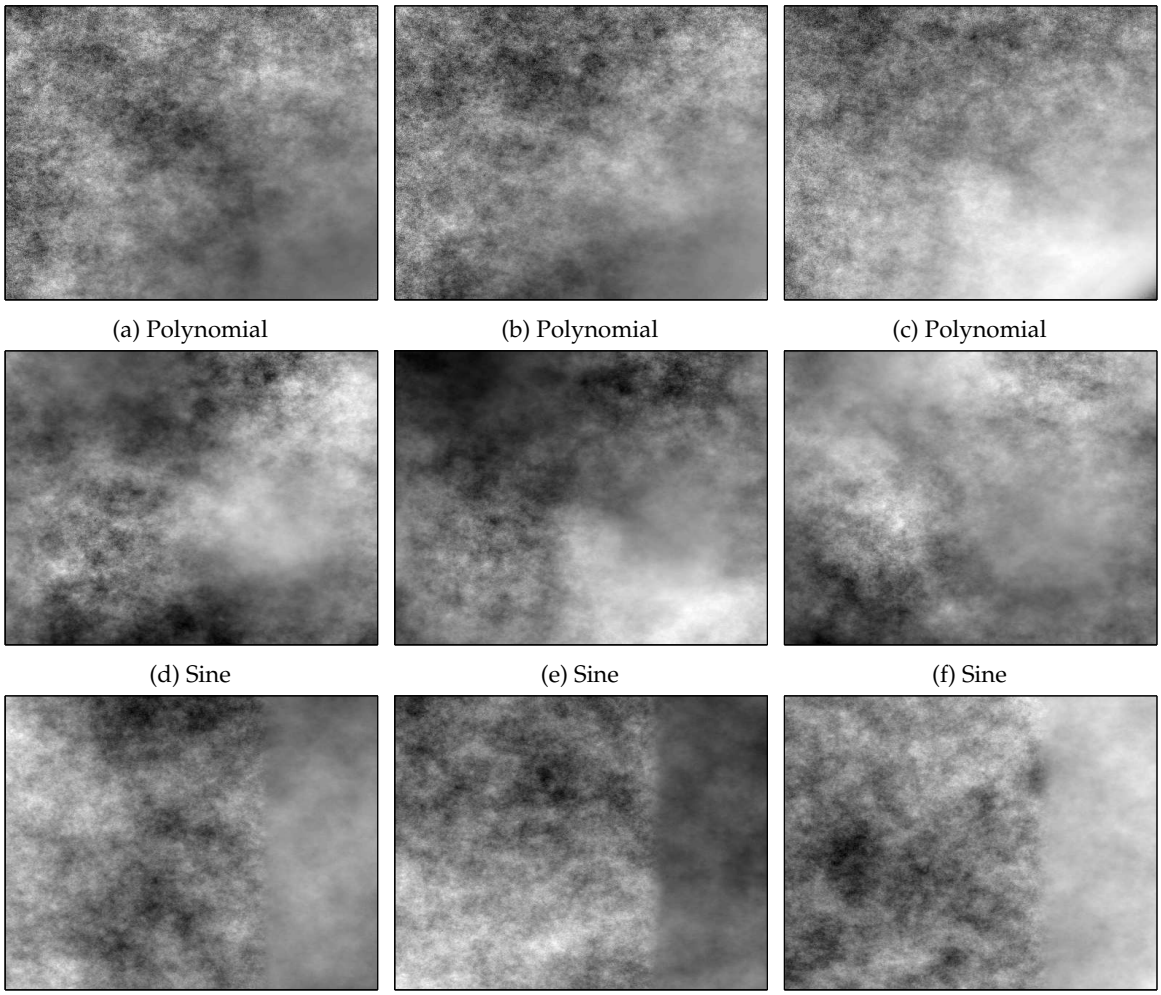
Figure 12.3 – Prescribed regularity of our experimental data.

Table 12.1 – GP parameters used in our experiments.

Parameter	**Description**
Population size	200 individuals
Iterations	100 generations
Initialization	*Ramped Half-and-Half*
Crossover probability	$p_c = 0.85$
Mutation probability	$p_\mu = 0.15$
Bloat control	Dynamic maximum tree depth
Initial max. depth	Six levels
Initial dynamic max. depth	Eleven levels
Max. tree depth	16 levels
Selection	Stochastic universal sampling
Survival	Elitism
Runs	Eleven

corresponding run number. These plots show similar convergence patterns in all of the runs, with HGP-7 being the best in both cases, and HGP-5 and HGP-6 not far behind based on the fitness computed with the training set. The first observation is that the evolutionary search shows a steady convergence over the specified number of generations, from which we can say that the proposed algorithm appears to be well suited for the stated problem. On the other hand, it could be said that the small number of runs can only provide a weak statistical inference, and that the termination criteria does not eliminate the possibility that further improvements could have been achieved. However, there is a practical restraint that accounts for these shortcomings, namely that the computation time for a single run is very long. In some experiments it required several days. Therefore, we believe that in such applications the somewhat small number of runs is justified.

Moving on, in Table 12.2 we show the result of testing our evolved operators on the additional test images, thirty in total. Here, we show the average RMSE computed for each of the three groups (polynomial, sine and exponential prescribed regularity functions), the standard deviation, and the corresponding fitness value. In these tests we use the best individual found at the end of each run. For comparisons we show the same statistics computed for the estimation obtained with the oscillations method. Among the evolved operators, HGP-7 also achieves the highest marks on the test images with polynomial regularity. It is also the second best when tested on the images with sine function regularity, while HGP-11 is the best in this case (the RMSE of both is of the same order). However, for the exponential function, HGP-7 is only average amongst this group, but still better that the oscillations method, the best operator for these images is HGP-1. A few remarks are relevant for these results. First, we can affirm that the evolved operators are indeed superior to oscillation-based estimation, in some cases the difference is one order of magnitude. Second, using the proposed training set we obtain an operator that achieves high fitness on the test images



(a) Polynomial (b) Polynomial (c) Polynomial

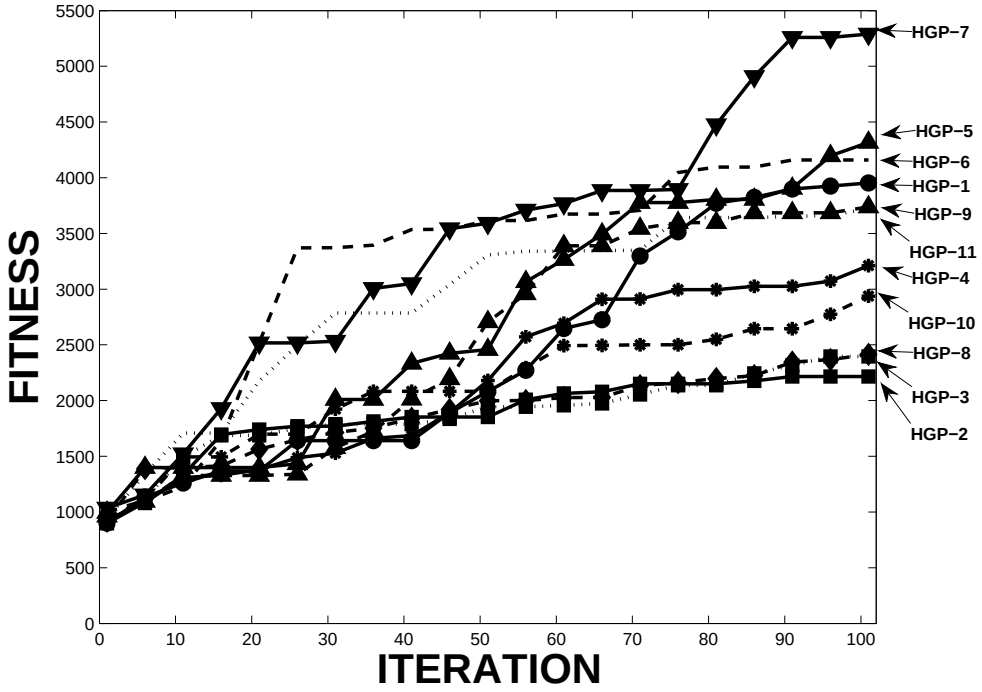
(d) Sine (e) Sine (f) Sine

(g) Exponential (h) Exponential (i) Exponential

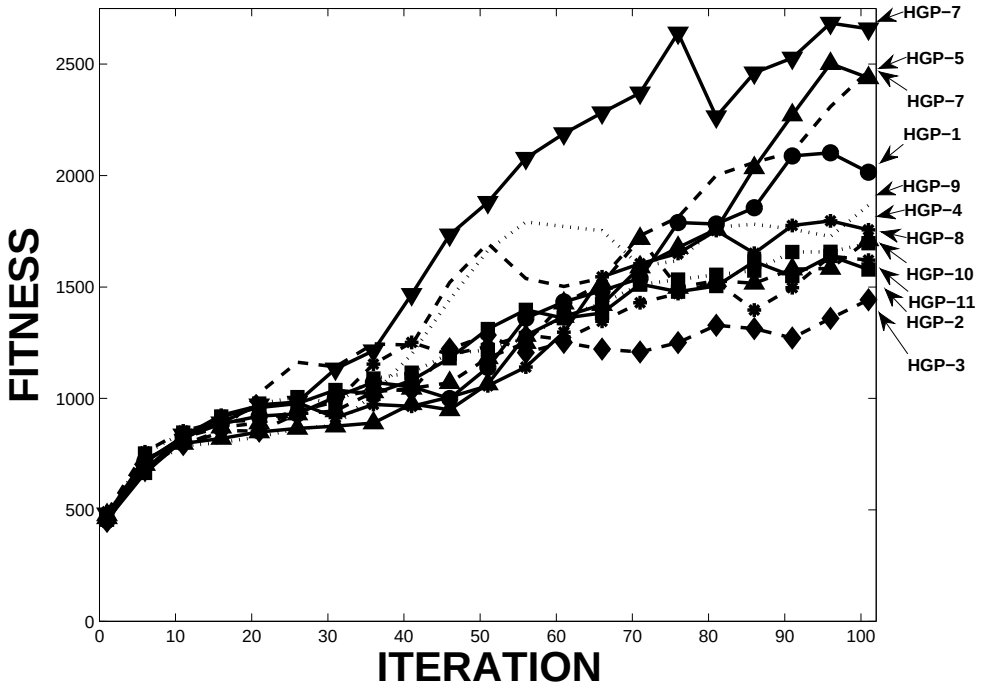
Figure 12.4 – These images have a prescribed regularity given by functions p_1 (Polynomial), p_2 (Sine) and p_3 (Exponential).

with polynomial and sine regularity. The former result was expected given the training set, and the latter is a good indication that GP does not over-fit the solutions that it generates. In the case of the exponential function, HGP-7 was not among the best but still reaches a better performance than the oscillation-based method. Third, we can see that some of the operators that achieve only average performance on the training set are able to achieve much better scores on other types of regularity functions. This further suggests that our algorithm does not over-fit the solutions to the specific problem given by the training set, it is a well-posed search for regularity estimator in a more general sense. Finally, in order to give a sense of what GP generates in work, the program tree of the best individual found is shown in Figure 12.6.

A qualitative comparison between our evolved operators and the oscillations method is shown in Figure 12.7, where the Hölderian regularity is estimated for three of the test images, one for each type of prescribed regularity. The first column of Figure 12.7 is the estimation for a test image with a prescribed regularity given by the polynomial function and the next column shows the difference between the estimated regularity and the prescribed regularity of Figure 12.3. The following two pairs of columns are similar for the sine and exponential cases. Four of the evolved operators are not included (HGP-2,3,8 & 10) because of paper length. Moreover, the excluded ones achieve the worst performance among the HGP estimators but are still better than the oscillations method. This comparison confirms



(a) Best fitness



(b) Population average

Figure 12.5 – Convergence plots for each of the eleven runs.

the numerical results of Table 12.2, from which we can reaffirm that the HGP operators are superior estimators of the pointwise Hölder exponent.

Finally, Table 12.3 provides a informal comparison of computation time between the HGP estimators, and the estimation obtained using Fraclab on a single test image, showing the average over five executions. These results were obtained using a standard Laptop-PC with Intel Dual Core 64-bit 2.39 GHz processor, 4 GM of shared RAM, and running 32-bit

Table 12.2 – Comparison of the best solution from each run with the oscillations method using the thirty test images from the three different functions for the prescribed regularity. The mean and std are scaled to 10^{-3}, and bold marks the best results.

	HGP-1	HGP-2	HGP-3	HGP-4	HGP-5	HGP-6	HGP-7	HGP-8	HGP-9	HGP-10	HGP-11	Osc.
Polynomial (six images)												
Mean	0.149	0.638	0.317	0.207	0.127	0.125	**0.087**	0.328	0.154	0.233	0.159	0.473
Std.	0.003	0.048	0.044	0.009	0.003	0.005	0.004	0.037	**0.002**	0.008	0.004	0.046
Fit.	4015	1355	2396	3254	4414	4452	**5343**	2337	3932	3001	3861	1744
Sine (twelve images)												
Mean	0.047	0.408	0.194	0.082	0.064	0.064	0.051	0.243	0.072	0.115	**0.028**	0.323
Std.	0.004	0.147	0.043	0.02	0.002	0.004	0.002	0.025	0.002	0.008	**0.003**	0.046
Fit.	6822	1969	3396	5484	6096	6105	6617	2915	5809	4645	**7799**	2364
Exponential (twelve images)												
Mean	**0.055**	0.395	0.165	0.102	0.103	0.103	0.166	0.282	0.078	0.141	0.06	0.329
Std.	0.004	0.184	0.013	0.03	**0.001**	0.003	0.001	0.049	0.003	0.007	0.002	0.029
Fit.	**6438**	2022	3770	4939	4923	4923	3766	2617	5615	4148	6256	2331

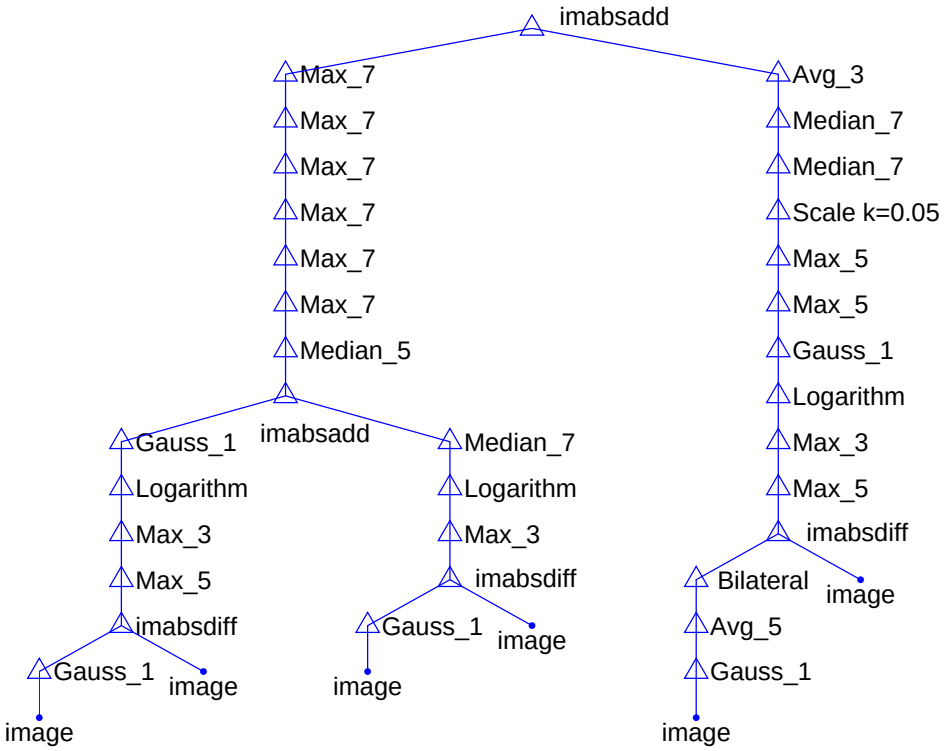
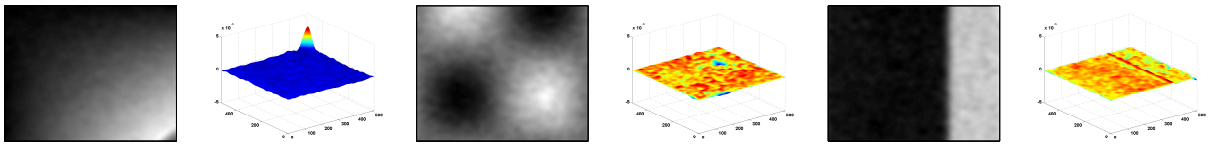


Figure 12.6 – The program tree for the HGP-7 estimator.

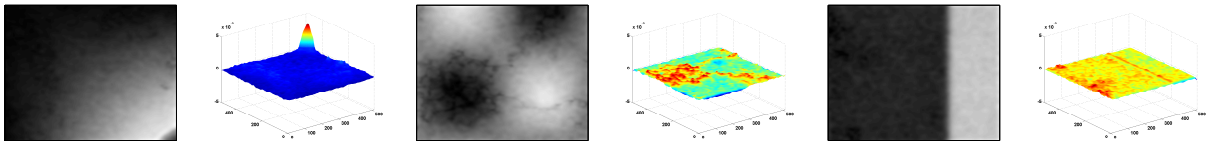
Win-XP SP3 and Matlab 2009a. Before evaluating these results, several observations are relevant. First, our estimators were not simplified, the complete GP trees are used in all tests. Second, time of computation was not included within our fitness criteria. And finally, neither our code nor the Fraclab functions for the oscillations method was optimized in any way. Nevertheless, the results presented in Table 12.3 do provide a rough estimate of which estimation method is more efficient, and under this comparison we can again conclude that GP produces better methods for estimation.

Table 12.3 – Run-time comparisons of the HGP estimators and the oscillations method from Fraclab; all values are provided in seconds and represent the average over five executions.

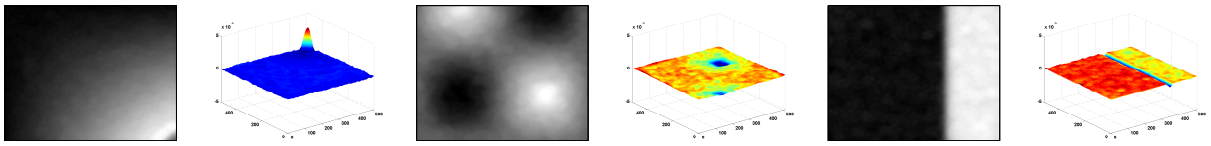
	HGP-1	**HGP-2**	**HGP-3**	**HGP-4**	**HGP-5**	**HGP-6**
Time	1.20	9.43	10.32	6.26	1.40	4.79
	HGP-7	**HGP-8**	**HGP-9**	**HGP-10**	**HGP-11**	**Osc.**
Time	9.34	5.08	**0.77**	12.61	5.81	365.2



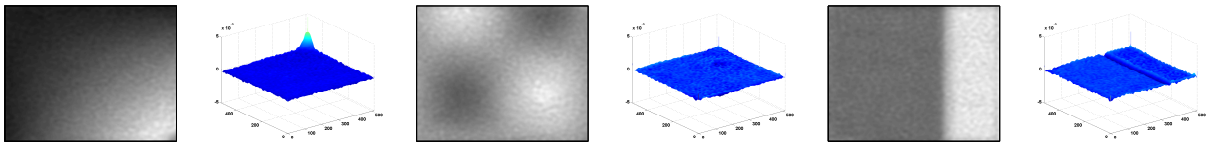
(a) HGP-1



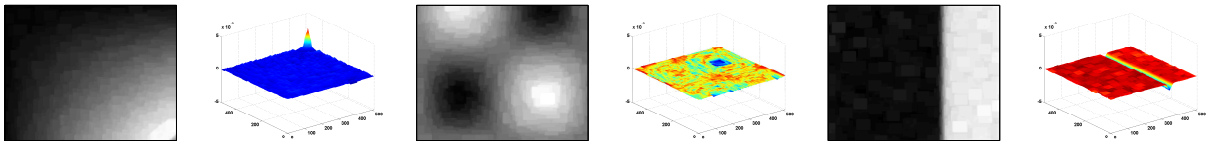
(b) HGP-4



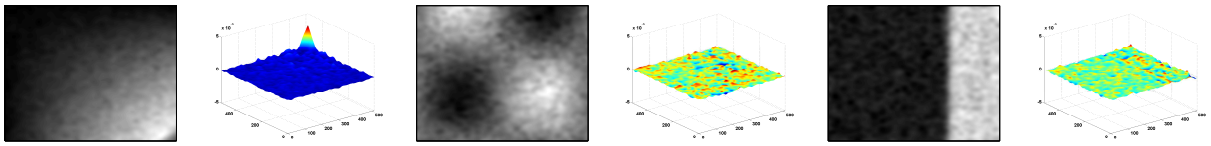
(c) HGP-5



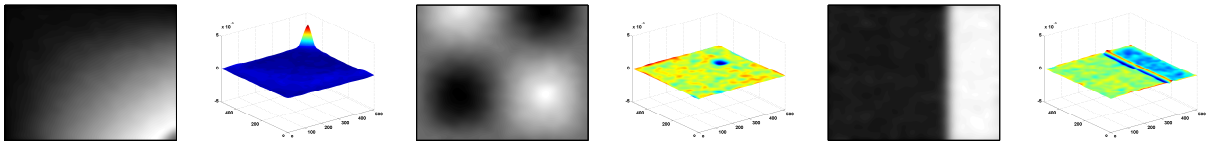
(d) HGP-6



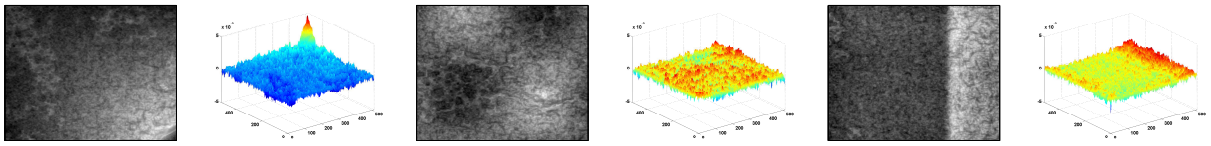
(e) HGP-7



(f) HGP-9



(g) HGP-11



(h) Oscillations

Figure 12.7 – Qualitative comparison between the evolved HGP estimators and the oscillations method using one test case from each prescribed regularity function. The first column shows the estimated regularity for a test image with a prescribed regularity given by the polynomial function, and the next column shows the difference between the estimated regularity and the prescribed regularity, see Figure 12.3. The following two pairs of columns are similar, the second pair is for the sine function, and the final pair for the exponential function. All of the difference plots share the same z-scale, $[-0.005, 0.005]$.

12.5 Concluding remarks

This chapter presents an approach that automatically synthesizes image operators that estimate the pointwise Hölder exponent for 2D signals. It employs a GP to search for operators that minimize the estimation error given a prescribed Hölderian regularity. Training is carried out using a small set of images, all of which have the same regularity given by a polynomial function. Then, the evolved estimators are tested on a new set of images, most of which have a different Hölderian regularity given by a sine and an exponential function. The experimental results show that the evolved HGP estimators produce a good estimation of the pointwise Hölder exponent, from both a quantitative and qualitative perspective. In fact, the HGP estimators consistently and significantly outperform the oscillations method by as much as one order of magnitude. Moreover, the GP algorithm is able to produce estimators that generalize quite well given the limited set of training examples, and the evolutionary process shows a steady and progressively improving convergence. These results suggest the following main conclusions. First, new estimators for the pointwise Hölder exponent can be developed using a GP-based search and optimization process. This gives further empirical evidence that confirms the applicability of GP to difficult mathematical problems in applied domains. Second, the GP-based approach could synthesize estimators that simplify the practical use of regularity-based analysis for a wider variety of application domains. Finally, we speculate that the estimators produced by GP might lead us towards new analytical approaches for regularity analysis, although this will be left as a topic for future research.

References

AYACHE, A. et J. LÉVY-VÉHEL. 2000, «The generalized multifractional brownian motion», *Statistical Inference for Stochastic Processes*, vol. 3, p. 7–8. 269

BALASUBRAMANIAM, P. et A. VINCENT ANTONY KUMAR. 2009, «Solution of matrix riccati differential equation for nonlinear singular system using genetic programming», *Genetic Programming and Evolvable Machines*, vol. 10, nº 1, p. 71–89, ISSN 1389-2576. 265

BARRIERE, O. 2007, *Synthése et estimation de mouvements browniens multifractionnaires et autres processus á régularité prescrite. Définition du processus auto-régulé multifractionnaire et applications*, thèse de doctorat, Ecole centrale de Nantes et Universite de Nantes, France. 265, 269

DE JONG, K. 2001, *Evolutionary Computation: A Unified Approach*, The MIT Press, 272 p.. 268

GUSTAFSON, S., E. K. BURKE et N. KRASNOGOR. 2005, «On improving genetic programming for symbolic regression», dans *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK*, IEEE, p. 912–919. 265

JAFFARD, S. 2004, «Wavelet techniques in multifractal analysis», dans *Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics*, vol. 72, p. 91–151. 264, 267

KEIJZER, M. 2004, «Scaled symbolic regression», *Genetic Programming and Evolvable Machines*, vol. 5, nº 3, p. 259–269, ISSN 1389-2576. 265

KEIJZER, M. et V. BABOVIC. 2002, «Declarative and preferential bias in gp-based scientific discovery», *Genetic Programming and Evolvable Machines*, vol. 3, nº 1, p. 41–79, ISSN 1389-2576. 265

KOZA, J. R. 1992, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 849 p.. 265, 268, 269

LEGRAND, P. 2004, *Debruitage et interpolation par analyse de la regularite Hölderienne. Application a la modelisation du frottement pneumatique-chaussee*, thèse de doctorat, Université de Nantes, France. 264, 267

LEGRAND, P. et J. LÉVY-VÉHEL. 2003, «Local regularity-based interpolation», dans *WAVELET X, Part of SPIE's Symposium on Optical Science and Technology*, vol. 5207. 264, 267

LEGRAND, P. et J. L. VEHEL. September 14-17, 2003, «Local regularity - based image denoising», *ICIP03, Spain, IEEE International Conference on Image Processing*, p. 377–380. 264, 267

LÉVY-VÉHEL, J. 1998, *Fractal Image Encoding and Analysis*, chap. Introduction to the Multifractal Analysis of Images, p. 299–341. 264, 267

LÉVY-VÉHEL, J. et P. LEGRAND. 2004, *Thinking in Patterns*, chap. Signal and Image Processing with FRACLAB, p. 321–322. Http://fraclab.saclay.inria.fr/homepage.html. 264, 269

MALLAT, S. 1999, *A wavelet tour of signal processing*, 2e éd., Elsevier, San Diego, CA, 637 p.. 264, 265, 266

MIKOLAJCZYK, K. et C. SCHMID. 2005, «A performance evaluation of local descriptors», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no 10, p. 1615–1630, ISSN 0162-8828. 264, 265

PEREZ, C. B. et G. OLAGUE. 2009, «Evolutionary learning of local descriptor operators for object recognition», dans *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, ISBN 978-1-60558-325-9, p. 1051–1058. 265

SILVA, S. et J. ALMEIDA. 2003, «Dynamic maximum tree depth.», dans *Proceedings of the Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference, Chicago, IL, USA, July 12-16, 2003, Part II*, édité par E. C.-P. et al., Lecture Notes in Computer Science, Springer-Verlag, p. 1776–1787. 269

SPECTOR, L., D. M. CLARK, I. LINDSAY, B. BARR et J. KLEIN. 2008, «Genetic programming for finite algebras», dans *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, ISBN 978-1-60558-130-9, p. 1291–1298. 265

TOMASI, C. et R. MANDUCHI. 1998, «Bilateral filtering for gray and color images», dans *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, ISBN 81-7319-221-9, p. 839. 268

TRICOT, C. 1995, *Curves and Fractal Dimension*, Springer-Verlag, ISBN 0387940952, 323 p.. 264, 266

TRUJILLO, L. et G. OLAGUE. 2006, «Synthesis of interest point detectors through genetic programming», dans *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Seattle, Washington, July 8-12*, vol. 1, édité par M. Cattolico, ACM, p. 887–894. 265

TRUJILLO, L. et G. OLAGUE. 2008, «Automated design of image operators that detect interest points», *Evolutionary Computation*, vol. 16, no 4, p. 483–507. 265

TRUJILLO, L., G. OLAGUE, P. LEGRAND et E. LUTTON. 2007, «Regularity based descriptor computed from local image oscillations», *Optics Express*, vol. 15, p. 6140–6145. 264, 265, 267

TRUJILLO, L., G. OLAGUE, E. LUTTON et F. FERNÁNDEZ DE VEGA. 2008, «Multiobjective design of operators that detect points of interest in images», dans *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Atlanta, GA, July 12-16*, édité par M. Cattolico, ACM, New York, NY, USA, ISBN 978-1-60558-130-9, p. 1299–1306. 265

ZHANG, Y. et P. I. ROCKETT. 2005, «Evolving optimal feature extraction using multiobjective genetic programming: a methodology and preliminary study on edge detection», dans *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, ISBN 1-59593-010-8, p. 795–802. 265

Chapter 13

Optimization of the Hölder Image Descriptor using a Genetic Algorithm

This chapter has been presented at the conference GECCO 2010, Jul 2010, Portland, Oregon, and received the Best Paper Award in the Track Real World Application. Work carried out with Leonardo Trujillo, Gustavo Olague and Cynthia Pérez.

Contents

Abstract

Local image features can provide the basis for robust and invariant recognition of objects and scenes. Therefore, compact and distinctive representations of local shape and appearance has become invaluable in modern computer vision. In this work, we study a local descriptor based on the Hölder exponent, a measure of signal regularity. The proposal is to find an optimal number of dimensions for the descriptor using a genetic algorithm (GA). To guide the GA search, fitness is computed based on the performance of the descriptor when applied to standard region matching problems. This criterion is quantified using the F-Measure, derived from recall and precision analysis. Results show that it is possible to reduce the size of the canonical Hölder descriptor without degrading the quality of its performance. In fact, the best descriptor found through the GA search is nearly 70% smaller and achieves similar performance on standard tests.

13.1 Introduction

Currently, a large part of computer vision research is devoted towards the development of recognition systems that rely on the analysis of salient local features. These features are commonly called interest points or interest regions SCHMID et collab. [2000]; TRUJILLO et OLAGUE [2006, 2007]. This local approach has gained a wide acceptance because it can help reduce the severity of several practical problems. For instance, it is less sensitive to partial occlusions within the scene LOWE [1999], it does not require traditional image segmentation SCHMID et MOHR [1997], and provides a higher invariance to geometric and photometric transformations MIKOLAJCZYK et SCHMID [2005]; SCHMID et collab. [2000]. Furthermore, this analysis is conceptually simple and can be easily adapted to different problem domains.

The basic approach consists of two phases, detection and description of locally salient features, see Figure 13.1. First, an operator identifies the position and scale of the salient image features TUYTELAARS et MIKOLAJCZYK [2008]. Afterwards, each region is normalized, adjusting for scale, rotation and illumination invariance. Finally, each normalized region is given as input to the description process, which then outputs a numerical vector called a local descriptor MIKOLAJCZYK et SCHMID [2005]. These descriptors extract a compact and unique representation of local image structure, they are required to be distinctive and informative. It is clear that the performance of a system that uses this approach will depend on the performance of the algorithms that are used for detection and description of the salient regions.

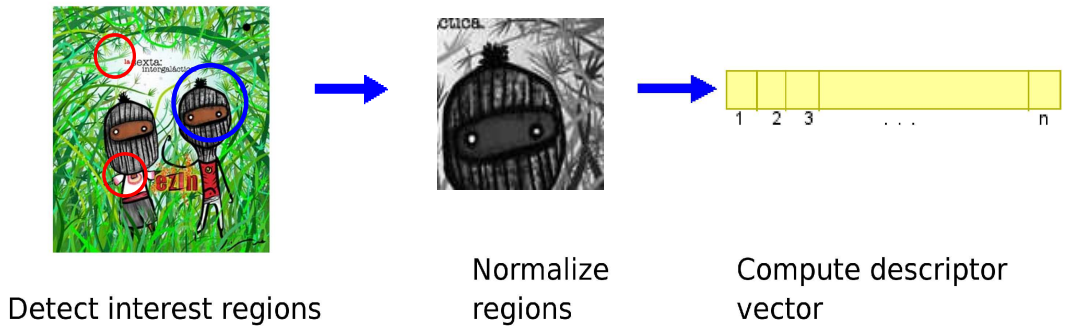


Figure 13.1 – Detection/description of local image features.

Keeping to the problem of region description, many proposals have been developed MIKOLAJCZYK et SCHMID [2005]. Probably the most widely used method is the Scale Invariant Feature Transform (SIFT) LOWE [1999], another example is the more recent Hölder descriptor TRUJILLO et collab. [2007]. In both cases, a measure of signal variation is used to build an histogram that characterizes the local shape and appearance; the former relies on the gradient orientation, and the latter on a measure of pointwise regularity. The performance of both descriptors has been shown to be quite similar when applied on standard tests TRUJILLO et collab. [2007]. On the other hand, one drawback shared by both descriptors is that current implementations are relatively slow, especially when considering real-time applications. However, one important practical difference between the two is that SIFT uses a very elaborate algorithm that is not easy to reproduced, while the Hölder descriptor employs a much simpler and direct algorithm. As such, the latter is much more amenable to optimization.

Therefore, the goal of the present work is to develop an optimized version of the Hölder descriptor that might lead to a simpler description process without decreased performance. This goal is posed as a combinatorial search problem, in order to find the optimal number of dimensions for the Hölder descriptor. It is hypothesized that the optimal size of the descriptor might be smaller than the original proposal. This hypothesis is based on the assumption that a local descriptor might be redundant, and empirical evidence supports

this claim KE et SUKTHANKAR [2004]. If this is true for the Hölder descriptor, it could lead to a more compact description of local image features.

In order achieve the goal stated above, we propose to use a genetic algorithm (GA) because of the combinatorial nature of the problem and the limited knowledge of the search space. Fitness is assigned based on the number of correct matches that are computed between two images using the descriptor. This criterion is quantified using an analysis of recall and precision statistics with the F-Measure, following the work of PEREZ et OLAGUE [2009]. The evolutionary process could eliminate redundant and unnecessary dimensions if such a representation achieves optimal performance, thereby compressing image information even further. As such, the current proposal is closely related with a long list of GA-based methods developed for dimensionality reduction, see for example BALA et collab. [1996]; HERNÁNDEZ et collab. [2007]; SIEDLECKI et SKLANSKY [1989]; SUN et collab. [2004]; TRUJILLO et collab. [2008b]. The results presented in this work show that the GA is indeed capable of finding a smaller local descriptor that achieves a similar performance.

The remainder of this chapter proceeds as follows. Section 13.2 reviews the topic of local image descriptors, explains how descriptors can be evaluated MIKOLAJCZYK et SCHMID [2005], and outlines how they can be optimized PEREZ et OLAGUE [2009]. Section 13.3 introduces the concept of Hölderian regularity and describes the canonical Hölder descriptor. Then, Section 13.4 describes the problem of this work and presents the proposed solution using a GA. Details of our implementation and the experiments are presented in Section 13.5. Finally, Section 13.6 contains our concluding remarks.

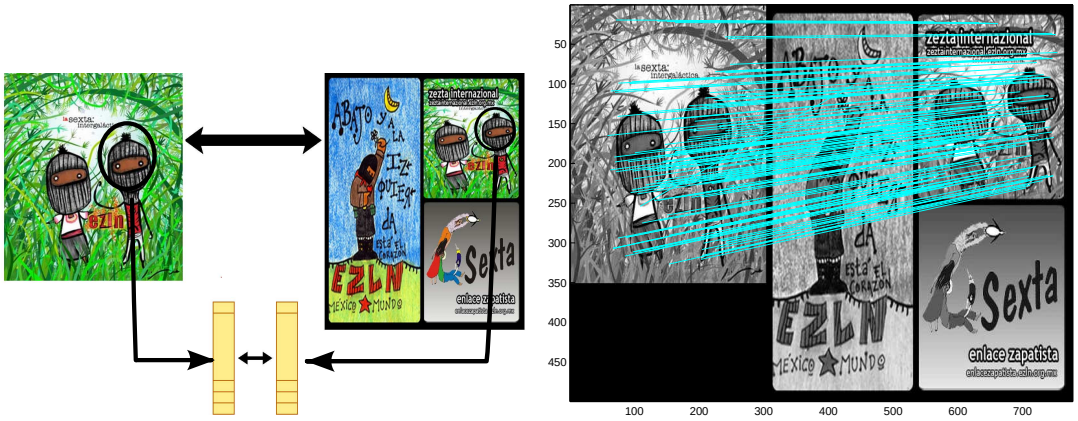
13.2 Local image descriptors

This section presents a brief overview of the state-of-the-art in local image descriptors, and describes a common approach for evaluation and comparison.

13.2.1 Previous work

Computer vision literature that focuses on the detection and description of local features has grown rapidly over the last ten years. A comprehensive review on these topics can be found in TUYTELAARS et MIKOLAJCZYK [2008] for detection algorithms, and in MIKOLAJCZYK et SCHMID [2005] for description methods. Keeping to the latter, it is possible to identify four main groups of methods: distribution-based, spatial-frequency techniques, differential descriptors, and others. In this discussion we will only deal with the first group, because they have shown to be better at extracting distinctive image information MIKOLAJCZYK et SCHMID [2005]. Distribution-based descriptors use histograms to represent local image shape or appearance. Currently it is widely accepted that the SIFT descriptor is the best histogram based method, and it is probably the most widely used approach in computer vision research. The SIFT descriptor is a 3D histogram of gradient locations and orientations, where the contribution to each bin in the histogram is weighted by the gradient magnitude.

The success of SIFT has prompted many researchers to propose variations and improvements over the basic SIFT method, for examples see BAY et collab. [2008]; KE et SUKTHANKAR [2004]; MIKOLAJCZYK et SCHMID [2005]. However, despite its success, there is one practical drawback to SIFT, it is a complex and relatively slow algorithm. Therefore, it is not a simple task to reproduce the original code, and it is not feasible to use SIFT in demanding applications that require real-time output CALONDER et collab. [2008]. Therefore, several researchers have proposed to reduce the dimensions of the SIFT vector KE et SUKTHANKAR [2004] or to use simplified implementations BAY et collab. [2008].



(a) Single match (b) All of the matches

Figure 13.2 – The matching process with local descriptors.

13.2.2 Evaluation method

Let us now return to the topic of establishing an overal performance measure for local image descriptors. In this respect we follow MIKOLAJCZYK et SCHMID [2005], that bases the evaluation on recall and 1-precision curves. The problem used for evaluation is the matching of local regions between two different images of the same scene; see Figure 13.2.

For example, region $\mathbf{A}$ from image I_1 is matched with region $\mathbf{B}$ from image I_2 if the Euclidean distance between their corresponding descriptors, $D_{\mathbf{A}}$ and $D_{\mathbf{B}}$, is below a certain threshold h, and if $D_{\mathbf{B}}$ is the nearest neighbor of $D_{\mathbf{A}}$. When the geometric transformation between I_1 and I_2 is known beforehand, then it is possible to determine if each match is correct MIKOLAJCZYK et SCHMID [2005]. From this, recall and 1-precision values can be easily obtained using

$$recall = \frac{\#\ correct\ matches}{\#\ true\ correspondences}\ ,$$

$$1 - precision = \frac{\#\ false\ matches}{\#\ correct\ matches + \#\ false\ matches}\ .$$

A performance curve for a descriptor can be built by varying the matching threshold h. In this work, we use twenty different values, following MIKOLAJCZYK et SCHMID [2005].

However, evaluating descriptors in this way does have some limitations. More notably, it causes ambiguities when the curves of two different descriptors intersect. In order to simplify the comparison between two curves we could use the F-Measure, as done in PEREZ et OLAGUE [2009]. The F-Measure is a concept commonly used in information retrieval, it gives an estimation of the accuracy of a test; a perfect accuracy would produce an F-Measure equal to one, and zero in the opposite case. The F-Measure is defined as,

$$F(precision, recall)_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{\beta^2\cdot\text{precision} + \text{recall}}\ , \tag{13.1}$$

where if $\beta = 1$ we obtain a symmetric balance between precision and recall.

13.2.3 Optimization of the detection/description methods

The large number of proposed methods for detecting and describing locally salient features has necessitated the development of experimental evaluation methods such as the one described above. The goal of such measures of performance was to provide objective criteria that could be used to make an informed decision when choosing a method for a particular vision application. On the other hand, these measures have also facilitated the development of

automatic design algorithms that automatically synthesize detection and description methods. For example, SCHMID et collab. [2000] proposed a performance measure for interest point detection based on point repeatability. Afterwards, this measure has been used to pose single TRUJILLO et OLAGUE [2006, 2008] and multi-objective TRUJILLO et collab. [2008a] optimization problems that search for optimal interest point detectors. The evolved detectors achieve state-of-the-art performance, in other words they are human competitive results. Another example is the proposal made in PEREZ et OLAGUE [2009], that uses the evaluation method described above to synthesize a novel weight operator for the SIFT algorithm. Whereas SIFT uses the gradient magnitude to weigh the contributions made to each bin in the histogram, the weight operators evolved with GP produced significant performance gains. It is of interest to note that previous proposals to enhance the SIFT descriptor failed to notice that the weighting function could, or should, be improved. Indeed, the original proposal seems reasonable to a human expert; fortunately, however, evolution need not be hindered by such *reasonable assumptions*, evidenced by the counter-intuitive operators found by the GP search PEREZ et OLAGUE [2009]. From this it follows that further inquiry is still necessary regarding the development of new local descriptors. In this sense, we argue that the Hölder descriptor could prove to be a viable alternative for future work in this area TRUJILLO et collab. [2007].

13.3 The Hölder descriptor

In this section the concepts of local regularity and the Hölder exponent are introduced, and then the Hölder-based descriptor is described in detail.

13.3.1 Holderian regularity

It is known that most of the useful information contained within a signal is located within the irregular or singular regions. In images, for instance, such regions correspond with edges, corners and interest points. Hölderian regularity provides a characterization of such singular structures [MALLAT, 1999]. It can be quantified, for example, by the pointwise Hölder exponent which is defined as follows.

Definition 1:Let $f : \mathbb{R} \to \mathbb{R}$, $s \in \mathbb{R}^{+*} \setminus \mathbb{N}$ and $x_0 \in \mathbb{R}$. $f \in C^s(x_0)$ if and only if $\exists \eta \in \mathbb{R}^{+*}$, and a polynomial P of degree $< s$ and a constant c such that

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \leq c|x - x_0|^s \,, \tag{13.2}$$

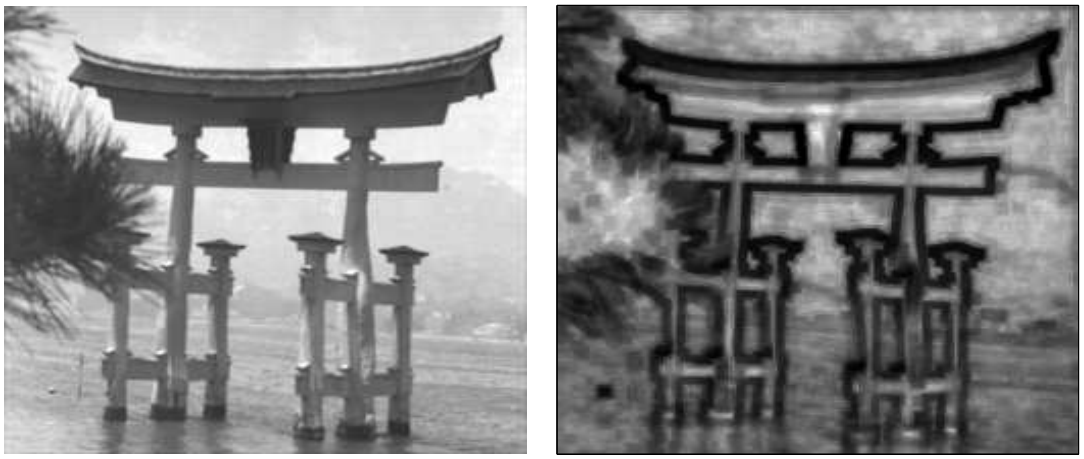
where $B(x_0, \eta)$ is the local neighborhood around x_0 with a radius η. The pointwise Hölder exponent of f at x_0 is $\alpha_p(x_0) = \sup_s \{f \in C^s(x_0)\}$.

Hölderian regularity refines the concept of the Taylor series approximation of a function by also accounting for non-differentiable points [MALLAT, 1999]. The pointwise Hölder exponent has proven to be useful in several tasks of image analysis, such as noise removal LEGRAND et VEHEL [September 14-17, 2003], interpolation LEGRAND et LÉVY-VÉHEL [2003], and edge detection LÉVY-VÉHEL [1998]. However, it can only be computed analytically for a small set of signals. Therefore, in order to use Hölderian regularity the exponents must be estimated. Here we review the oscillations method for estimation, which is directly derived from the definition given above TRICOT [1995].

13.3.1.1 Estimation through oscillations

The Hölder exponent of function $f(t)$ at t is the $sup(\alpha_p) \in [0, 1]$, for which a constant c exists such that $\forall\, t'$ in a neighborhood of t,

$$|f(t) - f(t')| \leq c|t - t'|^{\alpha_p} \,. \tag{13.3}$$



(a) Original Image (b) Hölder Image

Figure 13.3 – The pointwise Hölder exponent.

In terms of signal oscillations, a function $f(t)$ is Hölderian with exponent $\alpha_p \in [0,1]$ at t if $\exists c$ $\forall \tau$ such that $osc_\tau(t) \leq c\tau^{\alpha_p}$, with

$$osc_\tau(t) = \sup_{t',t'' \in [t-\tau, t+\tau]} |f(t') - f(t'')| \,. \tag{13.4}$$

Now, if $t = x_0$ and $t' = x_0 + h$ in 13.3, we can also write that

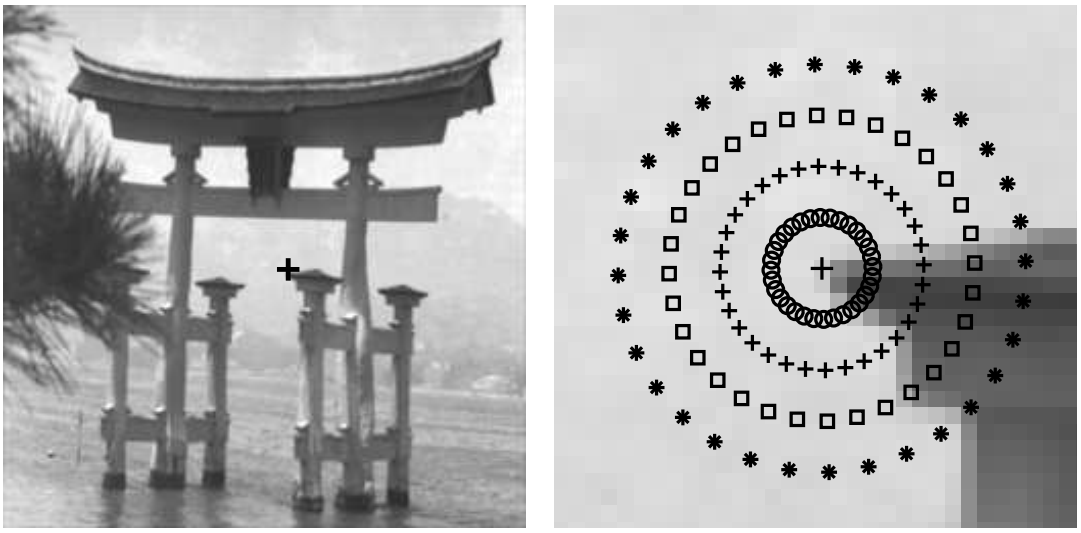
$$\alpha_p(x_0) = \text{liminf}_{\,h \to 0} \frac{\log |f(x_0+h) - f(x_0)|}{\log |h|} \,. \tag{13.5}$$

Therefore, the problem is that of finding an α_p that satisfies 13.3 and 13.4, and in order to simplify this process we can set $\tau = \beta^r$. Then, we can write $osc_\tau \approx c\tau_p^\alpha = \beta^{(\alpha_p r + b)}$, which is equivalent to $log_\beta(osc_\tau) \approx \alpha_p r + b$.

An estimation of the regularity can be built at each point by computing the slope of the regression between the logarithm of the oscillations osc_τ and the logarithm of the dimension of the neighborhood at which the oscillations τ are computed; we use least squares regression with $\beta = 2$ and $r = 1, 2, \ldots, 7$. Also, it is preferable not to use all sizes of neighborhoods between two values τ_{min} and τ_{max}. Hence, we calculate the oscillation at point x_0 only on intervals of the form $[x_0 - \tau_r : x_0 + \tau_r]$. For a 2D signal, x_0 defines a point in 2D space and τ_r a radius around x_0, such that $d(t', t) \leq \tau_r$ and $d(t'', t) \leq \tau_r$, where $d(a, b)$ is the Euclidean distance between a and b. Figure 13.3 presents the estimation of the Hölder exponent for an image. This method has proven to be superior in some cases to the wavelet leaders method JAFFARD [2004]; LEGRAND [2004], and it is used to construct the Hölder descriptor we introduce below.

13.3.2 Hölder descriptor

The descriptor based on Hölderian regularity is very simple and easy to construct TRUJILLO et collab. [2007]. The idea is to uniformly sample the value of the Hölder exponent using a circular grid within each region. For instance, Figure 13.4a shows an interest point detected within a test image, and Figure 13.4b presents the local region around it. The descriptor is then constructed by sampling the exponent of the central point and at 32 equidistant points at four different radii, this gives a vector dimension of 129, Figure 13.4b illustrates this process. The Hölder descriptor has two useful properties for region description. First, because the exponent is estimated using oscillations which are relative intensity differences within the region, there is no need to normalize the descriptor for uniform intensity variations. Second, rotation invariance can be obtained by ordering the values in the descriptor based



(a) Interest point (b) Sample points

Figure 13.4 – Sampling used with the Hölder descriptor.

on the principal orientation of the gradient within each region. Additionally, it compares favorably with SIFT in two important aspects. On the one hand, experimental results have confirmed that the Hölder descriptor can achieve comparable results on standard tests TRUJILLO et collab. [2007]. On the other hand, the Hölder descriptor is constructed using a much simpler algorithm, this makes it easy to implement and replicate.

The biggest drawback of the Hölder descriptor is the high computation time it requires. Similar to SIFT, it is not feasible to use the Hölder descriptor in real-time. However, unlike SIFT, because the algorithm used to build the descriptor is very simple, there are two obvious ways in which to speed-up the process. One option is to devise a faster estimation method, however this is not a trivial task and is left for future work. Another option is to reduce the number of sample points used to build the descriptor. In KE et SUKTHANKAR [2004] the dimensions of the SIFT descriptor were reduced through PCA, and they showed that some of the dimensions were not necessary to uniquely describe a region. Similarly, we suggest that this is a real possibility for the Hölder descriptor, and we expect that the optimal number of sample points might not be the 129 points used by the canonical version of the descriptor.

13.4 The search problem and the proposed solution

Given the above arguments, the goal of this work is to find the optimal set of sample points that should be used to build the Hölder descriptor. If we set the maximum number of points to the original 129 used by the canonical descriptor, then we can propose a combinatorial search problem where the goal is to find the optimal subset of these points using as objective the performance criteria presented in Section 13.2.2. Notice that the problem stated in this way only considers performance on the matching tests, it does not explicitly search for the smallest number of dimensions. The search problem was posed in this way for the following reasons. First, we are interested in finding the best possible descriptor, one that achieves a performance that is comparable to the canonical Hölder descriptor. In fact, if the search converges towards a descriptor of dimension ≈ 129, then so be it. This would confirm that the original descriptor was the best possible construction given the proposed sampling grid. Second, if we add a second objective, namely the size of the descriptor, then we are faced with a multi-objective problem. However, from a practical perspective a Pareto front of solutions would only add another level of analysis that goes beyond the goal of this work.

Parameter	Description and value
Representation	Binary string.
Population size	100.
Generations	100.
Selection	Fitness proportional.
Crossover	Mask crossover; $p_c = 0.9$.
Mutation	*Single bit mutation;* $p_\mu = 0.1$.
Survival	Elitism of the best 15%.
Training pairs	$N = 3$.

Table 13.1 – GA run-time parameters.

Name	Transformation	No. of Images
Nueva York	Rotation	35
Van Gogh	Rotation	17
Monet	Rotation	18
Graph	Illumination	12
Mosaic	Illumination	18

Table 13.2 – Image sequences used to evaluate the performance of the H-GA descriptor

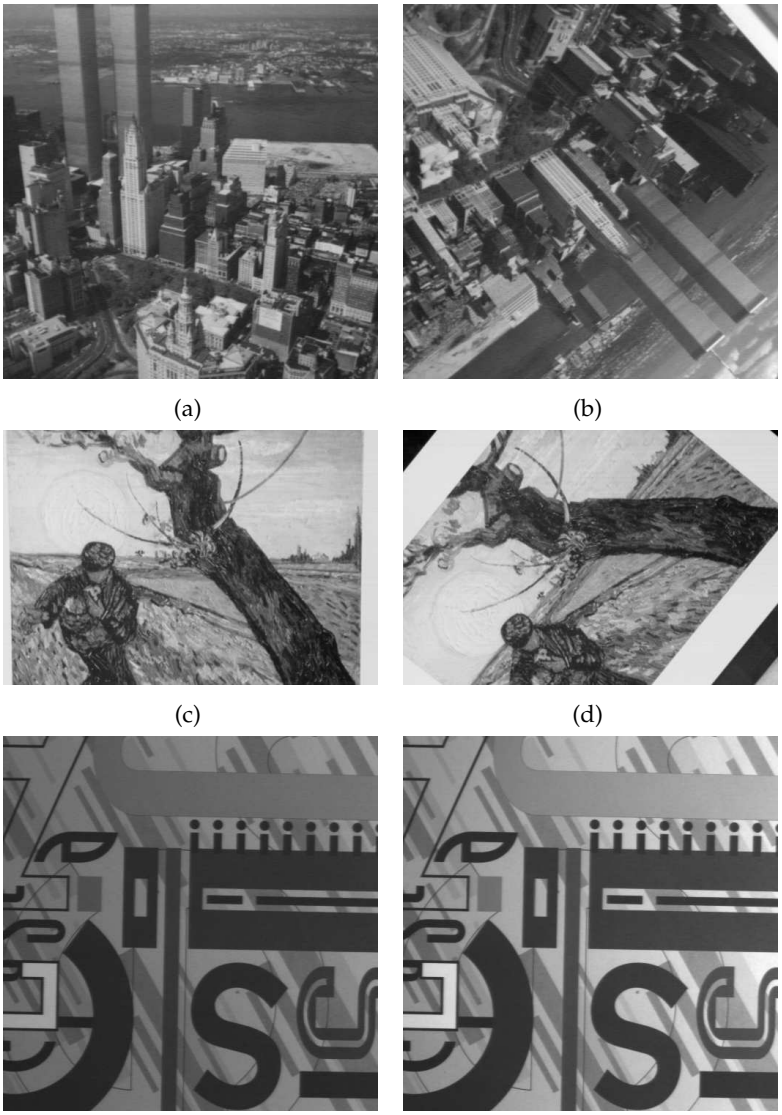
13.4.1 The genetic algorithm

The problem described above is a combinatorial search in 129 dimensions, exactly the kind of search problem in a high-dimensional space in which a GA thrives. Therefore, the task of choosing which points will be used to build the descriptive vector is assigned to a GA. Figure 13.4b shows the original sample points used by the Hölder descriptor, these act as the upper bound for the GA search. Therefore, the chromosome of each individual is expressed as a binary string $B = (b_1, b_2, ...b_{129})$ of 129 bits. Each bit is associated with one of the admissible sample points. Hence, when a bit is set to 1 then the corresponding sample point is used to build the descriptor. Conversely, if a bit is zero then the sample point is not considered.

The fitness of each individual is based on the F-Measure, given in Equation 13.1. It is important to note that each F_β value depends on a single recall/1-precision pair, and that the recall/1-precision curves contain 20 such points. Therefore, the mean value of the F-Measure $\overline{F}_\beta$ is used to characterize each performance curve. Moreover, if N pairs of images are used to train each individual, we then have the same number of recall/1-precision curves and corresponding $\overline{F}_\beta$s. The final fitness measure for each individual is posed for as a minimization task with the following cost function,

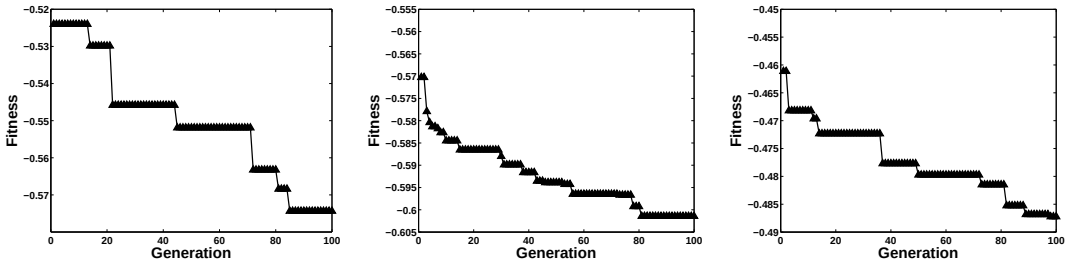
$$f(B) = -\frac{1}{N}\sum_{i=1}^{N}\overline{F}_\beta^i \; . \tag{13.6}$$

Note that the cost function does not explicitly favor individuals that produce smaller descriptors. The assumption is that evolution will be less constrained, and will tend to favor individuals that produce the best performance on the matching tests. The goal is to find the optimal subset of sample points. The GA was configured using the parameters shown in Table 13.1, from which we can see that the algorithm is basically standard. The fitness function uses three pairs of images for training, each pair is shown in Figure 13.5. Using this setup each run required between eight and ten hours of computation time, and a total of twenty runs of the algorithm were carried out in order to validate the performance of the proposal. Despite the long run-times, it is important to remember that this should be considered as a training step, and normal use of the final solutions does not require similar executions.



(a) (b) (c) (d) (e) (f)

Figure 13.5 – Training pairs of images used to assign fitness, each pair has a reference and a transformed image. (a,b) New York image with rotation transformation; (c,d) Van Gogh image with rotation; and (e,f) Graph image with illumination change.

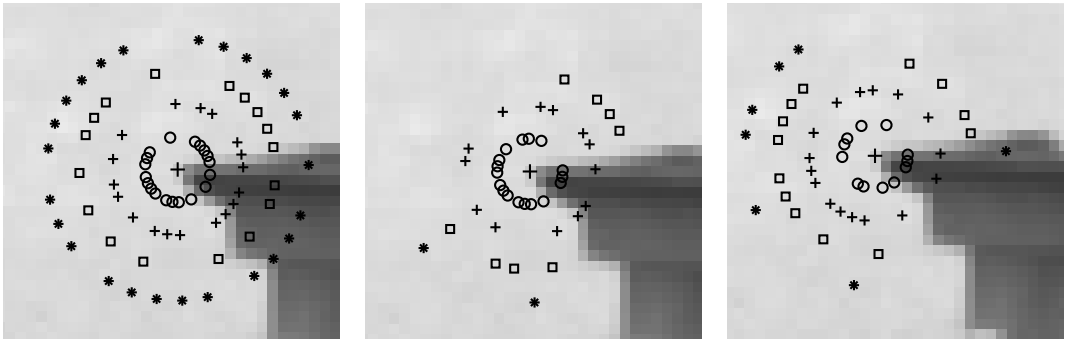


(a) (b) (c)

Figure 13.6 – The convergence graphs of the three best experiments.

13.5 Experiments and results

In this section we present the best three results found using the proposed GA search. Figure 13.6 shows the convergence plots of the best fitness at each generation for each of the three best experiments, (a), (b) and (c). In all three cases we can see a steady and progressively



(a) 80 points (b) 40 points (c) 48 points

Figure 13.7 – The phenotype of the best individual from each run. The figure shows the sample points that are used to build the descriptor.

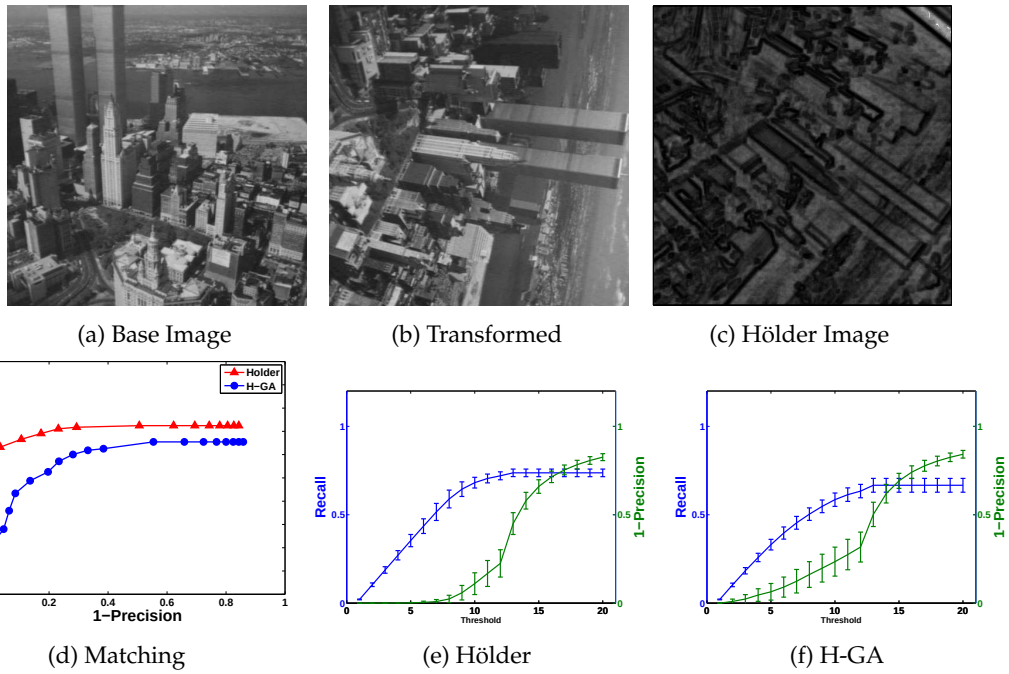
improving convergence.

Then, in Figure 13.7 we show the phenotype of the best individual found in each of these runs. For comparison, the sample points chosen by each individual are marked using the same region presented earlier in Figure 13.4, and the total number of points is specified.

The first observation we can make is that even if all three of the solutions are different, they all construct a smaller vector than the canonical Hölder descriptor. For instance, the best solution from run (a) uses 75% of the original sample points, the best solution from run (b) uses only 31%, and the solution found in run (c) uses only 37% of the maximum number of sample points. In all three cases the reduction in descriptor size is significant, particularly for runs (b) and (c).

However, the reduction in size and the good fitness scores do not imply that the descriptors will achieve a high level of performance on a wider variety of test cases. Therefore, the performance of the evolved descriptors must be validated on more images and compared relative to the performance of the canonical Hölder descriptor. For such a comparison we use similar criteria as those used in MIKOLAJCZYK et SCHMID [2005]; TRUJILLO et collab. [2007], testing on different image pairs and testing over complete image sequences which contain a base image and a series of progressively transformed images. However, in order to simplify the following discussion, we only present the best solution found in all of the runs, which was the sampling pattern obtained in run (b), which we denote as $H - GA$ (Hölder descriptor with Genetic Algorithm).

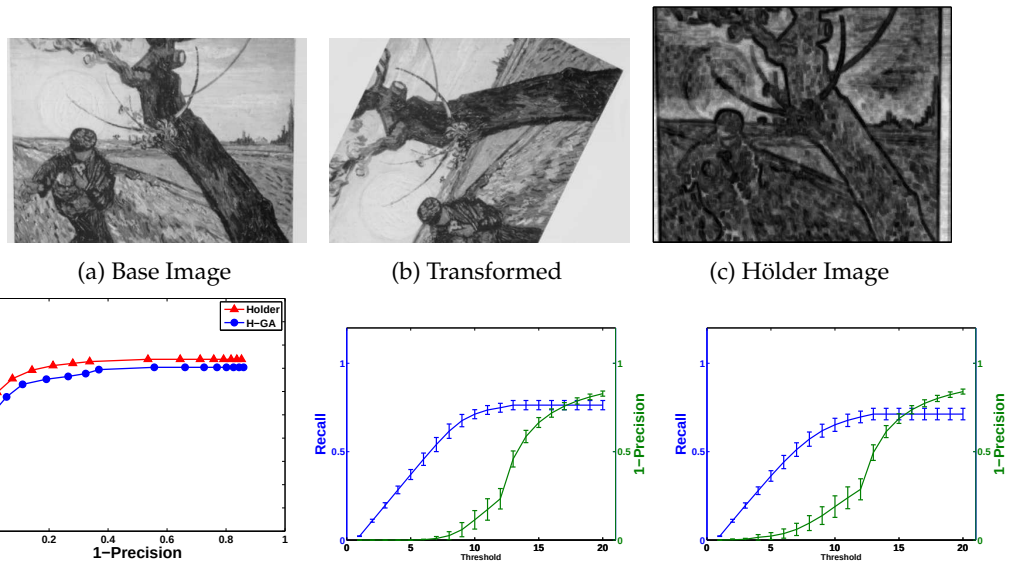
Before we compare with the canonical Hölder descriptor a few comments are necessary. First, solutions (a) and (c) were inferior to (b) based on their performance plots obtained with the matching tests. Therefore, we can say that the algorithm sometimes converged towards local-optima. Second, surprisingly the best performance was obtained using the solution that uses the least amount of sample points, which is strong evidence that suggests that the dimensions of the Hölder descriptor can be significantly reduced. Moreover, it is informative to see the spatial distribution of sample points suggested by the $H - GA$ descriptor, see Figure 13.7b. The test region contains a typical corner structure, with the actual corner nearly corresponding with the center of the detected region. It is evident that most of the points are located very close to the center of the region, in the first two concentric rings. This result is consistent with the SIFT algorithm, where the contribution that each point has towards building the descriptor is inversely proportional with its distance to the central point. Thus, the descriptor is building a description of the local region that heavily relies on the appearance of the central part of the region. On the other hand, the points that are sampled in the final two rings, farthest away from the central point, seem to be distributed in an almost symmetric manner along a tangential line to the corner. The distribution of points does not seem to be arbitrary, because the organization is relative towards the principal direction of the gradient within the region. Moreover, most points appear almost entirely outside of the



(a) Base Image (b) Transformed (c) Hölder Image

(d) Matching (e) Hölder (f) H-GA

Figure 13.8 – Comparison for the New York sequence with rotation transformation.



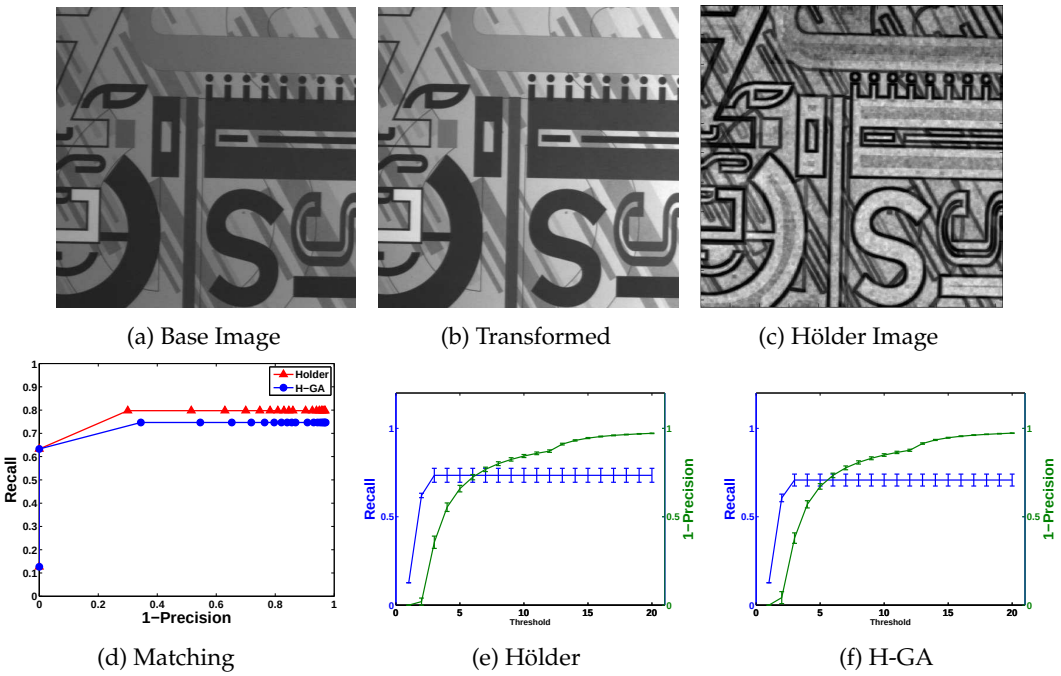
(a) Base Image (b) Transformed (c) Hölder Image

(d) Matching (e) Hölder (f) H-GA

Figure 13.9 – Comparison for the Van Gogh sequence with rotation transformation.

inner surface of the corner structure. This is consistent with the assumption that most corner structures will tend to be homogeneous, and that discriminative information contained around a corner will not be within its inner flat structure.

Now, the comparison between $H - GA$ and the canonical Hölder descriptor is presented in Figures 13.8, 13.9, 13.10, 13.11 and 13.12. In each test we compare the descriptors using a sequence of progressively transformed images and show the base image, one test image, a sample Hölder image, the corresponding recall vs. 1-precision curve between the base and test image, and two plots that show the average performance of each descriptor computed for the complete test sequence; these last two plots require further explanation. The plots have a double y-axis that show the average recall and 1-precision scores computed for all of the images in each sequence, they also show the standard deviation for these measures. The x-axis in these plots corresponds with the different thresholds used for matching, see

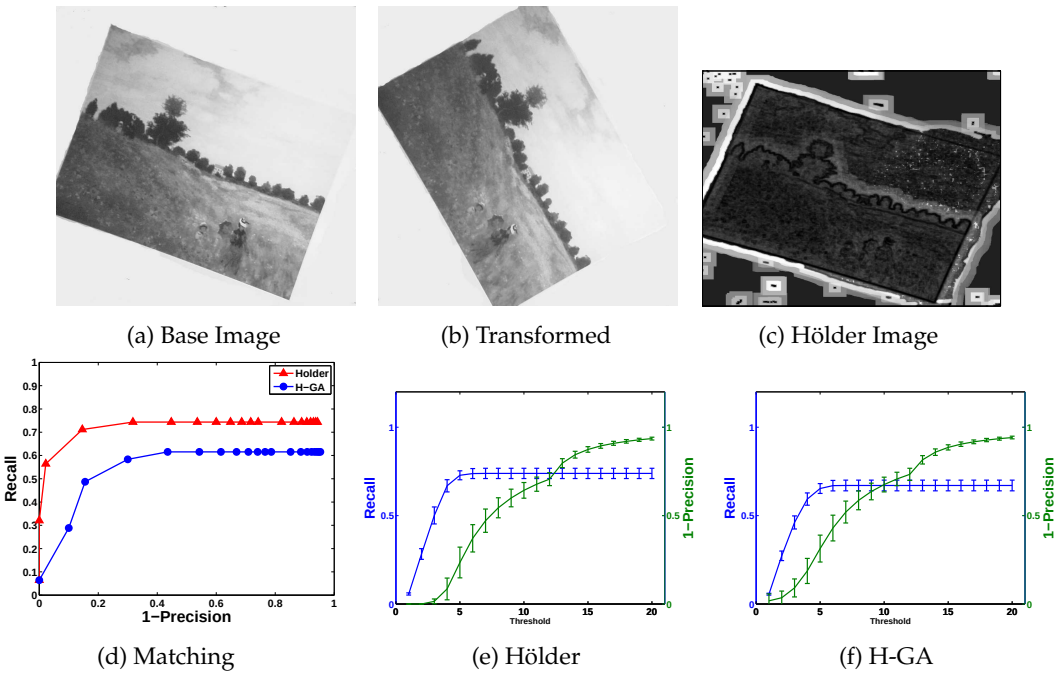


(a) Base Image (b) Transformed (c) Hölder Image

(d) Matching (e) Hölder (f) H-GA

Figure 13.10 – Comparison for the Graph image sequence with illumination change.

Section 13.2.2. In these plots an optimal performance is a horizontal recall curve close to one, and horizontal 1-precision curve close to zero. Table 13.2 summarizes the image sequences used to perform our experimental comparisons, it gives the name of the sequence, the type of image transformation and the number of images in each sequence.

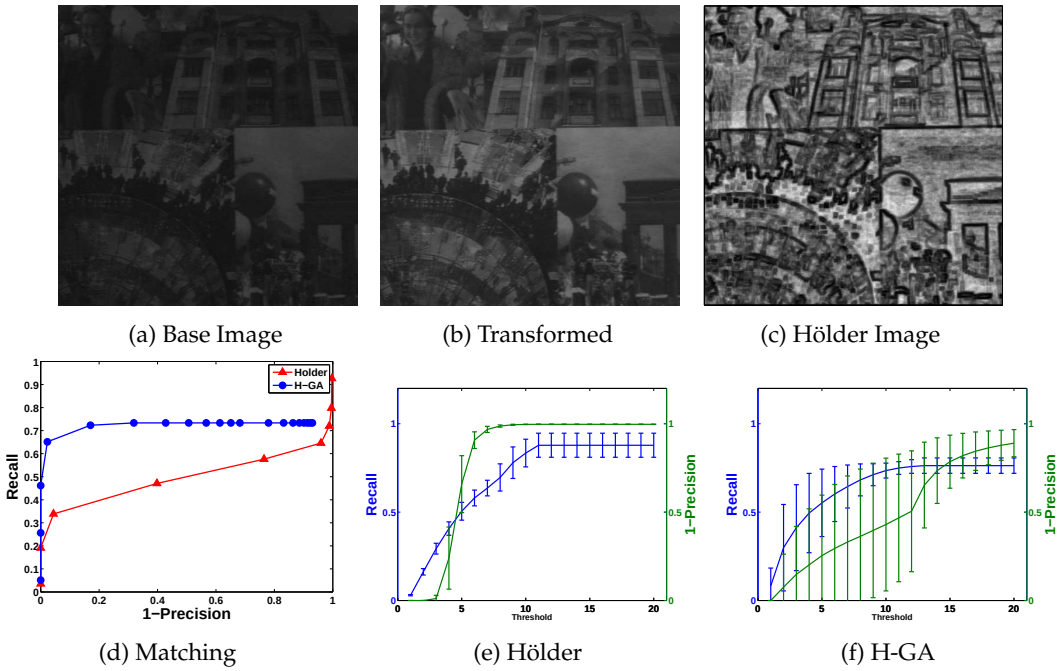


(a) Base Image (b) Transformed (c) Hölder Image

(d) Matching (e) Hölder (f) H-GA

Figure 13.11 – Comparison for the Monet image sequence with rotation transformation.

Figures 13.8, 13.9, 13.10 use the same image sequences from which the three training pairs were obtained. However, in all cases the test image that is used is different than the one that was used to compute the value of the cost functions, those are shown above in Figure 13.5. Several observations are pertinent here. First, we can appreciate that in some cases the performance between both descriptors is very similar, see Figures 13.9 and 13.10.



(a) Base Image (b) Transformed (c) Hölder Image

(d) Matching (e) Hölder (f) H-GA

Figure 13.12 – Comparison for the Mosaic image sequence with illumination change.

Particularly, we can see that on average the performance is very similar over the complete sequences. In other cases, namely in Figure 13.8 and 13.11, the canonical descriptor is better. And still yet, in Figure 13.12 the H-GA achieves a better performance. However, as is obvious from previous comparative works MIKOLAJCZYK et SCHMID [2005] most of these differences should be negligible, and in the performance can be regarded as equivalent. Moreover, an important part of these comparisons is the manner in which the recall/1-precision curves behave, not just the level of recall that they reach MIKOLAJCZYK et SCHMID [2005]. Obviously, a perfect descriptor would achieve a recall equal to 1 for any precision, however this should not be expected in a real-world test. Therefore, for practical purposes what is desired is a horizontal curve that achieves a high (above 0.6) recall that remains steady. Conversely, a slowly increasing curve shows that the descriptor is affected by the image degradation induced by the transformation; i.e., the descriptor is less invariant. Under such considerations, we can see that in fact the performance of both descriptors is quite similar. However, H-GA achieves these performance scores using nearly 70% less information than the canonical descriptor.

13.6 Summary and conclusions

This work addresses the problem of optimizing a descriptor for local image features. The study focuses on the Hölder descriptor because it achieves state-of-the-art performance, and because it relies on a very simple algorithm. The goal is to find the optimal set of sample points from which to compute the Hölder exponent and construct the Hölder descriptor. This task is posed as a combinatorial search problem and solved using a genetic algorithm. Fitness depends on the F-Measure of the descriptor computed on standard tests of region matching. The GA search produced several solutions that produce much more compact region descriptors. In fact, the best solution found by the GA, here called H-GA, uses only 31% of the dimensions from the canonical version of the descriptor and still achieves a similar performance. This suggests that the problem we have posed is multi-modal and that at least two optima exist, one of which is significantly more compact, and more efficient, than the other.

References

BALA, J., K. D. JONG, J. HUANG, H. VAFAIE et H. WECHSLER. 1996, «Using learning to facilitate the evolution of features for recognizing visual concepts», *Evol. Comput.*, vol. 4, nº 3, doi:http://dx.doi.org/10.1162/evco.1996.4.3.297, p. 297–311, ISSN 1063-6560. 281

BAY, H., A. ESS, T. TUYTELAARS et L. V. GOOL. 2008, «Speeded-up robust features (surf)», *Comput. Vis. Image Underst.*, vol. 110, nº 3, doi:http://dx.doi.org/10.1016/j.cviu.2007.09.014, p. 346–359, ISSN 1077-3142. 281

CALONDER, M., V. LEPETIT et P. FUA. 2008, «Keypoint signatures for fast learning and recognition», dans *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-540-88681-5, p. 58–71, doi:http://dx.doi.org/10.1007/978-3-540-88682-2_6. 281

HERNÁNDEZ, B., G. OLAGUE, R. HAMMOUD, L. TRUJILLO et E. ROMERO. 2007, «Visual learning of texture descriptors for facial expression recognition in thermal imagery», *Computer Vision and Image Understanding*, vol. 106, nº 2-3, doi:http://dx.doi.org/10.1016/j.cviu.2006.08.012, p. 258–269, ISSN 1077-3142. 281

JAFFARD, S. 2004, «Wavelet techniques in multifractal analysis», dans *Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics*, vol. 72, p. 91–151. 284

KE, Y. et R. SUKTHANKAR. 2004, «Pca-sift: A more distinctive representation for local image descriptors», dans *Proceedings of CVPR*, vol. 2, IEEE Comp. Soc., ISSN 1063-6919, p. 506–513, doi:http://doi.ieeecomputersociety.org/10.1109/CVPR.2004.183. 281, 285

LEGRAND, P. 2004, *Debruitage et interpolation par analyse de la regularite Hölderienne. Application a la modelisation du frottement pneumatique-chaussee*, thèse de doctorat, Université de Nantes, France. 284

LEGRAND, P. et J. LÉVY-VÉHEL. 2003, «Local regularity-based interpolation», dans *WAVELET X, Part of SPIE's Symposium on Optical Science and Technology*, vol. 5207. 283

LEGRAND, P. et J. L. VEHEL. September 14-17, 2003, «Local regularity - based image denoising», *ICIP03, Spain, IEEE International Conference on Image Processing*, p. 377–380. 283

LÉVY-VÉHEL, J. 1998, *Fractal Image Encoding and Analysis*, chap. Introduction to the Multifractal Analysis of Images, p. 299–341. 283

LOWE, D. G. 1999, «Object recognition from local scale-invariant features.», dans *Proceedings of ICCV*, vol. 2, IEEE Computer Society, p. 1150–1157. 280

MALLAT, S. 1999, *A wavelet tour of signal processing*, 2e éd., Elsevier, San Diego, CA, 637 p.. 283

MIKOLAJCZYK, K. et C. SCHMID. 2005, «A performance evaluation of local descriptors», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, nº 10, doi:http://dx.doi.org/10.1109/TPAMI.2005.188, p. 1615–1630, ISSN 0162-8828. 280, 281, 282, 288, 291

PEREZ, C. B. et G. OLAGUE. 2009, «Evolutionary learning of local descriptor operators for object recognition», dans *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, ISBN 978-1-60558-325-9, p. 1051–1058. 281, 282, 283

SCHMID, C. et R. MOHR. 1997, «Local grayvalue invariants for image retrieval», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, nº 5, p. 530–534. URL http://perception.inrialpes.fr/Publications/1997/SM97. 280

SCHMID, C., R. MOHR et C. BAUCKHAGE. 2000, «Evaluation of interest point detectors», *International Journal of Computer Vision*, vol. 37, nº 2, doi:http://dx.doi.org/10.1023/A:1008199403446, p. 151–172, ISSN 0920-5691. 280, 283

SIEDLECKI, W. et J. SKLANSKY. 1989, «A note on genetic algorithms for large-scale feature selection», *Pattern Recogn. Lett.*, vol. 10, nº 5, doi:http://dx.doi.org/10.1016/0167-8655(89)90037-8, p. 335–347, ISSN 0167-8655. 281

SUN, Z., G. BEBIS et R. MILLER. 2004, «Object detection using feature subset selection», *Pattern Recognition*, vol. 37, nº 11, p. 2165–2176. 281

TRICOT, C. 1995, *Curves and Fractal Dimension*, Springer-Verlag, ISBN 0387940952. 283

TRUJILLO, L. et G. OLAGUE. 2006, «Synthesis of interest point detectors through genetic programming», dans *Proceedings of GECCO*, vol. 1, édité par M. Cattolico, ACM, p. 887–894. 280, 283

TRUJILLO, L. et G. OLAGUE. 2007, «Scale invariance for evolved interest operators», dans *Proceedings of EvoIASP 2007*, LNCS, Springer-Verlag, p. 423–430. 280

TRUJILLO, L. et G. OLAGUE. 2008, «Automated design of image operators that detect interest points», *Evolutionary Computation*, vol. 16, nº 4, p. 483–507. 283

TRUJILLO, L., G. OLAGUE, P. LEGRAND et E. LUTTON. 2007, «Regularity based descriptor computed from local image oscillations», *Optics Express*, vol. 15, p. 6140–6145. 280, 283, 284, 285, 288

TRUJILLO, L., G. OLAGUE, E. LUTTON et F. FERNÁNDEZ DE VEGA. 2008a, «Multiobjective design of operators that detect points of interest in images», dans *Proceedings of GECCO*, édité par M. Cattolico, ACM, New York, NY, USA, ISBN 978-1-60558-130-9, p. 1299–1306, doi:http://doi.acm.org/10.1145/1389095.1389344. 283

TRUJILLO, L., G. OLAGUE, F. FERNÁNDEZ DE VEGA et E. LUTTON. 2008b, «Selecting local region descriptors with a genetic algorithm for real-world place recognition», dans *Proceedings of EvoIASP 2008*, LNCS, p. 325–334. 281

TUYTELAARS, T. et K. MIKOLAJCZYK. 2008, «Local invariant feature detectors: a survey», *Found. Trends Comput. Graph. Vis.*, vol. 3, nº 3, doi:http://dx.doi.org/10.1561/0600000017, p. 177–280, ISSN 1572-2740. 280, 281

Chapter 14

Interactive evolution for cochlear implants fitting

This chapter has been published in the journal paper Genetic Programming and Evolvable Machine, GPEM in 2007. Work carried out with Claire Bourgeois-Republique, Vincent Péan, Esther Harboun Cohen, Jacques Levy-Vehel, Bruno Frachet, Evelyne Lutton and Pierre Collet.

Contents

Abstract

Cochlear implants are devices that become more and more sophisticated and adapted to the need of patients, but at the same time they become more and more difficult to parameterize. After a deaf patient has been surgically implanted, a specialised medical practitioner has to spend hours during months to precisely fit the implant to the patient. This process is a complex one implying two intertwined tasks: the practitioner has to tune the parameters of the device (optimisation) while the patient's brain needs to adapt to the new data he receives (learning). This chapter presents a study that intends to make the implant more adaptable to environment (auditive ecology) and to simplify the process of fitting. Real experiments on volunteer implanted patients are presented, that show the efficiency of interactive evolution for this purpose.[1]

14.1 Introduction

Cochlear Implants (CI) allow totally deaf people to hear again provided their auditory nerve and cochlear are still functional: a computer processes sounds picked up from a microphone, to stimulate directly the auditory nerve through several electrodes inserted inside the cochlea (cf. fig. 14.1).

As one can imagine, there are hundreds of parameters that can be tuned, and in the same time the patient has to learn to "hear" using new informations provided to his auditory nerve. The tuning of such a device is thus extremely complex, and highly dependent on the patient. This process is currently done "by hand" by medical practitioners, and looks like an optimisation process based on "trial and error." This process is so delicate that sometimes, no satisfactory fitting can be found for some patients.

Hence, it seems interesting to use an interactive evolutionary algorithm (IEA) to help finding the best values for implant parameters. This is the main topic of the HEVEA project, which is a collaboration between computer scientists, signal processing experts and medical researchers. The aim is actually twofold: to facilitate the initial fitting of cochlear implants, and to automatise the adaptation of cochlear implants to various sound environments. A simple IEA was developed with this in mind, and tested on a very basic feature, the range of intensities that a specific electrode can take when stimulating the auditory nerve. The IEA has been implemented on a PDA and tests have been performed on volunteering patients with satisfying results.

The chapter is organised as follows: section 14.2 presents cochlear implants, and section 14.3 describes how they are currently tuned by medical practitioners. The approach of the HEVEA project is developed in section 14.4, and a first implementation of an IEA is detailed in section 14.5. Experiments on several patients are reported in section 14.6, yielding good results as well as important conclusions on manual fitting procedures. This first validation step is important: an analysis of the success and failures raises new questions that are developed in section 14.7, related to the well-known "user fatigue" problem of IEAs, and to the fact that different sound environments have an important influence on the fitting of implants. Automatic adaptation of the device to sound has been investigated, based on a sound signal classification scheme, which is detailed in section 14.7. Conclusions and perspectives are described in section 14.8.

14.2 Cochlear Implants

A cochlear implant is a surgically implantable device GALLEGO et collab. [1998] that provides hearing sensations to individuals with severe to profound hearing loss, who cannot

[1]This work has partially been funded by the French ANR - RNTS HEVEA project 04T550

benefit from hearing aids. In a normal ear, sound energy is converted to mechanical energy by the middle ear, which is then converted to electrical impulses by the inner ear (see figure 14.1). In order to perform this last stage, the cochlea (part of the inner ear) contains a fluid which is set into motion by the oval window which is connected to the middle ear. Within the cochlea, sensory cells (inner and outer hair cells) are sensitive transducers that convert the mechanical fluid motion into electrical impulses conveyed to the brain by the auditory nerve. Cochlear implants are designed to be a substitute for the middle ear, cochlear mechanical motion, and sensory cells, transforming directly sound energy into electrical energy that will initiate impulses in the auditory nerve COHEN [1989]; MOORE [1995] thanks to a digital signal processor.

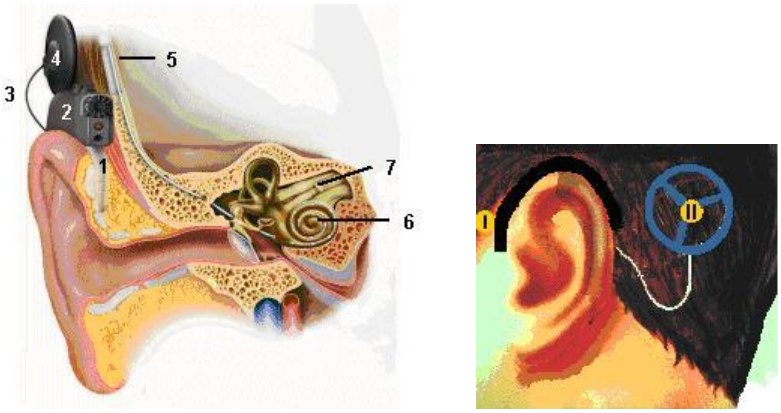


Figure 14.1 – All implant devices have the following features in common : sound is collected by a microphone (1) and sent to electronic components within a speech processor (2). The speech processor analyzes the input signal (sound) and converts it into an electronic signal (electrical). This code travels along a cable (3) to the transmitting coil (4) and is sent across the skin via frequency modulated (FM) electro-magnetic waves to the implant package (5). Based on characteristics of the code transmitted to the internal device, electrode contacts within the cochlea (6) provide electrical stimulation to the spiral ganglion cells and dendrites extending into the modiolus. Electrical impulses then travel along the auditory nerve (7), ascending auditory pathways to the brain.

Cochlear implants have been very successful in restoring partial hearing to profoundly deaf people ARCHBOLD et collab. [1995]; OSBERGER [1997]. In 2006, around 70 000 deaf people are implanted with such devices around the world. Efficiency is quite variable, ranging from totally deaf patients that have fully recovered their audition and are capable to follow telephone conversations and enjoy music, to others who hear strange sounds they can't benefit from, to a point where they prefer to switch off the implant CHOUARD et collab. [1995]; ROMAN [1998].

For many people, it is still difficult to fully take advantage of the device because it is not easy to tune the parameters of digital signal processor and adjust them for the characteristics for each patient, since all patients are different (cause of deafness, number of years between total deafness and implantation, age, depth of electrode insertion,...).

Research has been going on since nearly 50 years ago on how to electrically stimulate the auditory nerve to give a totally deaf patient sound sensations LOIZOU et collab. [2000]. Even though the early devices stimulated the auditory nerve with one electrode only, some lucky patients managed to hear again and even understand speech. Nowadays, it is technologically possible to use more than one electrode, in order to stimulate more of the thousands of neurons the auditory nerve is made of CHOUARD et collab. [1983]; PIALOUX et collab. [1979]. However, the more electrodes, the more parameters to tune.

The cochlea is used to interface electrodes and the auditory nerve. The cochlea is a biological device that mainly allows to map different sound frequencies onto different neurons. It is shaped like a snail shell. Only long wavelengths (low frequency sounds) can reach the

far end of the cochlea, while short wavelengths (high frequency sounds) are stopped at the entrance of the cochlea. The idea is then for surgeons to use this frequency discriminator and insert into the cochlea a thin silicon wire, bearing an array of ring-shaped electrodes.

The array of electrodes (cf. fig. 14.1/6) is then connected to an antenna inserted under the skin of the patient, in a cavity created by the surgeon in the skull bone of the patient, just above his external ear (cf. fig. 14.1/5). On the outer side of the skin, the patient wears another antenna (centered over the inner antenna thanks to a powerful magnet (cf. fig. 14.1/4)) that is itself connected to a digital signal processor (DSP) which uses a microphone as input(cf. fig. 14.1/2).

When a sound is received by the microphone, the DSP processes it and sends electrical impulses to the external antenna, that are received by induction by the implanted antenna. when the microphone picks up a low frequency sound, the DSP will stimulate electrodes introduced deeply in the cochlea (that will make the patient hear a low pitch sound) while on the contrary, high pitch sounds received by the microphone will have the DSP stimulate electrodes closer to the entrance of the cochlea (that will make the patient hear a high pitch sound).

14.3 Cochlear Implant fitting

14.3.1 Complexity of the problem

Being able to use more than one electrode to stimulate different neuron areas is indeed a great improvement, but the number of parameters to tune increases drastically. Concerning electrodes only, many questions arise, among which:

- Which frequencies should be mapped to which electrodes ?
- Which range of intensities should be applied to which electrodes ?
- How many electrodes should be stimulated simultaneously ?
- Should the processor prohibit neighbour electrodes to be stimulated simultaneously in order to avoid diaphony (crosstalk between nearby electrodes) ?

Finding good answers to these questions is a difficult optimisation problem. This not only due to the extremely large size of the search space but to several other reasons. First of all, the quality of a fitting is a two stage process where subjectivity plays a large role: the practitioner has to interpret the quality of the fitting (second subjective process) from the answers given by the patient (first subjective process). The disparity of patient behaviour with respect to language and sensitivity to various thresholds, as well as the character of the practitioner deeply influences the results. For example the well known psychological "Pygmalion" effect biases answers of the patient, who often unconsciously tries to satisfy the practitioner's expectations.

The sound environment is another cause of variability of results, as the fitting session usually takes place in a small room at hospital with the practitioner. However the cochlear implant must also be used in real life, and a correct fitting at hospital may reveal very uncomfortable or unuseful when in the street, or in a restaurant.

Fatigue and brain adaptation are also other sources of trouble: it is impossible to test many possible parameter sets during a single session, so the process is very long and needs sometimes weeks to obtain a satisfying result. In the same time, a fitting that may not appear immediately as satisfying, may improve when testing it on a longer period (brain has a plasticity that cannot be neglected).

There are many factors that make this problem highly irregular. However, it has been proved that an acceptable or even good fitting is reachable by a manual search conducted

by an experienced practitioner. We describe below this manual fitting technique, which is mainly a human-guided "trial and error" process, resembling a local search.

14.3.2 Manual fitting

Nowadays, depending on the manufacturer, the number of electrodes varies between 8 and 22. Cochlear implant "fitting" is performed by an expert practitioner, who proceeds in the following way:

- Right after the surgical intervention, the practitioner tries to determine which electrodes are functional (an electrode is functional if the patient hears a sound when current is applied to the electrode).

- For each functional electrode, the practitioner tries to determine the range of intensities that can be used. The lowest intensity above which the patient perceives a sound is called T (for Threshold). The maximum comfortable intensity (loudest sound the patient can bear for a reasonable amount of time) is called C (for Comfort threshold).

  Determining the T and C values for each electrode takes time (communication with a deaf patient, a young child, or with an old patient can be difficult), and due to the increasing number of electrodes, some manufacturers now advise to determine T and C values for one every three or four electrodes, and extrapolate the values for the other electrodes. See ROUX [2001], HESSE [2002] for more informations on this topic.

  Other manufacturers even set average values for T and C, based on neural response or even statistics.

- Then, once the $C-T$ range is maximised for all the electrodes, the "real" fitting begins. The practitioner uses his expertise to map frequency bands logarithmically onto the different functional electrodes, and starts to tune the gain and sensitivity depending on sound frequencies, then tunes the number of simultaneously active electrodes,... while at the same time asking the patient whether they understand better or worse, whether the sound quality is comfortable or not, a.s.o.. Of course, misunderstandings between the practitioner and the deaf patient may occur here (elderly patients, small children, ...) that may affect the quality of the fitting. In certain cases, the practitioner will slightly reduce the $C-T$ range for some electrodes, when he has the feeling that the "neurologic" bandwidth is limited, and that the neurons facing the electrode are getting saturated at only moderate auditory levels.

Results are variable, but often good. Usually, a fitting session starts with the practitioner asking whether the current fitting is better or worse than the previous one. The best of the recent fittings is taken as a basis that the practitioner will try to improve, resulting in some sort of hill climbing process.

The patient tries to describe the quality of his audition, and the practitioner tries to modify some parameters to help solving the problems. Two or three parameters can be changed during a 30 to 90 minutes fitting session. Then, the patient leaves with the new settings that he keeps for a couple of months, before he comes back for another fitting session. The whole process is therefore very long (several years for problematic patients).

14.4 Description of the Problem

As seen above, fitting cochlear implants is done through a set of correlated parameters LOIZOU et collab. [2000], and perception and comfort thresholds are linked to histopathological factors specific to the patient KAWANO et collab. [1998]. In most cases, the fitting

strategy simply consists in maximising the number of electrodes and maximising their dynamic range BLAMEY et collab. [1992]. This often gives good results, but for some patients this approach does not work. Moreover, the following observations have also been reported:

- Better results might be obtained by decreasing the dynamic range FRANCK et collab. [2003].
- Only using a subset of electrodes might improve speech recognition ZWOLAN et collab. [1997].
- Holes in spectral representation can exist in tonotopic representation (mapping of the sound frequencies on the electrodes) and spectral information redistribution around the holes does not increases results SHANNON et collab. [2002].

Moreover:

- Most of the patients do not use all the information given by the electrodes FISHMAN [1996].
- All the electrodes are not necessary to obtain maximal speech perception performance in silent DORMAN et collab. [1989]; FISHMAN [1996]; KIEFER et collab. [2000]; LAWSON et collab. [1996] and noisy environments FRIESEN et collab. [2001] (part of this could be due to electrical interaction between channels STICKNEY et collab. [2006]).

These published observations show that choosing a good subset of electrodes can have an influence on speech understanding, as well as the dynamic range on the electrodes. Finally, taking into account a real sound environment could increase speech understanding for some patients.

The work presented in this chapter will try to address both the problems of choosing a good subset of electrodes, and taking into account a real life sound environment.

14.5 Description of the Interactive Evolutionary Algorithm

Before this work was started, several fitting sessions were observed, with patients who were not satisfied with their cochlear implant. During these experimentations, it really seemed that the fitting was stuck in a local optimum, since the expert's heuristics looked quite like what is known in computer science as a local search (trial of neighbours of the current best fitting) that would not bring any improvement.

This triggered the idea to use evolutionary algorithms, that are both quite good at optimising parameters and not easily trapped in local optima. The genetic loop is the following: the EA "suggests" a set of parameters that are directly uploaded into the Cochlear Implant's processor, and waits for an evaluation.

Other works have been conducted on interactively fitting hearing aids with evolutionary algorithms, DURANT [2002]; TAKAGI [2001], but they concern only conventional hearing aids, with a relatively small number of parameters that can be tuned. To our knowledge, nobody has tried to apply evolutionary algorithms to Cochlear Implants fitting.

14.5.1 Managing the runs

In an interactive evolutionary algorithm, a human user evaluates the different individuals proposed by the algorithm.

Thomas Bäck's results BÄCK [2005], suggest that an evolutionary algorithm may do as well (if not better) than a human expert on a number of evaluations of the same order than the number of real parameters to optimise. Therefore, if the problem has around 100 parameters to tune, performing only 100 evaluations may allow to obtain interesting results. If it

is possible to find an evaluation procedure that takes around 5mn, a run would last around 8 hours.

However, it is also important to take psychology and human fatigue into account: a well tuned convergence speed over 100 evaluations could seem discouraging for a human patient, who may think that improvement is too slow. Besides, since it is not possible to have an 8 hour run in one go, an elegant solution consists in fractioning the experimentation into several partial fast-converging runs, with a restart at the end of each run JANSEN [2002]. Dividing the 8 hour run into 5 makes for 5 1h30 runs, that are quite manageable.

Rather than finding ways to avoid premature convergence, it is on the contrary a very fast convergence that is sought on these short runs of approximately 20 generations. This feature is easily obtainable with evolutionary algorithms, since they are known to converge quite fast, if no counter-measures are taken.

This policy allows to use a very fast converging algorithm trying to exploit local minima, rather than a slow converging algorithm trying to widely explore the search space, looking for the global minimum. The consequences of premature convergence are dealt with thanks to the periodical restarts. During the last run, one can restart the algorithm with the best individuals found in the 4 first runs, so as to benefit from the results previously found.

Population size and number of children per generation. For an identical number of evaluations, two possibilities exist: either many children per generation and a small number of generations, or a small number of children per generation and many generations.

Out of these two possibilities, it is the algorithm that maximises the number of generations that will favour most convergence. This suggests a *SteadyState* replacement policy, or a $(\mu + \lambda)$ with a very reduced λ (number of children) BÄCK [1996]. Then in order not to spend too many evaluations in the initial population, one can also reduce it as is done in *micro-GAs* KRISHNAKUMAR [1989].

Extremely low values can be used, such as 3 to 6 individuals for the initial population, with 1 to 3 children per generation. For the fifth run, 4 individuals could be used for the initial population, taken from the best individuals of the 4 previous runs.

The algorithm chosen for this specific interactive optimisation will therefore be a modern evolutionary algorithm, in the sense that it does not take after any of the four usual paradigms (Evolution Strategies, Genetic Algorithm, Genetic Programming, Evolutionary Programming) DE JONG [2005].

According to Bäck BÄCK [2005], using an Evolution Strategy paradigm for 100 evaluations should allow to optimise up to 100 real variables. In Cochlear Implants fitting, however, one can start with trying to find the best T and C values for each electrode. With the MXM 15 electrodes CI used for this experiment, the genome is therefore an array of only 30 real values, meaning that the chances to find a good fitting are much higher.

14.5.2 Initialisation

One hard constraint needs to be respected: the algorithm should not go beyond the maximum intensity for each of the electrodes for fear of destroying some of the patient's auditory neurons. Therefore, for each new patient, a first session with a practitioner is realised to determine the maximum admissible intensity for each electrode, that is called a *psychophysical test*. In order to reduce the search space, a minimal intensity below which the patient does not hear anything is also determined.

The initialisation of each individual therefore simply consists, for each of the 15 electrodes, to pick up two random values within the $[min, max]$ interval determined during the psychophysical test, and to take the lower value as a T threshold, and the higher value as a C threshold for the each of the 15 electrodes.

14.5.3 Selection of the parents

Parents selection is different from the replacement stage, in that it can select an individual several times. Whenever a child must be created, two different individuals are selected among the parent's population, that can be selected again to create another child.

Since the selection pressure of proportional selection depends on the fitness landscape of the problem to be solved (which is unknown), a stochastic tournament is selected BLICKLE et THIELE [1996], with a 90% probability, that consists in randomly selecting 2 individuals and to take the best of the two with a 90% probability.

14.5.4 Crossover

The genes are real values, which could have suggested some kind of barycentric crossover (such as used in Evolution Strategies), where each gene of the child is an average between the two genes of his parents. But since it is intervals that must be evolved, this type of crossover would have led to reducing the intervals progressively.

The chosen crossover is that of genetic algorithms, which exchange the parent's genes after a crossover point (*locus*) chosen randomly. A mono-point crossover was chosen, as a multiple crossover would have had a tendency to break efficient genomes, and would have turned the crossover in a kind of macro-mutation.

In this same attempt to not break good configurations, the determination of the locus is made electrode by electrode (the two T and C values are not separated). Since we are using a $(\mu + \lambda)$ evolutionary engine, with a number of children smaller than the size of the population, the crossover is called to create each child (100% probability).

14.5.5 Mutation

Mutation is also called with a 100% probability on each created child. In the proposed algorithm, each gene has a 10% probability to be mutated. Since there are 30 genes, each child undergoes 3 mutations in average. This may seem important, but due to the large epistasis, modifying a threshold on the global genome only has a limited influence on the global evaluation. This high mutation rate allow to keep a reasonable exploratory character to the algorithm, in spite of the very small number of evaluations.

14.5.6 Replacement

A *Steady State*-like replacement is used, i.e. with a very small number of children per generations, in order to promote a fast convergence. During a strict Steady State replacement, only one child would be created, that would replace the worst of both parents. Since we decided to have several children per generation, it is a $(\mu + \lambda)$ replacement scheme that is used, with only 2 or 3 children per generation (where Evolution Strategies usually create more children than there are individuals in the population).

14.5.7 Evaluation

It is possible to memorize 2 or 3 fittings on modern cochlear implant processors (called $P1, P2, P3$). Until this research was conducted, the evaluation of the patient's understanding was done by two different ways. Either the patient was sent home with the new fitting on $P1$ and the previous fitting on $P2$, which allowed him to compare both fittings in his environment, or an evaluation was done by an orthophonist with intensive tests during more than one hour.

Even though an interactive evolutionary algorithm requires a reduced number of evaluations TAKAGI [1998] none of these methods were suitable for an interactive evolutionary

algorithm, so various evaluation protocols have been devised and will be described in details in section 14.6.

14.5.8 Execution

The evolutionary algorithm has been implemented both on a regular Personal Computer and on a PDA so that it is possible for a patient to tune his cochlear implant in a real environment (in a train station, for instance, if the patient works there and really needs a specific fitting for this particular environment).

The first versions have been implemented using the EASEA[2] COLLET et collab. [2000] language in combination with the GALib library WALL. Later versions have been completely re-implemented from scratch in C++, because the GALib library has a bug that makes it ignore the evaluation of the initial population. Although this is not very important for applications where evaluations are easy to obtain, losing n interactive and tiring human evaluations was too high a cost in this special case.

14.6 Experiments

The first three sub-sections present results obtained with patient A, that were conducted by Claire Bourgeois-République, as part of her PhD. thesis of the Université de Bourgogne. These results have already been presented in several papers BOURGEOIS-RÉPUBLIQUE et collab. [2005]; BOURGEOIS-RÉPUBLIQUE [2004]; BOURGEOIS-RÉPUBLIQUE et collab. [2005].

The following experiments have been conducted by Vincent Péan and Pierrick Legrand within the RNTS HÉVÉA project, funded by the French Ministry of Health.

14.6.1 Presentation of Patient A

Patient A has received an MXM cochlear implant 10 years ago in 1994. Unfortunately, he has not recovered a perfect audition (he understands some words quite well, but not others), although he is able to hold a conversation over the telephone, which is already quite a feat.

He was initially given a waist processor (called *Boîtier*) to be carried attached to his belt, until MXM recently came up with a tiny "Behind The Ear" BTE processor. In 2003, patient A has received a BTE with the hope that new technology would allow him to hear better.

Unfortunately, this is not the case. After many disappointing fitting sessions with an expert practitioner, he still feels uncomfortable with the BTE and apparently cannot follow a conversation with it. He therefore keeps the BTE in a drawer and uses the old *Boîtier* for every day life.

The automatic fitting algorithm described in this paper was developed with the latest MXM technology, i.e. BTEs. It was thought that *Patient A* could be a good patient to test the evolutionary algorithm, with the remote hope to find parameters that would allow him to hear with his state of the art BTE at least as well as with his old *Boîtier*.

To start with, *Patient A* came to the hospital for yet another fitting session with a practitioner, with the aim to determine the minimum and maximum (C and T) intensity values for each of the electrodes for his BTE, to feed the evolutionary algorithm (cf. table 14.1).

Electrodes 10, 11 and 12 have C and T values of 0 because the auditory neurons they face have apparently been damaged (Patient A does not hear anything whatever intensity is applied to these electrodes).

In order to be able to compare fittings, evaluations were done with the best fittings on the *Boîtier* and the BTE. The results corresponded to his claims. With the 78%/22% evaluation described above:

[2] http://sourceforge.net/projects/easea or
http://complex.inria.fr/cgi-bin/twiki/view/Complex/SoftwareEASEA

Electrode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Min	6	6,5	6,5	9	9	9	8	8	8	0	0	0	7	6	5
Max	9,5	13	13	18	20	21,5	21,5	18	16,5	0	0	0	12	10	9

Table 14.1 – Minimum and maximum intensity (C and T values) for each electrode for *Patient A*. A pulse stimulation is defined by two parameters: Pulse height = intensity in mA (between about ten microA and 2 mA). Pulse width = duration in microseconds with a step of 0,5 microsecond. The min and max units are the min and max of the pulse width (in microsecond) of the stimulation for a given pulse height.

- The *Boîtier* obtained an evaluation of 53% (slightly more than 50% of the 78 words were understood).
- The BTE obtained an evaluation of 48.5% (fewer words were understood and the BTE is less confortable).

Although the results above are not statistically significant (only one evaluation for the *Boîtier* and the BTE), they matched well with the patient's feelings.

14.6.2 First set of experiments

14.6.2.1 Evaluation for the Patient A.

A new evaluation protocol have been devised, using calibrated sentences extracted from a list of "cochlear" sentences elaborated by Pr. Lafon LAFON [1964], that are supposed to contain representative syllable of the French language allowing to evaluate pathological cochlea. Ten sentences were selected, for a total of 78 words, that would give 78 points if all words were correctly understood.

A comfort mark between 0 and 10 completes the evaluation, as an uncomfortable fitting will not be used by the patient. The comfort mark is multiplied by 2.2 so that the global evaluation is made of 78 points coming from the recognised words + 22 points coming from the comfort of the tested fitting.

Tests have shown that this evaluation procedure takes slightly less than 4 minutes. This is clearly not enough to obtain a fine evaluation of the audition of the patient, but it allows to perform 100 evaluations in 6h40mn only (i.e. 1h20mn per run if the 100 target evaluations are decomposed in 5 runs). If this reduced protocol is enough to guide the evolutionary algorithm and allow it to improve the fitting over 100 such evaluation, the aim is reached.

Such an aim is different from the aim of the complete evaluation of a standard practitioner, because due to the very small number of fittings they can perform in a year (about 10 fitting sessions per year and per patient), they need a very precise evaluation procedure in order to test the quality of the audition of the patient.

The experiments below were made in order to find the good values for the parameters of the interactive evolutionary algorithm (size of the population, number of children per generation, mutation rate, ...), hence the different tested values.

14.6.2.2 Experiment 1 and results.

For the first experiment with patient, the size of the population was limited to 3 individuals and the evolutionary algorithm was asked to create 3 children per generation. Mutation rate was 0.1 and crossover rate was 1.

On the first evaluation (of a randomly created individual) 42 words were understood on a total of 78. Patient A gave an evaluation mark of only 1 (over 10) because even though he could understand more than half of the words, the BTE sound was resonating and feeling uncomfortable. The global evaluation was therefore of $42+1\times2.2=42.2$.

On this first experiment, 12 evaluations were performed, which is a large number, knowing that preparation and evaluation of one fitting takes between 15 and 20 mn for an experienced practitioner. With the evolutionary algorithm, only 4 mn were needed per fitting.

The result of the evaluation is given in the table .

Fitting	1	2	3	4	5	6	7	8	9
Evaluation	44,2	21,2	9,2	31,4	55,6	46,4	74,8	74,8	58,4
Fitting	10	11	12						
Evaluation	81	81	79,8						

Table 14.2 – Experiment 1 -patient A

The first three evaluations (44.2, 21.2, 9.2) correspond to random individuals. Artificial evolution starts on fitting number 4, with 3 children per generation (generations are marked with a double vertical bar).

From the 5th evaluation onwards, obtained results are better or equivalent to the best fitting performed by the medical practitioner (48.5).

Fittings 7 and 8 are nearly identical, as well as fittings 10, 11 and 12. These results have never been approached by the expert neither with the BTE nor with the *Boîtier*.

Patient A is enthusiastic, and a second experiment is started with 6 individuals, to avoid premature convergence.

14.6.2.3 Experiment 2 and results.

The only changes that have been made are a population size of 6 individuals and 4 children per generation (generations are marked with double vertical bars).

Fitting	1	2	3	4	5	6	7	8	9	10
Evaluation	24	17	30	19	53.2	37.4	22.6	24	33.4	32
Fitting	11	12	13	14	15	16	17	–	–	–
Evaluation	9	27.4	34	34.5	12	27	32	–	–	–

Table 14.3 – Experiment 2 - patient A

The first four random individuals get poor results. Then, crossover and mutations have difficulties creating better individuals, with some really poor individuals (fittings 11 and 15).

Patient A gets tired and disappointed. The test is stopped after the 17th fitting.

14.6.2.4 Experiment 3 and results.

For the 3rd test, the population is reduced back to three individuals, but with 2 children per generation. Mutation rate is increased to 0.6 and roulette-wheel is used as a selector in order to increase the selective pressure when choosing parents.

The three initial individuals obtain great values (54, 33 and 26.5). The second generation obtains values near 50. Then evaluations increase towards 60s and 70s without dropping below 50 again.

Around generation 10 or 11 (fittings 20, 21, 22), evaluations seem to stabilise near 70 without beating value 73 of fitting 16.

Fitting	1	2	3	4	5	6	7	8	9	10	11
Evaluation	54	33	26.5	48	52	51.6	54.6	62.8	59.6	65.6	60.1
Fitting	12	13	14	15	16	17	18	19	20	21	22
Evaluation	60	72	69.4	53.4	73	67	50.1	62	68.3	67.3	65

Table 14.4 – Experiment 3 - patient A

14.6.2.5 Experiment 4 and results.

For the fourth experimentation, population size is set to four individuals and four children per generation. Mutation rate is brought back to 0.1 and parents selection is set back to Tournament.

Fitting	1	2	3	4	5	6	7	8	9	10	11	12
Evaluation	59.4	62.2	57.3	58.9	57	62.3	65	73	75.3	65.2	83.1	68
Fitting	13	14	15	16	17	18	19	20	21	22	23	24
Evaluation	75.4	91	91.5									

Table 14.5 – Experiment 4 - patient A

In average, the first four individuals present an average evaluation of 59.5 and all subsequent values are above 56.5.

Values of 91 and 91.5 are obtained at the end of generation 4. *Patient A* is tired but extremely satisfied and surprised by such results. He leaves for lunch with the BTE.

However, when he returns a couple of hours later, he says that the fitting is not very efficient in noisy environments, and feels like he still prefers his *Boîtier*, as it feels much more comfortable to wear, as he has used it for the past 10 years.

14.6.2.6 Experiment 5 and results.

Population is now of 5 individuals, with two children per generation, a tournament selector and a mutation probability of 0.1.

Fitting	1	2	3	4	5	6	7	8	9	10	11	
Evaluation	18.6	53	70.1	9	71.9	58.4	60.3	58	51	57.3	48.2	
Fitting	12	13	14	15	16	17	18	19	20	21	22	23
Evaluation	36	36.2	50	29	33.5	50.3	40.2	44.5	48.3	49.3	45.2	50

Table 14.6 – Experiment 5 - patient A

Among the first five random individuals, two show a surprising evaluation of 70.1 and 71.9, which raises questions on the original fitting of the expert for the BTE, which only gets 48.5 on the exact same data. Some answers to this will be provided in third and fourth sets of experiments below.

However, evolution does not seem to find any better individuals.

14.6.2.7 Discussion on obtained results

14.6.2.7.1 Fitness evolution: The evolution of the best individual for the five runs The evolution of the best individual for each of the runs is shown figure 14.2. Fitness increases on all experiments but exp. 2, which is a nice result for such a small number of evaluations, meaning that the educated guesses made on the IEA implementation were probably good. It seems that the correct population size is 3 or 4 individuals, with 2 to 4 children per generation.

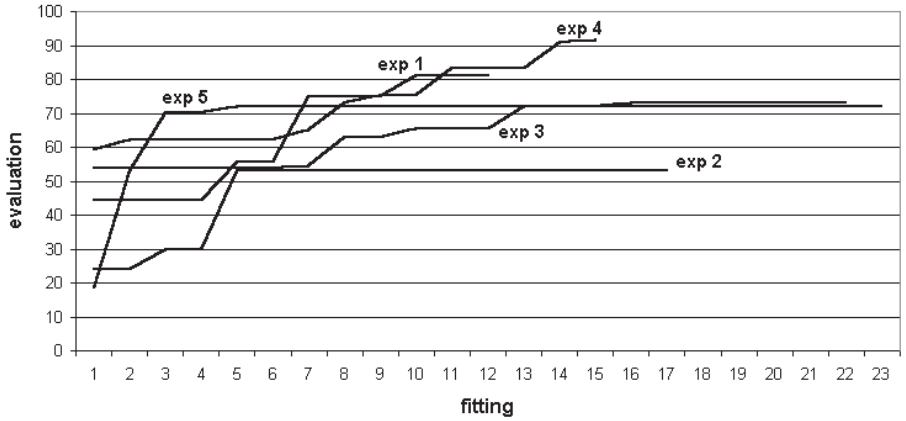


Figure 14.2 – Evolution of the best individuals per evaluations, for each experimentation.

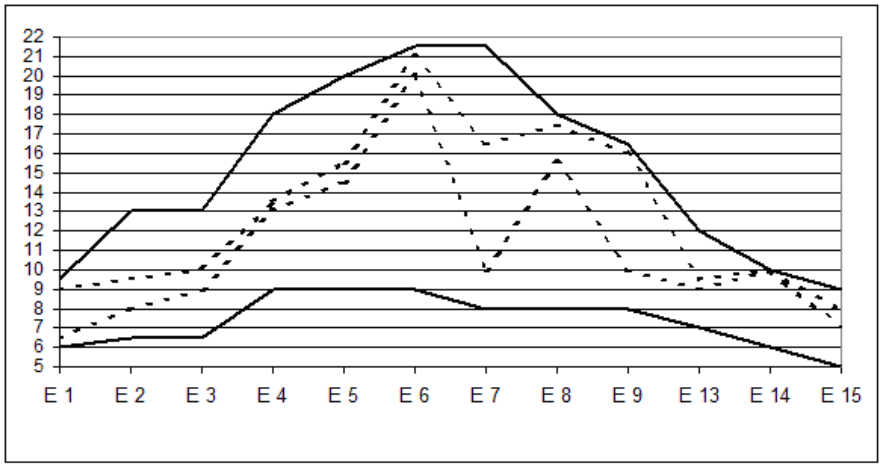


Figure 14.3 – X-axis: Electrodes, Y-axis: Intensity. The bold curves represent the maximum allowed envelope (T and C) for each electrode and the curves in dotted lines represent the best obtained individual (T and C).

14.6.2.7.2 Analysis of the best obtained individual: Analysis of the T/C values of the best individual is intriguing (Figure 14.3: Electrodes 10, 11 and 12 have been omitted as they are not functional). Sometimes, experts reduce the C - T range for some electrodes when they feel that the neural "bandwidth" is too narrow and there is a possibility of saturation if the auditory information is too important. In the fitting found by the IEA, however, many of the C - T ranges are reduced down to 1.5, 1, 0.5 and even 0. In fact, only electrodes 1, 7 and 9 have significant ranges (over 2.5, cf. circled arrows). Other good fittings show wider ranges for electrodes 7 and 9 and narrower ranges for the other electrodes, which raises a hypothesis: What if, for this precise patient, some electrodes had a negative influence on speech understanding ? If this were the case, the current practice (that has been going on for

many years) of maximising the range of as many electrodes as possible would also maximise the range of "wrong" electrodes that prevent the patient of understanding speech. After this first evolutionary fitting session, the patient went back home with the original settings in his CI.
This experiment raises several questions:

- Is minimising the $T - C$ interval equivalent to shutting down an electrode ?
- Could there be a diaphony problem (crosstalk) between the electrodes ?
- Could the problem be combinatorial ?

14.6.3 Second set of experiments.

A second set of experiments has been conducted in order to verify some hypotheses that arose after the first set of experiments. The tests have been conducted with the same patient and with the same evaluation protocol, but one month later. It is important to note that between the two sets of experiments, the patient has used his old *Boîtier* and has resumed his normal life, meaning that it is very probable that no neuronal plasticity has had any chance to occur. The evaluation basis are therefore comparable. In the text below, the first set of experiments is noted C_1 while the second set is noted C_2.

14.6.3.1 Experiment 6.

Surprisingly enough, the best individual obtained during the fourth run was virtually using only three of the 12 functional electrodes: Electrodes 1, 7 and 9. Each electrode corresponds to a given frequency band:

- Electrode 1: 4089 - 5798 Hz,
- Electrode 7 : 671 - 916 Hz
- Electrode 9 : 427 - 549 Hz

Since electrode 1 was mapped onto very high frequency sounds that are not discriminant for speech, the number of functional electrodes could be reduced to only 2. In order to confirm this strange result, the first deterministic test maximises electrodes 7 and 9 only (using the maximum C - T range of table 14.1), giving only a small range to electrode 1 figure 14.5. For all the other electrodes, T and C values are set to 1 and 1.5, i.e. much below the threshold, in order to cancel them totally. This setting obtains an evaluation of 82, which is much better than with all activated electrodes (best fitting of 48.5 obtained by the expert). Nearly 90% of the words were understood, and the fitting was rated as not very comfortable. This allows to conclude that for this patient, using only three electrodes out of 15 allows him to understand speech better than with all functional electrodes set to nearly maximum range.

14.6.3.2 Experiment 7: On the influence of electrode 8.

In the C_1 set of experiments, the evolutionary algorithm seems to hesitate a bit on electrode 8. In order to test its real contribution, the electrode 8 is added to the 1, 7 and 9 electrodes, by maximising its $C - T$ interval (using the values of table 14.1). The obtained evaluation is 81, and the patient finds that the fitting is slightly less comfortable than the previous one. Speech understanding is comparable. The electrode 0 does not seem to have an important role in speech understanding.

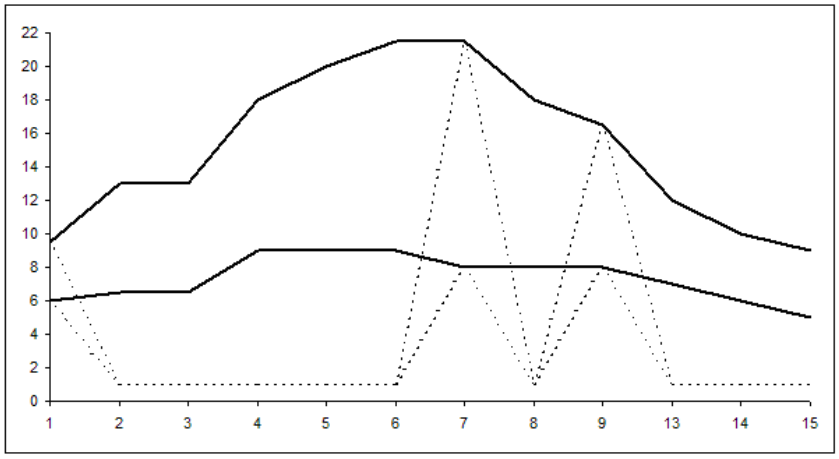


Figure 14.4 – Experiment 6. X-axis: Electrodes, Y-axis: Intensity. Testing with electrodes 1, 7 and 9 only. The bold curves represent the envelope (T and C) for each electrode. The dotted lines correspond to the manually tested T and C values for each electrode.

14.6.3.3 Experiment 8: Is there any diaphony between the electrodes ?

In order to explore this hypothesis, even electrodes are suppressed (by setting T and C values below the T liminary values for the patient), and the odd electrodes are maximised (using the values of table 14.1), so as to space active electrodes (cf. figure 14.5).

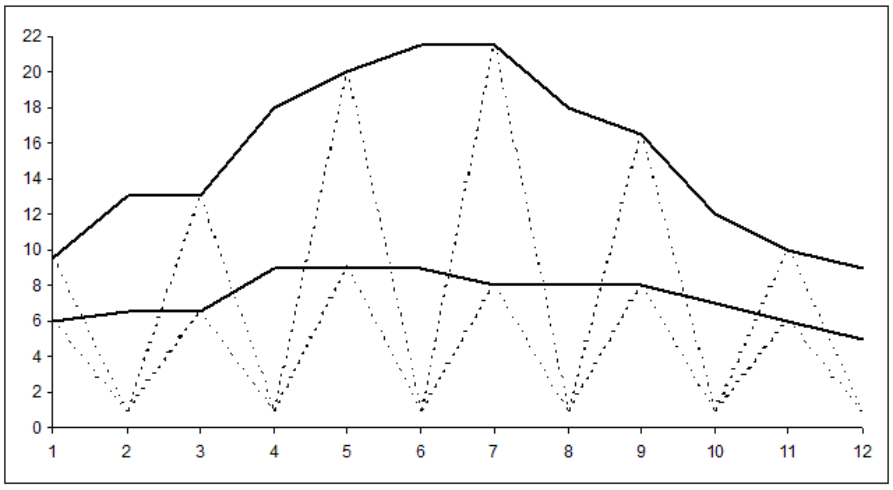


Figure 14.5 – Experiment 8. X-axis: Electrodes, Y-axis: Intensity. Checking for diaphony.

This fitting obtains an evaluation of 78.8, and is judged less comfortable by the patient. The result is therefore not as good as those obtained during experiments 6 and 7. Adding other electrodes does not seem to add much. The result is however still much better than the one obtained by the practitioner with the BTE (48.5).

14.6.3.4 Experiment 9: Spacing electrodes even more.

This time, 2 electrodes out of 3 are canceled, by setting their T and C values to 1 and 1.5 (cf. figure 14.6). Therefore, electrodes 1, 4, 7 are activated. It was chosen to keep electrode

9 active, so as to keep a common comparison basis with the previous experiments. Finally, electrode 15 is maximised figure 14.6. This fitting obtains an evaluation of only 58.5, i.e. clearly not as good as the previous ones, and the patient rates it as quite uncomfortable. This is very surprising, as the only difference with the first test (that had obtained an evaluation of 82) is that electrodes 4 and 15 have been added. Clearly, not only is there no diaphony problem (spacing active electrodes did not improve evaluation), but it can be concluded that for this patient, electrodes 4 and 15 contribute negatively to speech understanding. The fact that functional electrodes can contribute negatively to speech understanding is a totally new concept in the cochlear implant medical field.

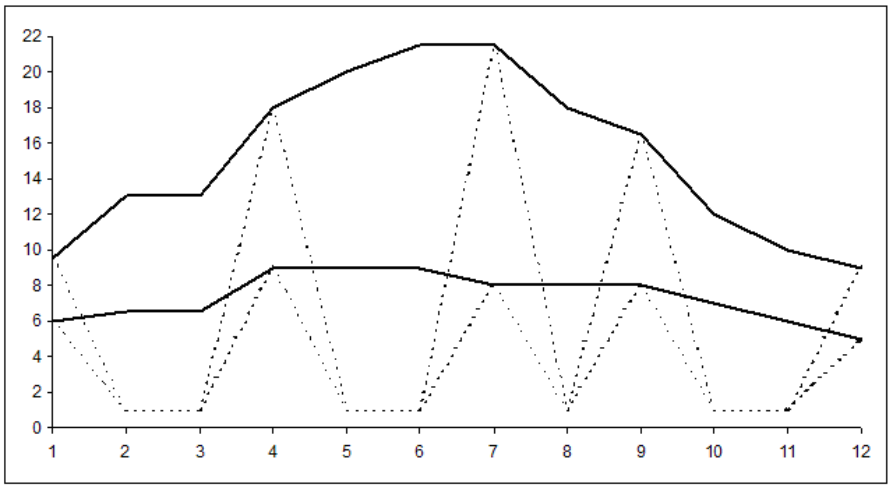


Figure 14.6 – Experiment 9. X-axis: Electrodes, Y-axis: Intensity. Checking for diaphony by selecting only one every 3 electrodes, and keeping electrode 9.

14.6.3.5 Experiment 10: Evaluation of the best individual of C_1.

In order to test the evaluation procedure, the best individual of the set of experiments C_1 is tested again, one month later, and without telling anything to the patient.

The speech understanding test is again very good (94% of the words are understood, which is even better than one month before) but the comfort mark (over 22, cf. section 14.6.2) is not as good, resulting in a slightly lower evaluation of 86.2%. All in all, this value is slightly lower than the one obtained during C_1, but it is the best value obtained during C_2.

14.6.3.6 Experiment 11: Evaluation of the practitioner's fitting.

This time, it is the practitioner's original fitting that is tested again (the one that more or less maximised all electrodes, and that had obtained 48.5 during C_1).

Here again, the number of recognised words is very low (only 33%) and comfort gets a bad 4/10 evaluation. The global evaluation is 41.8, which is also slightly worse than during C_1.

All in all, in one month, the best fitting found by the IEA went down from 91.5 to 86.2, while at the same time, the practitioner's fitting also went down from 48.5 to 41.8. This suggests that the proposed quick 4mn evaluation is quite reliable, as the results seem to be reproducible one month later, while the patient used his old *Boîtier* in the meantime.

14.6.3.7 Other tests.

In order to verify that values obtained by the evolutionary algorithm are better than random ones, other experiments have been conducted with random values for T and C for all electrodes. Evaluations range from average to bad, although often greater than those obtained by the practitioner (48.5). The patient finds that these fittings are not comfortable.

14.6.4 Third set of experiments with others patients

In order to verify the gain obtained with computer-aided CI fitting, and develop its use at hospital, new experiments have been carried out with others patients. This set of experiments C_3 is conducted with 2 new patients: Patient B and patient C. For these experiments, the parameters of the IEA are the following:

Population	3
Children	2
Mutation rate	0.1
Crossover rate	1

The new population is obtained by taking the best individuals of the intermediate population consisting of the 3 parents and the 2 children (i.e. in the style of a (3+2)ES).

14.6.4.1 Corpus and methodology.

Both patients have received MXM cochlear implants some years ago, but they are not satisfied with their devices and have no good results (general evaluation by the practitioner is less than 50%). The IEA has been used to try to determine optimal C (Comfortable) and T (Threshold intensity) values for each of the electrodes of the CI.

To start with, the patients came to the hospital for a fitting session with a practitioner, and minimum and maximum intensity values for each electrodes of their BTE have been determined, to give boundaries to the evolutionary algorithm.

For these 2 patients (B and C), the same procedure that was used for patient A (a set of calibrated sentences) has been tested. Unfortunately, the results are disappointing as patients B and C recognise but a few words, meaning that this test is too hard for them.

Therefore, a new evaluation procedure was set up, based on a weighted evaluation of the results of:

- A discrimination test (ASSE) on 7 items. The ASSE test consists in emitting a sound n times (an [i] for instance), and within these occurrences, replacing one of the [i] with an [a] (for the following sequence: i i i i a i i). The patient needs to detect that one of the sounds was different. The ASSE test counts for 20% of the evaluation.

- A VCV (Vowel/Consonant/Vowel) test ([APA], [ATA], ...), where the patient must recognise the consonant between the two vowels. In one VCV test, each VCV is repeated 3 times, meaning that 48 VCVs are proposed to the patient (because in French, there are 16 different phonetic consonants). This test counts for 50% of the evaluation.

- A comfort evaluation with a mark from 0 to 10, that counts for 30% of the evaluation.

Unfortunately, the complete evaluation takes a long time (much more than 4 minutes), and the patients are less compliant than patient A, so it is impossible to get around 100 evaluations (as for patient A).

After the first sessions, the P1 and P2 settings of the CI were loaded with respectively the fitting obtained with the IEA, and the manual fitting of the practitioner, after which the patients were sent home with the instruction to use the best fitting of P1 or P2.

After two weeks, the patients came back for new tests:

1. a discrimation test with P1 and with P2,
2. a VCV recognition test with P2 and with P1,
3. a sentence recognition test with 10 sentences using the P1 setting (IEA).

14.6.4.2 Third set of experimentations with patients B and C

- First session for Patient B:

Eval Nb	Manual	1	2	3	4	5	6	
ASE Result	4/7	5/7	5/7	5/7	6/7	5/7	7/7	
VCV Result	33%	31%	25%	18%	29%	31%	31%	fatigue
Comfort	7/10	6/10	7/10	5/10	5.5/10	6/10	8/10	
Evaluation		5	5	4	5	5	6	

- Second session for Patient B three days later:

Setting	Manual	1	2	3	4	5	6	7	
ASE Result		6/7	7/7	7/7	7/7	6/7	5/7	5/7	
VCV Result	35%	25%	27%	10%	18%	18%	20%	27%	fatigue
Comfort		5/10	6/10	6/10	5/10	5/10	5/10	5/10	
Notation		4	5	4	4	4	4	4	

The best obtained fitting (6th fitting of the first session) was loaded in memory P1 of the BTE and the patient was sent home for two weeks for a long evaluation of the new fitting.

- First session for Patient C:

A first set of independent random tests has been performed, to be compared to the manual fitting results, in the table below:

Setting	Manual	Random								
ASE Result	5/7	6/7	5/7	5/7	5/7	5/7	4/7	5/7	6/7	5/7
VCV Result	45%	33%	29%	22%	39%	31%	18%	29%	39%	35%
Comfort		4/10	5/10	5/10	5/10	5/10	4/10	5/10	5/10	5/10
Notation		5	4	4	5	4	3	4	5	4

Then the IEA is used, but only based on a VCV evaluation, to shorten the time of evaluation.

Setting	1	2	3	4	5	6	7	
VCV Result	35%	41%	39%	33%	20%	43%	37%	Fatigue
Notation	4	4	4	3	2	4	4	

As for Patient B, Patient C is sent back home for a long evaluation of fitting obtained in setting 6 (VCV result = 43%).

After two weeks, patients B and C come back to hospital with the following results for patient B:

Test	ASE	VCV	Words/list	Comfort
Auto	3/7	33%	7	n/a
Manual	5/7	27%	10	n/a

And for Patient C:

Test	ASE	VCV	Words/list	Comfort
Auto	3/7	52%	1	8/10
Manual	4/7	37%	2	8/10

Remarks:

1. Both patients preferred to use the P1 fitting, i.e. the one found by the interactive evolutionary algorithm (although the very small number of evaluations due to the long evaluation makes it more like a random search than an evolutionary search).

2. Random fitting can do really well, sometimes slightly better than what the practitioners do when they maximise the number of electrodes and their dynamic.

3. Comfort is too difficult to evaluate accurately for the patients.

4. As for the interactive evolutionary algorithm, each of these non word based evaluations is much too long to obtain, meaning that patients get tired very rapidly and the run is stopped before a significant number of evaluations could be done that could allow the evolutionary algorithm to suggest other fittings than random ones.

The results obtained by this random search again question the maximisation of the number of electrodes and the maximisation of their dynamic range.

These random tests also show that the ranges of possible parameters values is well chosen, providing a search space having many “average good” solutions, but with a rather “flat” search landscape. In these conditions, and considering the parameter setting of the IEA (a (3+2) Evolution Strategy), time for convergence is too short to really obtain the beginning of a convergence. The problem of user fatigue makes it impossible to obtain meaningful results. Additionally it can be argued that the evaluation is not discriminant enough to provide an efficient fitness landscape to the IEA.

New tests have been designed, taking these results into account.

14.6.5 Fourth set of experiments

The same patients (Patient B and C) were tested. The parameters of the IEA are the following:

Population	4
Children	3
Mutation rate	0.8
Crossover rate	1

The new population is obtained by an elitist binary tournament between a population made by the parents and the children. The elitism is "soft," in the sense that it is the best individual of the 7 individuals that is taken to be part of the next generation (and not the best of the parents only). The three other individuals are selected by a standard binary tournament.

14.6.5.1 Corpus and methodology.

Each trial was based on the results of a VCV recognition test such as [APA], [ATA] ... The patient has to recognise the consonant in the VCV. Each VCV is proposed once, meaning that there are only 17 items in a test. The result over the 17 VCV counts for 100% of the evaluation.

14.6.5.2 Experiments

- Patient B:
  The IEA fitting tested over two weeks obtains an evaluation of 2 over the 17 tested VCVs. The expert fitting is tested again, and here again, only 2 of the 17 tested VCVs were recognised. A new run gives the following results:

Evaluation	1	2	3	4	5	6	7	8	9	10
VCV Result	2	3	3	2	3	4	3	4	4	3

  And after one hour break and restart:

Evaluation	1	2	3	4	5	6	7	8	9
VCV Result	4	2	3	2	4	2	2	3	3

  Several fittings were found where 4 VCVs were recognised rather than only 2 previously, but it must be noted that these fittings were found at random. Patient B was satisfied with this result.

- Patient C:
  The IEA fitting tested over two weeks obtains an evaluation of 8 over the 17 tested VCVs. A new run gives the following results:

Evaluation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
VCV Result	6	5	5	5	4	7	5	7	7	8	7	4	6	4	4	7	5	4

  After a lunch break, the algorithm is restarted, initialised with two individuals which are the IEA fitting the patient had been using for the previous week and the best fitting of the previous run (fitting 10).

Evaluation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VCV Result	3	5	6	6	7	8	7	7	6	3	5	8	6	6	6	4

  Then after another break, the algorithm is restarted again to produce the following results:

Evaluation	1	2	3	4	5
VCV Result	6	3	7	6	8

Remarks:

- The IEA was working fine, although no real improvement could be seen, even during the longest runs (like the first run of patient C, i.e. 18 evaluations, i.e. evolution during five generations).
- The probable explanation is that the chosen VCV evaluation is too difficult for both patients, and the algorithm cannot find any fitting leading to a stable improvement of the audition of the patients.

14.7 Actual work and perspectives

Concerning the evolutionary runs, the evaluation function is very important. If for these patients, the VCV test is really too hard, the IEA will not be able to find any improvements (the fitness landscape is too flat to give a direction for improvement to the algorithm).

It seems important to spend some time to set up an evaluation function specific to each patient, that can return an average value, neither too low, like 3/17 or 5/17 (because this would mean that the test is too difficult) or too high, like 15/17 or 16/17 (because this would not leave any room for improvement).

The evaluation function must be quick. If it is too slow, the patient will get tired before any significative number of evaluations are done (in the set of experiments 3).

Finally, until the IEA procedure is routinely giving good enough results, it may be interesting to choose "easier" patients, i.e. patients for whom the cochlear implantation works slightly better ...

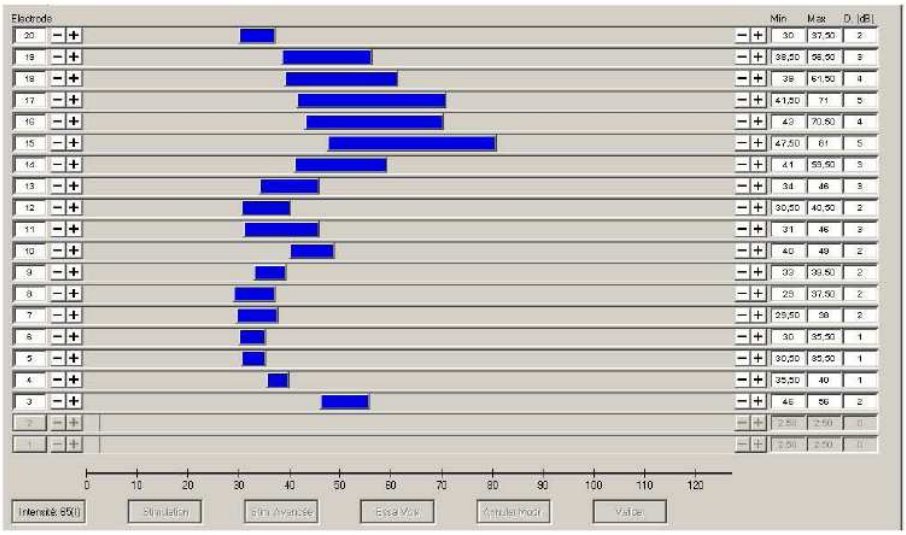


Figure 14.7 – Best fitting found by the practitioner for patient C: each rectangle represents the $[T, C]$ interval for each electrode.

Even though the sets of experiments 3 and 4 have not been really satisfying evolutionary-wise, the results are very interesting on a medical point of view, since it has confirmed that narrower intervals (or even removal of one or several electrodes) can lead to better speech understanding.

In all tested patients (of which A B C were a subset), it was possible to find fittings that were working at least as well as manual fittings maximising the dynamics for all electrodes, and in many cases, these fittings were simply random fittings !

In order to have a visual example, fig. 14.7 shows the intervals for all the electrodes of patient C on the best fitting found by the practitioner, while fig. 14.8 shows the best fitting obtained ... randomly, that gives better results than the practitioner's. Please note

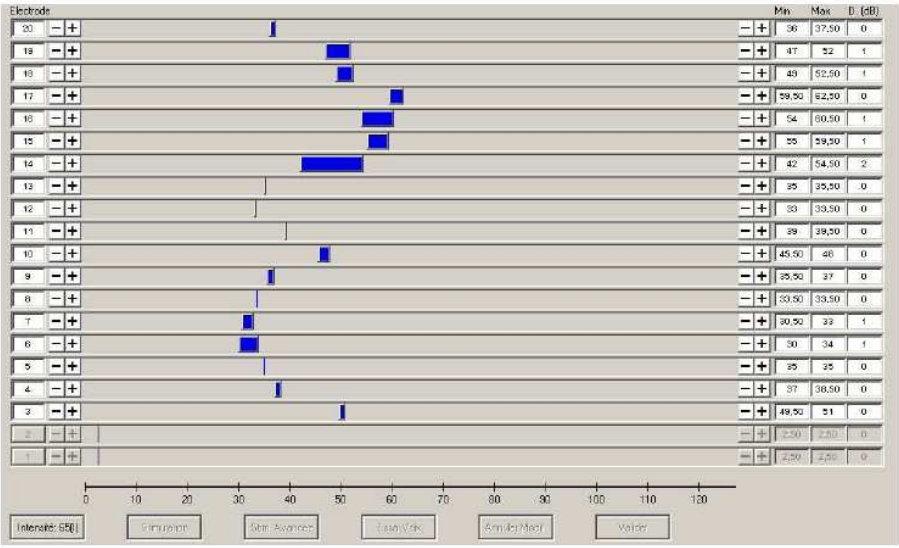


Figure 14.8 – Best fitting found at random for patient C, that beats the best fitting found by the practitioner: each rectangle represents the $[T, C]$ interval for each electrode.

the skinny intervals compared to those of fig. 14.7. In some cases, some electrodes are virtually cancelled (electrodes 5, 8, 11, 12 and 13), which goes against reason (and against what is advocated by the cochlear implants manufacturers).

14.7.1 Classification of sound environments

Many users of cochlear implants or hearing aids find that the parameter setting of their device is not perfectly adapted to all situations of their everyday life: in restaurants, they find clicking cutlery aggressive, and they have a hard time following a conversation, in the street, some noises are nearly unbearable, and so on. Some patients may need a setting for a quiet environment (such as home) but may work in a noisy environment (metal industry, garage, other noisy environments) so there is no miracle solution.

The aim of the HEVEA project is to improve hearing with cochlear implants by several means. One is to help the expert find good fittings using an interactive evolutionary algorithm BOURGEOIS-RÉPUBLIQUE et collab. [2005], and another is to integrate into the processor a small signal analysis software that would be able to recognise the sound environment and automatically select a fitting accordingly, among a set of available fittings corresponding to different situations.

In order to achieve this second task, several stages must be performed :

1. The medical team must determine with the patient a number of common environments for which the patient would need a specific fitting, for instance: home, work, supermarket, cinema, ...

   The number of specific environments should be limited, because for each of the specified environments, a special set of parameters needs to be found for the cochlear implant, and finding a good set of parameters can be a long and difficult task (even with the help of an evolutionary algorithm).

2. For each of the specified environments, the patient must take a number of sound samples to bring back to hospital.

3. Specific parameters must be found, to deal with each of the specified environments (possibly with the help of an interactive evolutionary algorithm).

4. In parallel, the different samples must be analysed to extract some common features, so that a classifying algorithm can determine them in which category falls the sound environment that is surrounding the patient.

5. Finally, the characteristics and parameters for the different environments must be uploaded into the cochlear implant processor, along with a signal processing program that will automatically choose the correct parameters to match the environment in which the patient is evolving.

The result is an "intelligent" cochlear implant that can automatically switch between potentially different sets of parameters, depending of the sound environment surrounding the patient.

This section presents the sound sampling, characterization and classification stage. It starts with a description of the specific sound sampler developed for this application, followed by a sub-section recalling the wavelet theory on which the scientific work is based. Then, a third sub-section describes how the energy content of a sample can characterise a sound environment. Finally, results are presented on the classification of different environments using a standalone piece of software.

14.7.1.1 Development of an *a posteriori* sound sampler.

In this application, sound sampling is essential to provide accurate data for two orthogonal needs:

1. The sound environment must be accurately recorded so that it can be recognised in the future by the processor with sufficient confidence to switch between different sets of parameters.

2. Particularities must be also recorded so that a specific fitting can be found that will help to cope with the current environment.

This distinction must be made because it is necessary to tune the Cochlear Implant (CI) on possibly transient sounds that are not representative of the general sound environment. For instance, one patient currently switches off his cochlear implant whenever cycling to work, because the sound of a motorbike passing by is too stressful to be bearable with his usual CI fitting. Choosing to switch off his CI (and becoming totally deaf) in a street environment is quite radical, but shows how much an adaptive and "intelligent" CI would be needed for this patient.

So it would be necessary for the adaptive CI to recognise a street environment, in order to choose for a fitting that would allow to cope with passing motorbikes, although passing motorbikes are exceptional in a street. One must therefore find a fitting adapted to an exceptional event, that should be selected when a sound environment (that has nothing to do with the exceptional event) is detected.

14.7.1.1.1 Sampling the regular environment for characterization. The sampling must be as accurate as possible, so that the processor can select the correct parameters without making any mistakes. Therefore, recording a sound environment on an old tape recorder may not be sufficient. A small jack plug has been added to the processor of the CI so that it could output directly the sound picked up by the microphones of the CI to a digital sampler.

Then, a sampling software has been developed on a PDA (Personal Digital Assistant) that the patient plugs directly onto the CI processor in order to sample the exact sound that is received by the processor (cf. fig. 14.9).

Figure 14.9 – A sampling software has been developed on a PDA that the patient plugs directly onto the CI processor in order to sample the exact sound that is received by the processor.

14.7.1.1.2 Sampling the exceptional event for CI fitting. Then, another problem arises: whenever an exceptional event occurs for which the CI should be tuned, it is often too late (the unbearable motorbike sound has vanished before the patient could record it, or in a crowded restaurant, the words that have not been understood cannot be repeated in exactly the same manner). A solution could be to sample the street (or the restaurant) for a long enough time, but here again, it is difficult do predict when the right motorbike will appear (or when the waiter will speak in an unintelligible way), and this could result in hours of recording, and hours to replay the records to find the relevant information.

A special sampling software has therefore been developed that constantly records the current sound for a period of n seconds. When the patient hits the `record` button, whatever happened during the previous n seconds is stored in a file, for future use. 30 seconds seems to be a correct period, so that when the patient uses the PDA to record precise sounds, he has 30 seconds to press on the button after he noticed that some interesting sound has occurred.

These very transient sounds samples (motorbike) have a different content than the samples that are used to characterise the general environment ("standard" street noise).

14.7.1.2 Characterisation of a sound environment.

We distinguish two steps in the problem of "sound environment classification". The first step is the extraction of the characteristics, in order to build the representation's space. The second step is to find a classification method which allows to fit each point of this space with a probability of being in a specified family. We can extract a lot of information from a sound in order to make a classification. For example, one can use the frequential content, the cepstral characteristics, the loudness, the pitch ...

From what is know from the human perception, spectral characteristics are discriminant for the recognition of all kind of sounds. This is the reason why we decided to use spectral measurements for artificial characterisation.

For this work we will analyse the frequential content at each dyadic scale because the implant performs the same kind of analysis. We will use a wavelet transform in order to perform a multiscale analysis (see DAUBECHIES [1992] and MEYER [1990]). We could use a simple Fourier Transform but we prefer keep the possibility to use the time localisation provided by the wavelet transform for a future work. In fact, Wavelet analysis allows to adjust the width of analysis windows, and achieves a perfect localisation in time and frequency. Logically, temporally extended windows are used to study low frequencies, while narrower windows are used for higher frequencies. This localisation property makes wavelet theory predominant in several areas of signal processing.

14.7.1.2.1 Continuous Wavelet Transform (CWT). A wavelet is a "wave localised in time." More precisely, it is a function $\psi \in L2(\mathbf{R})$ such that $\int_{\mathbf{R}} \psi(t)dt = 0$.
If $\int_{\mathbf{R}} \psi^2(t)dt = 1$, then we use normalized wavelets.

The continuous wavelet transform of a signal f is given by:

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right) dt$$

In this expression, a is a scale factor and b is a translation parameter (temporal shift). Variable a represents the inverse of the frequency: the smaller a, the (temporally) narrower the wavelet (i.e. the analysing function).

Therefore, one can see this expression as the projection of the signal on a family of analysing functions:

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

constructed by widening and translation from the original ψ wavelet.

14.7.1.2.2 Discrete Wavelet Transform. In this work we use a discrete wavelet transform which is faster than the continuous transform. The Discrete Wavelet Transform can be obtained thanks to the discretization of the parameters of resolution (a) and position (b). Let $a = a_0^m$ with m an integer, a_0 a resolution step greater than 1 and $b = nb_0a_0^m$ with n an integer and $b_0 > 0$.

Furthermore, if $a = 2$ and $b = 1$, the transform is called "dyadic." One then has:

$$C_{j,k} = 2^{-\frac{j}{2}} \int_{-\infty}^{\infty} f(t)\psi(2^{-j}t - k)dt$$

If $\psi_{j,k} = 2^{-\frac{j}{2}}\psi(2^{-jt} - k)$ we get a tiling of the time-frequency space called a dyadic grid (see fig 14.10).

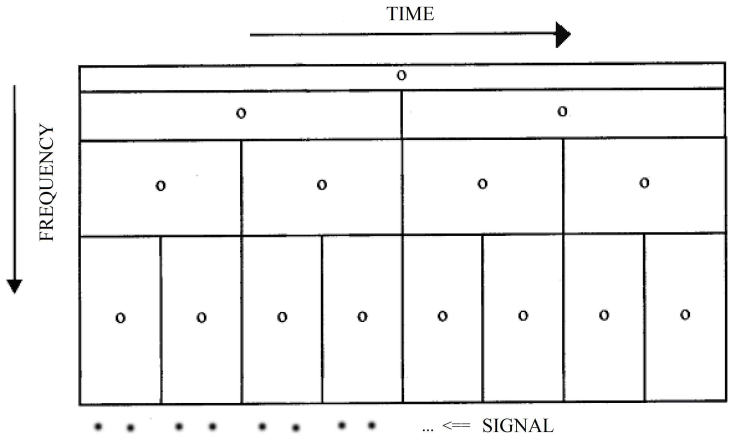


Figure 14.10 – Dyadic grid. X-axis: Time, Y-axis: Frequency. At the bottom, each point is a point of the signal. The matching discret wavelet coefficients are the circle in the grid. At low frequencies, the computation of the wavelet coefficient uses large windows in time, then we only have few coefficients. On the opposite, at high frequencies the computation uses small windows.

14.7.1.2.3 Energy of a signal. For a given scale, if we use a normalized wavelet, the energy of the signal can be obtained from the continuous wavelet transform. More precisely: one can compute the energy of the a scale by adding the squares of the wavelet coefficients of the continuous transform at this scale:

$$E_a^2 = \int [CWT(a,b)]^2 db \tag{14.1}$$

where E_a^2 is the energy at scale a. If we use the discrete wavelet transform, we get:

$$E_j^2 = \sum_{k=1}^{2^{j-1}} [C(j,k)]^2 \tag{14.2}$$

where E_j^2 is the energy at scale j.

14.7.1.2.4 Characterisation of a class by its energy content. As said above, a class will be characterised by its energy content. Let us consider a sound environment $S1$. The patient records a collection of `*.wav` files, that are chopped into a family of n_1 subsignals of 2^{14} points (almost 3 seconds for each subsignal, the 2^{14} number of points being chosen because it is a good compromise between quantity of information and computing speed). If one computes the discrete wavelet transform of theses signals and the energy of each of the obtained frequency bands during multi-resolution analysis, one then gets n_1 vectors of 14 coordinates. We choose to characterize a class by the mean value of these vectors. We obtain for each class a value at each dyadic bandwidth frequency (see fig 14.11).

14.7.1.3 Classification of sound environments.

The aim is to create a class for a specific environment, by using a collection of `.wav` files as input. The set of sounds chosen below are part of a patient's environment.

When the patient is in a new environment, he uses the sound sampler and records a sample of this environment. A `.wav` file is imported and chopped into 2^{14} micro-samples. When clicking on `compute`, each of the mini-samples is associated with the family that matches the sample best.

A ratio is then displayed, that presents the number of samples that corresponded to each family, and the results are displayed in a bar-chart. The bar-chart provides us the matching family with a certain confidence. For example if 80% of the micro-sample are classified in the class $S1$, then the sample will be classified in the class $S1$ with a confidence of 80%.

14.7.1.4 Results.

For each family, available `.wav` files have been chopped into mini-samples of 2^{14} points. 66% of the mini-samples chosen randomly are used for the learning set, and 33% for the test set. The results are presented in the following table:

Family	Learning set	Test set	matching family	Confidence
Car-radio	16	8	Car-radio	100%
Cross-roads	24	13	Crossroads	84 %
Birds	12	7	Birds	100%
School-yard	22	11	School-yard	100%
Supermarket	35	15	Supermarket	100%
Lawn-mower	10	5	Lawn-mower	80%

This set of sound was chosen because they were part of the sound environment of the tested patient.

All samples have been correctly classified. For Car-radio, Bird, School-yard, and Supermarket environments we have 100% of confidence. The worst results are for the Crossroad

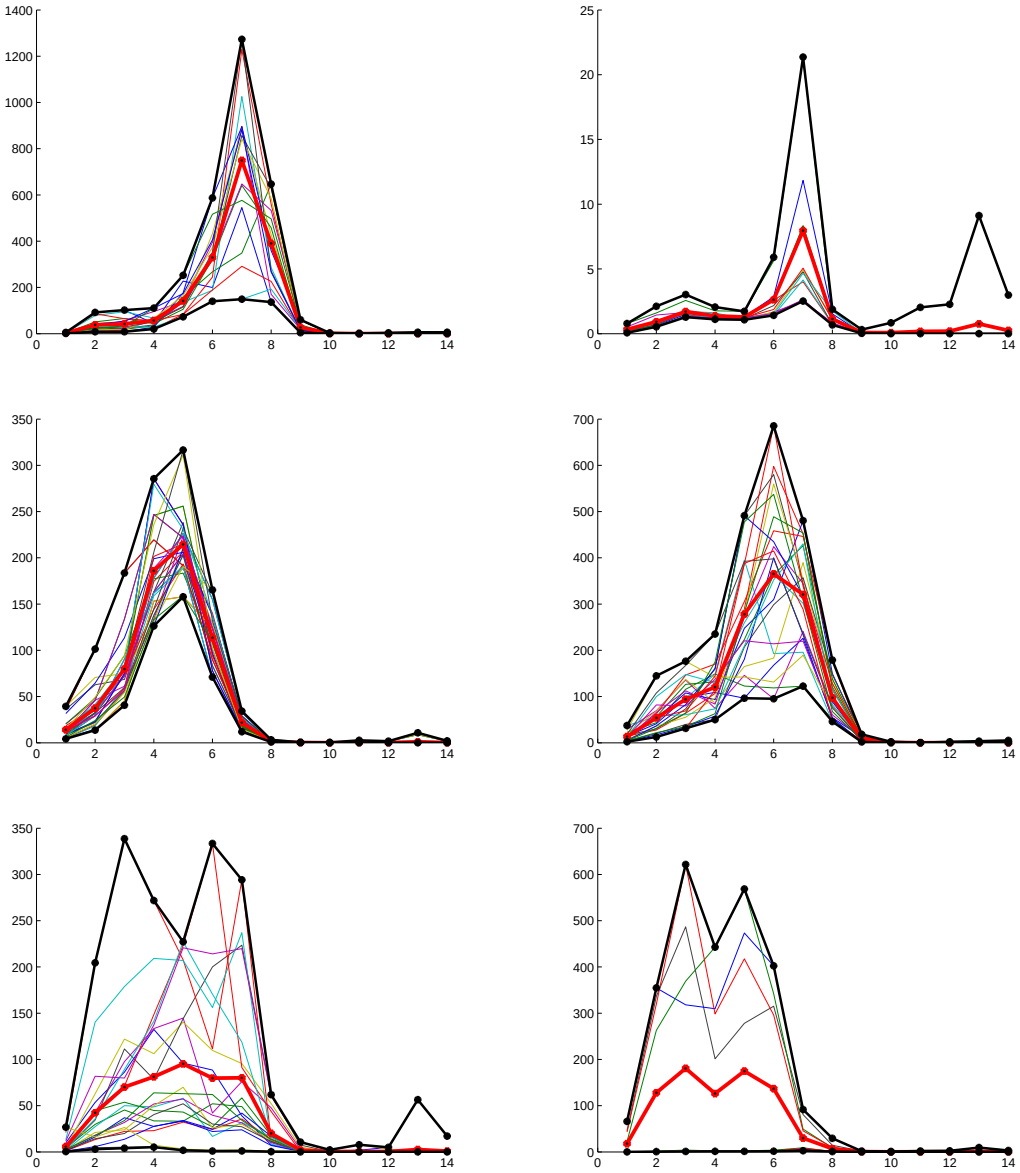


Figure 14.11 – X-axis: frequency, Y-axis: Energy. Left up: "Car-radio" environment. Right up: "Birds" environment. Left middle: "Supermarket" environment. Right middle : "road corner" environment. Left down: "School-yard" environment. Right down: "Lawn mower" environment. Set of values of the energy for each frequency (fine lines), envelope and mean criterion (thick lines).

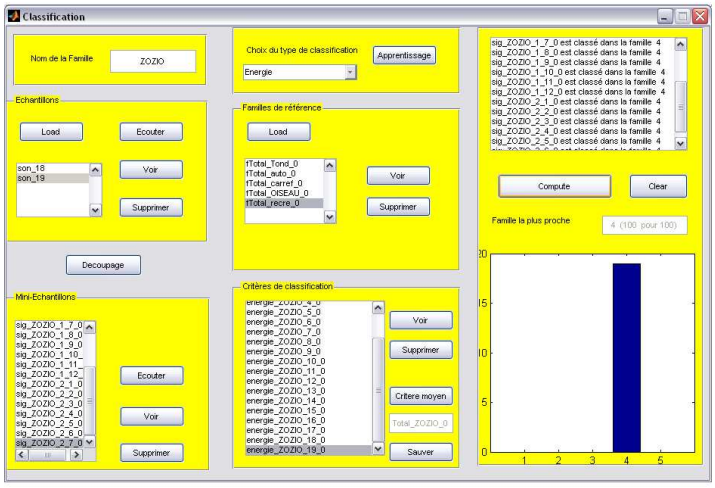


Figure 14.12 – Graphical Interface for the classification toolbox.

and Lawn-mower environments, the sample have been correctly classified with a confidence of respectively 84% and 80% (on the 13 Crossroad test samples, one is identified as a Supermarket environment and another one as a lawn-mower, and on the lawn-mower, one out of 5 samples is classified as being a crossroad).

14.7.1.5 Future work.

What needs now to be done for the scheme to be fully functional is to connect the PDA to the cochlear implant, so that if the PDA is able to classify an environment with a confidence rate greater than 50%, it selects automatically the corresponding CI fitting adapted to this sound environment and it uploads it into the CI.

If, on the contrary, the confidence rate is less than 50%, the sound environment is sampled and memorized, so that it can be classified later on (which may require to create a new sound class).

14.8 Conclusion

The problem of cochlear implants fitting belongs to a class of very difficult problems, impossible to solve in a deterministic way in a limited time, for at least two reasons:

- The function to be optimised cannot be modeled. It is extremely variable, because it is dependent on the patient and linked to a subjective evaluation of his auditive sensations.
- The search space is very large, therefore, strict optimality is out of reach.

The work presented in this paper describes an approach of this problem, based on an interactive evolutionary algorithm with a micro-population. The first results with patient A are promising: evolution has taken place (as the curves show in fig. 14.2) and the obtained results were far better than those obtained by an expert practitioner.

However, this experiment showed that it was possible to obtain good fittings by simply selecting values at random, which questions the usual aim, that is to maximise the number and range of electrodes to improve audition and comprehension. A number of other experiments has been conducted that shows that indeed, the strategy advocated by CI manufacturers may not be the best, which is a new result in the medical field.

But this work is obviously a preliminar one, that needs to be confirmed with additional experimental analysis on other patients, having various profiles. Moreover, the aim of this project is to make cochlear implants more adaptive to patients and to their environments: The adaptation to audio environment that has been sketched in section 14.7, needs now to be tested by patients in real environments.

Other points of improvements are more technical and relate to the heart of the interactive optimisation method. The real experiments presented in this paper actually prove the importance of user fatigue, which is a general problem in IEAs. But in the case of audio interaction this problem is even more crucial, for two reasons: only one signal can be evaluated at once (on the contrary to visual evaluations), and the attention needed to correctly evaluate a fitting is extremely demanding for implanted patients.

Usually, one copes with user fatigue in three ways POLI et CAGNONI [1997]; TAKAGI [1998]:

- reduce the size of the population and the number of generations,
- choose specific models to constrain the research in a priori "interesting" areas of the search space,
- perform an automatic learning (based on a limited number of characteristic quantities) in order to assist the user and only present to him the most interesting individuals of the population, with respect to previous votes of the user.

In this paper we have used the first item, i.e. a micro-EA. The experimental analysis that has been presented proves the necessity to try other strategies. According to the third item above, experiments have been conducted on another application (image denoising) with a fitness map technique LUTTON et collab. [2005], where the fitness rating has been extended to individuals of a larger population via the analysis of the user judgment on a small sample of individuals. Future work on cochlear implants could use a similar strategy, in order to evolve a larger population of parameter settings while keeping a low number of user evaluations.

Additionally, other strategies to better exploit the user interactions should be considered, such as using partial evaluations (shorter audio tests), and refinements of audition, understanding and comfort evaluations only on areas of the search space that have been identified as promising by the IEA.

Acknowledgements

We would like to thank *Neurelec* (an *MXM* company, http://www.neurelec.com) who provided us with equipment that made this research possible.

References

ARCHBOLD, S., M. LUTMAN et D. MARSHALL. 1995, «Categories of auditory performance», dans *International cochlear implants, speech and hearing symposium, Melbourne. Annals of Otology, Rhinology and Laryngology*, vol. 104, édité par G. Clarck et R. Cowan, p. 312–314. 297

BÄCK, T. 1996, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New-York. 301

BÄCK, T. 2005, «Tutorial on evolution strategies», Genetic and Evolutionary Computation Conference Gecco'05. 300, 301

BLAMEY, P., B. PYMAN, M. GORDON, G. CLARK, A. BROWN, R. DOWELL et R. HOLLOW. 1992, «Factors predicting postoperative sentence scores in postlinguistically deaf adult cochlear implant patients», dans *Annals of Otology, Rhinology and Laryngology*, vol. 101(4), p. 342–348. 300

BLICKLE, T. et L. THIELE. 1996, «A comparison of selection schemes used in evolutionary algorithms», dans *Evolutionary Computation*, vol. 4, p. 361–394. URL citeseer.ist.psu.edu/blickle97comparison.html. 302

BOURGEOIS-RÉPUBLIQUE, C., B. FRACHET et P. COLLET. 2005, «Using an interactive evolutionary algorithm to help fitting a cochlear implant», dans *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, ACM Press, New York, NY, USA, p. 133–139, doi:http://doi.acm.org/10.1145/1102256.1102287. 303, 316

BOURGEOIS-RÉPUBLIQUE, C. 2004, *Plateforme de réglage automatique et adaptatif d'implant cochléaire par algorithme évolutionnaire interactif*, Phd thesis, University of Bourgogne, Fr. 303

BOURGEOIS-RÉPUBLIQUE, C., G. VALIGIANI et P. COLLET. 2005, «An interactive evolutionary algorithm for cochlear implant fitting: first results.», dans *Proceedings of the 2005 ACM symposium on Applied computing*, p. 231–235. 303

CHOUARD, C., C. FUGAIN, B. MEYER et H. LACOMBE. 1983, «Long term results of the multichannel cochlear implant», dans *Annals of the New-York Academy of Sciences*, vol. 405, p. 387–411. 297

CHOUARD, C., M. OUAYOUN et B. MEYER. 1995, «Speech coding strategies of the digisonic fully digitized cochlear implant», dans *Acta Otolaryngol, Stockholm*, vol. 115, p. 264–268. 297

COHEN, L. 1989, «Time frequency distribution a review», dans *Proceedings IEEE volume 77*, p. 941–981. 297

COLLET, P., E. LUTTON, M. SCHOENAUER et J. LOUCHET. 2000, «Take it EASEA», dans *Parallel Problem Solving from Nature - PPSN VI 6th International Conference, LNCS*, vol. 1917, édité par M. Schoenauer, K. Deb, G. Rudolph, E. L. X. Yao, J. J. Merelo et H.-P. Schwefel, Springer-Verlag, Paris, France, p. 891–901. URL http://minimum.inria.fr/evo-lab/Publications/PPSNVI.ps.gz. 303

DAUBECHIES, I. 1992, *Ten Lectures on Wavelets*, vol. 61, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia. 318

DE JONG, K. 2005, *Evolutionary Computation: a Unified Approach*, MIT Press. 301

DORMAN, M., K. DANKOWSKI, G. MCCANDLESS et S. L.M. 1989, «Consonant recognition as a function of the number of channels of stimulation by patients who use the symbion cochlear implant», dans *Ear and Hearing*, vol. 10(5), p. 288–291. 300

DURANT, E. 2002, *Hearing Aid fitting with Genetic Algorithms*, Phd thesis, University of Michigan, USA. 300

FISHMAN, G. 1996, *Monte-Carlo: Concepts, Algorithms and Applications*, Springler-Verlarg, New York. 300

FRANCK, K., L. XU et B. PFINGST. 2003, «Effect of stimulus level on speech perception with cochlear prostheses», dans *J. Assoc. Res. Otolaryngol.*, vol. 4(1), p. 49–59. 300

FRIESEN, L., R. SHANNON, D. BASKENT et X. WANG. 2001, «Speech recognition in noise as a function of the number of spectral channels: comparison of acoustic hearing and cochlear implants», dans *J. Acoust. Soc. Am.*, vol. 110(2), p. 1150–1163. 300

GALLEGO, S., B. FRACHET, C. MICHEYL, E. TRUY et L. COLLET. 1998, «Cochlear implant performance and electrically-evoked auditory brain-stem response characteristics», dans *Electroencephalography neurophysiologie and clinical neurophysiology*, vol. 108, p. 521–525. 296

HESSE, G. 2002, *Programmation des seuils liminaires de l'implant cochléaire MED-EL Tempo+*, Thèse de doctorat de médecine, Université de Rennes, Fr. 299

JANSEN, T. 2002, «On the analysis of dynamic restart strategies for evolutionary algorithms.», dans *Parallel Problem Solving from Nature - PPSN VII, Granada, Spain*, vol. 2439, édité par B. Springer, p. 33–43. 301

KAWANO, A., H. SELDON, G. CLARK, R. RAMSDEN et C. RAINE. 1998, «Intracochlear factors contributing to psychophysical percepts following cochlear implantation», dans *Acta Otolaryngol.*, vol. 118(3), p. 313–326. 299

KIEFER, J., C. VONILBERG, V. RUPPRECHT, J. HUBNER-EGNER et R. KNECHT. 2000, «Optimized speech understanding with the cis sampling speech coding strategy in patients with cochlear implants: effect of variations in stimulation rate and number of channels», dans *Annals of Otology, Rhinology and Laryngology*, vol. 109(11), p. 1009–1020. 300

KRISHNAKUMAR, K. 1989, «Micro-genetic algorithms for stationary and non-stationary function optimization», dans *SPIE Procedings: Intelligent Control and Adaptive Systems*, vol. 1196, Philadelphia, PA, p. 289–296. 301

LAFON, J. 1964, *Le test phonétique et la mesure de l'audition*, Ed. Centrex, Eindhoven. 304

LAWSON, D., B. WILSON, M. ZERBI et C. FINLEY. 1996, «Third quaterly progress report: Speech processors for auditory prostheses», cahier de recherche NIH Contract N01-DC-5-2103. 300

LOIZOU, P., O. POROY et M. DORMAN. 2000, «The effect of parametric variations of cochlear implant processors on speech understanding», dans *Journal of Acoustical Society of America*, vol. 108(2), p. 790–802. 297, 299

LUTTON, E., M. PILZ et J. LEVY-VEHEL. 2005, «The fitness map scheme. application to interactive multifractal image denoising.», dans *CEC2005, IEEE Congress on Evolutionary Computation is in Edinburgh, UK*, vol. 3, p. 2278–2285. 323

MEYER, Y. 1990, *Ondelettes et Opérateurs*, Hermann, Paris. 318

MOORE, B. 1995, *Perceptual consequences of cochlear damage*, Oxford Medical Publication. 297

OSBERGER, M. 1997, «Cochlear implantation in children under the age of two years: candidacy considerations», dans *Otolaryngol of Head and Neck Surgery*, vol. 117, p. 145–149. 297

PIALOUX, P., C. CHOUARD, B. MEYER et C. FUGAIN. 1979, «Indications and results of the multichannel cochlear implant», dans *Acta Otolaryngol*, vol. 87(3-4), p. 185–189. 297

POLI, R. et S. CAGNONI. 1997, «Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement», dans *Genetic Programming 1997: Proceedings of the Second Annual Conference*, édité par J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba et R. L. Riolo, Morgan Kaufmann, Stanford University, CA, USA, p. 269–277. URL `citeseer.ist.psu.edu/article/poli97genetic.html`. 323

ROMAN, S. 1998, *La réhabilitation des sourds profonds par implant cochléaire*, Thèse de doctorat de médecine, Université de Marseille, Fr. 297

ROUX, G. 2001, *Synthèse et réalisation d'études cliniques sur l'implant cochléaire*, Thèse de doctorat de médecine, Université de Rennes, Fr. 299

SHANNON, R., J. R. GALVIN et D. BASKENT. 2002, «Holes in hearing», dans *J. Assoc. Res. Otolaryngol.*, vol. 3(2), p. 185–199. 300

STICKNEY, G., P. LOIZOU, L. MISHRA, P. ASSMANN, R. SHANNON et J. OPIE. 2006, «Effects of electrode design and configuration on channel interactions», dans *Hearing Research*, vol. 211, p. 33–45. 300

TAKAGI, H. 1998, «Interactive evolutionary computation: System optimization based on human subjective evaluation», dans *Proceedings of the IEEE International Conference on Intelligent Engineering Systems (INES'98)*, Vienna, Austria, p. 1–6. 302, 323

TAKAGI, H. 2001, «Interactive evolution computation: Fusion of the capabilities of ec optimization and human evaluation», dans *Proceedings of the IEEE*, vol. 89, p. 1275–1296. 300

WALL, M. «Matthew's genetic library», Http://lancet.mit.edu/ga/. 303

ZWOLAN, T., L. COLLINS et G. WAKEFIELD. 1997, «Electrode discrimination and speech recognition in postlingually deafened adult cochlear implant subjects», dans *J. Acoust. Soc. Am.*, vol. 102(6), p. 3673–3685. 300

Chapter 15

Feature extraction and classification of EEG signals. The use of a genetic algorithm for an application on alertness prediction

This chapter is related to the PhD thesis of Laurent Vezard, the PSI region project and the ACOBSEC European project. A slightly different version has been published in a book chapter. Eduardo Miranda; Julien Castet; Benjamin Knapp. Guide to Brain-Computer Music Interfacing, Springer, 2014. Work carried out with Laurent Vézard, Marie Chavent, Frédérique Faïta-Aïnseba and Leonardo Trujillo.

Contents

15.1 Introduction

Over the last decade, Human-Computer Interaction (HCI) has grown and matured as a field. Gone are the days when only a mouse and keyboard could be used to interact with a computer. The most ambitious of such interfaces are Brain-Computer Interaction (BCI) systems. The goal in BCI is to allow a person to interact with an artificial system using only brain activity. The most common approach towards BCI is to analyze, categorize and interpret Electroencephalographic (EEG) signals, in such a way that they alter the state of a computer.

In particular, the objective of the present work is to study the development of computer systems for the automatic analysis and classification of mental states of vigilance; i.e., a person's state of alertness. Such a task is relevant to diverse domains, where a person is expected or required to be in a particular state. For instance, pilots, security personnel or medical staffs are expected to be in a highly alert state, and a BCI could help confirm this or detect possible problems.

It is possible to assume that the specific topic presented in this chapter lies outside the scope of the book entitled "Guide to Brain-Computer Music Interfacing" (where this work has been published). Nevertheless, from our point of view, many tasks have to be accomplished before any interaction between a person's brain and music can be done by using EEG signals. Suppose that we wish to develop a musical instrument that can generate music that is specifically related to the alertness of a subject. For such a system, a first objective should be to classify the EEG signals of a subject based on different levels of alertness. In order to reach this objective, informative features have to be extracted, particularly since processing raw EEG data is highly impractical, and then proceed to a final classification step using relevant mathematical concepts. However, this problem is by no means a trivial one. In fact, EEG signals are known to be highly noisy, irregular and tend to vary significantly from person to person, making the development of general techniques a very difficult scientific endeavor. Then, it is important to find a method that is adaptable to different persons and that it provides a rapid and accurate prediction of the alertness state.

15.1.1 Electroencephalographic signals and previous works

The electrical activity of the brain is divided into different oscillatory rhythms characterized by their frequency bands. The main rhythms in ascending order of frequency are delta (1-3.5 Hz), theta (4-8 Hz), alpha (8-12 Hz) and beta (12-30 Hz). Alpha waves are characteristic of a diffuse awake state for healthy subjects and can be used to discern the normal awake and relaxed states, which is the topic of this experimental study. The oscillatory alpha rhythm appears as visually observable puffs on the electroencephalogram, especially over the occipital brain areas at the back of the skull, but also under certain conditions in more frontal recordings sites. The distribution of cortical electrical activity is taken into account in the characterization of an oscillatory rhythm. This distribution can be compared between studies reported in the literature through the use of a conventional electrode placement; the international system defined by JASPER [1958] and shown in Figure 15.1.

Furthermore, the brain electrical activity is non-stationary, as specified in SUBASI et collab. [2005]; i.e., the frequency content of EEG signals is time varying. EEG signals are almost always pre-treated before any analysis is performed. In most cases, the Fourier transform or discrete wavelet decomposition (DWT) are used (see section 15.4.1). In SUBASI et collab. [2005], authors use a DWT to pick out the wavelet subband frequencies (alpha, delta, theta and beta) and use it as an input to a neural networks classifier. In HAZARIKA et collab. [1997], coefficients of a DWT are used as features to describe the EEG signal. These features are given as an input to an artificial neural network.

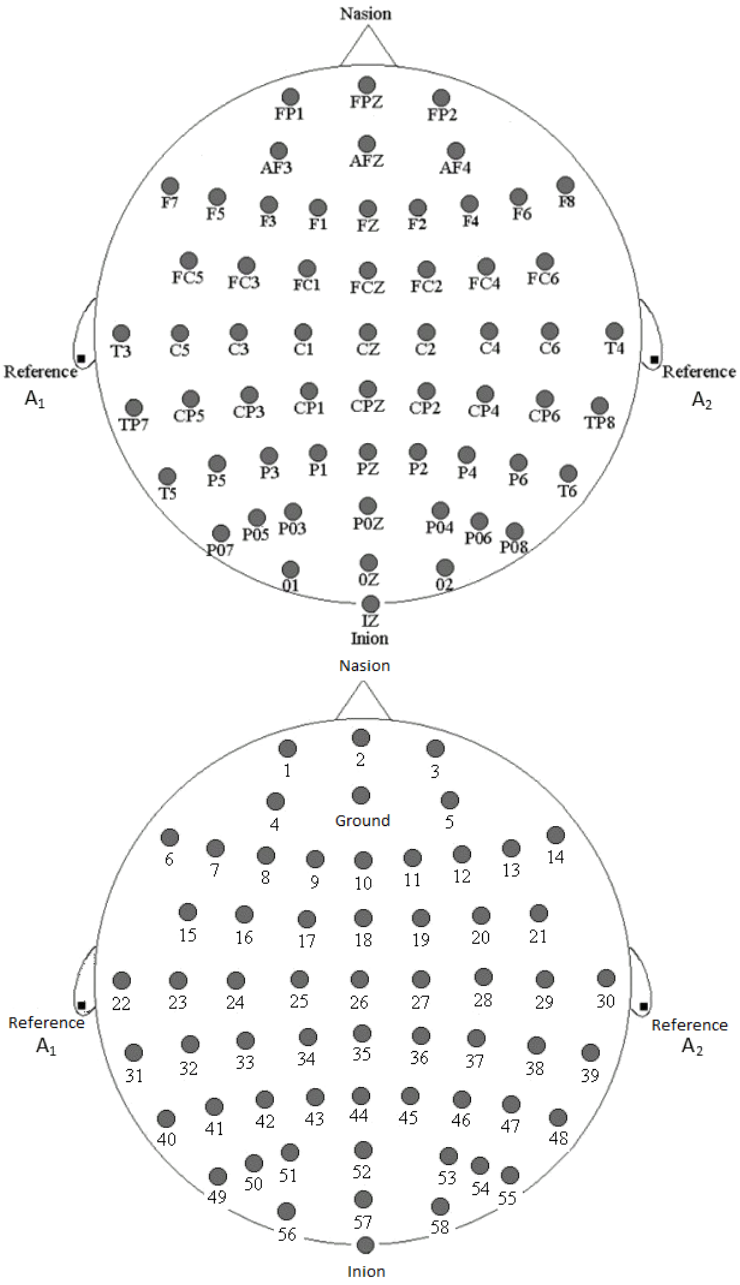


Figure 15.1 – Representation of the distribution of electrodes in the international system 10/10.

In BEN KHALIFA et collab. [2005], the EEG signal is decomposed in 23 bands of 1 Hz (from 1 to 23Hz) and a Short Term Fast Fourier transformation (STFFT) is used to calculate the percentage of the power spectrum of each band. In CECOTTI et GRAESER [2008], a Fourier transform is used between hidden layers of a convolutional neural network to switch from the time domain to the frequency domain analysis in the network.

To predict the state of alertness, the most common method is neural networks (see for example SUBASI et collab. [2005] or VUCKOVIC et collab. [2002]). However, the disadvantage of this approach is that it requires having a large set of test subjects relative to the number of predictive variables. To avoid this problem, the authors of SUBASI et collab. [2005] and VUCKOVIC et collab. [2002] split their signal into several segments of a few seconds, called “epochs". Other approaches use different statistical methods. For example, YEO et collab. [2009] uses Support Vector Machine, ANDERSON et SIJERCIC [1996] uses autoregressive models (AR) and OBERMAIER et collab. [2001] use hidden Markov chains.

15.1.2 Main contributions

The aim of the work presented in this chapter is to construct a model that is able to predict the alertness state of a human using one electrode; and this model will be used in real time applications. That is why, the two main objectives are:

- Reduce the time needed to install the EEG cap on a participant using a variable selection method in order to choose the best electrode (based on classification rate). In fact, in real world applications, it is necessary to reduce the number of electrodes needed because the cap installation process has to be short. A long installation of the EEG cap can cause a disturbance of the mental state of the person that we want to study (pilots or surgeons for example).
- To obtain a model (decision rule) which is able to give a reliable prediction of the alertness state of a new participant.

To achieve these objectives, we apply a wavelet decomposition as a pre-processing step and a new criterion for state discrimination is proposed. Then, several standard methods for supervised classification (binary decision tree, random forests and others) are used to predict the state of alertness of the participants. The criterion is then refined using a genetic algorithm to improve the quality of the prediction. Finally, this work presents results that are part of a broader research program that is being investigated by the lead authors, focusing on the development of BCIs. In particular, this chapter contains a detailed description of the system originally presented in VÉZARD et collab. [2013], where critical aspects were not discussed in detail.

The remainder of this chapter proceeds as follows. The data acquisition protocol is precisely detailed in the section 15.2. The validation of the data is described in the section 15.3. A data pre-processing is proposed in section 15.4 and a feature extraction is performed section 15.5 in order to compute a first attempt of classification of EEG signals. Section 15.6 contains the general principles of a genetic algorithm and presents how this stochastic optimization method improves the results obtained in the previous section. Finally, Section 15.7 presents a summary of this work and discusses our main conclusions.

15.2 Data acquisition

This work is based on real data that we have collected. This section will describe the data acquisition and data validation steps.

15.2.1 Participants

This work uses 44 participants, with ages between 18 and 35, all are right-handed, to avoid variations in the characteristics of the EEG due to age or handedness linked to a functional interhemispheric asymmetry.

15.2.2 Procedure

The experiment was conducted individually in a soundproof room, where the participant was comfortably seated in front of the computer screen (see Figures 15.2 and 15.4).

It takes approximately two hours and a half to place the EEG cap, to perform the experiment and to have a final explanatory interview with the participant. This interview occurred at the end of the whole data acquisition procedure to not affect EEG records. Data collection was controlled by the acquisition system Coherence 3NT (Deltamed, `http://www.natus.com/`). The data acquisition procedure is composed of five steps which are represented in Figure 15.3:

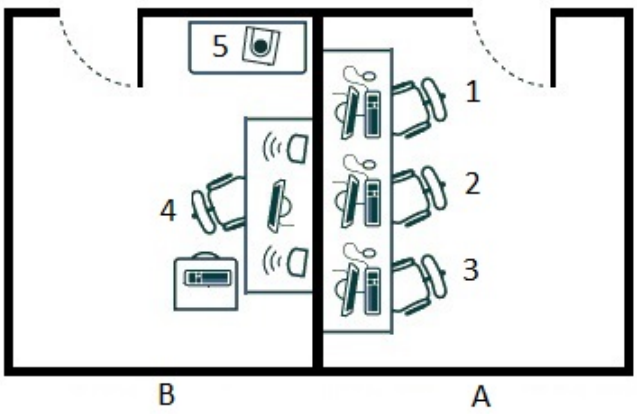


Figure 15.2 – Experimentation rooms. A: Control room. B: Room of the participant. 1: Recording computer. 2: Computer devoted to the relaxation process. 3: Control computer linked to the control camera, 4: Participant. 5: control camera.

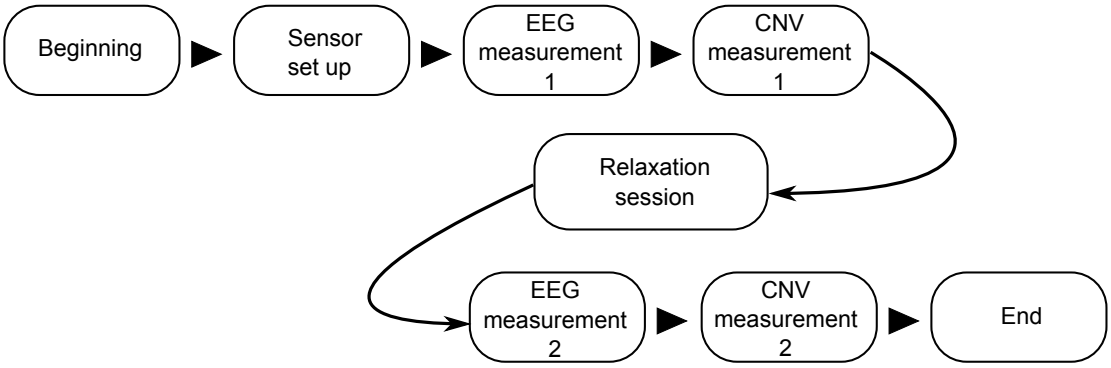


Figure 15.3 – Diagram of the data acquisition procedure.

1. First EEG: the participant has to look at a cross (fixation point) at the center of the screen to reduce eye movements. This first recording corresponds to the reference state, considered as the normal vigilance state of the participant. A photograph of a member of our team, took to represent the conditions of an EEG recording, is given in Figure 15.4.

2. Attentional task devoted to collect contingent negative variation (CNV): the participant was instructed to press the space bar as quickly as possible after each time the cross was replaced by a square on the screen. For each appearance of this square, a warning sound (beep), presented 2.5 seconds before, allowed the participant to prepare his response. The experimental session included 50 pairs of stimuli (S1: beep, S2: square), with a random amount of time elapsing between each pair. The purpose of this task is specified in Subsection 15.3.1.

3. Relaxation session: the participant was fully guided by a soundtrack broadcast through loudspeakers placed in the room. The soundtrack suggested the participant to perform three successive exercises of self-relaxation, based on muscular relaxation and mental visualization. The first exercise is the autogenic training SCHULTZ [1958]. In this exercise, the participant has to mentally repeating some sentences like for example "I am calm" or "my arms and legs are heavy". The second exercise is the progressive relaxation JACOBSON [1974]. It consists in tense and unflex some muscles of the body. The last exercise is the mental visualization. The participant imagines that he is moving in a familiar and lovely place. The purpose of this relaxation session is to try to bring the participant to a lower level of vigilance, qualified as the “relaxed" state.

Figure 15.4 – Photograph that represents the conditions during an EEG recording.

4. Second EEG recording: 3 minutes of EEG were recorded with the same protocol as in the step 1. This second recording should reflect the relaxed state of the participant's brain if it was reached in the prior step.

5. Second CNV task: CNV is collected using exactly the same protocol as in step 2.

15.3 Data validation

15.3.1 Contingent negative variation extraction

For a given participant, the CNV analysis will allow us to determine if the relaxation step was effective. CNV extraction has been performed by applying the Event-Related Potentials (ERPs) method ROSENBLITH [1959]. It consists, in the present experimental design, on averaging the electrical activity recorded in synchrony with all warning signals (S1: beep) until the response stimulus (S2: square). Such average allows event-related brain activity components, reflecting stimulus processing, to emerge from the overall cortical electrical activity, unrelated to the task performed. Thus in our paradigm, a negative deflection of the averaged waveform, called CNV, is obtained WALTER et collab. [1964]. This attentional component has the property of decreasing in amplitude when the participant is less alert, either because he is distracted TECCE [1979], is deprived of sleep NAITOH et collab. [1971] or is falling asleep TIMSIT-BERTHIER et collab. [1981]. This fundamental result is shown in Figure 15.5. In this Figure, the CNV is plotted as a dotted line for an alert participant and as a solid line for a participant which is less alert. The amplitude of the CNV is proportional to the alertness of the subject.

That is why, although the instruction given to the participant during CNV acquisition was to press the space bar as quickly as possible after the square appearance, the reaction time is not investigated in this study. However, the way the participant prepares to perform the task is observed.

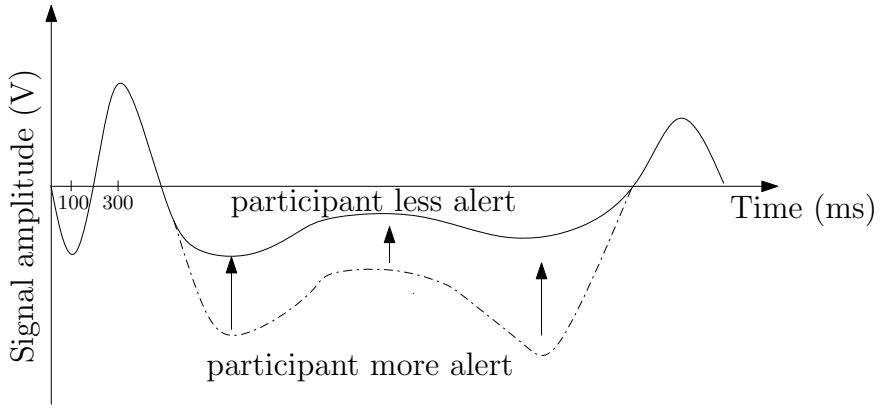


Figure 15.5 – Representation of the amplitude variation of the CNV with respect to the alertness of a participant.

The comparison of the amplitude of the CNV between tasks performed in steps 2 and 5 is used to determine if the alertness of a participant has changed. It allows us to know if he is actually relaxed. Only the positive cases, for which the amplitude of the CNV has significantly declined, were selected for comparative analysis of their raw EEG's (stages 1 and 4). Their EEG were then tagged respectively as "normal" or "relaxed" state. An example of a participant kept after studying his CNV is shown in Figure 15.6 and an example of a rejected participant is given in Figure 15.7.

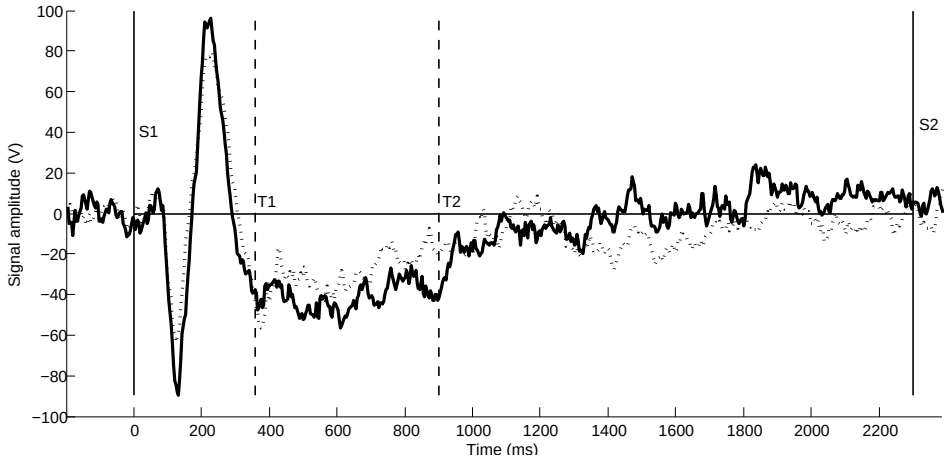


Figure 15.6 – Representation of CNV recorded on participant 4 during steps 2 (solid curve) and 5 (dotted curve). The solid vertical lines correspond to warning signals (S1: beep, S2: square). This participant is kept because the solid curve is mainly below the dotted curve between T1 and T2 (framed by the dotted vertical lines).

In these Figures, the solid curve represents the CNV recorded during step 2 and the dotted curve represents the CNV recorded in step 5. The solid vertical lines correspond to warning signals (S1: beep, S2: square). The area between the curve and the x-axis is calculated between T1 and T2 (section framed by the dotted vertical lines). A participant is kept if the area calculated with the CNV recorded in step 5 is lower than the area calculated with the CNV recorded in step 2. To facilitate this validation step an allow a visual inspection of the curves, a graphical user interface was created. This interface is given in Figure 15.8. Using this interface, an user can easily plot the CNV curve for a given participant. The top right of Figure 15.8 is a topographic map. At a given period, it represents the electrical activity recorded on the scalp of a participant. It allows to view the appearance of the CNV on the scalp and thus to locate brain regions involved in the CNV appearance.

The study of CNV was performed on the 44 participants of the study and 13 participants

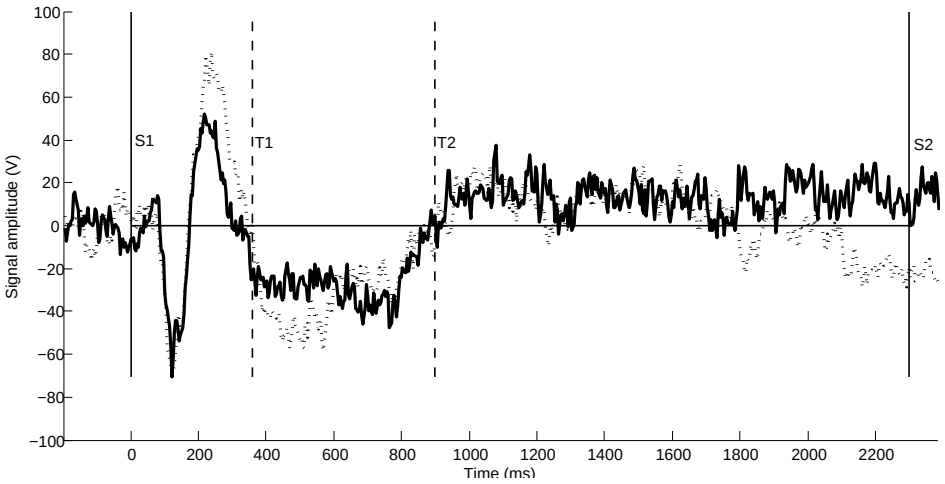


Figure 15.7 – Representation of CNV recorded on participant 9 during steps 2 (solid curve) and 5 (dotted curve). The solid vertical lines correspond to warning signals (S1: beep, S2: square). This participant is rejected because the solid curve is mainly above the dotted curve between T1 and T2 (framed by the dotted vertical lines).

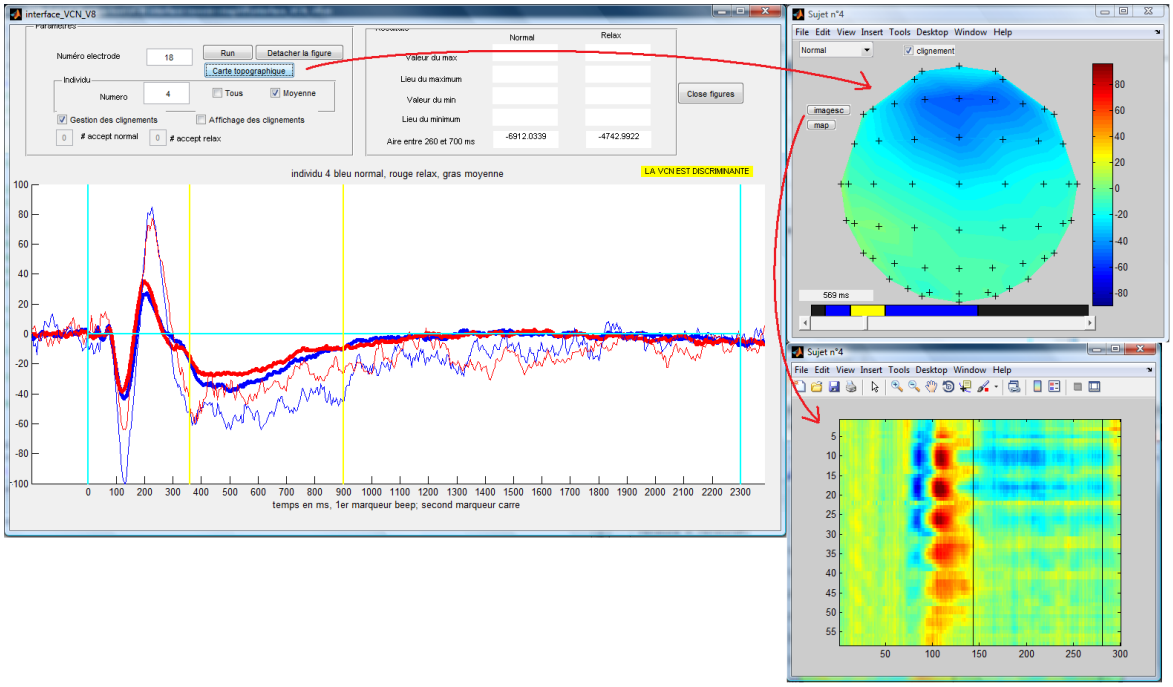


Figure 15.8 – Graphical user interface for the CNV display.

were kept for further analysis. Thus, an important number of participants are rejected. The stress due to the experiment and the duration of the installation of the cap may be factors that deteriorate the efficiency of the relaxation session. To limit the duration of the cap wearing, the relaxation session is relatively short. Thus, it is possible that the duration of the relaxation session (20 minutes) is too short to achieve fully relax these subjects. The participants selected are those that have special abilities to relax in stressful conditions and in a relatively short period of time. Those points can explain the high proportion of rejected participants in our study.

15.3.2 Data

Finally, the data consist of 26 records of 3 minutes of raw EEG signals from the 13 selected participants (one “normal" EEG and one “relaxed" EEG for each participant). Each record contains variations of electric potential obtained with a sampling frequency of 256 Hz with 58 active electrodes placed on a cap (ElectroCap). Using this sampling frequency, each signal recorded by an electrode for a given subject in a given alertness state contains 46000 data points. A representation of the data matrix is given in Figure 15.9.

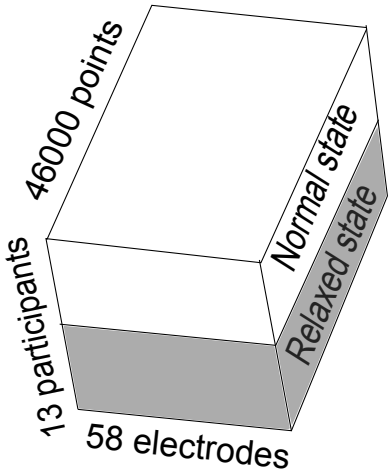


Figure 15.9 – Representation of the data matrix. There are three dimensions: one for the participants, one for the time (46000 points corresponding to the number of points in each 3 minutes EEG signals recorded using a sampling frequency of 256 Hz) and one for the electrodes.

15.4 Data pre-processing

The data is specified in 3 dimensions (time, electrodes and participants). The proposed approach is to extract a feature in 2 dimensions to implement common classification tools. To do this, the signal energy, obtained by the wavelet decomposition, is considered.

15.4.1 Wavelet decomposition

Wavelet decomposition DAUBECHIES [1992], MALLAT [2008] is a method widely used in signal processing. Its main advantage is that it can be used to analyze the evolution of the frequency content of a signal in time. It is therefore more suitable than the Fourier transform for analyzing non-stationary signals.

A wavelet is a function $\psi \in L^2(\mathbb{R})$ such that $\int_{\mathbb{R}} \psi(t)dt = 0$. The continuous wavelet transform of a signal X can be written as:

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} X(t)\psi\left(\frac{t-b}{a}\right) dt$$

where a is called the scale factor that represents the inverse of the signal frequency, b is a time-translation term and function ψ is called the mother wavelet. The mother wavelet is usually a continuous and differentiable function with compact support. Several families of wavelet mother exist such as Daubechies wavelets or Coiflets.
Some wavelets are given Figure 15.10.

It is also possible to define the discrete wavelet transform, starting from the previous formula and discretizing parameters a and b. Then, let $a = a_0^j$, where a_0 is the resolution parameter such as $a_0 > 1$ and $j \in \mathbb{N}$ and let $b = kb_0a_0^j$, where $k \in \mathbb{N}$ and $b_0 > 0$. It is very

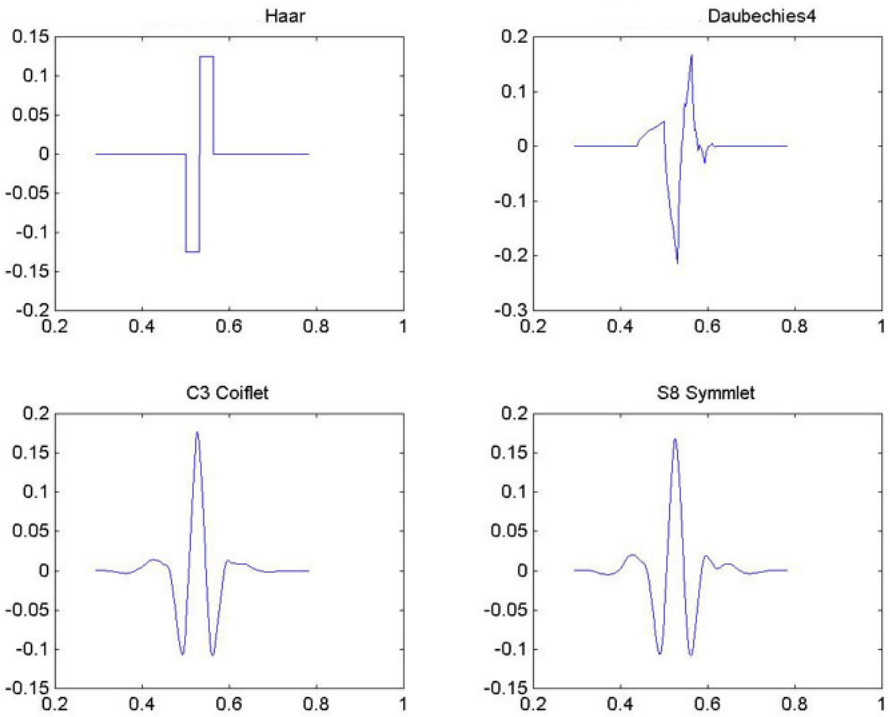


Figure 15.10 – Some wavelets.

common to consider the “dyadic" wavelet transform which corresponds to the case where $a_0 = 2$ and $b_0 = 1$. In this case, $j = 1, 2, ...n$, where n is the base-2 logarithm of the number of points forming the signal and $k = 1, 2, ..., 2^{j-1}$. Then, the dyadic discrete wavelet transform is:

$$x_{j,k} = 2^{-\frac{j}{2}} \int_{-\infty}^{\infty} X(t)\psi(2^{-j}t - k)dt$$

where j is the decomposition level (or scale) and k the time lag. The maximal number of decomposition levels, n, is the log_2 of the number of points forming the signal. The discrete wavelet transform is faster than the continuous version and also allows for an exact reconstruction of the original signal by inverse transformation. The dyadic grid provides a spatial frequency representation of discrete dyadic wavelet transform (see Figure 15.11). In this Figure, the x-axis corresponds to time, the y-axis represents the frequencies and the circles correspond to the wavelet coefficients $x_{j,k}$. The signal points are represented below the last level of decomposition. At each additional level, the frequency is doubled.

The dyadic grid allows us to visualize the frequency content of the signal and to see when these frequencies appear. For example, Figure 15.12 represents a signal and his DWT computed by the toolbox FracLab LEVY VEHEL et LEGRAND [2004], http://fraclab.saclay.inria.fr/ (see Figure 15.13). Below, the dyadic grid is presented, containing the absolute value of the discrete wavelet coefficients of the above signal. The high coefficients values are in red and the low values in blue. In Figure 15.12, the second level of decomposition, related to low frequencies, contains high absolute coefficients values on the complete signal. The fifth scale contains mid-range value coefficients in the last part of the signal. Finally the last scale allows to visualize the high frequency content appearing at the beginning and at the end of the signal.

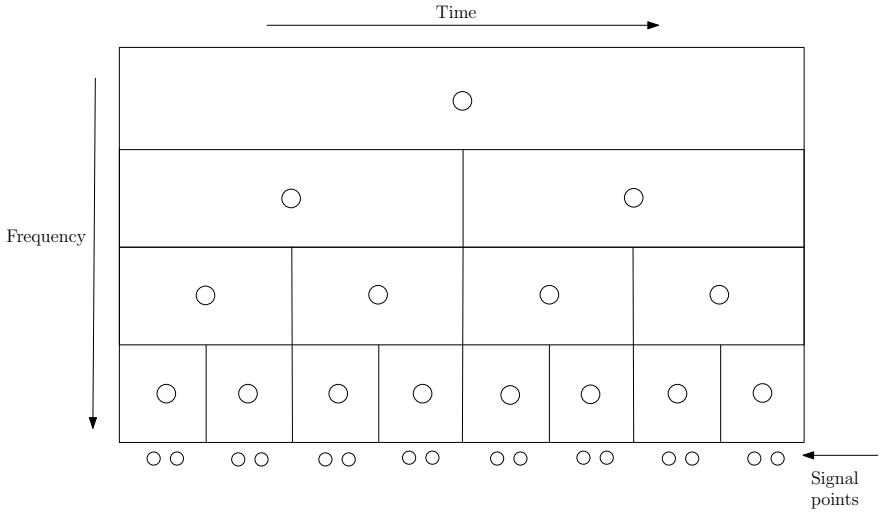


Figure 15.11 – Representation of the dyadic grid with 4 levels of decomposition (4 scales).

15.4.2 Signal Energy

Wavelet decomposition can also be used to calculate the energy of a signal for each level of decomposition. Thus, the energy e_j^2 of the signal X in the scale j is given by:

$$e_j^2 = \sum_{k=1}^{2^{j-1}} x_{j,k}^2, \forall j \in \{1, ...2^{j-1}\}.$$

In other words, from the dyadic grid, the energy associated with the scale j (decomposition level j) is equal to the sum of the squares of the coefficients of the line j. The use of signal leads to a loss of the temporality information. It is also possible to obtain this result using a Fourier transform, however, the discrete wavelet decomposition provides more opportunities for further work. For example, the wavelet decomposition could be useful if the temporal evolution of the frequency content of signals is investigated in a future work.

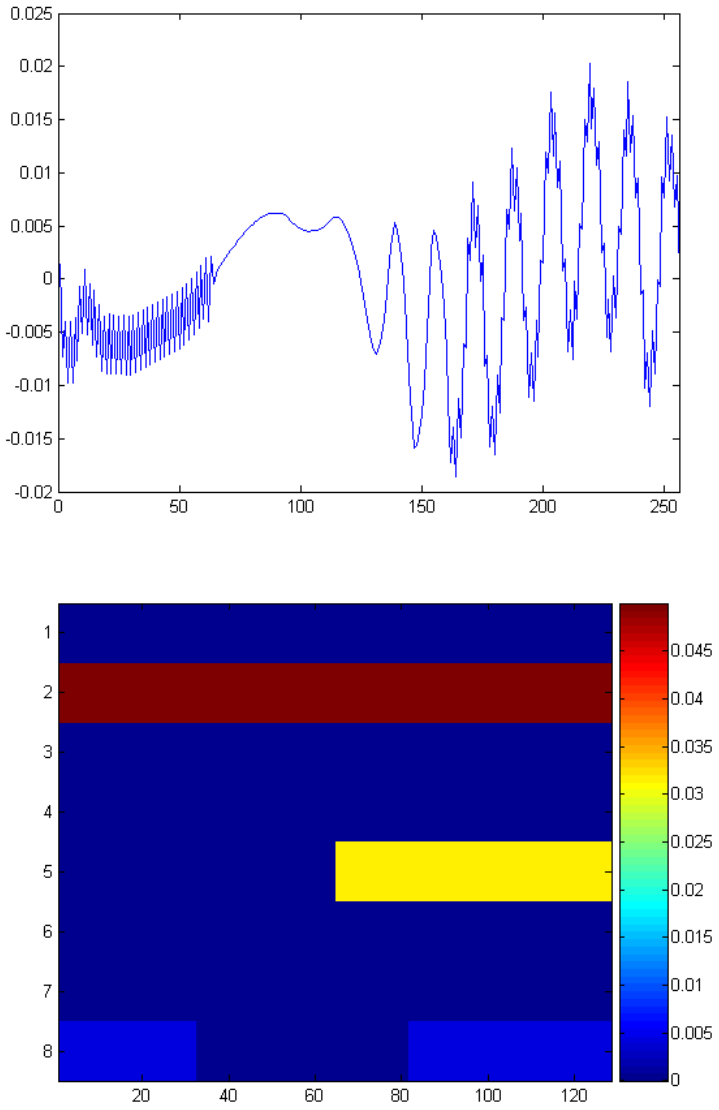


Figure 15.12 – A signal generated with the toolbox FracLab. Below, the dyadic grid, containing the absolute value of the discrete wavelet coefficients of the signal. The large coefficients are in red and the smallest values in blue.

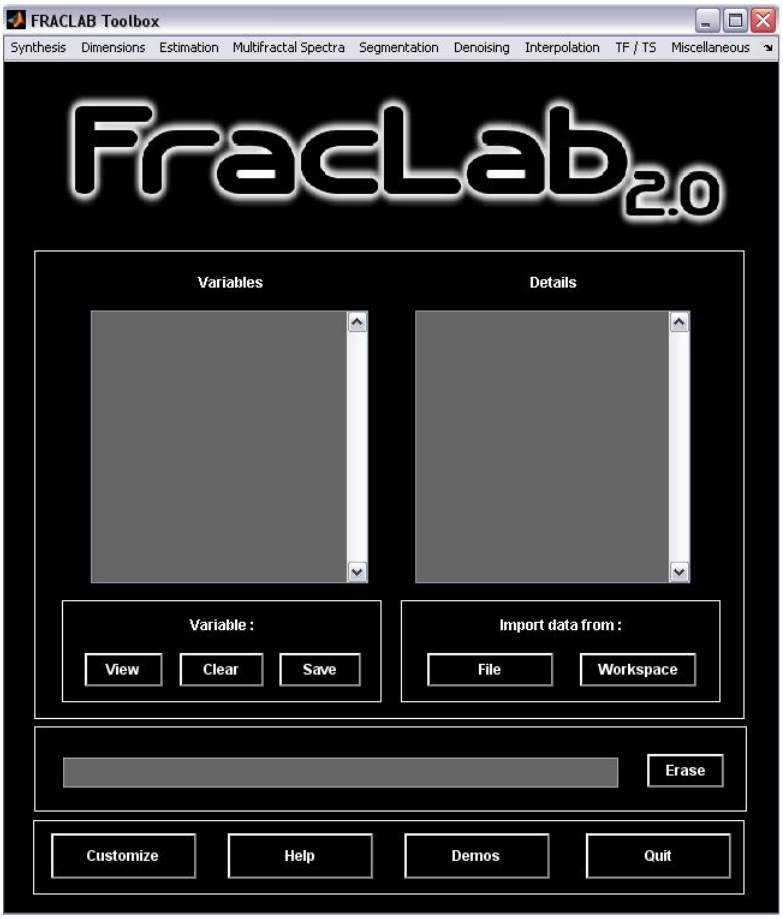


Figure 15.13 – The graphical user interface of the FracLab toolbox.

15.5 Examples of feature extraction

15.5.1 Slope criterion

For a given participant i ($i = 1, ..., 13$) in a given state (normal or relaxed), each electrode m ($m = 1, ..., 58$) provides a signal X_m. A discrete dyadic wavelet decomposition is performed on this signal by considering 15 scales ($15 = \lfloor log_2(46000) \rfloor$), where 46000 is the number of points in each 3 minutes EEG signals and where $\lfloor . \rfloor$ is the integer part). From the coefficients obtained, the energy of the signal is calculated for each scale. Figure 15.14 presents these energies as a function of frequency.

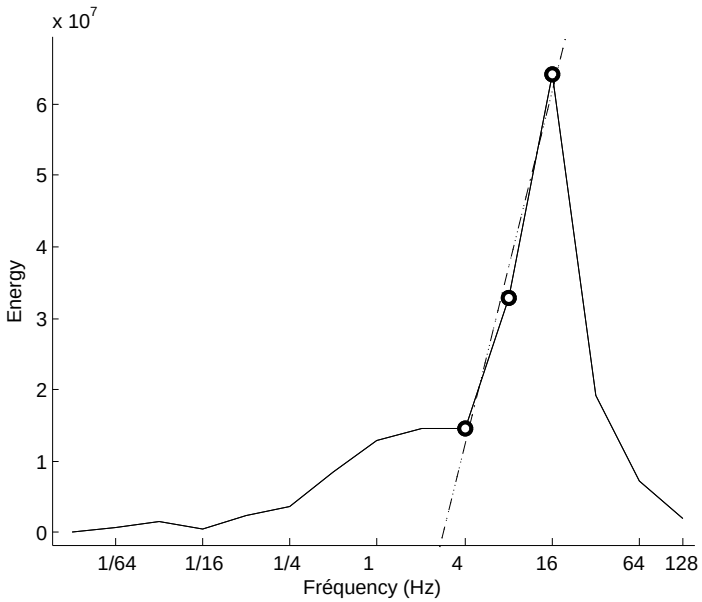


Figure 15.14 – Representation of the energy of signal X_m obtained using a discrete dyadic wavelet decomposition as a function of frequency. To calculate the slope criterion, a simple regression is performed (dotted line) on the energies calculated for 4, 8 and 16 Hz (circles).

The Alpha waves are between 8 and 12 Hz. Thus, according to the literature, only the energies calculated for 4, 8 and 16 Hz are used (black circles in Figure 15.14). Then, a simple regression is performed (dotted line in Figure 15.14) and the slope is retained. This coefficient is representative of the evolution of signal energy in the frequency considered. By repeating this process for each electrode, a feature of 58 coefficients (one per electrode) is obtained for an individual in a given state. Thus, a matrix of size 26×58 is obtained, representing the slope criterion.

The Figure 15.15 give a representation of the data matrix after a dimension reduction. On the left, the data obtained after a discrete wavelet transform. There are still three dimensions: one for the participants, one for the 15 frequencies and one for the electrodes. Compared to the Figure 15.9, we switched between time (46000 points) and frequencies (15 scales). On the right, after the calculus of the slope coefficient, only two dimensions are remaining: one for the participants, one for the electrodes.

To construct a model able to predict the alertness state, some usual classification tools (classification and regression trees or k nearest neighbors for example) will be applied on this matrix in 2 dimensions.

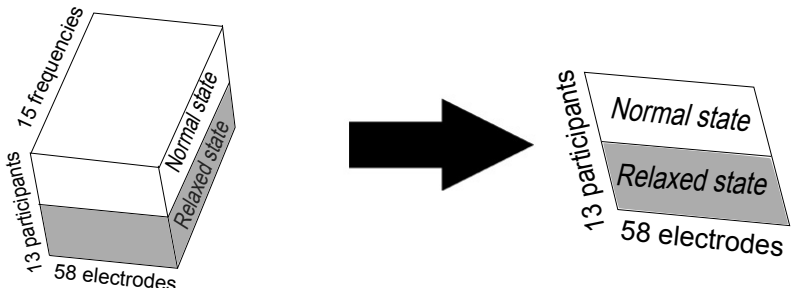


Figure 15.15 – Representation of the data matrix after a dimension reduction. On the left, the data obtained after a discrete wavelet transform. There are still three dimensions: one for the participants, one for the 15 frequencies and one for the electrodes. On the right, after the calculus of the slope coefficient, only two dimensions are remaining: one for the participants, one for the electrodes.

15.5.2 Hölder exponent criterion and Alpha criterion

Previously, other approaches to obtain a summarized data matrix in two dimensions have been tested on similar signals VÉZARD [2010]. The goal was to obtain an approach which allows separating the two alertness states and reducing the inter individual variability observed. One of these approaches was based on the use of the Hölder regularity of the signal. The Hölder exponent JAFFARD et MEYER [1996], LEVY VEHEL et SEURET [2004] is a tool to measure the regularity of a signal at a given point. The smaller the Hölder exponent (respectively large) is, the more irregular (respectively smooth) is the signal. The Hölder exponent was estimated as defined in LEGRAND [2004]. The aim was to summarize the signal recorded by an electrode in its global regularity. An average of Hölder exponents for each point of the signal provided by an electrode was calculated.

Another approach was to analyze the alpha wave content in signals. Alpha rhythm is the classical EEG correlate for a state of relaxed wakefulness. When the person is relaxed, the neurons are synchronized and operate at a particular and identical rhythm. This rhythm appears to be responsible for the more pronounced appearance of Alpha waves NIEDERMEYER et LOPES DA SILVA [2005]. When the person is forced to perform a task that can break the relaxed state, the functioning of neurons vary widely. They seem to act by groups which do not work at a similar rhythm. Alpha waves are then masked by the more pronounced appearance of other waves (like Beta waves). Thus, the idea was to measure the proportion of alpha waves in the signal (alpha waves divided by the sum of all waves: alpha, beta, teta and delta).

These two approaches gave a data matrix in two dimensions like that obtained with the slope criterion. However, they did not seem to work as well as the matrix of slopes to discriminate the two states of vigilance VÉZARD [2010]. Therefore, the slope criterion is investigated in this chapter.

15.5.3 Preliminary results

The relevance of the slope criterion is illustrated in Figures 15.16 and 15.17. Figure 15.16 provides for each participant, in his state of “normal" alertness and his state of “relaxed" alertness, the sum of the slope criterions on all electrodes. It appears that for a given individual, the slope criterion is almost always lower when the individual is in the normal state than when he is in the relaxed state. Thus, by comparing, for a given individual, the values of the slope criterion for the normal and relaxed states it is possible to effectively distinguish the two states. However, for a new individual, a single record is known and the problem remains unsolved. Figure 15.17 shows for each electrode the sum of the slopes of the participants in a “normal" alertness state and participants in a “relaxed" state. The previous observation is also true at the electrode level. In fact, for a given electrode, the

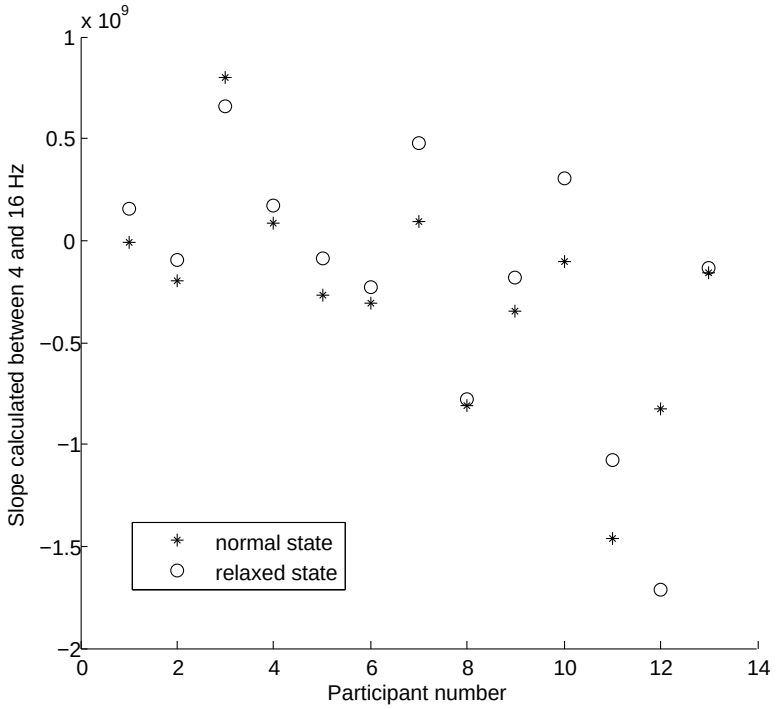


Figure 15.16 – Slope criterion summed over all electrodes for each of 13 participants.

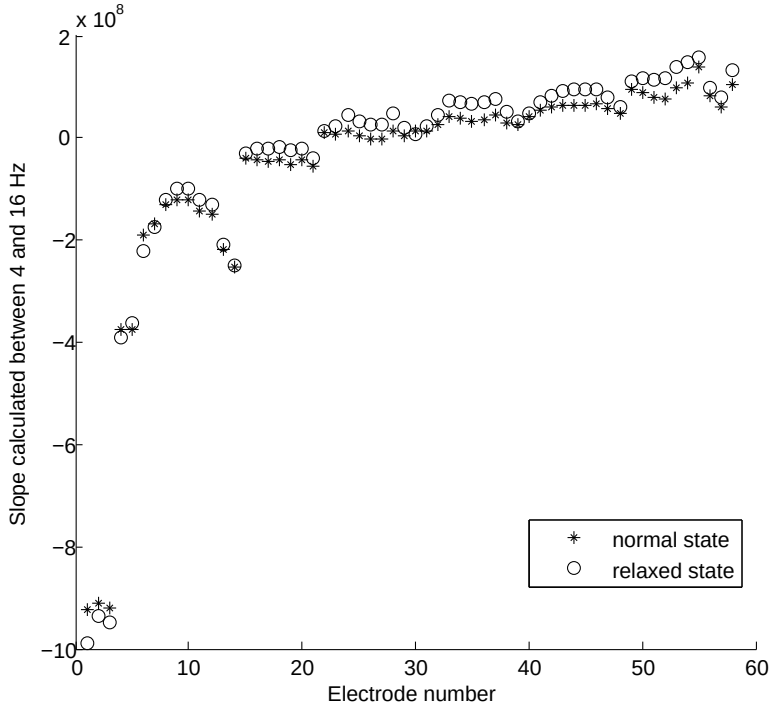


Figure 15.17 – Slope criterion summed over all participants for each of 58 electrodes.

slope criterion is higher when considering the record obtained by this electrode after the relaxation. Thus, the slope criterion can effectively discriminate the two states of alertness for an individual. However, a strong inter-individual variability can be observed in Figure 15.16. Because of this strong individual variability, we cannot plot a line on Figure 15.16 which separates the two alertness states (represented by cross and circles). Then, for a given subject with two EEG records, the slope criterion allows determining which record corresponds to the record done in the relaxed state. However, when only one record is known (new subject), we cannot classify it effectively.

Common classification methods were initially used on the slope matrix to predict the alertness state of the participants. Predictive performance of k nearest neighbors (presented in HASTIE et collab. [2009]), binary decision tree BREIMAN et collab. [1984] (CART), random forests BREIMAN [2001], discriminant PLS (by direct extension of the regression PLS method described in TENENHAUS [1998] recoding the variable to explain using dummy variables) and discriminant sparse PLS LÉ CAO et collab. [2008] were studied. R packages "class", "rpart", "randomForest", "pls" and "SPLS" were respectively used to test these methods. Random forests have been applied by setting the number of trees at 15000 and leaving the other settings by default. Other methods were tuned by applying a 10 folds cross-validation on the training sample (number of neighbors for k nearest neighbors, complexity of the tree for CART, number of components for the discriminant PLS, number of components and value of the thresholding parameter for discriminant sparse PLS). The PLS method has been adapted for classification by recoding the variable to predict (alertness) using a matrix formed by an indicator of the modality ("normal" or "relaxed"). To compare the results, these methods were evaluated on the same samples (learning and test). A 5 fold cross-validation was used to calculate a classification rate. This operation was repeated 100 times to study the stability of classification methods with respect to the data partitioning.

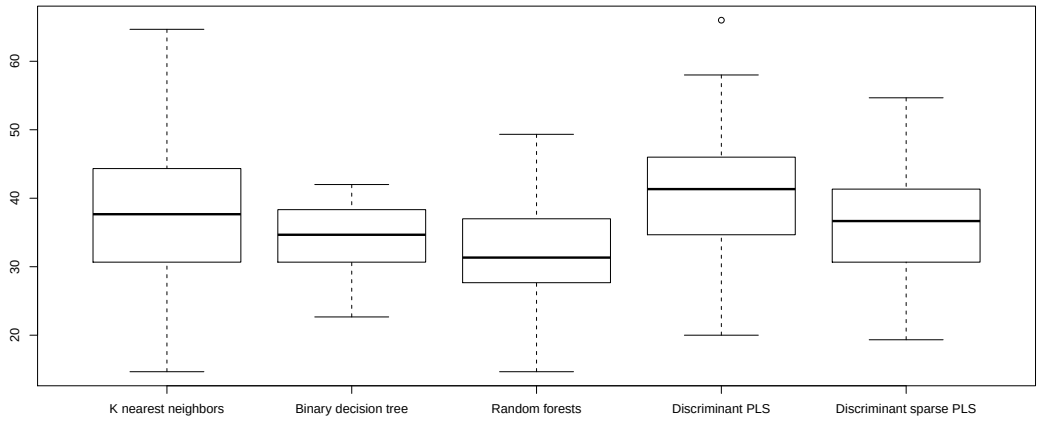


Figure 15.18 – Correct classification rate for the classification methods on the slope criterion.

The results are given by the boxplots in Figure 15.18.

	K nearest neighbors	Binary decision tree	Random forests	Discriminant PLS	Sparse discriminant PLS
Mean	37.28	33.98	32.03	40.63	36.25
Standard dev.	10.47	5.15	6.46	8.55	7.96

Table 15.1 – Means and standard deviations of correct classification rate for the classification methods on the slope criterion.

It appears that the median correct classification rate is very disappointing. It does not exceed 40% for most methods. Table 15.1 summarizes the means and standard deviations obtained using classification methods on the slope criterion. Large standard deviations reflect the influence of the data partitioning on the results. In the case of a binary prediction, these results cannot be satisfactory. It is likely that the inter-individual variability observed in Figure 15.16 has affected the performance of the classification methods. This inter-individual variability is very difficult to include in the classification methods with the available data for this study. Therefore, the pre-processing has been refined to obtain improved classification rates. Specifically, a genetic algorithm has been used as a feature selection process,

to determine the electrode and the frequencies that provide the best discrimination for the slope criterion.

15.6 Feature Selection with a genetic algorithm

In this section, a genetic algorithm is used to improve the slope criterion. So far, previous work in the field, which suggested to focus on the alpha waves, was used. For this reason, the regression was done using frequencies between 4 and 16 Hz. Given the results, this approach will be refined. The algorithm searches for the best range of frequencies (not necessarily adjacent) to perform the regression. Similarly, so far all electrodes were kept. However, one objective of this work is to remove some electrodes to reduce the time required for the installation of the cap. Thus, the best combination electrode/frequencies based on the quality of the prediction is searched for. In this work, 58 electrodes and 15 decomposition levels are available. Then, $58 * 2^{15} = 1900544$ ways exist to choose an electrode and a frequency range. To avoid an exhaustive search, the proposed approach is to use a genetic algorithm to perform a feature selection (BROADHURSTA et collab. [1997] and CAVILL et collab. [2009]).

15.6.1 General principle of a genetic algorithm

These optimization algorithms DE JONG [1975], HOLLAND [1975] are based on a simplified abstraction of Darwinian evolution theory. The general idea is that a population of potential solutions will improve its characteristics over time, through a series of basic genetic operations called selection, mutation and genetic recombination or crossing. From an algorithmic point of view, the general principle is depicted in Figure 15.19.

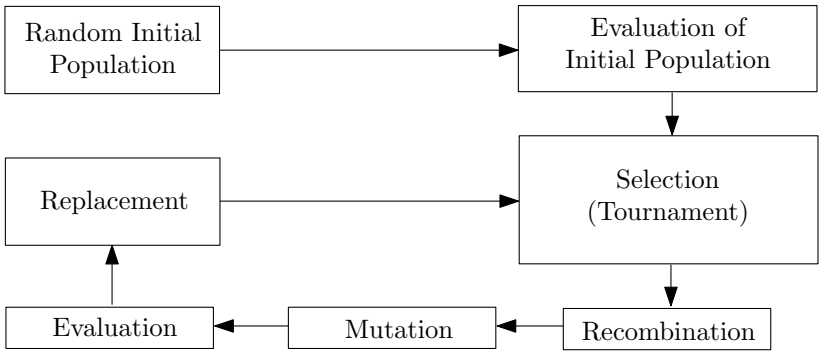


Figure 15.19 – Evolutionary loop of a basic Genetic Algorithm.

The purpose of these algorithms is to optimize a function (fitness) within a given search space of candidate solutions. Solutions (called individuals) correspond to points within the search space, a random set of which are generated, this seeds the algorithm with an initial Population (set of individuals). They are represented by the genomes (binary codes or reals, with a fixed or variable size). All individuals are evaluated using a problem specific objective function called fitness. Individuals are selected based on their fitness (using a series of tournaments), these selected individuals are called Parents. These parents are used to generate new individuals using two basic genetic (search) operations, recombination (random recombination of two or more individuals) and mutation (random modification of a single individual). These newly generated individuals are called Offspring, since they share (genetic) similarities with the Parents used to generate them. Finally, the best individuals (amongst Parents and Offspring) are selected and replace the initial population. The algorithm is iterated until a stop criterion is reached; for instance, when all individuals are identical (convergence of the algorithm) or after a pre-specified number of iterations.

15.6.2 Algorithmic choices

In this work, the genome is composed of 16 variables: the first, an integer ranging from 1 to 58, characterizes the number of the electrode selected, the 15 others are binary and correspond to the inclusion (or not) of each frequency to compute the slope criterion. An example of a genome is given in Figure 15.20. Each genome defines the electrode and the frequencies on which to perform the regression as illustrated in Figure 15.21. In the example illustrated in this figure, a discrete wavelet decomposition is performed on the EEG signals collected by the 15 electrode for each of the subjects (electrode FC5, the correspondence between the electrode number and its positioning being provided by Figure 15.1). The energies corresponding to the 15 levels of decomposition are calculated. For a subject in a given state, a regression on the energies corresponding to the frequencies associated with 1 in the genome is performed and the reference coefficient is maintained (slope criterion). In the figure, the frequencies (1/8, 1/4 and 1 Hz) are associated with a 1 in the genome. It is therefore on the energies calculated for these frequencies that the regression is performed. This procedure is performed on the n subjects in both states of vigilance and a $2n$ length vector is obtained. A classification method will be used to obtain a Correct Classification Rate for each of the genomes (see Section 15.6.2.2).

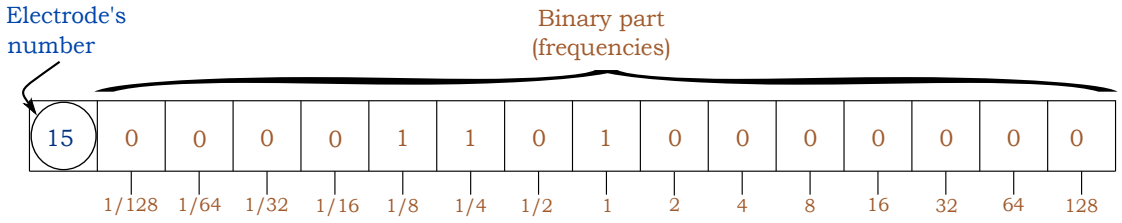


Figure 15.20 – Example of a genome in the genetic algorithm.

15.6.2.1 Genetic Operators

The main operators are mutation and crossover (recombination). Crossover is the genetic operator that allows the characteristics of both parents to be shared in order to create a child that looks like them. In this way, if the parents have good attributes, the child will benefit. The purpose of the mutation is to introduce differences between the child and the parents who generated it. This will give the child new attributes. This operator is particularly important when minimizing or maximizing functions. Indeed, these functions can have local optima that should be avoided if the overall optimum of the function is to be sought. The mutation will make it possible to prevent the solutions tested (children) from stagnating around these local optima by changing their characteristics somewhat and thus allowing them to exceed these optima. The mutation therefore allows a better exploration of the research space. It is also an operator that promotes local research. Indeed, if all the individuals in a population are very similar, mutation is the only operator that will allow differences to be introduced in the created children, thus favouring the path of the research space.

15.6.2.1.1 Crossover To create a child, two parents are randomly selected. A tournament is performed to keep only the best individual (the one with the highest rating based on fitness). The selection pressure is not high (the best of 2) to maintain a high diversity in the population. The selection and the tournament is repeated twice in order to select two parents (tournament “winners"). Both parents are crossed and create a child. The child inherits the electrode which is located halfway between the electrodes of both parents. The frequency crossover is done using a logical operator given in Table 15.2. For a given position in the genome, when both parents have the same binary value, the child inherits it (lines 1

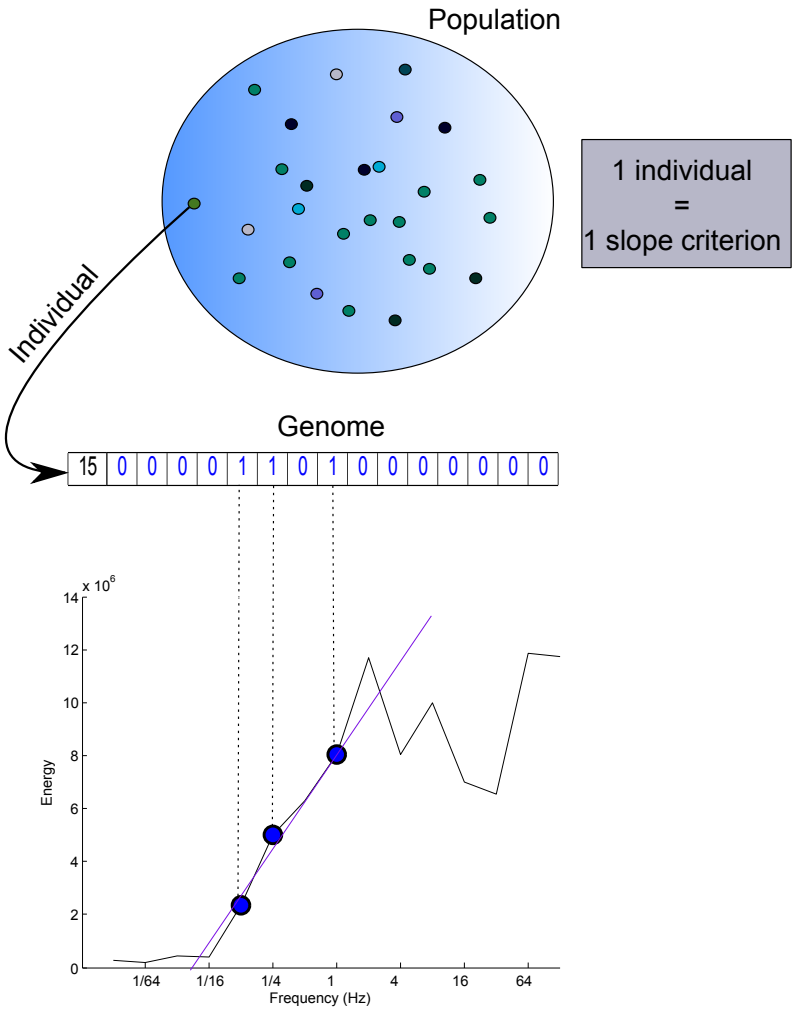


Figure 15.21 – Relationship between the genome and the slope criterion.

and 4 of the Table 15.2). When both parents have different values (lines 2 and 3 of the Table 15.2), a Bernoulli's law of probability $\frac{1}{2}$ is used with it to determine the child's component. Thus, the latter inherits a 0 (respectively a 1) with a probability of $\frac{1}{2}$.

Parent 1	Parent 2	Child
0	0	0
1	0	Bern($\frac{1}{2}$)
0	1	Bern($\frac{1}{2}$)
1	1	1

Table 15.2 – Logical operator used for the frequencies during the crossover.

15.6.2.1.2 Mutation Once the child is established, a mutation is applied. Each component of the genome of the child mutate with probability $1/8$. Thus, each child is, on average, affected by two mutations. When a mutation reaches the electrode number, a random number (drawn between 1 and 58) replaces the child electrode number. For the binary part, a mutation is the change of the binary variable (the 0 becomes 1 and vice versa).

15.6.2.2 Evaluation functions

The genetic algorithm searches for the best combination of electrode / frequency range which achieves the highest prediction accuracy. Thus, it seems natural to rely on the correct classification rate (CCR). Then, the fitness function corresponds to the CCR obtained for each genome. These are then ranked in descending order of CCR. To compare each genome, the same samples are used to calculate the CCR using a 5 fold cross-validation. The evaluation step is done for each child at each iteration. Thus, it is necessary to use a fast classification method as evaluation function. In this work, two methods have been tested (see algorithms 15.6.2.2.1 and 15.6.2.2.2). The first is the single variable classification (SVC) GUYON et ELISSEEFF [2003], a method to predict from a single variable. The average for each modality (normal or relaxed) is calculated on the individuals in the training set for the variable (feature). Individuals of the test sample are then assigned to the class corresponding to the nearest average. The prediction is compared to ground truth which gives a CCR. The second method is the binary decision tree (CART) BREIMAN et collab. [1984]. Here, the algorithm is used with a single variable which guarantees fast calculation. Then, the fitness function for each genome x is written as:

$$f(x) = \frac{\#\text{ well classified participants of the test set}}{\#\text{ participants in the test set}}.$$

The genetic algorithm searches for the genome which maximizes f.

15.6.2.2.1 Single Variable Classifier (SVC)

This is a method of producing a prediction from a single variable. A graphical representation of the method is provided in Figure 15.22. In Figure 15.22, the individuals in the learning sample in the normal and relaxed state are represented by blue and red circles respectively. For a given genome, a slope criterion is obtained. The averages of this criterion are calculated on individuals in the learning sample in the normal and relaxed state (blue and red triangle in the figure). The individuals in the test sample are then assigned to the class corresponding to the nearest average. In Figure 15.22, the normal state is predicted for the individual in the represented test sample (grey circle). The predictions obtained are compared with reality in order to obtain a CCR.

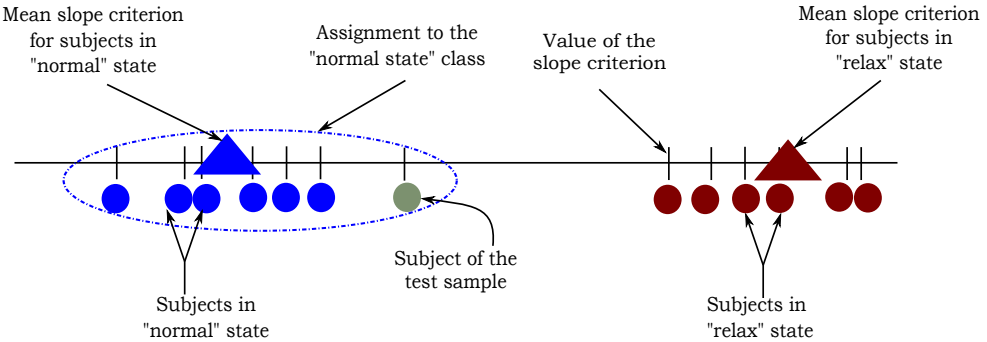


Figure 15.22 – Graphical representation of the Single Variable Classifier method. The subjects in the normal and relaxed learning sample are represented by blue and red circles respectively. The averages of the slope criterion are represented by a blue (normal state) or red (relaxed state) triangle. An individual in the test sample (grey circle) is assigned to the class corresponding to the nearest average

15.6.2.2.2 CART

In this work, the CART algorithm is used with only one variable (slope criterion) which guarantees speed of execution. The construction of the decision binary tree is based on the formation of a series of decision nodes. Let's note n^t the number of subjects that reach the node t. The n subjects are presented at the root of the tree ($n^1 = n$). Subjects are asked a binary question and divided into two subgroups (or two branches) noted n_d^t and n_g^t (where n_d^t represents the right branch and n_g^i the left and $n^t = n_d^t + n_g^t$) based on their answer. In our case, the formation of a question corresponds to the definition of a threshold on the slope criterion. The question is then "is the value of the slope criterion on the subject under consideration below or above the threshold?". The threshold is chosen so that, within each of the two subgroups, the subjects are as homogeneous as possible with respect to the variable to be explained (state of vigilance).

There are several measures of heterogeneity, the most commonly used, in the case where the variable to be predicted is qualitative, are the Gini diversity index and the Shannon entropy. The Gini diversity index at node t, noted $I_{Gini}(t)$, is defined by the quantity :

$$I_{Gini}(t) = \sum \sum_{j=1}^{J} \frac{n^t(j)}{n^t} \left(1 - \frac{n^t(j)}{n^t}\right),$$

where J represents the number of modalities of the variable to be predicted (2 in our case) and $n^t(j)$ represents the number of subjects who reach the node by being in the j class. Shannon's entropy at node t, noted $I_{Entropy}(t)$, is defined by the quantity :

$$I_{Entropy}(t) = -\sum_{j=1}^{J} \frac{n^t(j)}{n^t} \log\left(\frac{n^t(j)}{n^t}\right).$$

These measures of heterogeneity of a node t are minimal if the node is pure (totally homogeneous). In our study, entropy was used. The threshold is therefore chosen in order to make the generated nodes as pure as possible.

The process of creating a node is iterated until the nodes obtained are terminal. A node becomes a terminal when any new separation of the sample applied to it does not improve the homogeneity already achieved. A terminal node is then called "sheet". Each leaf of the tree is associated with a value (in the case where Y is quantitative) or a modality of Y (in the case where Y is qualitative). Thus, in the case of a quantitative Y, the value associated with the leaf is the average of the individuals who have reached it. In the case of a variable to be explained qualitatively, the modality associated with the leaf is the one that is mainly represented among the individuals present on the leaf.

Once the complete tree is obtained, pruning is done to reduce the complexity of the tree and avoid overlearning (tree too close to the data and very unstable to predict data that was not used in its construction). The predictive performance of all sub-trees (tree at 1 node, 2 nodes, ..., d nodes where d represents the number of nodes in the complete tree) are evaluated by cross-validation of type 5 *folds* on the learning sample. The prediction of the class of a subject in the test sample is obtained by running him through the tree and assigning him to the modality associated with the leaf he reaches. The subtree that provides the best TBC is preserved and represents the pruned tree.

The terminal nodes of this one are assigned to one of the Y terms ("normal" or "relaxed"). This is a majority vote among individuals n^T at the terminal node T in the qualitative case and an average of the Y response of individuals n^T in the quantitative case.

A representation of a pruned binary decision tree is provided in Figure 15.23. In this figure, subjects in normal and relaxed state are represented by blue and red circles respectively. The figure can be read from top to bottom. The top of the figure represents the root of the tree. The question asked there is $P > \mu_1$?, where P represents the size vector $2n$ containing the values of the slope criterion for the subject n in both states of vigilance. Depending on

their answer, the subjects are distributed in the branches of the tree and thus descend the tree to the terminal nodes according to the answers to the questions encountered. In Figure 15.23, it is assumed that the tree is already pruned. Thus each leaf of the tree is associated with a modality (normal or relaxed) represented by a triangle (blue or red). The prediction of the class of a subject in the test sample (grey circle) is obtained by making him walk through the tree and assigning him to the modality associated with the leaf reached by it. On the figure, the individual is assigned to the class "relaxed".

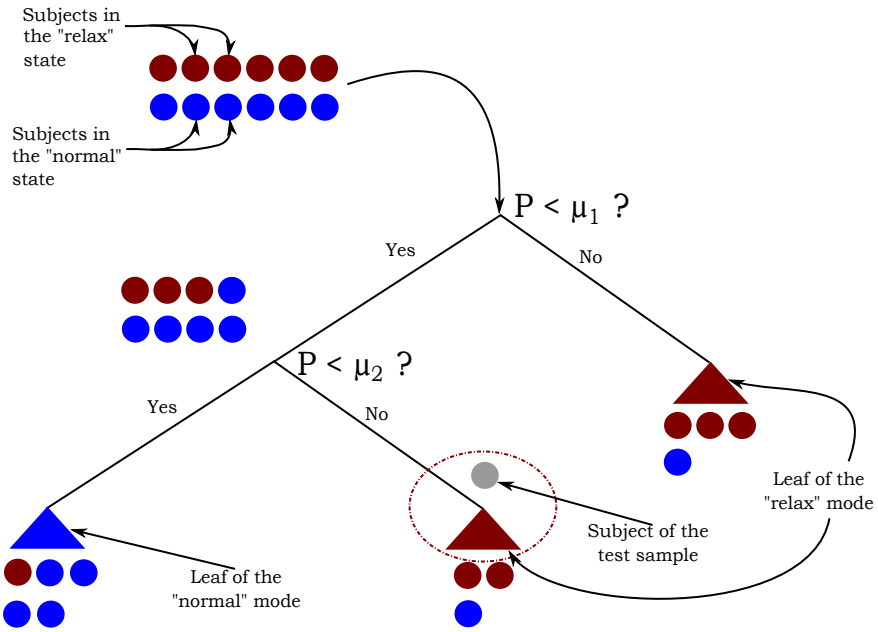


Figure 15.23 – Representation of a pruned binary decision tree. Subjects in normal and relaxed state are represented by blue and red circles respectively. Each leaf of the tree is associated with a modality (normal or relaxed) represented by a triangle (blue or red). The prediction of the class of a subject in the test sample (grey circle) is obtained by making him walk through the tree and assigning him to the modality associated with the leaf reached by it (class "relaxed" in this case).

15.6.2.3 Stop criterion

The algorithm stops if one of the following three conditions is satisfied:

- The number of iterations exceeds 1000.
- Parents are the same for 10 generations.
- The number of differences among the parents is less than 3.

To calculate the number of differences for a given population, denoted D, the genomes of the population at iteration i are stored in a matrix, denoted by P^i. Let P^i_j be the column j of the matrix P^i (where $j = 1, ..., 16$). Then $D = D_b + D_{elec}$ where:

- D_b is the number of differences for the binary part of P^i_j (columns 2 to 16). The number of differences for column P^i_j (where $j = 2, ...16$) is min(number of 0 in P^i_j, number of 1 in P^i_j).
- D_{elec} is the number of differences in P^i_1 (column corresponding to the electrode component). Then, D_{elec} is the number of individuals who have a electrode which is different from the electrode most selected in the population.

15.6.3 Results

The algorithm, programmed using Matlab, is run 100 times for each evaluation method with 300 parents and 150 children. The training and test sets are different for two different runs. Figure 15.24 gives CCR values for each run of the genetic algorithm with CART (stars) and SVC (circles).

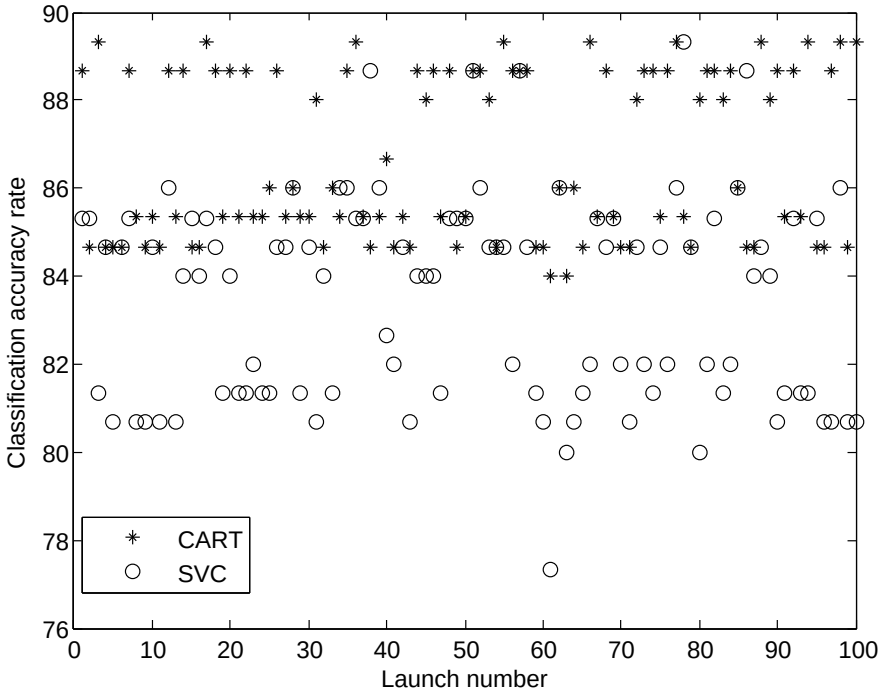


Figure 15.24 – Correct classification rates calculated with CART (stars) and SVC (circles) for each run of the genetic algorithm with 300 parents and 150 children.

For each run, the algorithm is launched two times (one time with CART and the other time with SVC). During a run, CART and SVC use the same training and test sets in order to obtain comparable results. The correct classification rate obtained by CART (mean of 86.68% and standard deviation of 1.87%) exceed significantly (Mann-Whitney paired test with a p-value $= 5.57 * 10^{-14}$) those obtained by SVC (mean of 83.49% and standard deviation of 2.37%), as mentioned in Table 15.3. At the end of the algorithm, some of the best genomes have the same evaluation (due to the low number of individuals and the evaluation method). It is therefore necessary to choose a genome (BEST) among those who have the same score. Thus, the best genomes at the end of each run of the algorithm are stored. The genome that appears most often is considered as the BEST for the evaluation method considered. The two BEST (for CART and SVC) get a correct classification rate equal to 89.33%. For CART, the BEST is obtained by performing regression between $1/8$, $1/4$, 2, 4 and 64 Hz on electrode F4 (right frontal area on Figure 15.1). For SVC, the BEST is obtained from electrode F2 (right frontal area) and the regression between $1/32$, $1/16$, 2, 4, 8, 64 and 128 Hz (see Table 15.4). Frequencies chosen for these genomes are more extensive than those used in the preliminary study.

Figure 15.25 gives the occurrence of the electrodes in the best genome over the 100 runs. When some genomes have the same CCR at the end of the run, we select the electrode chosen most often among the genomes with equal CCR. The algorithm running with CART selects the electrodes around the number 10 (FZ in Figure 15.1), 17 (FC1) or 30 (T4). With the SVC method, the electrodes around the 2 (FPZ), the 11 (F2) or the 48 (T6) are mostly chosen. Finally, on average, the population of the evolutionary algorithm converges in less than 50 iterations for both methods. Figure 15.26 gives the number of differences among parents for

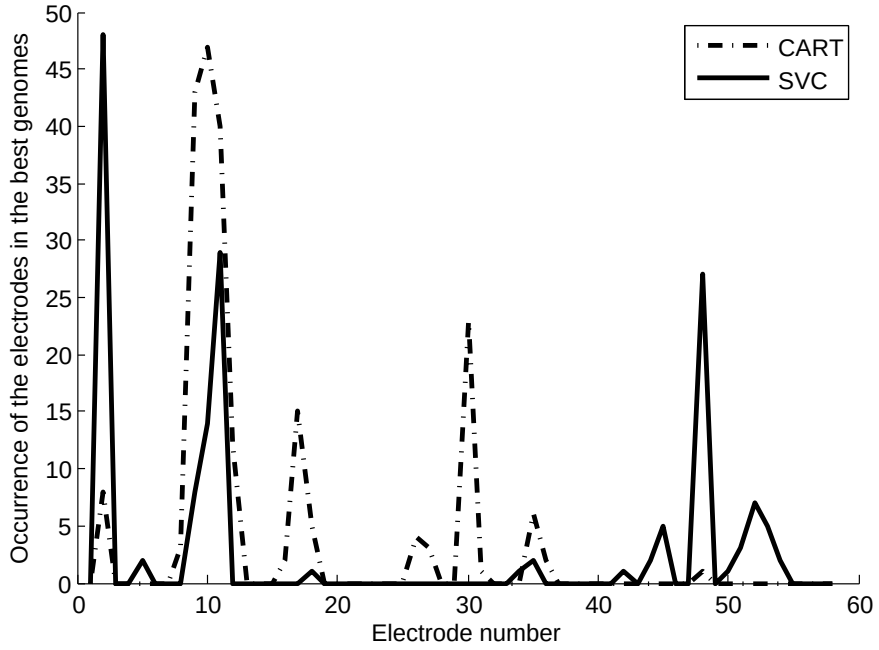


Figure 15.25 – Occurrence of the electrodes in the best genomes for each electrodes during the 100 runs of the genetic algorithm with 300 parents, 150 children and CART (dash-dotted curve) or SVC (solid curve).

one run of the algorithm. It shows that the number of differences among parents decreases very rapidly and falls below the threshold of 3 differences in less than 40 iterations. Then, one of the three stop conditions is satisfied and the algorithm stops.

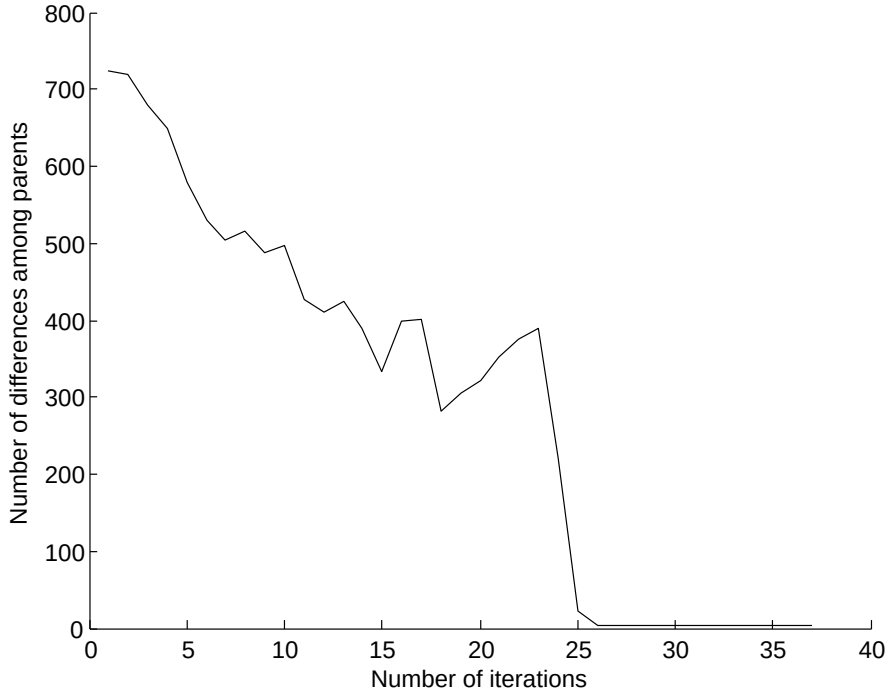


Figure 15.26 – Number of differences among parents for a run of the genetic algorithm with 300 parents, 150 children and SVC.

Evaluation methods	CCR	
	mean	standard deviation
CART	86.68	1.87
SVC	83.49	2.37

Table 15.3 – CCR for the two evaluation methods.

Evaluation methods	BEST genome		
	electrode selected	frequency selected (Hz)	CCR
CART	F4	$1/8$, $1/4$, 2, 4 et 64	$89,33\%$
SVC	F2	$1/32$, $1/16$, 2, 4, 8, 64 et 128	$89,33\%$

Table 15.4 – Summary table of results for best genomes.

Tables 15.3 and 15.4 summarize the CCR obtained by the genetic algorithm, which are better than those obtained (see Figure 15.18) with the criterion of the slopes calculated for frequencies between 4 and 16 Hz (alpha waves). Moreover, Table 15.5 shows that the genetic algorithm allows for a dimension reduction. SVC classifier can not be used with more than one variable. Then, Table 15.5 only shows a comparison between the results obtained in section 15.5.3 and those obtained with the genetic algorithm for the CART classifier.

Evaluation methods	Number of electrodes in the predictive model	CCR	
		mean	standard deviation
CART	58	33.98	5.15
CART	1	86.68	1.87

Table 15.5 – Comparison between CCR obtained in the preliminary study (1st row) and CCR obtained with the genetic algorithm (2nd row).

It also appears that it is more appropriate to use a regression on frequencies of $1/8$, $1/4$, 2, 4 and 64 Hz for the signal of electrode $F4$ and the CART classifier. Then, this work allows to accurately predict the state of alertness of a new individual. In fact, this electrode and this range of frequencies will be used to calculate the slope criterion for this individual. The CART decision tree, built on the sample formed by the 26 signals (13 study participants in both states of alertness) will be used as a classifier to predict his state of alertness.

15.7 Conclusions

This chapter presents a system for the automatic detection of human mental states of alertness using EEG data and wavelet decomposition. This contribution is also coupled with a complete protocol of data acquisition, a data validation procedure and a feature selection strategy. Initially, we proposed a criterion to obtain a summarized data matrix in two dimensions. Given the disappointing results obtained by classifying all of the available data, a genetic algorithm was used as a feature selection step to refine it. This allowed obtaining a reliable classification model that achieves average of classification accuracy equal to 86.68% with a standard deviation of 1.87%. The algorithm also selects only a single electrode from the 58 that were initially available; this greatly enhances the possibility of applying the proposed system in real-world scenarios.

An exchange with neurobiologists now seems necessary to link the results obtained by the genetic algorithm to human physiology. A new campaign to collect EEG data and increase the number of participants included in the study has been undertaken in 2012. Increasing the number of data should allow us to improve the precision of the estimate of CCR

and thus reduce the number of solutions that have the same score at the end of the genetic algorithm execution. In addition, an increase of the number of participants allows us to provide an external validation set for the CCR at the end of the genetic algorithm execution.

Moreover, it is possible to improve the genetic algorithm proposed in this chapter. In fact, improving genetic operators and testing other evaluation criteria are all paths that remain to be explored. A final interesting point concerns the transformation of the prediction obtained ("normal" state of alertness or "relaxed") to a probability using linear discriminant analysis or logistic regression as evaluation functions.

The authors wish to thank Vérane Faure, Julien clauzel and Mathieu Carpentier, who collaborated as interns in the research team during the development of parts of this work.

References

ANDERSON, C. et Z. SIJERCIC. 1996, «Classification of EEG signals from four subjects during five mental tasks», *Proceedings of the Conference on Engineering Applications in Neural Networks, London, United Kingdom*, p. 407–414. 329

BEN KHALIFA, K., M. BÉDOUI, M. DOGUI et F. ALEXANDRE. 2005, «Alertness states classification by SOM and LVQ neural networks», *International Journal of Information Technology*, vol. 1, p. 131–134. 329

BREIMAN, L. 2001, «Random forests», *Machine Learning*, vol. 45, p. 5–32. 343

BREIMAN, L., J. FRIEDMAN, R. OLSHEN et C. STONE. 1984, «Classification and regression trees», *Wadsworth Advanced Books and Software*. 343, 347

BROADHURSTA, D., R. GOODACREA, A. AH JONESA, J. J. ROWLANDB et D. B. KELP. 1997, «Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry», *Analytica Chimica Acta*, vol. 348, p. 71–86. 344

CAVILL, R., H. C. KEUN, E. HOLMES, J. C. LINDON, J. K. NICHOLSON et T. M. EBBELS. 2009, «Genetic algorithms for simultaneous variable and sample selection in metabonomics.», *Bioinformatics*, vol. 25, p. 112–118. 344

CECOTTI, H. et A. GRAESER. 2008, «Convolutional neural network with embedded fourier transform for EEG classification», *International Conference on Pattern Recognition, Tampa, Florida*, p. 1–4. 329

DAUBECHIES, I. 1992, *Ten Lectures on Wavelets*, SIAM. 335

DE JONG, A., K. 1975, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, thèse de doctorat, University of Michigan. 344

GUYON, I. et A. ELISSEEFF. 2003, «An introduction to variable and feature selection», *Journal of Machine Learning Research*, vol. 3, p. 1157–1182. 347

HASTIE, T., R. TIBSHIRANI et J. FRIEDMAN. 2009, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction.*, 2e éd., Springer. 343

HAZARIKA, N., J. CHEN, C. TSOI et A. SERGEJEW. 1997, «Classification of EEG signals using the wavelet transform», *Signal Processing*, vol. 59, p. 61–72. 328

HOLLAND, H., J. 1975, *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press. 344

JACOBSON, E. 1974, *Biologie des ï¿½motions. Les bases thï¿½oriques de la relaxation.* 331

JAFFARD, S. et Y. MEYER. 1996, «Wavelet methods for pointwise regularity and local oscillations of functions.», *Mem. Amer. Math. Soc.*, vol. 123, nº 587. 341

JASPER, H. H. 1958, «Report of the committee on methods of clinical examination in electroencephalography», *Electroencephalography and clinical neurophysiology*, vol. 10, p. 1 – 370. 328

LÉ CAO, K.-A., D. ROSSOUW, C. ROBERT-GRANIÉ et P. BESSE. 2008, «Sparse PLS: Variable selection when integrating omics data», *Statistical Applications in Genetics and Molecular Biology*, vol. 7, nº Article 35. 343

LEGRAND, P. 2004, *Débruitage et interpolation par analyse de la régularité Höldérienne. Application à la modélisation du frottement pneumatique-chaussée*, thèse de doctorat, École Centrale de Nantes et Université de Nantes. 341

LEVY VEHEL, J. et P. LEGRAND. 2004, «Signal and image processing with fraclab», *Complexity and Fractals in Nature, 8th International Multidisciplinary Conference*. 336

LEVY VEHEL, J. et S. SEURET. 2004, «The 2-microlocal formalism», *Fractal geometry and Applications : A jubilee of Benoit Mandelbrot, Proc. Sympos. Pure Math.*, vol. 72-2, p. 153ï¿½215. 341

MALLAT, S. 2008, *A Wavelet Tour of Signal Processing.*, 3e éd., Academic Press. 335

NAITOH, P., L. C. JOHNSON et A. LUBIN. 1971, «Modification of surface negative slow potential (CNV) in the human brain after total sleep loss.», *Electroencephalography and Clinical Neurophysiology*, vol. 30, p. 17–22. 332

NIEDERMEYER, E. et F. LOPES DA SILVA. 2005, *Electroencephalography, basic principles, clinical applications and related fields*, 5e éd.. 341

OBERMAIER, B., C. GUGER, C. NEUPER et G. PFURTSCHELLER. 2001, «Hidden markov models for online classification of single trial EEG data», *Pattern recognition letters*, vol. 22, p. 1299–1309. 329

ROSENBLITH, W. 1959, «Some quantifiable aspects of the electrical activity of the nervous system (with emphasis upon responses to sensory stimuli)», *Revs. Mod. Physics*, vol. 31, p. 532–545. 332

SCHULTZ, J. H. 1958, *Le Training autogene*, PUF. 331

SUBASI, A., M. AKIN, K. KIYMIK et O. EROGUL. 2005, «Automatic recognition of vigilance state by using a wavelet-based artificial neural network», *Neural Comput and Applic*, vol. 14, p. 45–55. 328, 329

TECCE, J. J. 1979, «A CNV rebound effect.», *Electroencephalography and Clinical Neurophysiology*, vol. 46, p. 546–551. 332

TENENHAUS, M. 1998, *La régression PLS, Théorie et Pratique*. 343

TIMSIT-BERTHIER, M., A. GERONO et H. MANTANUS. 1981, «Inversion de polarité de la variation contingente négative au cours d'état d'endormissement.», *EEG Neurophysiol*, vol. 11, p. 82–88. 332

VÉZARD, L. 2010, *Réduction de dimension en apprentissage supervisé. Applications à l'étude de l'activité cérébrale.*, mémoire de maîtrise, INSA de Toulouse. 341

VÉZARD, L., P. LEGRAND, M. CHAVENT, F. FAÏTA AÏNSEBA, J. CLAUZEL et L. TRUJILLO. 2013, «Classification of EEG signals by evolutionary algorithm», *Advances in Knowledge Discovery and Management*, vol. 4. 330

VUCKOVIC, A., V. RADIVOJEVIC, A. CHEN et D. POPOVIC. 2002, «Automatic recognition of alertness and drowsiness from EEG by an artificial neural network», *Medical Engineering and Physics*, vol. 24, p. 349–360. 329

WALTER, W. G., R. COOPER, V. ALDRIDGE, W. C. MCCALLUM et A. WINTER. 1964, «Contingent negative variation: An electric sign of sensorimotor association and expectancy in the human brain.», *Nature*, vol. 203, p. 380–384. 332

YEO, M., X. LI, K. SHEN et E. WILDER-SMITH. 2009, «Can SVM be used for automatic EEG detection of drowsiness?», *Safety Science*, vol. 47, p. 115–124. 329

Chapter 16

Regularity and Matching Pursuit Feature Extraction for the Detection of Epileptic Seizures

This work, related to the PhD of Emigdio Z. Flores, has been published in the Journal of Neuroscience Methods, Volume 266, 15 June 2016, Pages 107-125. Work carried out with Emigdio Z. Flores, Leonardo Trujillo, Arturo Sotelo and Luis N. Coria.

Contents

Abstract

The neurological disorder known as epilepsy is characterized by involuntary recurrent seizures that diminish a patient's quality of life. Automatic seizure detection can help improve a patient's interaction with her/his environment, and while many approaches have been proposed the problem is still not trivially solved.
In this work, we present a novel methodology for feature extraction on EEG signals that allows us to perform a highly accurate classification of epileptic states. Specifically, Hölderian regularity and the Matching Pursuit algorithm are used as the main feature extraction techniques, and are combined with basic statistical features to construct the final feature sets. These sets are then delivered to a Random Forests classification algorithm to differentiate between epileptic and non-epileptic readings.
Several versions of the basic problem are tested and statistically validated producing perfect accuracy in most problems and 97.6% accuracy on the most difficult case. A comparison with recent literature, using a well known database, reveals that our proposal achieves state-of-the-art performance.
The experimental results suggest that using a feature extraction methodology composed of regularity analysis, a Matching Pursuit algorithm and time-domain statistic measures together with a classifier produces a system that can predict epileptic states with competitive performance that matches or even surpass other novel methods.

16.1 Introduction

Epilepsy is a type of neurological disorder that it is characterized by an enduring predisposition to generate unprovoked seizures, each occurring more than 24 hours apart [FISHER et collab., 2014]. Normal brain activity is considered to be non-synchronous, but during epileptic seizures a group of neurons begins firing in an abnormal, excessive and synchronized manner. This is opposed to what normally happens, when an excitatory neuron fires it becomes resilient to firing again for a short period of time [ELSE et HAMMER, 2013]. There are approximately 65 million people worldwide living with epilepsy [THURMAN et collab., 2011]. It varies from region to region, for instance in the United States the annual incidence of epilepsy is 48 per 100000 inhabitants, whereas the prevalence approximates 710 per 100000 [HIRTZ et collab., 2007]. Diagnosing epilepsy after a single unprovoked seizure, when there is a high risk for recurrence, may or may not lead to a decision to initiate treatment by the epileptologist. Nonetheless, the diagnosis helps the physician assess the balance between the possible avoidance of a second seizure and the associated risks of actually occurring. According to EADIE [2012] about 70% of the cases can be controlled by medication, but even then the patient might suffer negative side effects.

Automated systems can help develop an appropriate therapy plan that could eventually improve the balance of the associated risks and potentially enhance the patient's quality of life [FISHER et collab., 2014]. Such systems would be able to detect a bodily response that matches the epileptic symptoms, and distinguish them from signals present during a patient's regular activity. There are non-invasive methods that can sense epileptic states before they manifest, as well as to detect the seizure physiologic state, which is also known as the Ictal state. One such method is the Electroencephalogram (EEG), which records electrical brain activity along the scalp. Other methods are also available, such as the Electrocorticogram (ECoG) that is recorded by electrodes that are inserted trough the skull [SOTELO et collab., 2013, 2015]. ECoG provides more localized readings but can be undesirable for the patient since it is invasive [BALL et collab., 2009].

EEGs are very popular because they have several advantages compared to other methods [ACHARYA et collab., 2012b; AHAMMAD et collab., 2014; GÜLER et UBEYLI, 2005; GULER et collab., 2005; GUO et collab., 2010; KAMATH, 2015; LIMA et collab., 2010; ORHAN et col-

lab., 2011; RAJENDRA ACHARYA et collab., 2012; TZALLAS et collab., 2009; ÜBEYLI et GÜLER, 2007], these are: (1) hardware costs are relatively low; (2) electrodes can be positioned based on application needs; (3) it does not expose the patient to high-intensity magnetic fields like magnetoencephalography; and (4) it is non-invasive, relative with other methods. There are some drawbacks as well, like the inherent presence of noise in the signals. Nevertheless, the present work focuses on the use of EEG recordings for automatic epilepsy analysis.

Epileptic seizures can be identified in EEG signals by experienced physicians, but automatic recognition is still not a trivial task. Research for the development of computational systems that perform these tasks usually focuses on three fronts. Either by exploiting the morphology of the signals recorded during the epileptic crisis [YADAV et collab., 2012], by applying analytic or numerical methods [RAMGOPAL et collab., 2014], or a combination of the two [ACHARYA et collab., 2012b; AHAMMAD et collab., 2014; DIVYA, 2015; GÜLER et UBEYLI, 2005; GULER et collab., 2005; GUO et collab., 2010; KAMATH, 2015; KUMAR et collab., 2014; LIMA et collab., 2010; MURUGAVEL et RAMAKRISHNAN, 2014; ORHAN et collab., 2011; RAJENDRA ACHARYA et collab., 2012; TZALLAS et collab., 2009; ÜBEYLI et GÜLER, 2007].

We propose a new methodology for feature extraction, which incorporates two powerful signal analysis tools to construct specialized domain features, namely Hölderian regularity [MALLAT et HWANG, 1992] and the Matching Pursuit (MP) algorithm [MALLAT, 1993]. Each of these tools tackles the posed problem from different perspectives. The former incorporates a local signal regularity measure while the latter employs a time-frequency analysis. These two methods are related, since MP can be seen as a global regularity measure. Both of these techniques are considered to be highly nonlinear in their core design, thus the proposed methodology is well suited to analyze a nonlinear process, like the one that produces an epileptic seizure. Moreover, we combine these nonlinear features with much simpler features computed as statistics of the raw signals in time domain, which have been shown to be useful in related tasks [SOTELO et collab., 2013].

After the proposed feature extraction process, a classifier is used to solve the automatic detection problem. Experimental results achieve a perfect performance for three of the four tested problem instances. Moreover, on the fourth test case, the most complex, classification accuracy is 97.6%, competitive with the state-of-the-art [ACHARYA et collab., 2012b; AHAMMAD et collab., 2014; GULER et UBEYLI, 2007; GÜLER et UBEYLI, 2005; GULER et collab., 2005; GUO et collab., 2010; KAMATH, 2015; LIMA et collab., 2010; ORHAN et collab., 2011; RAJENDRA ACHARYA et collab., 2012; TZALLAS et collab., 2007, 2009; ÜBEYLI et GÜLER, 2007]. All of these conclusions are derived from a rigorous experimental validation and comprehensive statistical tests.

In the following subsections a brief description of epileptic states is presented, as well as a short review of previous works related to techniques employed for EEG analysis.

16.1.1 Epileptic states

Epiletic seizures are unintentional and disruptive events of mental activity that impair a patient's motor, sensorial and autonomic functions. Seizures develop over several states [FRANASZCZUK et collab., 1998]: (1) the Basal state; (2) the Pre-Ictal state; (3) the Ictal state; (4) the Post-Ictal state; and (5) the Inter-Ictal state. These states can be identified by the symptoms exhibited by the patient and the morphology of the EEG signals. The Basal state corresponds to normal brain functions, in this state brain signals are characterized by a low amplitude and a relatively high frequency. The Pre-Ictal state refers to the time period before the seizure symptomatology is evident. Here, the signal amplitude is higher than in the Basal state, with the presence of spikes and transitory activity, also called recruiting rhythms [KOHSAKA et collab., 2002]. The Ictal state is the prominent period where the symptomatology is evident. The EEG signal magnitude is higher than in any other state and displays a dominant low frequency rhythm. The Post-Ictal state refers to the span of time when an

altered state of consciousness exists after the active portion of the seizure ended. This period is variable and depends of the seizure duration. The overall amplitude of the signal decreases and the frequency increases. Several symptoms appear during this state, like migraines, depression and a loss of motor functions [FISHER et SCHACHTER, 2000]. Finally, the Inter-Ictal state refers to the period of time between seizures.

The problem of automatically detecting the states of epileptic seizures can be posed in different ways [SOTELO et collab., 2013, 2015]. Here, we recognize that the identification of Ictal states is an important task, as a way to prevent or prepare for a seizure. Moreover, the effects on the patient are most severe in this state. Consequentially, this work focuses on the automatic recognition of Ictal activity among other EEG readings. For this work, we use the Bonn data set [ANDRZEJAK et collab., 2001], a public database that contains several types of EEG recordings, to test the proposed approach.

16.1.2 Previous work

The automatic classification of epileptic seizures has received much attention over recent years. The problem has been handled from many perspectives, focusing on different aspects of the problem, while using a variety of signal processing and pattern recognition paradigms. Since the number of reported studies is large, here we review the most recent and relevant examples. Moreover, for comparative reasons, we mostly limit to works based on the Bonn data set [ANDRZEJAK et collab., 2001]. ACHARYA et collab. [2013] present an extensive survey that summarizes a variety of different methods on how this problem has been addressed. In that survey, the authors also summarize the classification accuracy from each reviewed work, comparing their quality and experimental work. Even though some works achieve strong performance, there is still room for developing new domain-specific patterns recognition methods. Moreover, the insights gathered in epileptic states detection might be extended to other areas of EEG analysis or application domains [VEZARD et collab., 2014].

Before attempting to solve the classification task, feature extraction must be performed. The simplest features are extracted in the time domain. For instance, XIE et KRISHNAN [2014] proposed an improved Dynamic Principal Component Analysis (DPCA) by means of a non-overlapping moving window. Because they show a highly effective feature extraction, a simple nearest neighbor classifier produces good results. This shows that one of the most difficult tasks is deriving an optimum feature extraction method. KAMATH [2015] uses the Hilbert Transform (HT) as a method for a time to time-domain transformation, where the frequency is expressed as a rate of phase change, a dispersion entropy measure, dispersion complexity and forbidden count. These features are then used to distinguish between the Ictal and Inter-Ictal states.

A more common approach is to use time-frequency analysis for epilepsy detection, such as wavelet analysis [ACHARYA et collab., 2012b; AHAMMAD et collab., 2014; CHEN et collab., 2014; GANDHI et collab., 2012; GUO et collab., 2010; KUMAR et collab., 2014; KUMARI et PRABIN, 2011; MURUGAVEL et RAMAKRISHNAN, 2014; NUNES et collab., 2014; ORHAN et collab., 2011; TZALLAS et collab., 2009; ZAINUDDIN et collab., 2013]. A related technique is used by KOVACS et collab. [2014], the Short Time Fourier Transform (STFT), achieving good results and confirming that well established methods are still relevant in this domain. Packet Wavelet Decomposition is also used in this domain, since it is a variant of the same underlying concept [RAJENDRA ACHARYA et collab., 2012]. FAUST et collab. [2015] present a recent comprehensive review of wavelet-based methods used to solve the problem of automatic seizure detection. It is clear that the robust time-frequency analysis provided by the wavelet decomposition is a popular technique among the reviewed literature, because its properties are useful in many areas of signal processing.

Other nonlinear methods not directly related to time-frequency analysis are also used in this domain, ACHARYA et collab. [2013] suggests that given the nonlinearity of EEG signals

these techniques might yield better results, although simpler methods sometimes can outperform them. For example, some works [ALAM et BHUIYAN, 2013; BAJAJ et PACHORI, 2012; DIVYA, 2015] use the Empirical Mode Decomposition (EMD) to decompose EEG signals into a set of intrinsic mode functions, a method that performs well with highly non-stationary and nonlinear signals, while using a variety of classifiers. GULER et collab. [2005] make use of another nonlinear approach, the Lyapunov exponent approximation as a measure of signal chaosity, with promising results. The use of different type of entropies like Kolmogorov-Sinai Entropy (KSE), Approximate Entropy (ApEn), Sample Entropy (SampEn), Spectral Entropy (SE) among others have been the subject of study in this domain [ACHARYA et collab., 2012a,b; MARTIS et collab., 2013; NICOLAOU et GEORGIOU, 2012]. ACHARYA et collab. [2015] survey recent works that apply entropy methods for epilepsy detection. Other related methods include works that use the High Order Spectra (HOS), employing the third order cumulant as a feature [ACHARYA et collab., 2012a], Recurrence Quantification Analysis (RQA), an analysis based on the topologies derived from recurrence plots in dynamical systems [NIKNAZAR et collab., 2013], and the Higuchi Fractal Dimension for measuring signal complexity through fractal concepts [MARTIS et collab., 2013].

The above cited works recognize that a critical step in identifying a brain-related phenomenon using EEG recordings is the feature extraction phase, the main emphasis of the current work. In the following section we present our proposed feature extraction approach and afterwards we evaluate our classification results. Moreover, we will compare our results with those previously reported in the literature, showing that our proposal achieves state-of-the-art performance.

The reminder of this chapter is organized as follows. In Section 16.2 the employed EEG data set is described, the proposed feature extraction methods are presented and the classification task is posed. In Section 16.3 we discuss the experimental details and its results are presented in Section 16.4. Finally, in Section 16.6 we present our conclusions and outline future work.

16.2 Materials and methods

16.2.1 Epilepsy EEG Data set

The data set used in this work was published by the Bonn University [ANDRZEJAK et collab., 2001]. The data set includes five subsets of signals (denoted as Z, O, N, F and S), each containing 100 single-channel EEG segments with a duration of 23.6s. All segments were recorded using an amplifier system, digitized with a sampling rate of 173.61 Hz and 12-bit A/D resolution, and filtered using a 0.53-40 Hz (12 dB/octave) band pass filter. The normal (Basal) segments (Sets Z and O) were taken from five healthy subjects. The standard surface electrode placement scheme (the international 10-20 system) was used to obtain the EEG from the healthy cases. Volunteers were relaxed in an awake state with eyes opened (Z) and eyes closed (O), respectively. Both states exhibit different characteristics, the EEG readings with the eyes closed show a higher magnitude than when the eyes are opened, as well as the presence of alpha waves which are common in a relaxation state [BALL et collab., 2009]. Both the Inter-Ictal and Ictal segments were obtained from five epileptic patients. The Inter-Ictal segments were recorded during seizure-free intervals from the depth electrodes that were implanted into the hippocampal formations (Set N) and from the epileptogenic zone (Set F). The Ictal segments (Set S) were recorded from all sites exhibiting Ictal activity using depth electrodes and also from strip electrodes that were implanted into the lateral and basal regions of the neocortex. For convenience, we refer to subsets Z, O, N, F and S, as class A, B, C, D and E respectively; see Table 16.1.

Table 16.1 – Summary of the Bonn data set. All classes have 100 epochs per class and 4096 samples per epoch.

Data set label	Class	Description
Z	A	Normal, eyes open
O	B	Normal, eyes closed
N	C	Seizure free, depth electrodes, hippocampal formations
F	D	Seizure free, depth electrodes, epileptogenic formations
S	E	Epiletic activity, depth electrodes, eptic formations

16.2.2 Proposed feature extraction

An appropriate feature extraction is one of the most important tasks in designing classification systems. We propose a feature extraction approach, that to the authors knowledge, has not been used before in the problem domain studied here. In particular, our approach is based on combining the MP algorithm [MALLAT, 1993] and signal regularity analysis [JAFFARD et MEYER, 1996]. The proposed system is depicted in Figure 16.1. Some versions of each method have been used before trying to either analyze or classify EEG recordings, specially using the MP algorithm [DURKA et BLINOWSKA, 1995]. However, this is the first time that they are used simultaneously to undertake the epilepsy detection problem. The motivation for the simultaneous use of both techniques is that together they can complement the level of detail extracted from the signal, from a broader to a local characterization. This span of detail might produce a robust feature extraction outcome, and its effectiveness is actually validated by experimental results.

Durka has been efficiently using MP decomposition for EEG signal analysis for several years [DURKA et BLINOWSKA, 1995; DURKA et collab., 2001, 2005; FRANASZCZUK et collab., 1998]. In particular, he worked on EEG signals derived from subjects exhibiting an epileptic condition with competent results [DURKA, 2004]. More recently, additional research has been focused on extending Durka's work, like improving the residual inference using source deflation [WU et SWINDLEHURST, 2013] or by imposing a restricted dictionary on the MP [PICOT et collab., 2012], among other proposals [BÉNAR et collab., 2009]. It is worth mentioning that the measures derived from the MP decomposition used in our proposal differ from Durka's works, as seen further in Section 16.2.2.2. MP is effective at detecting epileptic spikes due to its ability in finding base functions closely similar to the analyzed signal in time-frequency space [DURKA, 2004]. In our work, we use a simple approach: a canonical MP (as originally proposed by Stéphane Mallat [MALLAT, 1993]) decomposition is performed over an EEG recording (no additional complexity is added) and several statistics are extracted from the obtained decomposition.

Similarly, regularity of EEG signals has also been studied before [MATHUVANESAN et JAYASANKAR, 2013; MIKAILI et GOLPAYEGANI, 2002; NATARAJAN et collab., 2004; POPIVANOV et collab., 2006], however this topic is broad and there are several ways to measure a signal regularity; including Shannon entropy, spectral entropy, ApEn, Lempel-Ziv complexity and Higuchi fractal, among others. Some works have focused on ApEn concerning EEG analysis [ABÁSOLO et collab., 2005; CHUCKRAVANEN, 2014; FAN et collab., 2011]. Closely related to signal regularity, there is also some research for EEG analysis by means of mutifractal theory. For instance, a tool for extracting the regularity spectrum of a time series, the so called Multifractal Detrended Fluctuation Analysis (MFDFA) is employed in different works, by approximating the Hurst exponent [FIGLIOLA et SERRANO, 2007; KANTELHARDT et collab., 2002; ZORICK et MANDELKERN, 2013]. Another method for extracting the fractal dimension of a signal, an approach for regularity analysis, is the Wavelet Transform Modulus Maxima (WTMM), used in several works [DICK et SVYATOGOR, 2012; MA et collab.,

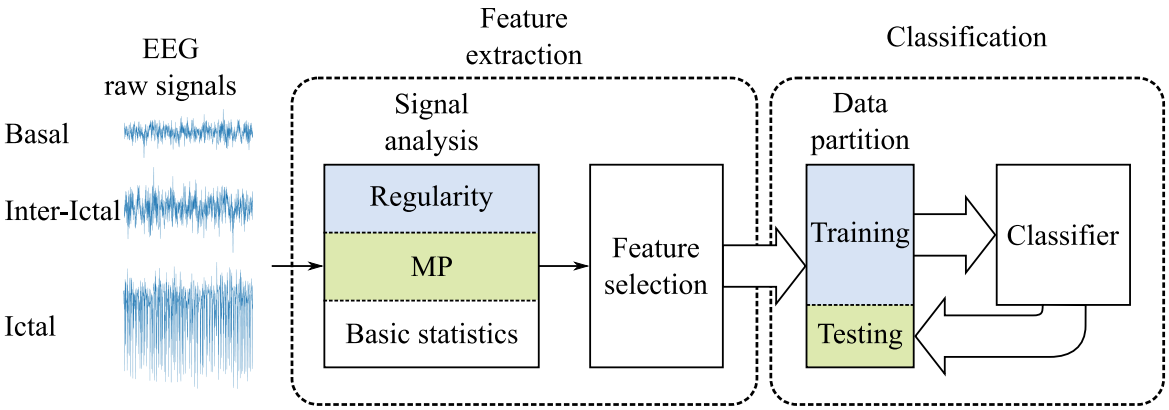


Figure 16.1 – Proposed system for automatic epileptic seizures detection.

2006; POPIVANOV et collab., 2006; SONG et LEE, 2005]. However, this work represents the first time that the approach by JAFFARD [2004] is employed to solve the problem of detecting epileptic states.

In summary, the MP algorithm and regularity analysis complement each other by accessing different layers of information embedded in the signals. Hölderian exponent calculation works as a local regularity measure while MP acts as a global regularity measure. In the following subsections we formally describe both feature extraction methods.

16.2.2.1 Hölder exponent

In general, the regularity of a signal can be described as the characterization of the singularities it contains. These singularities often carry valuable information, specially for nonlinear and non-stationary signals [MALLAT, 2008].

There are several methods to calculate the regularity of a non-stationary time series, either of form local or pointwise [MALLAT, 2008]. The precise calculus of them requires of complex numerical methods which are not practically viable and sometimes are not even possible to calculate it. Rather, an approximation is commonly used, where the computational resources are feasible and its characterization is quite accurate. In this work we make use of the regularity measure given by the pointwise Hölder exponent, approximated through a discrete wavelet decomposition. The pointwise Hölder exponent of a function is defined as:

Definition 15 *Let* $f : \mathbb{R}^d \to \mathbb{R}, s \in \mathbb{R}^{+*}\backslash \boldsymbol{N}$ *and* $x_0 \in \mathbb{R}^d$. *Then,* $f \in C^s(x_0) \Leftrightarrow \exists \eta \in \mathbb{R}^{+*}$, *a polynomial* P *of degree* $< s$ *and a constant* c *such that*

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \leq c|x - x_0|^s. \tag{16.1}$$

The pointwise Hölder exponent of f at x_0 is $\alpha_p = sup_s\{f \in C^s(x_0)\}$.

In some cases, it is necessary to have information of the regularity of a signal, not in a point, but in a neighborhood of that point, namely a local Hölder exponent. Its definition is:

Definition 16 *Let* f *be a function on the neighborhood of* x_0. *Let* $\{I_n\}_{n\in\boldsymbol{N}}$ *be a decreasing sequence of open intervals converging toward* x_0. *The local Hölder exponent of the function* f *at* x_0 *is*

$$\alpha_l(x_0) = \sup_{n\in\boldsymbol{N}}[\alpha_f(I_n)] = \lim_{n\to+\infty} \alpha_f(I_n) \tag{16.2}$$

where

$$\alpha_f(\Omega) = sup\{0 < s < 1 : f \in C^s(\Omega)\} \tag{16.3}$$

with $f \in C^s(\Omega)$ *if there exist a constant* c *such that, for any couple* (x, y) *in* Ω *an open subset of* $\mathbb{R}$, $|f(x) - f(y)| \leq |x - y|^s$.

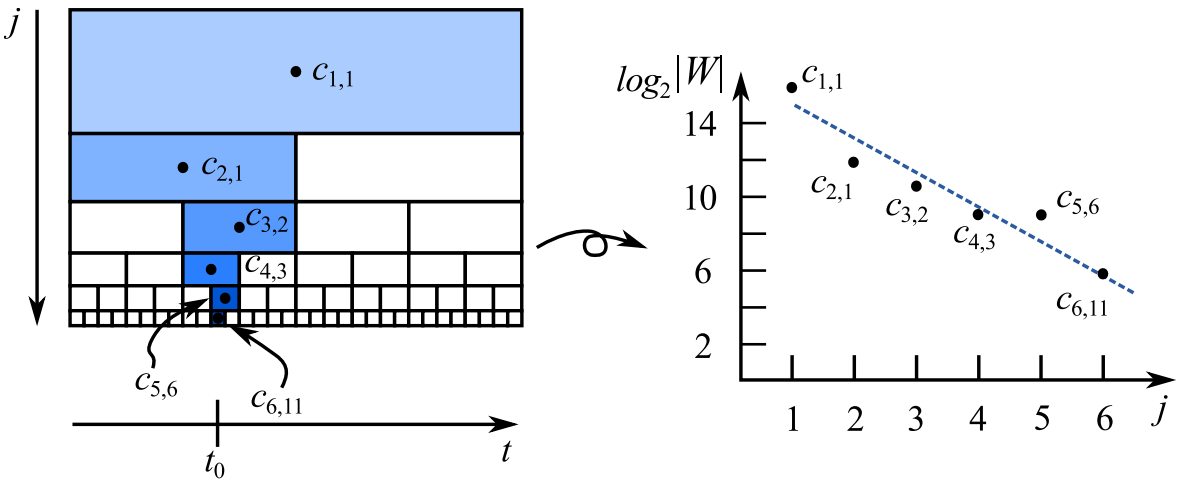
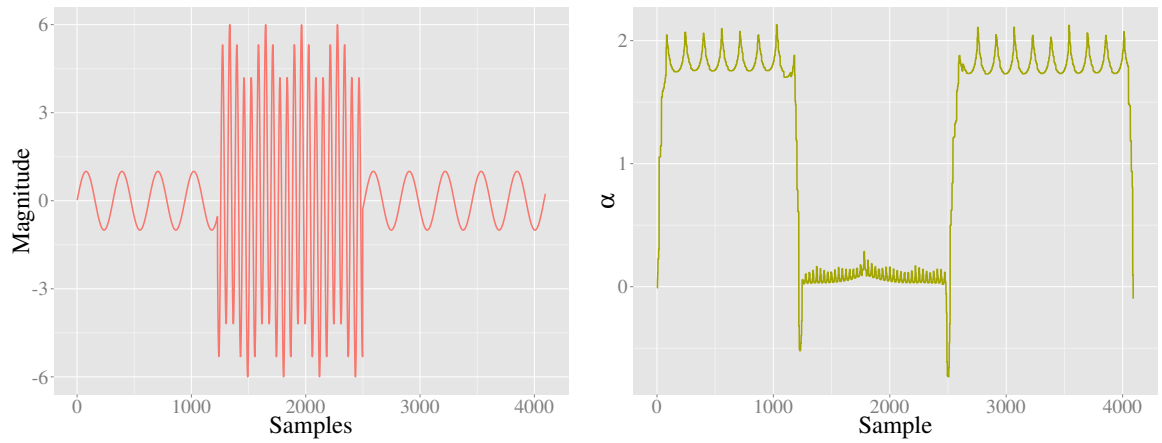


Figure 16.2 – Regression calculated over a point of the signal. Left image shows a dyadic wavelet decomposition, and the right image display the actual regression calculated over the point t_0, where each dot corresponds to each log_2 of the wavelet coefficient magnitude located approximately above t_0.



(a) [Raw signal, superposition of two waveforms with different frequencies.]

(b) [Regularity measure.]

Figure 16.3 – (b) Hölderian regularity calculated over a sample signal (a), where α is the estimated Hölder exponent.

There are, as well, different techniques to estimate the pointwise regularity by considering Hölder spaces. One of them is by using the oscillation method; a mechanism that closely follows the Hölder exponent definition [JAFFARD et MEYER, 1996; TRICOT, 1995]. Another method is the use of discrete wavelet decomposition (DWT). Wavelet analysis can be used to compute an approximation of the Hölder exponent by estimating the decay rate of its wavelet coefficients versus the scales, corresponding to the wavelets localized near the considered point [JAFFARD, 2004]. Here, we choose the latter method as the preferred way to achieve the regularity estimation of our signals, and specifically using a dyadic decomposition methodology.

Being Ψ a mother wavelet of the classical form $\{\Psi_{j,k}\}_{j,k}$ that makes an orthonormal base of $\mathbf{L}^2$, wavelet coefficients are denoted as $c_{j,k}$ of f where j corresponds to the scales and k corresponds to the temporal location.

Theorem 1 *Let f be a uniformly Hölderian function and α is the pointwise Hölder exponent of f at point t_0, then exists a constant $c > 0$ such that the wavelet coefficient satisfy*

$$|c_{j,k}| \leq c2^{-j(\alpha+\frac{1}{2})}(1 + |2^j t_0 - k|)^{\alpha} \forall j, k \in \mathbb{Z}^2 \; ; \tag{16.4}$$

reciprocally,

$$\text{if } \forall j,k \in \mathbb{Z}^2 \text{ we have } |c_{j,k}| \leq c2^{-j(\alpha+\frac{1}{2})}(1+|2^j t_0 - k|)^{\alpha'} , \tag{16.5}$$

for an $\alpha' < \alpha$ the Hölder exponent of f at t_0 is α. A function f is uniformly Hölderian is there is $\varepsilon > 0$ such that $f \in C^{\varepsilon}(\mathbb{R})$.

Equation 16.4 states that the wavelet coefficients decrease in absolute value by an amount that depends on the Hölder exponent. From this theorem, if we make the hypothesis that the global and local exponents are the same then we are only interested in indices j, k such that $|k - 2^j t_0| < c$ for the point t_0. Then, Equation 16.5 implies the existence of coefficients of the order of $2^{-j(\alpha+\frac{1}{2})}$. This simplifying assumption is satisfied if and only if the local exponent is equal to the pointwise exponent at t_0 [VÉHEL et LEGRAND, 2004].

Under this assumption, an estimation of the Hölder exponent is obtained by means of the slope p of the $log_2|c_{j,k}|$ regression with $j : \alpha(n, t_0) = -p - \frac{1}{2}$, $n = \lfloor log_2(N) \rfloor$ being the number of decomposition levels and N the length of the signal.

Theorem 2 *In each point t_0 of the signal, decomposed in n scales, the regularity estimation is given by*

$$\alpha(n, t_0) = -\frac{1}{2} - K_n \sum_{j=1}^{n} s_j log_2|c_{j,k}| , \tag{16.6}$$

with $K_n = \frac{12}{n(n-1)(n+1)}$ and $s_j = j - \frac{n+1}{2}$. $c_{j,k}$ are the wavelet coefficients located over t_0. The value of k is given by $\lfloor \frac{t_0+1}{2^{n-j+1}} \rfloor$.

Figure 16.2 illustrates this method [LEGRAND, 2004].

Figure 16.3 presents an example of regularity computation on a time series. The input signal is composed by the superposition of two waveforms with different amplitudes, frequencies and starting times. The regularity measure characterizes the signal singularities. The amplitude given by α corresponds to the regularity of the signal around a given point. Clearly, the low frequency waveform is more regular than the high frequency one. The hard transition between both waveforms is also captured by the presence of highly irregular spikes.

16.2.2.2 Matching Pursuit

Matching pursuit refers to a family of greedy algorithms that compute the best nonlinear approximation of a signal [HUSSAIN et SHAWE-TAYLOR, 2009]. One of these algorithms was introduced by MALLAT [1993], with the goal of decomposing a signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions.

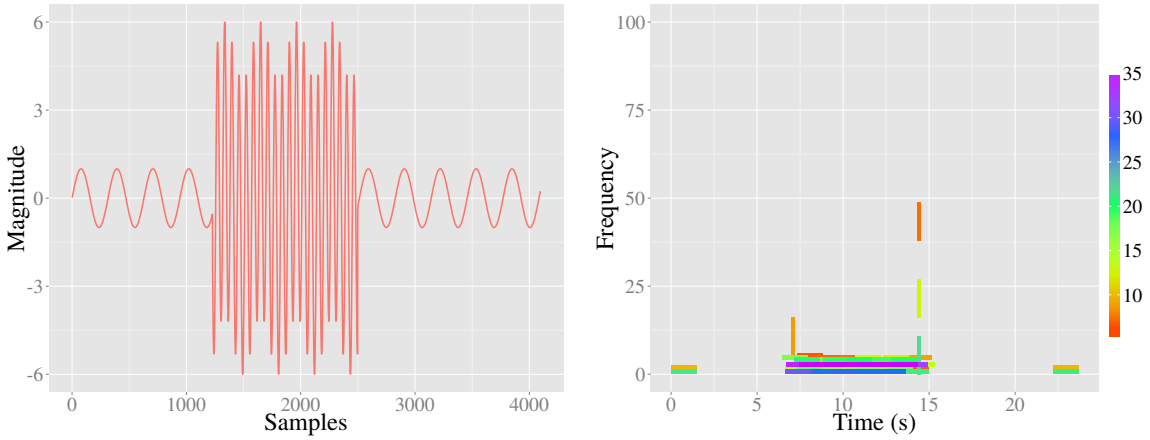
An unknown signal can be expanded in terms of functions $g_{\gamma n}$, called time-frequency atoms [CHEN et collab., 1998], given by

$$f(t) = \sum_{n=-\infty}^{+\infty} a_n g_{\gamma n}(t) , \tag{16.7}$$

where a_n is an expansion coefficient. In this way $f(t)$ can be explained using functions $g_{\gamma n}$ from a dictionary $\mathcal{D}$. The family $\mathcal{D}$= $[(g_\gamma(t))_{\gamma \in \mathbf{\Gamma}}]$, where the index γ is an element of the set $\mathbf{\Gamma} = \mathbb{R}^+ \times \mathbb{R}^2$, is highly redundant. In general, a family of time-frequency atoms can be generated by scaling, translating and modulating a single window function

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t} , \tag{16.8}$$

where s, u, ξ are the scale, translation and frequency modulation respectively, for the proposed window. The expansion coefficient a_n in Equation 16.7 provide explicit information on certain types of properties of $f(t)$.



(a) [Raw signal.]

(b) [MP decomposition.]

Figure 16.4 – MP Heisenberg boxes over a sample signal.

Let $f \in \mathbf{H}$, where $\mathbf{H}$ is a Hilbert space, and in our case $\mathbf{H} = \mathbf{L}^2(\mathbb{R})$ (space of complex valued functions), the MP algorithm computes a linear expansion of f over a set of vectors selected from $\mathcal{D}$, by successive approximations of f with orthogonal projections on elements of $\mathcal{D}$. Let $g_{\gamma 0} \in \mathcal{D}$, the vector f can be decomposed into

$$f = \langle f, g_{\gamma 0} \rangle g_{\gamma 0} + Rf \ , \tag{16.9}$$

where Rf is the residual vector after approximating f in the direction of $g_{\gamma 0}$. The MP algorithm sub-decomposes the residue Rf by projecting it on a vector of $\mathcal{D}$ that best matches Rf, as it was done for f.

In general, after m iterations MP decomposes a signal f into

$$f = \sum_{n=0}^{m-1} \langle R^n f, g_{\gamma n} \rangle g_{\gamma n} + R^m f \tag{16.10}$$

or, in terms of energy,

$$\|f\|^2 = \sum_{n=0}^{m-1} |\langle R^n f, g_{\gamma n} \rangle|^2 + \|R^m f\|^2 \ . \tag{16.11}$$

In Figure 16.4, a visualization of the MP decomposition into time-frequency atoms of one signal is shown, by means of a Heisenberg boxes. Notice that the raw signal is the same used in the example of Figure 16.3. The boxes show specific waveform characteristics: its frequencies, positions, durations and amplitudes, by means of the atoms found during the decomposition. The high frequency spikes that exist at the transition of both waveforms are captured by the MP algorithm.

16.2.3 Proposed feature sets

The main contribution of this chapter is the exploitation of time-frequency content and regularity analysis of EEG signals for the detection of the Ictal state during an epileptic seizure. The pointwise Hölder exponent and the MP decomposition are not directly dependent on signal amplitude. This is important in real applications, since it is well known that EEG recordings can vary in amplitude as result of changes in the instrumentation or noise induction [USAKLI, 2010]. Usually, EEG recording systems require a calibration process which can be problematic [GRIZOU et collab., 2014]. Therefore, frequency and regularity-based techniques could help alleviate the previously mentioned complications.

Nonetheless, it is evident that the Ictal state presents high amplitude spikes in the time-domain, a characteristic that might help simplify the detection problem. Therefore, we

recognize that some useful, and probably necessary, information can only be obtained by explicitly considering the time-domain features.

Thus, we propose two sets of features extracted from each EEG epoch, which are evaluated and compared in our experimental work: (a) Hölder regularity and MP decomposition features (4 features); (b) Hölder, MP and statistical time-domain features (10 features). Additionally, we include an additional set which is automatically chosen by searching for the optimum combination of the proposed 10 feature set, using a meta-heuristic combinatorial optimization algorithm. All of these feature sets are summarized in Table 16.2 and described in the following subsections.

16.2.3.1 Hölderian regularity and MP decomposition (4 features set)

These features are presented in the second column of Table 16.2. For the regularity analysis, two statistical measures are used as features: the mean regularity of an epoch $\mu(\mathbf{H})$ and standard deviation calculated by the Median Absolute Deviation (MAD), MAD($\mathbf{H}$), being $\mathbf{H}$ the pointwise regularity vector for a given epoch with the same length as number of epoch samples. The MAD is commonly used instead of the standard deviation because it is more resilient to outliers. The MAD can be calculated with

$$\text{MAD}(\mathbf{H}) = (\mu_{1/2})_i(|H_i - (\mu_{1/2})_j(H_j)|) \,, \tag{16.12}$$

where $\mu_{1/2}$ is the median, i and j are indexes of samples in $\mathbf{H}$.

For the MP decomposition, we propose two basic features: the Gabor Atom Density (GAD), first introduced by KOUBEISSI et collab. [2009], and the frequency mean $\mu(\mathbf{F})$ of all atoms, where $\mathbf{F}$ is the atom frequencies vector. The frequencies mean is calculated with

$$\mu(\mathbf{F}) = \frac{1}{m}\sum_{n=0}^{m} F_n(g_{\gamma n}) \,, \tag{16.13}$$

where $F_n(g_{\gamma n})$ is the frequency of the n^{th} atom found during the decomposition.

The GAD is defined as the number of atoms m retrieved during the decomposition, divided by the size of the reconstructed time-frequency space. For each window, the range of time and frequency are, respectively, $R_t = N(1/F_s)$; $R_f = F_s/2$, where N is the number of points in the window and F_s the sampling frequency. The density of atoms over the space is

$$\text{GAD} = \frac{m}{R_t R_f} = \frac{2m}{N} \,. \tag{16.14}$$

16.2.3.2 Hölder, MP decomposition and time-domain features (10 features set)

This set is presented in the third column of Table 16.2. These include the regularity and MP features, as well as the mean amplitude of the MP decomposition

$$\mu(\mathbf{A}) = \frac{1}{m}\sum_{n=0}^{m} A_n(g_{\gamma n}) \,, \tag{16.15}$$

where $A_n(g_{\gamma n})$ is the amplitude of the n^{th} atom.

Additionally, five additional features which were successfully used in [SOTELO et collab., 2013] for epilepsy state identification are considered. The proposed features are basic statistical measures calculated explicitly in the time domain over the raw epochs $\mathbf{E}$, with $\mathbf{E}$ being the epoch vector in the time domain. They are: the mean $\mu(\mathbf{E})$, median $\mu_{1/2}(\mathbf{E})$, standard deviation $\sigma(\mathbf{E})$, skewness $\gamma_1(\mathbf{E})$ and kurtosis $\gamma_2(\mathbf{E})$.

Table 16.2 – Proposed feature sets. An additional set is selected automatically, presented in Section 16.3.6

Type	Number of features	
	4	10
Regularity statistics	MAD($\mathbf{H}$) $\mu(\mathbf{H})$	MAD($\mathbf{H}$) $\mu(\mathbf{H})$
MP statistics	GAD $\mu(\mathbf{F})$	GAD $\mu(\mathbf{F})$ $\mu(\mathbf{A})$
Raw signal basic statistics		$\mu(\mathbf{E})$ $\mu_{1/2}(\mathbf{E})$ $\sigma(\mathbf{E})$ $\gamma_1(\mathbf{E})$ $\gamma_2(\mathbf{E})$

16.2.3.3 Automatic feature selection with GA

An additional feature set is proposed, which is automatically chosen as a subset of the complete 10 feature set. To perform this automatic feature selection procedure we employ a Genetic Algorithm (GA) [GOLDBERG, 1989], a popular algorithm from the field of Evolutionary Computation (EC) [EIBEN et SMITH, 2015].

GAs are well established stochastic population-based optimization algorithms, where a large set of candidate solutions compete to survive in each iteration (generation) of the search process. Each solution is ranked by a fitness function that is defined based on the problem domain. For instance, for a classification task, fitness can be based on accuracy or total classification error. Each candidate solution (individual) is encoded using a domain specific representation called a genotype, that the search process can manipulate. The genotype is then decoded into a problem domain representation (phenotype), for which fitness can be computed. By means of the search (genetic) operators, solutions evolve by randomly swapping genetic material between pairs of solutions (crossover) or by randomly altering parts of existing solutions (mutation). The fitness function is the main determining factor for choosing which solutions will be subjected to the search operators, and to determine which solutions will be kept in the population or will be discarded before the next iteration. This produces a selective pressure towards high-performing solutions. Normally, the search is stopped when a certain number of iterations is reached, and the best solution found at that point is returned.

GAs have been widely used for automatic feature selection [LIN et collab., 2014; SUN et collab., 2004], in particular given that the canonical binary representation is well suited to represent solutions in this general task. Section 16.3.6 provides further details of our implementation and the resulting feature set.

16.2.4 Classification

Once the feature extraction process has been done and consequentially a set of features has been extracted, the classification task must be performed, depicted in the rightmost block of Figure 16.1. This task solves the problem of assigning a class label to unknown data, based on the set of known features and corresponding class labels, basically a supervised learning problem [DUDA et collab., 2000]. Machine learning literature includes a wide variety of supervised classification techniques, from simple Bayes classifiers, to more complex methods like hidden Markov models [ALPAYDIN, 2010] or genetic programming [SOTELO et collab., 2013]. In this work, we use the Random Forests™(RF) classifier by BREIMAN [2001], part of the family of decision trees classifiers [DUDA et collab., 2000]. Decision trees creates a model

that predicts the class label of an unknown feature vector based on rules derived from the training set and expressed as a tree where each internal node performs a decision based on a particular input feature, and each leaf is labeled with a class or a probability distribution over the classes [ROKACH et MAIMON, 2008]. RF creates a set of decision trees in an ensemble scheme, where a set of weak models work together to build a stronger classifier. The main algorithm parameter is the number of trees used.

This method exhibits several advantages that have made it popular in many domains [SCULLEY, 2011], including EEG analysis [CHEN et collab., 2014]. It is noteworthy to mention that to the author's knowledge RF has not been applied to the Bonn data set before [ACHARYA et collab., 2013]. The advantages of RF are [BREIMAN, 2001]: (1) accuracy on test data is superior to many classifiers when feature selection is optimal; (2) it performs implicit feature selection; (3) it can effectively estimate missing data while maintaining high accuracy; and (4) it has shown resiliency to over-fitting and class imbalance.

16.3 Experimental work

This section presents our experimental validation of the proposed feature extraction methods and classification scheme, evaluating each of the proposed feature sets, the effect of signal normalization and considering several problem formulations, using standard performance measures and statistical tests. Moreover, our results are compared with state-of-the-art works from recent literature. In all of the following work, the reported algorithms and tests were implemented in MATLAB [2014].

16.3.1 Classification problems

Following the suggestions in [ACHARYA et collab., 2013], the Bonn data set can be used to formulate four distinct problems of varying degrees of difficulty, these are summarized in Table 16.3. Problem 1 and Problem 2 are binary classification problems, the former between class A and E (see Table 16.1) and the latter between all Basal readings (classes A, B, C and D) and Ictal state readings (class E). On the other hand, Problems 3 and 4 are multi-class problems, the former considering classes A, D and E, while Problem 4 considers all five classes, the most difficult case. Note that each group (A-E) contains 100 epochs with 4096 samples each. Therefore, Problem 2 in particular presents an unbalanced classification task.

16.3.2 Epoch segmentation

Each epoch in the Bonn database is composed by 4096 samples or measurements, given the duration of each recording and the sampling frequency. However, we pose two different classification problems by treating the data differently. First, we consider each epoch as a single data vector, leaving 100 epochs per class. In the second approach, we split each epoch into 4 non-overlapping segments of roughly 5.9s each with 1024 samples, following [GÜLER et UBEYLI, 2005; ÜBEYLI et GÜLER, 2007]. Therefore, each class contains 400 total epochs (100×4). Notice that this increases the number of examples we have for each class, but each example contains less information, possibly making it more difficult to characterize each epoch. Hereafter, we refer to the second variant as the segmented method.

16.3.3 Experimental setup

For the computation of Hölder regularity we use a Daubechies 10 orthogonal wavelet family and least square linear regression for the coefficients slope calculation in the wavelet decomposition. When we segment the epochs, the maximum decomposition level is lowered because the epoch length is smaller by a factor of 4. The MP algorithm is setup with

Table 16.3 – Classification problems derived from the Bonn data set.

Name	Type	Classes
Problem 1	Binary	A - E
Problem 2	Binary	A,B,C,D - E
Problem 3	Multi-class	A - D - E
Problem 4	Multi-class	A - B - C - D - E

Table 16.4 – Configuration for MP algorithm.

Description	Value
Number of atoms per dictionary	5000
Sampling rate	173.61 Hz
Stopping criteria	25 dB SNR of reconstruction
Atom types in dictionary	Gabor with Gaussian window
	Gabor with Cosine window
	Chirp with Gaussian window
	Gabor with Rectangle window

the parameters in Table 16.4. The atom dictionary can capture different EEG patterns, this helps reduce the number of required iterations to obtain a satisfactory reconstruction.

Usually, there are two stopping criteria for the MP algorithm. One is the number of iterations and the other is the energy level of the signal reconstruction. Here the latter was chosen because we prefer to keep an uniform energy level after reconstruction for all epochs, regardless of its class.

Figure 16.5 shows the computation of the pointwise Hölder exponent and MP on the first epoch from each class (A-E). The pointwise Hölder exponent α are similar for class A, B, C and D, with small statistical differences. Class E is distinct, its regularity is noticeably different compared with other groups. For MP decomposition, we can see a clearer difference of atom distribution for all groups. Group B captures more content in higher frequencies, opposed to class E which has a more compact atom location toward lower frequencies despite the fact that it contains more atoms. Other groups exhibit an intermediate composition of atom frequencies, but each shows distinct patterns.

Figure 16.6 depicts pairwise projections of all possible combinations of the proposed features in Table 16.2, with their corresponding class emphasized by a different color. Effectively, the complete feature space can be visualized, depicting the underlying difficulty of the classification task. From this visualization we can see clearly that no single feature pair is enough to achieve strong discriminant effects. A common pattern found is that variance of class E is large and similar for all combinations of features, although its is usually clearly separated from other classes, thus the uniqueness of the signals generated during epileptic seizures.

16.3.4 Pre-processing

In this work we consider a simple pre-processing stage, where the amplitude is scaled within the range $[0, 1]$ using min-max normalization. To test the importance of this pre-processing step, the proposed features are evaluated both with and without normalization.

16.3.5 Classifier setup and performance measures

The experimental work uses the Random Forests™ implementation by JAIANTILAL [2012], configured to use a maximum of 200 decision trees. The EEG data was partitioned in train-

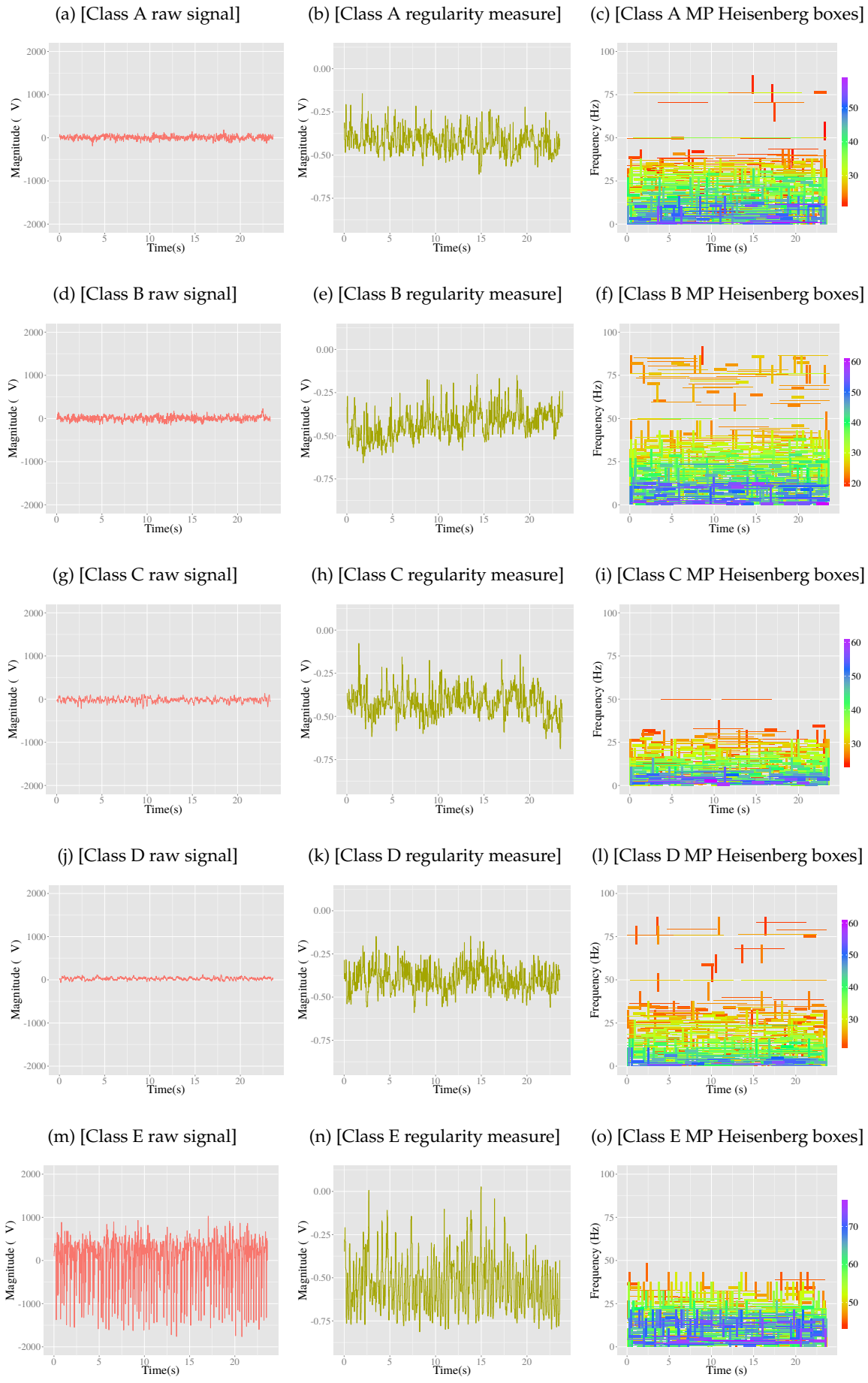


Figure 16.5 – The pointwise Hölder exponent α (second column) and MP Heisenberg boxes (third column) calculated over the first epoch of each group in the Bonn data set (first column).

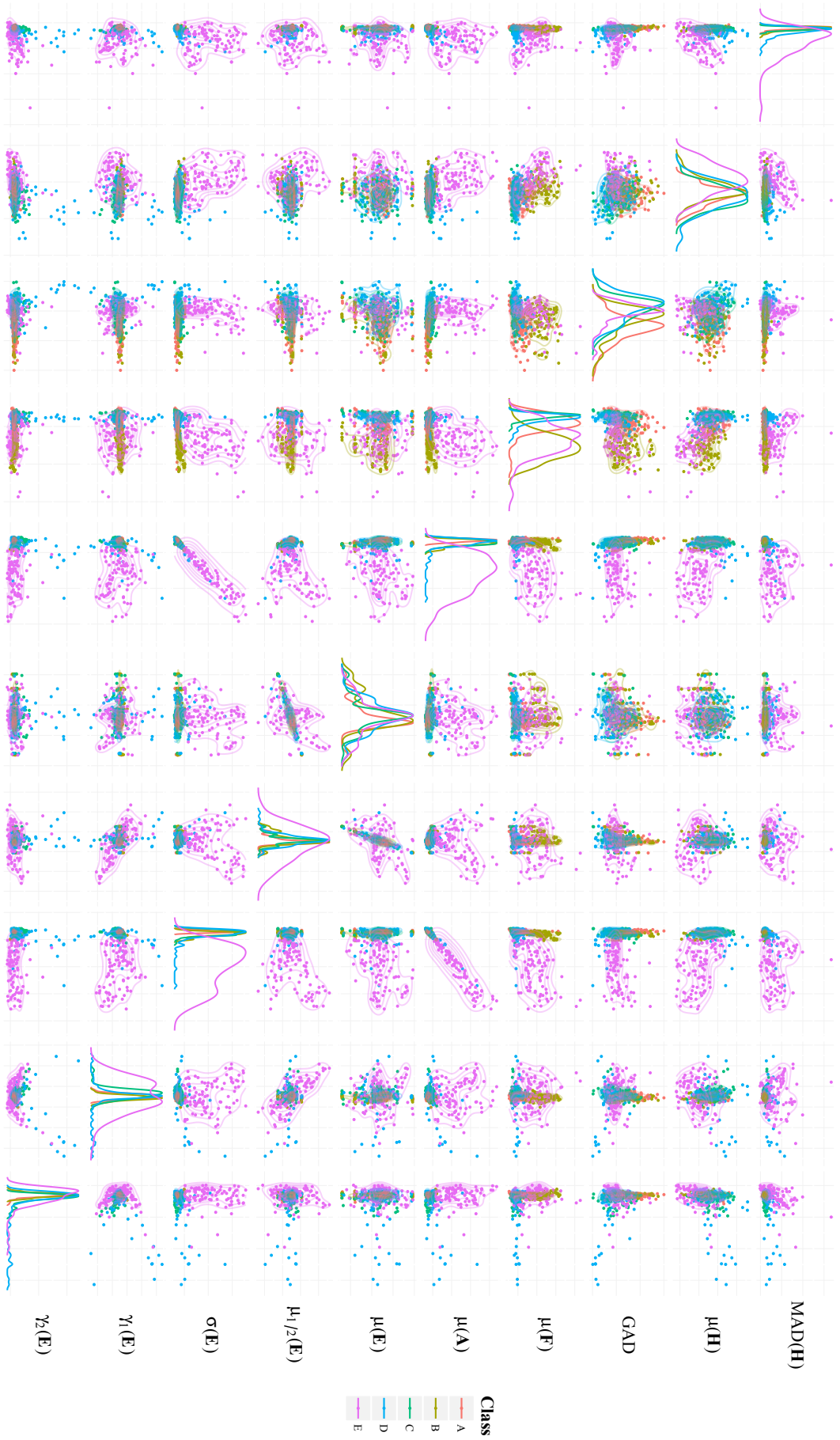


Figure 16.6 – Matrix of pairwise projections of all features for all five classes. The diagonal plots correspond to the density distribution per feature.

ing and testing sets using a 10-fold cross-validation to obtain a robust estimate of classifier performance. The cross-validation was performed 10 times, this gives us a total of 100 runs per problem instance. The following performance measures are reported based on statistics over all runs.

The main quality measure to determine the classification performance is the accuracy. Given a confusion matrix $A = [A(i,j)]$, where its element $A(i,j)$ is the number of data points whose true class label was i and were classified to class j, the overall accuracy can be calculated with

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{M} A(i,i)\ , \tag{16.16}$$

where N is the number of samples in the data set and $i \in M$ being the i-th label of M classes. Additionally, specificity, F-score and recall were calculated as test classification performance measures. Recall and specificity are measures developed for binary class problems, thus they are calculated for each class, where high values are desired. These are given by

$$\text{Recall}_i = \frac{A(i,i)}{A(i,i) + A(i,j)} \tag{16.17}$$

and

$$\text{Specificity}_i = \frac{A(j,j)}{A(j,j) + A(j,i)}\ . \tag{16.18}$$

The balanced F-score measure can be calculated based on precision and recall by

$$\text{F-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{16.19}$$

where precision is given by

$$\text{Precision}_i = \frac{A(i,i)}{A(i,i) + A(j,i)}\ . \tag{16.20}$$

16.3.6 Automatic feature selection (9 features set)

To pose the feature selection problem, the genotype of each candidate solution is given by a binary string $\mathbf{b} = [b_1, \ldots, b_{10}]$, where each b_i is associated with one element in the 10 feature set (see the third column of Table 16.2). Each bit in the string determines if the i-th feature is used ($b_i = 1$) or not ($b_i = 0$). The fitness function is based on the classification accuracy (Equation 16.16) achieved on the training set, considering all five classes (Problem 4). Moreover, we apply pre-processing and use the complete epochs. The parameters of the GA are given in Table 16.5. The GA was executed using 10-fold cross-validation, with the goal of detecting which features were chosen with the highest frequency.

Figure 16.7 depicts the frequency of the optimal set of features found by the GA over all folds of the training data. Notice that most features were used at least 80% of the time, with seven of them used with 100% frequency. The only exception was the $\gamma_2(\mathbf{E})$ feature, indicating that it is likely the least useful of all the features. Therefore, in this set we only use the first nine features; hereafter we will refer to this set as the 9 feature set or the automatic feature set.

Table 16.5 – GA parameters.

Parameter	Value
Population	100
Generations	50
Chromosome	Binary string
Crossover operator	Intermediate, 0.8 prob.
Mutation operator	Adaptive feasible, 0.15 prob.
Elitism	0.05%
Selection method	Tournament
Fitness function	RF classification accuracy

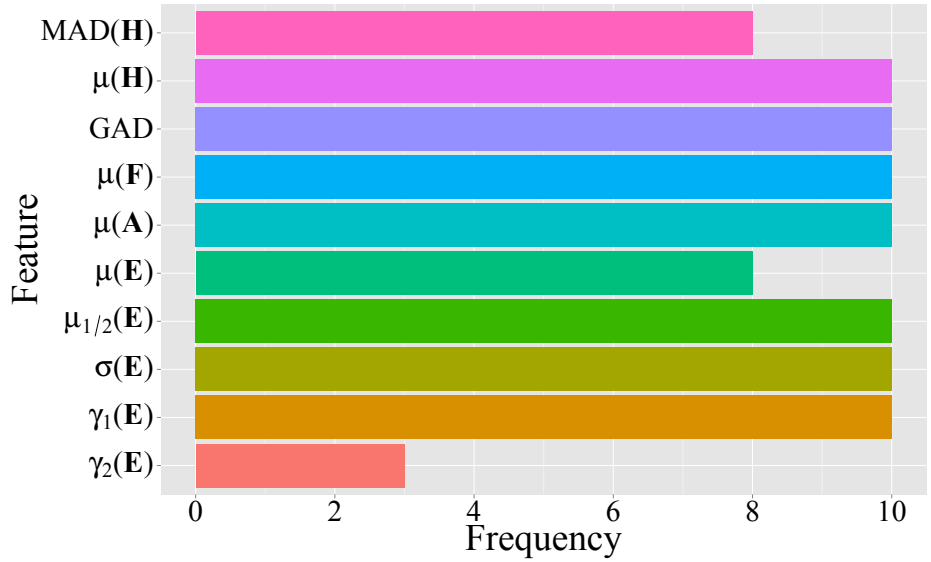


Figure 16.7 – Optimal feature frequency for GA algorithm. Each bar value is the accumulative feature appearance after running over all 10 folds.

16.4 Results

The experimental results are organized as follows. Firstly, we perform a statistical analysis on the complete feature set to detect any significant differences among the five classes. For this, the One-way Analysis of Variance (ANOVA) is employed over the feature matrix. These results are presented in Table 16.6.

Secondly, Table 16.7 summarize the performance of our classification system for each problem (1-4) using each feature set (4, 9 and 10). This table shows results obtained on the raw and normalized signals. In each case, the tables show the average Specificity, Recall and F-score computed over all runs. Moreover, the median, best, worst and inter-quartile spread is given. For the segmented approach, the same performance analysis is presented in Table 16.8 as well.

Finally, to validate our results, non-parametric statistical tests are used to compare the performance behavior of each system configuration based on the accuracy results. For every problem, the proposed approach was tested considering each feature set (3 variants), whether or not pre-processing is applied (2 variants) and if it is using the full epochs or the segmented approach (2 variants); a total of twelve different groups ($3 \times 2 \times 2$). Afterward, a Friedman test is used to calculate pairwise statistical differences. The p-values for each pairwise comparison are given in Table 16.9, applying the Benjamini-Hochberg correction. For these tests, we will reject the null hypothesis that two groups share the same median value with p-values below an $\alpha = 0.05$.

Table 16.6 – One-way ANOVA test for complete feature set. Mean and standard deviation per class are shown in columns 2 through 6. p-values for each feature are shown in last column.

Feature	Class					p-value
	A	B	C	D	E	
MAD($\mathbf{H}$)	0.07±0.01	0.07±0.01	0.07±0.01	0.08±0.02	0.12±0.05	0.0000
$\mu(\mathbf{H})$	-0.46±0.11	-0.53±0.11	-0.44±0.12	-0.45±0.15	-0.63±0.15	0.0000
GAD	0.10±0.02	0.09±0.02	0.06±0.03	0.03±0.01	0.09±0.05	0.0000
$\mu(\mathbf{F})$	0.00±0.00	0.01±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.0000
$\mu(\mathbf{A})$	6.46±0.74	7.12±0.69	5.30±0.44	4.90±0.24	6.50±0.49	0.0000
$\mu(\mathbf{E})$	-6.26±24.69	-12.51±30.57	-8.88±24.00	-6.20±23.40	-4.75±27.02	0.2437
$\mu_{1/2}(\mathbf{E})$	-6.23±24.63	-13.08±30.63	-8.42±24.04	-8.72±30.09	6.67±81.36	0.0226
$\sigma(\mathbf{E})$	40.73±8.26	61.11±18.12	50.83±19.17	65.62±57.53	306.61±147.23	0.0000
$\gamma_1(\mathbf{E})$	-0.02±0.11	0.06±0.13	-0.15±0.27	0.08±0.76	-0.06±0.76	0.0092
$\gamma_2(\mathbf{E})$	3.23±0.27	3.22±0.27	3.65±0.63	4.33±2.59	3.43±1.19	0.0000

Table 16.7 – Summary of classification performance processed over full length epochs, employing a 10-fold cross validation and 10 independent runs. Includes raw and normalized signals. Columns show the average Specificity, Recall, F-score and rank statistics of the accuracy, including median, best, worst and Interquartile Range (IQR).

(a) 4 features

		Raw							Normalized						
		Specificity	Recall	F-score	Accuracy				Specificity	Recall	F-score	Accuracy			
					media	best	worst	IQR				media	best	worst	IQR
Problem 1	Class A	0.9710	0.9700	0.9710	100	100	100	0.00	0.9750	0.9600	0.9668	96.7	100	95	3.80
	Class E	0.9700	0.9710	0.9692					0.9600	0.9750	0.9680				
	Average	0.9705	0.9705	0.9701					0.9675	0.9675	0.9674				
Problem 2	Class A,B,C,D	0.8200	0.9613	0.9582	92.8	92.8	92.8	0.00	0.8000	0.9545	0.9523	92.6	94	91	2.00
	Class E	0.9613	0.8200	0.8305					0.9545	0.8000	0.8040				
	Average	0.8906	0.8906	0.8943					0.8773	0.8773	0.8782				
Problem 3	Class A	0.9287	0.8430	0.8494	87	88.3	86.6	1.00	0.9897	0.9550	0.9665	95.8	96.7	95	0.98
	Class D	0.9203	0.8350	0.8371					0.9650	0.9390	0.9343				
	Class E	0.9655	0.9650	0.9531					0.9610	0.9390	0.9330				
	Average	0.9382	0.8810	0.8798					0.9719	0.9443	0.9446				
Problem 4	Class A	0.8941	0.7160	0.6960	69.6	71	68	2.00	0.8714	0.6420	0.6207	70.4	72	68	2.70
	Class B	0.9291	0.7290	0.7462					0.8593	0.4870	0.4758				
	Class C	0.8652	0.7380	0.6758					0.9171	0.6510	0.6783				
	Class D	0.9222	0.4930	0.5603					0.9370	0.9350	0.8892				
	Class E	0.9413	0.8970	0.8716					0.9480	0.7800	0.8045				
	Average	0.9104	0.7146	0.7100					0.9066	0.6990	0.6937				

(b) 9 features (Automatic)

Problem 1	Class A	1.0000	1.0000	1.0000	100	100	100	0.00	1.0000	1.0000	1.0000	100	100	100	0.00
	Class E	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
	Average	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
Problem 2	Class A,B,C,D	0.9680	0.9925	0.9923	98	98	98	0.00	0.9690	0.9950	0.9937	100	100	100	0.00
	Class E	0.9925	0.9680	0.9688					0.9950	0.9690	0.9735				
	Average	0.9803	0.9803	0.9805					0.9820	0.9820	0.9836				
Problem 3	Class A	0.9351	0.9410	0.9105	93	93.3	91.7	0.70	1.0000	1.0000	1.0000	100	100	100	0.00
	Class D	0.9625	0.8600	0.8863					0.9930	1.0000	0.9933				
	Class E	0.9926	0.9840	0.9854					1.0000	0.9860	0.9926				
	Average	0.9634	0.9283	0.9274					0.9977	0.9953	0.9953				
Problem 4	Class A	0.9291	0.8590	0.8179	80.7	82	79	1.06	0.9726	0.8970	0.8965	92.9	94	92	0.73
	Class B	0.9626	0.8650	0.8677					0.9807	0.8630	0.8886				
	Class C	0.9119	0.7970	0.7567					0.9729	0.9350	0.9183				
	Class D	0.9408	0.5770	0.6373					0.9819	0.9650	0.9502				
	Class E	0.9866	0.9520	0.9536					0.9946	0.9700	0.9741				
	Average	0.9462	0.8100	0.8067					0.9805	0.9260	0.9255				

(c) 10 features

Problem 1	Class A	1.0000	1.0000	1.0000	100	100	100	0.00	1.0000	1.0000	1.0000	100	100	100	0.00
	Class E	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
	Average	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
Problem 2	Class A,B,C,D	0.9680	0.9905	0.9912	98.6	99	98	0.70	0.9700	0.9950	0.9938	100	100	100	0
	Class E	0.9905	0.9680	0.9654					0.9950	0.9700	0.9747				
	Average	0.9793	0.9793	0.9783					0.9825	0.9825	0.9842				
Problem 3	Class A	0.9553	0.9260	0.9200	93.7	95	93.3	0.90	1.0000	1.0000	1.0000	100	100	100	0
	Class D	0.9530	0.8970	0.9006					0.9930	1.0000	0.9934				
	Class E	0.9927	0.9800	0.9828					1.0000	0.9860	0.9926				
	Average	0.9670	0.9343	0.9345					0.9977	0.9953	0.9953				
Problem 4	Class A	0.9504	0.8420	0.8326	83.4	85	82	1.90	0.9729	0.8830	0.8897	92.6	94	92	1.10
	Class B	0.9586	0.8770	0.8719					0.9752	0.8540	0.8702				
	Class C	0.9165	0.8250	0.7743					0.9740	0.9250	0.9143				
	Class D	0.9441	0.6300	0.6788					0.9809	0.9710	0.9522				
	Class E	0.9884	0.9650	0.9634					0.9946	0.9800	0.9800				
	Average	0.9516	0.8278	0.8242					0.9795	0.9226	0.9213				

Table 16.8 – Summary of classification performance processed over segmented epochs, employing a 10-fold cross validation and 10 independent runs. Includes raw and normalized signals. Columns show the average Specificity, Recall, F-score and rank statistics of the accuracy, including median, best, worst and Interquartile Range (IQR).

(a) 4 features

		Raw							Normalized						
		Specificity	Recall	F-score	Accuracy				Specificity	Recall	F-score	Accuracy			
					media	best	worst	IQR				media	best	worst	IQR
Problem 1	Class A	0.9625	0.9648	0.9634	97.3	97.5	96.9	0.62	0.9260	0.9263	0.9259	93.5	93.7	93.1	0.60
	Class E	0.9648	0.9625	0.9638					0.9263	0.9260	0.9260				
	Average	0.9636	0.9636	0.9636					0.9261	0.9261	0.9260				
Problem 2	Class A,B,C,D	0.8658	0.9781	0.9724	95.5	95.8	95.2	0.00	0.7580	0.9549	0.9476	91.6	92	91	0.50
	Class E	0.9781	0.8658	0.8863					0.9549	0.7580	0.7817				
	Average	0.9219	0.9219	0.9294					0.8564	0.8564	0.8647				
Problem 3	Class A	0.8866	0.8095	0.7977	83.8	84.6	83.3	0.83	0.9018	0.8320	0.8265	83.7	84.6	82.9	0.83
	Class D	0.9033	0.7663	0.7842					0.9076	0.7915	0.8062				
	Class E	0.9547	0.9420	0.9338					0.9100	0.8630	0.8531				
	Average	0.9149	0.8393	0.8386					0.9064	0.8288	0.8286				
Problem 4	Class A	0.8595	0.6643	0.6312	67.1	67.8	66.5	0.50	0.8070	0.4650	0.4605	56.4	57.2	55.5	0.75
	Class B	0.8933	0.6745	0.6777					0.8025	0.4130	0.4122				
	Class C	0.8743	0.6210	0.6172					0.8383	0.4888	0.5043				
	Class D	0.8765	0.4725	0.5020					0.8460	0.6280	0.6194				
	Class E	0.9522	0.8973	0.8885					0.9001	0.7998	0.7913				
	Average	0.8911	0.6659	0.6633					0.8388	0.5589	0.5575				

(b) 9 features (Automatic)

Problem 1	Class A	1.0000	1.0000	1.0000	100	100	100	0.00	1.0000	1.0000	1.0000	100	100	100	0.00
	Class E	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
	Average	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
Problem 2	Class A,B,C,D	0.9685	0.9956	0.9939	99	99.2	99	0.00	0.9780	0.9979	0.9962	99.5	99.5	99.2	0.00
	Class E	0.9956	0.9685	0.9752					0.9979	0.9780	0.9845				
	Average	0.9821	0.9821	0.9845					0.9879	0.9879	0.9904				
Problem 3	Class A	0.9476	0.9600	0.9307	94.3	95	94.2	0.42	1.0000	0.9998	0.9999	100	100	100	0.00
	Class D	0.9751	0.8815	0.9123					0.9999	1.0000	0.9999				
	Class E	0.9918	0.9903	0.9876					1.0000	1.0000	1.0000				
	Average	0.9715	0.9439	0.9435					1.0000	0.9999	0.9999				
Problem 4	Class A	0.9367	0.8930	0.8447	83.2	83.8	82.8	0.50	0.9926	0.9530	0.9617	97.6	97.8	97.5	0.25
	Class B	0.9676	0.8593	0.8711					0.9869	0.9683	0.9584				
	Class C	0.9303	0.7868	0.7708					0.9953	0.9723	0.9769				
	Class D	0.9400	0.6570	0.6968					0.9942	0.9840	0.9807				
	Class E	0.9945	0.9743	0.9781					0.9980	0.9928	0.9925				
	Average	0.9538	0.8341	0.8323					0.9934	0.9741	0.9740				

(c) 10 features

Problem 1	Class A	1.0000	1.0000	1.0000	100	100	100	0.00	1.0000	1.0000	1.0000	100	100	100	0.00
	Class E	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
	Average	1.0000	1.0000	1.0000					1.0000	1.0000	1.0000				
Problem 2	Class A,B,C,D	0.9700	0.9930	0.9928	98.8	99	98.5	0.25	0.9783	0.9983	0.9964	99.5	99.5	99.5	0.00
	Class E	0.9930	0.9700	0.9709					0.9983	0.9783	0.9854				
	Average	0.9815	0.9815	0.9818					0.9883	0.9883	0.9909				
Problem 3	Class A	0.9424	0.9655	0.9295	93.8	94.2	93.3	0.83	0.9970	0.9950	0.9945	99.8	100	99.6	0.42
	Class D	0.9743	0.8658	0.9023					0.9975	0.9940	0.9945				
	Class E	0.9884	0.9830	0.9809					1.0000	1.0000	1.0000				
	Average	0.9683	0.9381	0.9375					0.9982	0.9963	0.9963				
Problem 4	Class A	0.9378	0.8728	0.8360	82.8	83.5	82.5	0.50	0.9908	0.9398	0.9511	96.9	97.2	96.5	0.25
	Class B	0.9639	0.8480	0.8589					0.9827	0.9620	0.9478				
	Class C	0.9291	0.7810	0.7657					0.9927	0.9710	0.9711				
	Class D	0.9362	0.6608	0.6961					0.9929	0.9785	0.9756				
	Class E	0.9903	0.9728	0.9714					0.9984	0.9823	0.9880				
	Average	0.9515	0.8271	0.8256					0.9915	0.9667	0.9667				

16.5 Discussion

The main contribution of the proposed methodology is the proposed feature set, with Table 16.6 providing some interesting insights. Among all features only the mean of the epoch $\mu(\mathbf{E})$ suffers from poor discrimination properties. Indeed, the variance of the average amplitude of the raw signals is quite similar between epochs, thus it is hard to determine a statistically significant difference between the classes. Although the median is related to the mean, the latter is quite sensitive to outliers while the former is more robust, thus the difference in their corresponding p-value. There are also similarities in the skewness feature among the classes, although it is not statistically significant. Furthermore, there seems to be strong agreement between these results and the GA-based feature selection described in Section 16.3.6. Both the ANOVA test and the meta-heuristic feature selection confirm that 8 of the 10 features are relevant to the classification task, partially disagreeing on only two cases ($\mu(\mathbf{E})$ and $\gamma_2(\mathbf{E})$).

Turning to the classification results summarized in Table 16.7 and Table 16.8 (with statistical comparisons given in Table 16.9) it is clear that Problem 1 is the easiest and Problem 4 is the hardest, as expected. Indeed, for Problems 1-3, at least one configuration achieves perfect accuracy on the test set, while the best performance on Problem 4 is 97.6% accurate. Moreover, these results seem to depend on several factors. For instance, in most cases the full set of 10 features is required to achieve the best performance, with the automatic set of 9 features being the next best option. In all cases, except when using the 4 feature set, performing a pre-processing (normalizing) step is also necessary to achieve the best results.

An interesting result was the performance of the segmented approach. If we compare each row of Table 16.7 with each row of Table 16.8, particularly focusing on the normalized data, it is evident that the best results are achieved when we use the full epochs. This is consistent with the idea that with larger epochs the feature extraction process becomes more robust, and with a better ability to extract general properties from each class. There is one exception however, for Problem 4 the best performance is achieved using the segmented approach and the 9 feature set. It seems that the classifier benefits from having a larger amount of examples in this case.

To further validate our work, we take one of the best configurations (9 feature set, full length epochs and pre-processing) and compare it (informally) in Tables 16.10, 16.11, 16.12 and 16.13 with state-of-the-art results reported for the Bonn data set. In the case of Problem 4, we use the segmented variant that produced the best performance. These tables present the authors, the feature extraction approach, whether or not feature post-processing was performed, the classifier used, the experimental training and testing configurations, the number of features employed, and their performance based on average accuracy, recall and specificity. For a fair comparison, only works that explicitly reported all of the above configurations and performance measures are considered.

This comparison is quite favorable for the methodology described in our current work. First, for problems 1, 2 and 3 we achieve perfect performance based on classification accuracy, matching or surpassing all other previous works. Second, the performance on Problem 4 is quite competitive, though slightly outperformed by some works [Guler et Ubeyli, 2007; Güler et Ubeyli, 2005; Übeyli et Güler, 2007]. However, as noted in Tables 16.10, 16.11, 16.12 and 16.13, the experimental setup (regarding training and testing partitions) is sometimes not fully given [Ahammad et collab., 2014; Guler et Ubeyli, 2007; Güler et Ubeyli, 2005; Guler et collab., 2005; Guo et collab., 2010; Kamath, 2015; Lima et collab., 2010; Übeyli et Güler, 2007], making the relevance of the performance differences less clear in those cases. Furthermore, on Problem 4 our method uses a significantly smaller feature set (9) compared to other works summarized in Table 16.13, none of them use less than 20 features.

16.6 Summary and Conclusions

This chapter proposes a new feature extraction approach for the classification of EEG signals and the detection of epileptic seizures. The detection depends on the ability of the system to recognize seizures among other EEG recordings. The proposed system performs feature extraction using the MP algorithm and Hölderian regularity analysis. This work is the first to combine both methods to effectively extract meaningful traits from epileptic events that are captured by EEG recordings.

Results show that regularity analysis based on the Hölder exponent was able to capture enough information to make a local characterization of the signals and build useful features. One important characteristic of regularity based analysis is its invariance to signal scaling or amplitude changes, which makes it ideal for EEG processing given the noisy conditions. On the other hand, the MP decomposition provided a more global characterization of the analyzed signals, by building features over the obtained set of waveforms or so called atoms. Certainly, the feature extraction process was shown to be sufficiently informative to be able to simplify the classification problem and achieve state-of-the-art performance.

The experimental work allows us to conclude the following regarding the proposed approach. First, the preprocessing step produces statistically significant performance improvements. Second, using only MP decomposition and Hölderian regularity provided strong performance on all problems, but to achieve optimal accuracy additional information was required. In particular, statistical time-domain features provided sufficient information to solve most problems with state-of-the-art results, that compared favorably with recently published works. Third, the set of proposed features provides the necessary discriminant information to solve the studied problems, with both a statistical analysis and a combinatorial feature selection algorithm suggesting that almost all features are relevant to the epilepsy detection problem in EEG signals. Indeed, on the most difficult classification task (considering five classes), our results achieve close to perfect accuracy (97.6%), while using less than half of the total features used by other approaches that achieve similar performance.

One important aspect to consider for future work is to reduce the computational cost of the feature extraction process, for online or real time implementations. However, there are several ways to cut corners and produce a more efficient method. Hölderian regularity computation can be simplified substantially by relaxing the approximation accuracy and imposing a smaller analyzing window, as done in other works. Alternatively, accurate and efficient approximations can be obtained by using meta-heuristic methods (see Chapter 12 and [TRUJILLO et collab., 2012]), which the computational costs are dramatically reduced. MP can also be improved by using a pre-built dictionary employing an optimized family of functions according with a subset of already recorded signals; such task can be performed offline. Improvements of both techniques would assure an online implementation of the proposed approach.

Future work will also focus on feature post-processing, which might further improve classification accuracy. The post-processing might be useful to fully exploit the decorrelation hidden in the extracted features. Additional pre-processing techniques might also be effective in boosting performance; e.g., Hölderian regularity has proved to be a powerful tool for denoising signals [LEGRAND et VEHEL, 2003]. Moreover, future work will also explore other recently proposed classification algorithms [MUÑOZ et collab., 2015], and apply the proposed feature extraction methods to other problems that require automatic EEG analysis [VÉZARD et collab., 2015]. Finally, it is our intention to implement our methodology as part of a broader computer aided diagnosis tool, helping physicians to properly diagnose, monitor, treat and hopefully classify and predict epileptic seizures.

Table 16.9 – Friedman pairwise tests, showing adjusted p-values with the Benjamini-Hochberg correction; bold values indicate that the null hypothesis is rejected at the $\alpha = 0.05$ significance level.

(a) Problem 1

			Full						Segmented					
			Raw			Normalized			Raw			Normalized		
		# features	4	9	10	4	9	10	4	9	10	4	9	10
Full	Raw	4	-	**0.0000**	**0.0000**	**0.0072**	**0.0000**	**0.0000**	**0.0000**	**0.0029**	**0.0189**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	1.0000	**0.0000**	1.0000	1.0000	**0.0000**	**0.0189**	**0.0042**	**0.0000**	1.0000	1.0000
		10	-	-	-	**0.0000**	1.0000	1.0000	**0.0000**	**0.0189**	**0.0042**	**0.0000**	1.0000	1.0000
	Norm.	4	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	1.0000	**0.0000**	**0.0189**	**0.0042**	**0.0000**	1.0000	1.0000
		10	-	-	-	-	-	-	**0.0000**	**0.0189**	**0.0042**	**0.0000**	1.0000	1.0000
Segmented	Raw	4	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	0.0761	**0.0000**	**0.0189**	**0.0189**
		10	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0042**	**0.0042**
	Norm.	4	-	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	-	-	-	1.0000
		10	-	-	-	-	-	-	-	-	-	-	-	-

(b) Problem 2

			Full						Segmented					
			Raw			Normalized			Raw			Normalized		
		# features	4	9	10	4	9	10	4	9	10	4	9	10
Full	Raw	4	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2613	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	0.0577	**0.0000**	0.3139	0.8400	**0.0000**	0.4061	0.3502	**0.0000**	1.0000	0.8400
		10	-	-	-	**0.0000**	**0.0005**	**0.0023**	**0.0000**	0.5234	0.4600	**0.0000**	0.1089	0.0644
	Norm.	4	-	-	-	-	**0.0000**	**0.0000**	**0.0002**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	0.3091	**0.0000**	0.6431	**0.0022**	**0.0000**	0.3595	0.5492
		10	-	-	-	-	-	-	**0.0000**	0.2299	**0.0304**	**0.0000**	0.5340	0.8400
Segmented	Raw	4	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	**0.0034**	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**
	Norm.	4	-	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	-	-	-	0.2132
		10	-	-	-	-	-	-	-	-	-	-	-	-

(c) Problem 3

			Full						Segmented					
			Raw			Normalized			Raw			Normalized		
		# features	4	9	10	4	9	10	4	9	10	4	9	10
Full	Raw	4	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2141	**0.0000**	**0.0000**	**0.0009**	**0.0000**	**0.0000**
		9	-	-	**0.0036**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
	Norm.	4	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0001**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	0.5316	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0003**	0.0934
		10	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0309**	**0.0001**
Segmented	Raw	4	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0005**	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	**0.0435**	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**
	Norm.	4	-	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	-	-	-	**0.0000**
		10	-	-	-	-	-	-	-	-	-	-	-	-

(d) Problem 4

			Full						Segmented					
			Raw			Normalized			Raw			Normalized		
		# features	4	9	10	4	9	10	4	9	10	4	9	10
Full	Raw	4	-	**0.0000**	**0.0000**	0.5966	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	**0.0310**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.1111	0.8383	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0068**	**0.0079**	**0.0000**	**0.0000**	**0.0000**
	Norm.	4	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Segmented	Raw	4	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	0.1735	**0.0000**	**0.0000**	**0.0000**
		10	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**	**0.0000**
	Norm.	4	-	-	-	-	-	-	-	-	-	-	**0.0000**	**0.0000**
		9	-	-	-	-	-	-	-	-	-	-	-	**0.0079**
		10	-	-	-	-	-	-	-	-	-	-	-	-

Table 16.10 – Comparison of classification performance among several published approaches on problem 1, including our work. Accuracy values are shown as percentile. Best value is shown in bold. Compared works are sorted by their accuracy.

	Authors	Feature selection	Feature post-processing	Classifier	Training/Testing partition	Number of features	Accuracy	Recall	Specificity
Problem 1	Nicolaou et Georgiou [2012]	Permutation Entropy	No	SVM	60/40 random, 100 times	69-92	93.55	-	-
	Tzallas et collab. [2009]	Time Frequency analysis	PCA	ANN	50/50 random, 10 times holdout	16	94.3	100	100
	Divya [2015]	Empirical Mode Decomposition	No	ANN	[a]	15	96.88	93.3	100
	Zainuddin et collab. [2013]	DWT	No	Wavelet Neural Network	10-fold cross validation	20	98.87	94.96	99.43
	Murugavel et Ramakrishnan [2014]	Wavelet	Entropy	Optimized Extreme Learning Machine	50/50 cross fold	30	99	93.2	98.9
	Kovacs et collab. [2014]	Discrete Short Time Fourier Transform	No	Bagged Alternating Decision Trees	75/25	483	99.7	-	-
	Guo et collab. [2010]	Multiwavelet	Entropy	MLPNN	50/50 [a]	10	99.8	100	99.2
	Tzallas et collab. [2007]	Time Frequency analysis	PCA	ANN	50/50 random, 10 times holdout	13-66	**100**	100	100
	Lima et collab. [2010]	Wavelet	No	LS-SVM	50/50 [a]	5	**100**	100	100
	Kumari et Prabin [2011]	DWT	No	SVM	80/20	15	**100**	-	-
	Gandhi et collab. [2012]	DWT	Discrete Harmony Search	Probabilistic Neural Network	10-fold cross validation	45	**100**	-	-
	Alam et Bhuiyan [2013]	Empirical Mode Decomposition	No	ANN	60/35 + 5 of validation partition	48	**100**	100	100
	Kumar et collab. [2014]	DWT	Fuzzy Approximate Entropy	SVM	50/50 random, 10 times hold-out	6	**100**	100	100
	Chen [2014]	DTCWT	No	1-NN	50/50	2	**100**	-	-
	Kamath [2015]	Hilbert Transform Scatter Plot	No	Statistical analysis	[a]	>1000	**100**	100	100
	This work	Regularity & MP	No	RF	10-fold cross validation	9	**100**	100	100

[a] There is missing information of how the train/test partition was done or other details in the experimental setup.

Table 16.11 – Comparison of classification performance among several published approaches on problem 2, including our work. Accuracy values are shown as percentile. Best value is shown in bold. Compared works are sorted by their accuracy.

	Authors	Feature selection	Feature post-processing	Classifier	Training/Testing partition	Number of features	Accuracy	Recall	Specificity
Problem 2	NICOLAOU et GEORGIOU [2012]	Permutation Entropy	No	SVM	60/40 random, 100 times	69-92	86.1	-	-
	KOVACS et collab. [2014]	Discrete Short Time Fourier Transform	No	Bagged Alternating Decision Trees	75/25	483	96.7	-	-
	KUMAR et collab. [2014]	DWT	Fuzzy Approximate Entropy	SVM	50/50 random, 10 times hold-out	6	97.36	98.3	93.5
	GUO et collab. [2010]	Multiwavelet	Entropy	MLPNN	50/50 [a]	10	98.3	99	95.5
	MURUGAVEL et RAMAKRISHNAN [2014]	Wavelet	Entropy	Optimized Extreme Learning Machine	50/50 cross fold	30	99	-	-
	DIVYA [2015]	Empirical Mode Decomposition	No	ANN	[a]	15	99	93.3	100
	ORHAN et collab. [2011]	DWT	K-means	ANN	cross validation, 5000 times	18-56	99.6	100	98
	ALAM et BHUIYAN [2013]	Empirical Mode Decomposition	No	ANN	60/35 + 5 of validation partition	48	**100**	100	100
	CHEN [2014]	DTCWT	No	1-NN	50/50	2	**100**	-	-
	This work	Regularity & MP	No	RF	10-fold cross validation	9	**100**	100	100

[a] There is missing information of how the train/test partition was done or other details in the experimental setup.

Table 16.12 – Comparison of classification performance among several published approaches on problem 3, including our work. Accuracy values are shown as percentile. Best value is shown in bold. Compared works are sorted by their accuracy.

	Authors	Feature selection	Feature post-processing	Classifier	Training/Testing partition	Number of features	Accuracy	Recall	Specificity
Problem 3	AHAMMAD et collab. [2014]	Wavelet	No	Linear	62/38 [a]	18	84.2	83.5	85.6
	DIVYA [2015]	Empirical Mode Decomposition	No	ANN	[a]	15	93.9	97.1	89.1
	MARTIS et collab. [2013]	Higuchi Fractal Dimension, entropy over Intrinsic Time-Scale Decomposition	No	Decision Tree	[a]	6	95.67	99	99.5
	MURUGAVEL et RAMAKRISHNAN [2014]	Wavelet	Entropy	Optimized Extreme Learning Machine	50/50 cross fold	30	96	93.5	98.4
	ORHAN et collab. [2011]	DWT	K-means	ANN	random subsampling, cross validation, 5000 times & best model	18-56	96.7	93.5	98.3
	ACHARYA et collab. [2012b]	Entropies	No	Fuzzy	3-fold cross validation	4	98.1	99.4	100
	NIKNAZAR et collab. [2013]	DWT, Recurrence Quantification Analysis	Phase Space Reconstruction	Error-Correction Output Codes	70/30 random, 20 times	30	98.67	98.55	99.33
	RAJENDRA ACHARYA et collab. [2012]	Wavelet Packet Decomposition	PCA eigenvalues	Gaussian Mixture Model	10-fold cross validation	9	99	99	99
	ACHARYA et collab. [2012a]	Entropies, High Order Spectra, Hurst exponent, Higuchi Fractal Dimension	ANOVA filtering	Fuzzy	10-fold cross validation	7	99.7	100	100
	ALAM et BHUIYAN [2013]	Empirical Mode Decomposition	No	ANN	60/35 + 5 of validation partition	48	**100**	100	100
	This work	Regularity & MP	No	RF	10-fold cross validation	9	**100**	100	100

[a] There is missing information of how the train/test partition was done or other details in the experimental setup.

Table 16.13 – Comparison of classification performance among several published approaches on problem 4, including our work. Accuracy values are shown as percentile. Best value for each problem is shown in bold. Compared works are sorted by their accuracy.

	Authors	Feature selection	Feature post-processing	Classifier	Training/Testing partition	Number of features	Accuracy	Recall	Specificity
Problem 4	NUNES et collab. [2014]	DWT	Relief, InfoGain	Optimum Path Forest	10-fold cross validation	<40	89.2	-	-
	MURUGAVEL et RAMAKRISHNAN [2014]	Wavelet	Entropy	Optimized Extreme Learning Machine	50/50 cross fold	30	94	93.6	98.3
	ÜBEYLI et GÜLER [2007]	Eigenvector method	No	Modified of Mixture of expert model	50/50 [a]	64	98.6	98.6	99.6
	GÜLER et UBEYLI [2005]	Wavelet	No	ANFIS	50/50 [a]	20	98.7	98.7	99.7
	GULER et UBEYLI [2007]	Wavelet, Lyapunov exponents	No	SVM	50/50 [a]	24	**99.3**	99.3	99.8
	This work	Regularity & MP	No	RF	10-fold cross validation	9	97.6	97.8	97.5

[a] There is missing information of how the train/test partition was done or other details in the experimental setup.

Acknowledgment

Funding for this work was provided by CONACYT Basic Science Research Project No. 178323, DGEST (México) Research Project 5414.14-P, and FP7-PEOPLE-2013-IRSES project ACOBSEC financed by the European Commission with contract No. 612689.

References

ABÁSOLO, D., R. HORNERO, P. ESPINO, J. POZA, C. I. SÁNCHEZ et R. DE LA ROSA. 2005, «Analysis of regularity in the {EEG} background activity of alzheimer's disease patients with approximate entropy», *Clinical Neurophysiology*, vol. 116, nº 8, p. 1826–1834. 362

ACHARYA, U., H. FUJITA, V. SUDARSHAN et S. BHAT. 2015, «Application of entropies for automated diagnosis of epilepsy using eeg signals: A review», *Knowledge-Based Systems*, vol. 88, p. 85–96. 361

ACHARYA, U., S. SREE, P. ANG, R. YANTI et J. SURI. 2012a, «Application of non-linear and wavelet based features for the automated identification of epileptic eeg signals», *International Journal of Neural Systems*, vol. 22, nº 02, p. 1–12. 361, 383

ACHARYA, U. R., F. MOLINARI, S. V. SREE, S. CHATTOPADHYAY, K.-H. NG et J. S. SURI. 2012b, «Automated diagnosis of epileptic EEG using entropies», *Biomedical Signal Processing and Control*, vol. 7, nº 4, p. 401–408. 358, 359, 360, 361, 383

ACHARYA, U. R., S. VINITHA SREE, G. SWAPNA, R. J. MARTIS et J. S. SURI. 2013, «Automated EEG analysis of epilepsy: A review», *Knowledge-Based Systems*, vol. 45, p. 147–165. 360, 369

AHAMMAD, N., T. FATHIMA et P. JOSEPH. 2014, «Detection of epileptic seizure event and onset using EEG.», *BioMed research international*, vol. 2014. 358, 359, 360, 378, 383

ALAM, S. et M. BHUIYAN. 2013, «Detection of seizure and epilepsy using higher order statistics in the emd domain», *IEEE Journal of Biomedical and Health Informatics*, vol. 17, nº 2, p. 312–318. 361, 381, 382, 383

ALPAYDIN, E. 2010, *Introduction to Machine Learning*, 2e éd., The MIT Press, ISBN 026201243X, 9780262012430. 368

ANDRZEJAK, R. G., K. LEHNERTZ, F. MORMANN, C. RIEKE, P. DAVID et C. E. ELGER. 2001, «Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state.», *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 64, nº 6 Pt 1. 360, 361

BAJAJ, V. et R. PACHORI. 2012, «EEG signal classification using empirical mode decomposition and support vector machine», *Proceedings of the International Conference on Soft Computing*, p. 581–592. 361

BALL, T., M. KERN, I. MUTSCHLER, A. AERTSEN et A. SCHULZE-BONHAGE. 2009, «Signal quality of simultaneously recorded invasive and non-invasive EEG», *NeuroImage*, vol. 46, nº 3, p. 708–716. 358, 361

BÉNAR, C. G., T. PAPADOPOULO, B. TORRÉSANI et M. CLERC. 2009, «Consensus Matching Pursuit for multi-trial EEG signals», *Journal of Neuroscience Methods*, vol. 180, nº 1, p. 161–170. 362

BREIMAN, L. 2001, «Random forests», *Machine learning*, p. 5–32. 368, 369

CHEN, G. 2014, «Automatic eeg seizure detection using dual-tree complex wavelet-fourier features», *Expert Systems with Applications*, vol. 41, nº 5, p. 2391–2394. 381, 382

CHEN, S., D. DONOHO et M. SAUNDERS. 1998, «Atomic decomposition by basis pursuit», *SIAM journal on scientific computing*, vol. 43, nº 1, p. 129–159. 365

CHEN, W., Y. WANG, G. CAO, G. CHEN et Q. GU. 2014, «A random forest model based classification scheme for neonatal amplitude-integrated EEG», *BioMedical Engineering OnLine*, vol. 13, nº Suppl 2, p. 1–13. 360, 369

CHUCKRAVANEN, D. 2014, «Approximate Entropy as a Measure of Cognitive Fatigue: An EEG Pilot Study», *International Journal of Emerging Trends in Science and Technology*, p. 1036–1042. 362

DICK, O. E. et I. A. SVYATOGOR. 2012, «Potentialities of the wavelet and multifractal techniques to evaluate changes in the functional state of the human brain», *Neurocomputing*, vol. 82, p. 207–215. 362

DIVYA, S. 2015, «Classification of eeg signal for epileptic seizure detection using emd and elm», *International Journal for Trends in Engineering and Technology*, vol. 3, nº 2, p. 68–74. 359, 361, 381, 382, 383

DUDA, R. O., P. E. HART et D. G. STORK. 2000, *Pattern Classification (2Nd Edition)*, Wiley-Interscience, ISBN 0471056693. 368

DURKA, P. 2004, «Adaptive time-frequency parametrization of epileptic spikes», *Physical Review E*. 362

DURKA, P. et K. BLINOWSKA. 1995, «Analysis of eeg transients by means of matching pursuit», *Annals of Biomedical Engineering*, vol. 23, nº 5, p. 608–611. 362

DURKA, P., D. IRCHA et K. BLINOWSKA. 2001, «Stochastic time-frequency dictionaries for matching pursuit», *IEEE Transactions on Signal Processing*, vol. 49, nº 3, p. 507–510. 362

DURKA, P. J., A. MATYSIAK, E. M. MONTES, P. V. SOSA et K. J. BLINOWSKA. 2005, «Multichannel matching pursuit and EEG inverse solutions», *Journal of Neuroscience Methods*, vol. 148, nº 1, p. 49–59. 362

EADIE, M. J. 2012, «Shortcomings in the current treatment of epilepsy», *Expert Review of Neurotherapeutics*, vol. 12, nº 12, p. 1419–1427. 358

EIBEN, A. et J. SMITH. 2015, *Introduction to Evolutionary Computing, 2nd Edition*, Springer Verlag. 368

ELSE, T. et G. D. HAMMER. 2013, *Disorders of the Adrenal Cortex*, McGraw-Hill Education, New York, NY. 358

FAN, S., J. YEH, B. CHEN et J. SHIEH. 2011, «Comparison of eeg approximate entropy and complexity measures of depth of anaesthesia during inhalational general anaesthesia», *Journal of Medical and Biological Engineering*, vol. 31, nº 5, p. 359–366. 362

FAUST, O., U. ACHARIA, H. ADELI et A. ADELI. 2015, «Wavelet-based eeg processing for computer-aided seizure detection and epilepsy diagnosis», *Seizure*, vol. 26, p. 56–64. 360

FIGLIOLA, A. et E. SERRANO. 2007, «Study of EEG brain maturation signals with multifractal detrended fluctuation analysis», *XV Conference on Nonequilibrium Statistical Mechanics and Nonlinear Physics*, , nº 6 mm, p. 190–195. 362

FISHER, R. S., C. ACEVEDO, A. ARZIMANOGLOU, A. BOGACZ, J. H. CROSS, C. E. ELGER, J. ENGEL, L. FORSGREN, J. A. FRENCH, M. GLYNN, D. C. HESDORFFER, B. I. LEE, G. W. MATHERN, S. L. MOSHÉ, E. PERUCCA, I. E. SCHEFFER, T. TOMSON, M. WATANABE et S. WIEBE. 2014, «ILAE official report: a practical clinical definition of epilepsy.», *Epilepsia*, vol. 55, nº 4, p. 475–82. 358

FISHER, R. S. et S. C. SCHACHTER. 2000, «The postictal state: a neglected entity in the management of epilepsy.», *Epilepsy & Behavior*, vol. 1, nº 1, p. 52–59. 360

FRANASZCZUK, P. J., G. K. BERGEY, P. J. DURKA et H. M. EISENBERG. 1998, «Time-frequency analysis using the matching pursuit algorithm applied to seizures originating from the mesial temporal lobe», *Electroencephalography and Clinical Neurophysiology*, vol. 106, nº 6, p. 513–521. 359, 362

GANDHI, T., P. CHAKRABORTY, G. ROY et B. PANIGRAHI. 2012, «Discrete harmony search based expert model for epileptic seizure detection in electroencephalography», *Expert Systems with Applications*, vol. 39, nº 4, p. 4055–4062. 360, 381

GOLDBERG, D. E. 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1re éd., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, ISBN 0201157675. 368

GRIZOU, J., I. ITURRATE, L. MONTESANO, P.-Y. OUDEYER et M. LOPES. 2014, «Calibration-Free BCI Based Control», dans *Twenty-Eighth AAAI Conference on Artificial Intelligence*, Quebec, Canada, p. 1–8. 366

GULER, I. et E. UBEYLI. 2007, «Multiclass Support Vector Machines for EEG-Signals Classification», *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, nº 2, p. 117–126. 359, 378, 384

GÜLER, I. et E. D. UBEYLI. 2005, «Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients.», *Journal of neuroscience methods*, vol. 148, p. 113–121. 358, 359, 369, 378, 384

GULER, N., E. UBEYLI et I. GULER. 2005, «Recurrent neural networks employing Lyapunov exponents for EEG signals classification», *Expert Systems with Applications*, vol. 29, nº 3, p. 506–514. 358, 359, 361, 378

GUO, L., D. RIVERO et A. PAZOS. 2010, «Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks.», *Journal of neuroscience methods*, vol. 193, nº 1, p. 156–63. 358, 359, 360, 378, 381, 382

HIRTZ, D., D. J. THURMAN, K. GWINN-HARDY, M. MOHAMED, A. R. CHAUDHURI et R. ZALUTSKY. 2007, «How common are the "common" neurologic disorders?», *Neurology*, vol. 68, nº 5, p. 326–337. 358

HUSSAIN, Z. et J. SHAWE-TAYLOR. 2009, «Theory of matching pursuit», *Advances in Neural Information Processing Systems*, p. 1–8. 365

JAFFARD, S. 2004, «Wavelet techniques in multifractal analysis», *Proceedings of symposia in pure mathematics*. 363, 364

JAFFARD, S. et Y. MEYER. 1996, *Wavelet methods for pointwise regularity and local oscillations of functions*, Providence, R.I. : American Mathematical Society. 362, 364

JAIANTILAL, A. 2012, *RF Matlab interface, Version 0.02*. URL https://code.google.com/p/randomforest-matlab/. 370

KAMATH, C. 2015, «Analysis of EEG Dynamics in Epileptic Patients and Healthy Subjects Using Hilbert Transform Scatter Plots», *OALib*, vol. 02, p. 1–14. 358, 359, 360, 378, 381

KANTELHARDT, J. W., S. A. ZSCHIEGNER, E. KOSCIELNY-BUNDE, S. HAVLIN, A. BUNDE et H. E. STANLEY. 2002, «Multifractal detrended fluctuation analysis of nonstationary time series», *Physica A: Statistical Mechanics and its Applications*, vol. 316, nº 1-4, p. 87–114. 362

KOHSAKA, S., S. MIZUKAMI, M. KOHSAKA, H. SHIRAISHI et K. KOBAYASHI. 2002, «Widespread activation of the brainstem preceding the recruiting rhythm in human epilepsies», *Neuroscience*, vol. 115, nº 3, p. 697–706. 359

KOUBEISSI, M. Z., C. C. JOUNY, J. O. BLAKELEY et G. K. BERGEY. 2009, «Analysis of dynamics and propagation of parietal cingulate seizures with secondary mesial temporal involvement», *Epilepsy and Behavior*, vol. 14, nº 1, p. 108–112. 367

KOVACS, P., K. SAMIEE et M. GABBOUJ. 2014, «On application of rational discrete short time fourier transform in epileptic seizure classification», *Acoustics, Speech and Signal*, p. 5880–5884. 360, 381, 382

KUMAR, Y., M. DEWAL et R. ANAND. 2014, «Epileptic seizure detection using dwt based fuzzy approximate entropy and support vector machine», *Neurocomputing*, vol. 133, doi: 10.1016/j.neucom.2013.11.009, p. 271–279. 359, 360, 381, 382

KUMARI, S. et J. PRABIN. 2011, «Seizure detection in eeg using time frequency analysis and svm», *2011 International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, p. 626–630. 360, 381

LEGRAND, P. 2004, *Débruitage et interpolation par analyse de la régularité Hölderienne. Application à la modélisation du frottement pneumatique-chaussée.*, Theses, Ecole Centrale de Nantes, Université de Nantes. URL https://tel.archives-ouvertes.fr/tel-00643450. 365

LEGRAND, P. et J. VEHEL. 2003, «Local regularity-based image denoising», *Proceedings 2003 International Conference on Image Processing*, vol. 3, nº 1, p. 0–3. 379

LIMA, C. A. M., A. L. V. COELHO et M. EISENCRAFT. 2010, «Tackling EEG signal classification with least squares support vector machines: A sensitivity analysis study», *Computers in Biology and Medicine*, vol. 40, p. 705–714. 358, 359, 378, 381

LIN, C., H. CHEN et Y. WU. 2014, «Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection», *Expert Systems with Applications*, vol. 41, nº 15, p. 6611–6621. 368

MA, Q., X. NING, J. WANG et J. LI. 2006, «Sleep-stage characterization by nonlinear EEG analysis using wavelet-based multifractal formalism», *IEEE Engineering in Medicine and Biology Society*, p. 4526–4529. 362

MALLAT, S. 1993, «Matching pursuits with time-frequency dictionaries», *IEEE Transactions on Signal Processing*, vol. 41, nº 12, p. 3397–3415. 359, 362, 365

MALLAT, S. 2008, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3e éd., Academic Press, ISBN 0123743702, 9780123743701. 363

MALLAT, S. et W. L. HWANG. 1992, «Singularity detection and processing with wavelets», *IEEE Transactions on Information Theory*, vol. 38, nº 2 pt II, p. 617–643. 359

MARTIS, R., U. ACHARYA, J. TAN, A. PETZNICK, L. TONG, C. CHUA et E. NG. 2013, «Application of intrinsic time-scale decomposition (itd) to eeg signals for automated seizure prediction», *International Journal of Neural Systems*, vol. 23, nº 05, p. 1–13. 361, 383

MATHUVANESAN, C. et T. JAYASANKAR. 2013, «Performance Analysis Of Singularity And Irregular Detection In Human Health Monitoring Using Lipschitz Exponent Function», vol. 2, nº 6, p. 414–418. 362

MATLAB. 2014, *version 8.3 (R2014a)*, The MathWorks Inc., Natick, Massachusetts. 369

MIKAILI, M. et S. M. R. H. GOLPAYEGANI. 2002, «Assessment of the complexity/regularity of transient brain waves (eeg) during sleep, based on wavelet theory and the concept of entropy», *Iranian Journal of Science and Technology*, vol. 26, nº B4, p. 639–646. 362

MUÑOZ, L., S. SILVA et L. TRUJILLO. 2015, «M3gp multiclass classification with gp», dans *Genetic Programming*, *Lecture Notes in Computer Science*, vol. 9025, Springer International Publishing, p. 78–91. 379

MURUGAVEL, A. et S. RAMAKRISHNAN. 2014, «An optimized extreme learning machine for epileptic seizure detection», *IAENG International Journal of Computer Science*, vol. 41, p. 212–221. 359, 360, 381, 382, 383, 384

NATARAJAN, K., R. ACHARYA U, F. ALIAS, T. TIBOLENG et S. K. PUTHUSSERYPADY. 2004, «Nonlinear analysis of EEG signals at different mental states», *Biomedical engineering online*, vol. 3, nº 1, p. 1–11. 362

NICOLAOU, N. et J. GEORGIOU. 2012, «Detection of epileptic electroencephalogram based on permutation entropy and support vector machines», *Expert Systems with Applications*, vol. 39, nº 1, p. 202–209. 361, 381, 382

NIKNAZAR, M., S. MOUSAVI, B. VAHDAT et M. SAYYAH. 2013, «A new framework based on recurrence quantification analysis for epileptic seizure detection», *IEEE journal of biomedical and health informatics*, vol. 17, nº 3, p. 572–8. 361, 383

NUNES, T., A. COELHO, C. LIMA, J. PAPA et V. DE ALBUQUERQUE. 2014, «Eeg signal classification for epilepsy diagnosis via optimum path forest - a systematic assessment», *Neurocomputing*, vol. 136, p. 103–123. 360, 384

ORHAN, U., M. HEKIM et M. OZER. 2011, «EEG signals classification using the K-means clustering and a multilayer perceptron neural network model», *Expert Systems with Applications*, vol. 38, nº 10, p. 13 475–13 481. 358, 359, 360, 382, 383

PICOT, A., H. WHITMORE et F. CHAPOTOT. 2012, «Detection of cortical slow waves in the sleep EEG using a modified matching pursuit method with a restricted dictionary», *IEEE Transactions on Biomedical Engineering*, vol. 59, nº 10, p. 2808–2817. 362

POPIVANOV, D., V. STOMONYAKOV, Z. MINCHEV, S. JIVKOVA, P. DOJNOV, S. JIVKOV, E. CHRISTOVA et S. KOSEV. 2006, «Multifractality of decomposed EEG during imaginary and real visual-motor tracking», *Biological Cybernetics*, vol. 94, nº 2, p. 149–156. 362, 363

RAJENDRA ACHARYA, U., S. VINITHA SREE, A. P. C. ALVIN et J. S. SURI. 2012, «Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework», *Expert Systems with Applications*, vol. 39, nº 10, p. 9072–9078. 359, 360, 383

RAMGOPAL, S., S. THOME-SOUZA, M. JACKSON, N. E. KADISH, I. SÁNCHEZ FERNÁNDEZ, J. KLEHM, W. BOSL, C. REINSBERGER, S. SCHACHTER et T. LODDENKEMPER. 2014, «Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy», *Epilepsy & Behavior*, vol. 37, p. 291–307. 359

ROKACH, L. et O. MAIMON. 2008, *Data Mining with Decision Trees: Theroy and Applications*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, ISBN 9789812771711, 9812771719. 369

SCULLEY, D. 2011, «Results from a Semi-Supervised Feature Learning Competition», *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, p. 1–9. 369

SONG, I.-H. et D.-S. LEE. 2005, «Fluctuation Dynamics in Electroencephalogram Time Series», dans *First International Work-Conference on the Interplay Between Natural and Artificial Computation*, p. 195–202. 363

SOTELO, A., E. GUIJARRO, L. TRUJILLO, L. N. CORIA et Y. MARTÍNEZ. 2013, «Identification of epilepsy stages from ECoG using genetic programming classifiers», *Computers in Biology and Medicine*, vol. 43, nº 11, p. 1713–1723. 358, 359, 360, 367, 368

SOTELO, A., E. D. GUIJARRO et L. TRUJILLO. 2015, «Seizure states identification in experimental epilepsy using gabor atom analysis», *Journal of Neuroscience Methods*, vol. 241, p. 121–131. 358, 360

SUN, Z., G. BEBIS et R. MILLER. 2004, «Object detection using feature subset selection», *Pattern Recogn*, vol. 37, nº 11, p. 2165–2176. 368

THURMAN, D. J., E. BEGHI, C. E. BEGLEY, A. T. BERG, J. R. BUCHHALTER, D. DING, D. C. HESDORFFER, W. A. HAUSER, L. KAZIS, R. KOBAU, B. KRONER, D. LABINER, K. LIOW, G. LOGROSCINO, M. T. MEDINA, C. R. NEWTON, K. PARKO, A. PASCHAL, P.-M. PREUX, J. W. SANDER, A. SELASSIE, W. THEODORE, T. TOMSON et S. WIEBE. 2011, «Standards for epidemiologic studies and surveillance of epilepsy.», *Epilepsia*, vol. 52 Suppl 7, nº 1, p. 2–26. 358

TRICOT, C. 1995, *Curves and Fractal Dimension*, Springer-Verlag. 364

TRUJILLO, L., P. LEGRAND, G. OLAGUE et J. LÉVY-VÉHEL. 2012, «Evolving estimators of the pointwise Hölder exponent with Genetic Programming», *Information Sciences*. 379

TZALLAS, A. T., M. G. TSIPOURAS et D. I. FOTIADIS. 2007, «Automatic seizure detection based on time-frequency analysis and artificial neural networks.», *Computational intelligence and neuroscience*, vol. 2007. 359, 381

TZALLAS, A. T., M. G. TSIPOURAS et D. I. FOTIADIS. 2009, «Epileptic seizure detection in EEGs using time-frequency analysis.», *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 13, nº 5, p. 703–10, ISSN 1558-0032. 359, 360, 381

ÜBEYLI, E. D. et I. GÜLER. 2007, «Features extracted by eigenvector methods for detecting variability of EEG signals», *Pattern Recognition Letters*, vol. 28, p. 592–603. 359, 369, 378, 384

USAKLI, A. B. 2010, «Improvement of EEG signal acquisition: An electrical aspect for state of the Art of front end», *Computational Intelligence and Neuroscience*, vol. 2010. 366

VÉHEL, J. et P. LEGRAND. 2004, «Signal and Image processing with FracLab», *Thinking in Patterns, World Scientific*. 365

VEZARD, L., P. LEGRAND, M. CHAVENT, F. FAITA-AINSEBA, J. CLAUZEL et L. TRUJILLO. 2014, «Classification of EEG signals by evolutionary algorithm», dans *Advances in Knowledge Discovery and Management Volume 4*, *Studies in Computational Intelligence*, vol. 527, édité par F. Guillet, B. Pinaud, G. Venturini et D. Zighed, Springer, p. 133–153. 360

VÉZARD, L., P. LEGRAND, M. CHAVENT, F. FAÏTA-AÏNSEBA et L. TRUJILLO. 2015, «EEG classification for the detection of mental states», *Applied Soft Computing*, vol. 32, p. 113–131. 379

WU, S. C. et A. L. SWINDLEHURST. 2013, «Matching pursuit and source deflation for sparse EEG/MEG dipole moment estimation», *IEEE Transactions on Biomedical Engineering*, vol. 60, nº 8, p. 2280–2288. 362

XIE, S. et S. KRISHNAN. 2014, «Dynamic principal component analysis with nonoverlapping moving window and its applications to epileptic EEG classification.», *TheScientificWorldJournal*, vol. 2014, nº 1. 360

YADAV, R., A. K. SHAH, J. A. LOEB, M. N. S. SWAMY et R. AGARWAL. 2012, «Morphology-based automatic seizure detector for intracerebral EEG recordings», *IEEE Transactions on Biomedical Engineering*, vol. 59, nº 7, p. 1871–1881. 359

ZAINUDDIN, Z., L. HUONG et O. PAULINE. 2013, «Reliable epileptic seizure detection using an improved wavelet neural network», *The Australasian Medical Journal*, vol. 6, nº 5, p. 308–314. 360, 381

ZORICK, T. et M. A. MANDELKERN. 2013, «Multifractal Detrended Fluctuation Analysis of Human EEG: Preliminary Investigation and Comparison with the Wavelet Transform Modulus Maxima Technique», *PLoS ONE*, vol. 8, nº 7. 362

Part IV

Conclusions

Conclusions

This manuscript presents some advances in the field of artificial evolution. First, the building of models that can predict the performance of a GP-classifier without having to run the program or sample potential solutions in the research space is highlighted. Secondly, fitness case sampling methods were proposed and after an intensive comparison of various fitness case sampling methods, it appeared that the choice of fitness calculation method has an influence on the bloat, the calculation time and the quality of the result in terms of overfitting. Third, It has been shown that the Novelty Search can solve real and synthetic problems of classification, clustering and symbolic regression. In addition, a study on the bloat and research dynamics of GP-NS was conducted and two new high-performance versions of NS were proposed and tested. Fourth, it is shown that through the parameterization of trees (individuals) and local optimization, the search for the GP converges more quickly towards high quality solutions. As a first step, a parameter was simply added, a weight coefficient before each function of the set of functions (atoms available to build the tree). In a second step, we determined to which individuals and generations it was relevant to apply this local optimization. The results showed that the best strategy is to apply local optimization to all individuals or to a random sample of the best (in the sense of fitness) individuals in each generation. Finally, a hybrid methodology based on the RAndom SAmpling Consensus (RANSAC) algorithm and GP, which we call RANSAC-GP, is presented. This work presents the first application of RANSAC to symbolic regression with GP. The proposed algorithm is able to deal with extreme amounts of contamination in the training set, evolving highly accurate models even when the amount of outliers reaches 90%.

This manuscript also includes contributions in the field of signal regularity estimation. Apart from the two patents presented in this document (exclusively in the version submitted to the members of the jury), the following contributions were provided: First, an alternative version of the DFA: the CDFA (Continuous Detrended Fluctuation Analysis) has been invented. Secondly, a common mathematical formalization of the variants of the DFA has been proposed. Third, an interpretation of the fluctuation functions of the DFA and variants as the result of a filtering has been presented allowing to perform a filtering-based analysis comparing the DFA with the CDFA and other variants for Wide Sense Stationary Processes.

I hope that this document will have shown through Part III the interest of combining signal processing and artificial evolution tools to solve real problems. It is also important to note that effective resolution of real problems also requires a good understanding of the problem being addressed. This understanding is in the hands of the actors who are in the field and specialists in their area. Also, through these few lines, I would like to highlight the importance of exchanges with my collaborators doctors, industrialists, biologists, neurophysiologists, for the realization of the works, some of which are presented in Part III. On the other hand, the fundamental importance of the design of data acquisition protocols should not be overlooked. In particular, the acquisition of EEG signals is a difficult task, requiring accuracy and precision. In my opinion, without real signals, without good quality data acquisition, the field of signal processing and applied mathematics would lose some of their taste.

Perspectives

I hope to continue to be able to combine signal processing, fractal analysis and artificial evolution to solve real problems but also to make other theoretical contributions. Some of the themes presented in this document are still being worked on and articles are being published (On the topics related to Chapters 8, 10, 14, 15).

Nevertheless, I would like to briefly present in this section two projects that will soon start. This section provides an overview of these two upcoming projects. The first one is

related to the PhD thesis of **Jimmy Bondu** which will begin at the end of the year and the second is related to a collaboration with the hospital *Pellegrin* and the laboratory *EA 4136 Handicap Activity Cognition Santé*. As said in the introduction, choices had to be made and I apologize to the collaborators whose work I do not mention and who are building actually other interesting perspectives.

Presentation of the thesis topic: RADAR data processing by learning techniques

Supervision with **Audrey Giremus**, **Eric Grivel**, **Clément Magnant** and **Vincent Corretja**. In the context of maritime or ground surveillance by airborne radar, radar processing makes it possible, among other things, to establish a tactical situation in the area or areas of interest where the targets can be numerous and diverse. Classifying these targets then makes it possible to clarify this situation and thus reduce the operator's cognitive load so that he can identify targets of interest more quickly. This thesis aims to compare different classification approaches applied to radar data (measurements, radar images or other data of interest for radar processing) allowing the system to improve its performance through experience. Two philosophies are envisaged:

1. The first combines signature extraction and classification methods, also known as machine learning. Signatures can take different forms: parameters of a model representing the data being studied, characteristics of certain properties such as the rate of decrease of the autocorrelation function, projection coefficients of a representation in a given database, etc.
   As for classification methods, they can be distinguished according to several properties: supervised, semi-supervised or not, etc.

2. The second is based on a deep learning approach. This includes Convolutional Neural Networks (CNN or ConvNet). These approaches have been used in many contexts, from image and video recognition to natural language processing. They have also been combined with Probability Hypothesis Density (PHD) approaches for tracking objects in videos. They were also used to learn the representation in the state space of systems whose state is estimated by Kalman filtering COSKUN et collab. [2017].
   Their use is developing in the radar context. Without being exhaustive, we can mention for example PROTOPAPADAKIS et collab. [2017] ZHANG et collab. [2017] LIANG et collab. [2018] GALLEGO et collab. [2018].

In this thesis, we propose to evaluate or even compare these two types of approaches according to several cases of use in order to understand their respective relevance according to the applications. There are two main reasons for this approach. On the one hand, as Schegmann *et al.* points out in SCHWEGMANN et collab. [2017], deep learning should not be seen as an answer to all problems; often, a "simpler" approach in terms of computational complexity can be envisaged to successfully solve the problem. On the other hand, it is a question of integrating radar expertise as much as possible into the development of approaches based on artificial intelligence concepts.

Among the case studies identified, the following four will be addressed in the doctoral thesis:

1. Classification of radar measurements at the output of the radar signal processing block in order to discriminate between radar measurements (target or false alarm) or the type of targets (moving targets, fugitive targets, large targets, etc.);

2. Classification of object trajectories: in this case, it is a question of using the trajectories

estimated by the radar system or by another information system (AIS[1] data for instance - Automatic Identification System). This would make it possible to distinguish the type of targets and identify target trajectories with abnormal behaviour;

3. Selection of the best image in an ISAR (Inverse Synthetic Aperture Radar) context: this involves automatically determining the most representative image to present to the operator among a set of images representing the same target.

4. Generation of synthetic signals in a maritime surveillance context. The detection of marine targets is degraded by the presence of the signal back-diffused from the sea surface called sea clutter. Simulation of an environment similar to the operational operating conditions of a maritime surveillance radar consists of fine statistical modeling of sea clutter. Since flight tests to acquire real data are expensive, the generation of synthetic data is a major challenge for the development of new algorithms and their qualification.

Post-AVC project

Work carried out with **Eric Sorita** and **Marc-Michel Corsini**.
Eric SORITA, Doctor in Cognitive Sciences, Associate Researcher at EA 4136 Handicap Activity Cognition Santé - University of Bordeaux is the lead investigator.

In the context of certain disabilities resulting from a stroke, for example, we know that certain parts of the body, such as the upper limb, will be less spontaneously used, with risks of exclusion of these unused or underused parts, which will have an impact on performance and participation. The use of accelerometer wristbands at home in this context can then provide valuable insights into the degree of integration of the arm into daily life and the impact on the person's participation. The reliability of the actimetric data from these connected objects is extremely variable depending on the devices. Verification of the reliability of the actimetric data from these bracelets is therefore a necessary prerequisite to validate the interest of their use. One of the objectives of this study is therefore to collect accelerometric records during the performance of simple activities of daily living to verify the reliability of the data recorded in correspondence with the actions performed.

Subjects are equipped with accelerometric bracelets positioned bilaterally at the wrists. The recording frequency of the bracelets is set to 30 Hertz. The subjects are also equipped with a helmet with a GOPRO type camera oriented parallel to the axis of vision. The use of an on-board recording system is preferable to video recording from a fixed camera whose recording can be cut off from the action. The orientation of the on-board system in the axis of vision allows to observe the movements of the hands and forearms when the subject looks at his hands when performing the actions. It also makes it possible to observe the subject's movements when he/she goes to an activity station corresponding to one of the tasks. The subject's face cannot appear during the actions and remains anonymous.

The main contribution of this work on wristbands lies in demonstrating the clinical value of implementing simple and inexpensive devices to trace the activity of the arms in daily life in post-stroke patients. Currently, the only evaluations available to us on the spontaneous use of the hemiparetic arm in daily life are self-questionnaires and in particular the Motor Activity Log (MAL) questionnaire[2]. The use of wristbands at home would make it possible to provide more objective data and the low cost of the equipment could lead to a systematic approach to the management of these patients in France (the practice could, for example,

[1] AIS data, initially introduced to avoid collisions between vessels, make it possible to follow the movement of vessels over time. They include different types of information: dynamic parameters correspond to heading, speed and position over time, while static parameters refer to the identity of the ship, its dimensions or its origin and destination.

[2] https://www.uab.edu/citherapy/images/pdf_files/CIT_Training_MAL_manual.pdf

be recommended). Of course, it is necessary to check the question of the reliability of the equipment in order to have valid measurements. In addition, the most relevant and clinically informative data should be identified to show the added value of using these devices in addition to the usual recommendations. The graphical representation we seek to give to the data collected is in particular a first-rate clinical argument.

Among the technological obstacles that will have to be overcome, one of the most important will be to successfully identify voluntary movements and distinguish them from the rest of the recorded data (noise and unintentional movements). Currently, the discrimination filters used are only based on a threshold value that is difficult to justify.

In an epidemiological context where stroke is the leading cause of non-traumatic disability in adults, providing objective elements that allow a more appropriate orientation of patient care is also an issue that can have medical-economic as well as scientific and research benefits, simply because the intra-stroke population is ultimately very heterogeneous and providing data to better identify patient subgroups can also improve the quality and scope of patient inclusion criteria in clinical studies.

References

COSKUN, H., F. ACHILLES, R. DIPIETRO, N. NAVAB et F. TOMBARI. 2017, «Long short-term memory kalman filters: Recurrent neural estimators for pose regularization», *ICCV*.

GALLEGO, A.-J., P. GIL, A. PERTUSA et R. B. FISHER. 2018, «Segmentation of oil spills on side-looking airborne radar imagery with autoencoders», *Sensors*, vol. 18, (797), p. 1–16.

LIANG, P., W. SHI et X. ZHANG. 2018, «Remote sensing image classification based on stacked denoising autoencoder», *Sensors*, vol. 10, (16), p. 1–12.

PROTOPAPADAKIS, E., A. VOULODIMOS, A. DOULAMIS, N. DOULAMIS, D. DRES et M. BIMPAS. 2017, «Stacked autoencoders for outlier detection in over-the-horizon radar signals», *Computational Intelligence and Neuroscience*, p. 1–11.

SCHWEGMANN, C., W. KLEYNHANS et B. P. SALMON. 2017, «The development of deep learning in synthetic aperture radar imagery», *International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, p. 1–2.

ZHANG, Q., Y. SHAO, S. GUO, L. SUN et W. CHEN. 2017, «A novel method for sea clutter suppression and target detection via deep convolutional autoencoder», *International Journal of Signal Processing*, vol. 2, p. 35–40.