IHBA: An Improved Homogeneity-Based Algorithm for Data Classification

Fatima Bekaddour, Chikh Mohammed Amine

IHBA: An Improved Homogeneity-Based Algorithm for data classification

Fatima Bekaddour and Chikh Mohammed Amine

Abou Bekr Belkaid University, Tlemcen , Algeria
fatima.bekaddour@gmail.com
am_chikh@yahoo.fr

Abstract. The standard Homogeneity-Based (SHB) optimization algorithm is a metaheuristic which is proposed based on a simultaneously balance between fitting and generalization of a given classification system. However, the SHB algorithm does not penalize the structure of a classification model. This is due to the way SHB's objective function is defined. Also, SHB algorithm uses only genetic algorithm to tune its parameters. This may reduce SHB's freedom degree. In this paper we have proposed an Improved Homogeneity-Based Algorithm (IHBA) which adopts computational complexity of the used data mining approach. Additionally, we employs several metaheuristics to optimally find SHB's parameters values. In order to prove the feasibility of the proposed approach, we conducted a computational study on some benchmarks datasets obtained from UCI repository. Experimental results confirm the theoretical analysis and show the effectiveness of the proposed IHBA method.

Keywords: Metaheuristics, HBA, Improvement, Machine Learning, Medical Informatics.

1 Introduction

Nowadays, metaheuristics approaches represent a well-established method toward solving complex and challenging optimization problems. Offering suboptimal (optimal) quality solutions in a reasonable time, they may be considered as complement to exact optimization methods. Among popular metaheuristics, there are: Genetic Algorithm [1] emulates Darwinian evolution theory; Simulated Annealing imitates annealing process of melts [2] and Particle swarm optimization stems from biology where a swarm coordinates itself in order to achieve a goal [3].

Recently, Pham and Triantaphyllou [4][5][6]developed a new metaheuristic called HBA: Homogeneity-Based Algorithm. The Standard HBA metaheuristic (SHB) is used in conjunction with traditional data mining approaches (such as: ANN: Artificial Neural Network, DT: Decision Tree…) .The main idea of SHB algorithm is to simultaneously balance both fitting and generalization [5] by adjusting classification model through the use of the concept of Homogenous Set and Homogeneity Degree [4].This is done in order to reduce the total misclassification cost of the inferred mod-

els. However, a problem with SHB algorithm is that may not adopt computational complexity of the used classification model. This is due to the way objective function is defined. For the SHB metaheuristic, the total misclassification cost is described by computing only the three type of errors (false positive, false negative and the unclassifiable cases) with their penalty costs. Additionally, for this metaheuristic, only Genetic Algorithm (GA) is adopted to find optimally thresholds values, used to control the balance between the fitting and the generalization. This may reduce SHB's freedom degree.

In this article, we extend works in [4][5][6]. New contributions lies in (1) modifying the SHB's objective function to support structural complexity of the used classifier model (2) Proposing a meta-optimization based solution to the problem of tuning SHB's parameters. The IHBA (Improved Homogeneity-Based) algorithm enhances average results obtained in comparison to the standalone algorithms. Rest of this paper is organized as follows:

The standard HBA metaheuristic (SHB) is presented in the following section, before the proposed approach IHBA is elaborated. Section 3 describes some famous benchmark datasets used to test the proposed approach and explains respective results. Last section concludes the paper.

2 Methodology

2.1 Standard Homogeneity Based-Algorithm (SHB)

SHB is a recent metaheuristic, developed by Pham and Triantaphyllou in [4][5][6]. The main idea of SHB algorithm is to adopt a simultaneously balance between generalization in order to minimize total misclassification cost (TC) [4][5][6]. Let C_{FP}, C_{FN}, C_{Uc} be the penalty costs for the false positive, false negative and unclassifiable cases respectively. Also, let us denote RateFP, RateFN, RateUc as the false positive, false negative, unclassifiable rates, respectively. Then TC is defined as follow:

$$TC=\min (C_{FP}*Rate_{FP}+C_{FN}*Rate_{FN}+C_{UC}*Rate_{UC}) \quad (1)$$

SHB algorithm is used in conjunction with data mining techniques to create classification system that would be optimal in term of TC value. There is a fundamental key issue regarding the SHB algorithm [4][5][6]:

- The more compact and homogenous decision regions are, the more accurate the inferred models are. In addition, the denser the decision regions are, the more accurate the inferred models are.

The density measurement for a homogenous set is called Homogeneity Degree (HD) [4]. In [4][5][6] ,the authors proposed a way to compute HD as follow:

$$HD= \ln (nc) / h \quad (2)$$

Where *nc* is the number of points in a given set **C**, and *h* is defined in **Heuristic rule.**

The SHB algorithm stops when all of the homogenous sets have been treated. Note that SHB metaheuristic utilize GA (Genetic Algorithm) to find optimal values of the controlling threshold: β^-, β^+, α^-, α^+.

Heuristic Rule: if ***h*** is set equal to the minimum value in set ***C*** and this value is used to compute the density ***d(x)*** using equation 3, then ***d(x)*** approaches to a true density.

$$d(x) \approx \frac{1}{n * h^D} \sum_{i=1}^{n} \prod_{m=1}^{D} \varphi\left(\frac{x^m - x_i^m}{h}\right) \quad (3)$$

Where φ is the kernel function, defined in D-dimensional space and n is the number of points in a given set **C.**
The following pseudo-code describes the SHB algorithm:

```
Start
Initial parameters setting (α+, α-, β+, β -).
1. Apply a Data Mining approach on a training dataset T1
to infer positive and negative classification models.
2. Break the inferred models into hyper spheres.
3. For each hyper sphere C do:
  Determine whether C is homogenous or not.
     If so, computer HD using formula 2.
     Else fragment C into smaller hyper spheres.
4. Sort HD in decreasing order.
5. For each homogenous set C do:
  If [(HD ≥ β+(β-)] then
     Expand C using HD and α+(α-).
  Else
     Break C into smaller homogenous sets.
end
```

2.2 A modified SHB objective function

As presented above, SHB algorithm modifies an existing classification pattern such that the total misclassification cost TC (formula 1), will be optimized or significantly reduced. Nevertheless, SHB metaheuristic objective function neglects the structural complexity of a given classification model. For example, The ANN (Artificial Neural Network) structural complexity is defined as the total number of weights and bias, figured in its architecture and the time needed for network learning. It is proved by choosing theses parameters effectively minimize the network error and perform better results.

In this regards, we have proposed a modified objective function, adopting the computational complexity design function [7] to compute the penalty of a given pattern classification architecture as follow:

$$fobj = Penality * \frac{\alpha 1 * TC_{Training} + \alpha 2 * TC_{Generalization}}{\alpha 1 + \alpha 2} \quad (4)$$

Where $(\alpha 1, \alpha 2) > 0 \in \Re$ (usually $\alpha 1 \leq \alpha 2$), are factors indicating importance degree of the learning and the generalization errors respectively. Penalty presents the model architecture influence of the objective function value as follow [7]:

$$Penality = 5 * 10^{-8} * e^{f(x)} + 5 * 10^{-5} * y + 1 \quad (5)$$

Where: y is the number of epochs necessary in the model training; f(x) is the Structural complexity of a classification model.
Using different values of C_{FP}, C_{FN}, C_{Uc} in objective function formula (4) , we design others objectives functions formula (6-7-8) as follows:

$$fobj = Penality * \frac{\alpha 1 * (RateFP_{Train} + RateFN_{Train}) + \alpha 2 * (RateFP_{Gener} + Ra\,按eFN_{Gener})}{\alpha 1 + \alpha 2} \quad (6)$$

$$fobj = Penality * \frac{ア1 * TC1_{Train} + \alpha 2 * TC1_{Gener}}{\alpha 1 + \alpha 2} \quad (7)$$

Where: $TC1_{Train} = 3RateFP_{Train} + 3RateFN_{Train} + 3RateUc_{Train}$

$$TC1_{Gener} = 3RateFP_{Gener} + 3RateFN_{Gener} + 3RateUc_{Gener}$$

$$fobj = Penality * \frac{\alpha 1 * TC2_{Train} + \alpha 2 * TC2_{Gener}}{\alpha 1 + \alpha 2} \quad (8)$$

Where: $TC2_{Train} = RateFP_{Train} + 20RateFN_{Train} + 3RateUc_{Tr\,按in}$

$$TC2_{Gener} = RateFP_{Gener} + 20RateFN_{Gener} + 3RateUc_{Gener}$$

Note that, ($RateFP_{Train}$, $RateFN_{Train}$, $RateUc_{Train}$) represent FP, FN and Uc rates during the training phase and ($RateFP_{Gener}$, $RateFN_{Gener}$, $RateUc_{Gener}$) represent FP, FN and UC rates during the test phase.

- **In Formula 6:** we do not penalize Uc, but penalize the same cost for FP, FN.
- **In Formula 7:** we penalize all three error types by unit equal to three.
- **In Formula 8:** we penalize more FN than the other type of errors.

2.3 Tuning SHB parameters by means of metaheuristics

Within the scope of SHB algorithm, there are four parameters which are used to control the balance of fitting and generalization that would minimize (or significantly reduce) the total misclassification cost (TC):

- Two expansion factors α^-, α^+, to be used for expanding the negative and the positive homogenous sets.
- Two breaking factors β^-, β^+, to be used for breaking the negative and the positive homogenous sets.

Note that, if the expansion parameters values (α^-, α^+) are too high, then this would result in the oversimplification problem. On the contrary, too low expansion parameters values may not be sufficient to overcome the overfitting problem. The opposite situation is true with the breaking factors values (β^-, β^+). Authors in [4][5][6] propose

to only use genetic algorithm(GA) to find optimal threshold values for α^-, α^+, β^-, β^+. This may reduce the freedom degree of the SHB algorithm .

This article employs several metaheuristics approaches to formally test the existence of a relationship between performance and effective parameters values. In particular, (PSO: Particle Swarm Optimization, SA: Simulated Annealing and GA: Genetic Algorithm) metaheuristics are used for the SHB algorithm parameters α-, α+, β -, β+. That is these parameters represents individual variables and *fobj* described in formula 4 is taken as objective function. Since PSO, SA and GA metaheuristics approaches are tested using a dataset to find optimal values for (α-, α+, β -, β+), a calibration dataset is needed. This requirement can be fulfilled in the following way: the original training dataset T is divided into two datasets: T1 (for example: 90%) for training data mining models to infer positive and negative classification models, and T2 as a calibration dataset.

In the first phase, hyperspheres that cover decision regions are employed to obtain homogenous set (using step 3to 5 described in the pseudo-code of SHB algorithm) . Then, classification models (homogenous sets) are evaluated by using the calibration dataset T2 to compute *fobj*. Next, metaheuristic bloc could replace the default tuning parameters GA (Genetic Algorithm) and determine the new threshold values (α^-, α^+, β^-, β^+).

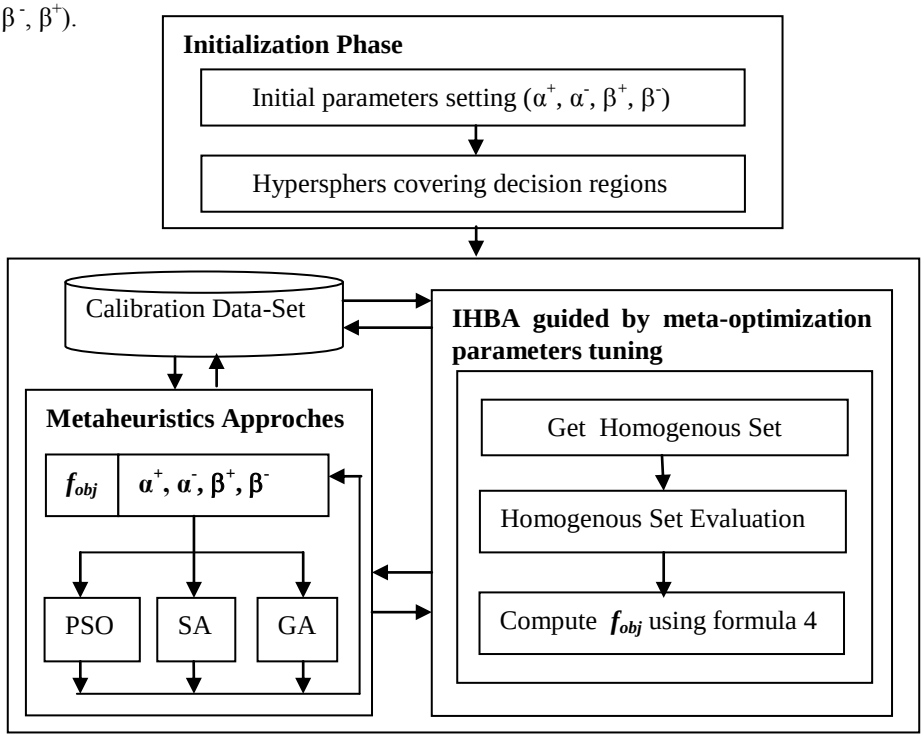

Fig. 1. Architecture of the proposed system to determine IHBA parameters

In fact, this leads to a meta-optimization approach, which means that any metaheuristic is used to search for the best tuning of parameters of metaheuristic in solving a

given optimization problem [3]. After a number of iterations, the proposed approach returns the optimal threshold values of (α^-, α^+, β^-, β^+). It is to be emphasized that by employing metaheuristics bloc during SHB algorithm iterations , permit to estimate effective parameters setting for SHB metaheuristic and therefore, allow to approximate a functional relationship between classifier's performance and effective parameters . The architecture of the overall system is depicted in figure 1.

3 Some computational results

3.1 Benchmark data sets

This paper studies two medical data sets: Appendicitis (AP), and Thyroid (TR). **Table 1** shows a summary of the main characteristics of these datasets. The benchmark chosen present a variety of descriptions (including number and type of attributes, number of instances…). The first dataset is Appendicitis, created by Kapouleas and Weiss (1989) [8] from Rutgers university. The features were obtained from laboratory tests as follow: WBC1, MNEP, MNEA, MBAP, HNEP and HNEA. The second medical dataset is thyroid disease, obtained from UCI repository [9]. It consists of five continuous attributes. The task is to identify whether a patient is normal or suffers from hyper (hypo) -thyroidism.

Datasets	No. Instances	No. Features	Training Dataset	Testing Dataset
Appendicitis	106	7	79	27
Thyroid	215	5	143	72

Table 1. Medical datasets characteristics

3.2 Results and Discussion:

The following are some computational results obtained from several experiments performed for each data mining approaches used in such work. Experiments were conducted with two datasets obtained from UCI repository [9]. As discussed before, we considered three scenarios for the IHBA objective function. Also, we choose different setting for (α_1, $\alpha 2$) factors that are used to weigh the importance degree attributed to the learning and the generalization errors respectively.

Initially, we assigned an equal weight ($\alpha_1=\alpha_2=1$) to the learning and the generalization errors for ANFIS (Adaptative Neuro-Fuzzy Inference System) [10], LVQ (Learning Vector Quantization) [11] and PMC (Perception Multi-layers). Then, we choosed a larger weight to the generalization than the training error ($\alpha_1=0.5$; $\alpha_2=1$). Finally, we attributed more importance to the ability of learning than the ability of finding correct output value for an unknown data sample ($\alpha_1=1$; $\alpha_2=0.5$).The results of these simulations are shown in Table 2, 3 and 4. According to those tables, it appears that α_1 and $\alpha 2$ factors have influence on the final results. Those Tables show the misclassification testing error rate ($\boldsymbol{TC_{Test}}$) and $\boldsymbol{f_{eval}}$ (the objective function evaluation)

obtained for original algorithms (ANFIS, LVQ, PMC) and the proposed IHBA approach. The colon improvement presents any improvement rate achieved by the IHBA when compared with that of the standalone algorithm.

Datasets	**Alg**	**α1 α2**	**Original-Alg** TC_{Test}	f_{eval}	**IHBA** TC_{Test}	f_{eval}	**Improv (%)**
AP	ANFIS	1 1	3	26.9	11.1	30.9	No.impr
		0.5 1	3	19.3	3.7	19.8	No.impr
		1 0.5	3	38.2	0.00	37.1	2.87
	LVQ	**1 1**	11	14.4	3.7	10.7	25.69
		0.5 1	3	7.08	3.7	7.55	No.impr
		1 0.5	11	13.8	3.7	11.3	18.11
	PMC	1 1	7	16.3	7.4	16.5	No.impr
		0.5 1	3	7.86	7.4	10.9	No.impr
		1 0.5	14	18.2	7.4	15.8	13.18
TR	ANFIS	1 1	69	40.8	30.5	21.5	47.30
		0.5 1	69	51.1	27.7	23.6	53.81
		1 0.5	69	35.1	27.7	21.3	39.31
	LVQ	1 1	25	23.4	26.3	24.0	No.impr
		0.5 1	69	69.6	26.3	41.0	41.09
		1 0.5	25	22.8	26.3	23.2	No.impr
	PMC	1 1	4	2.78	5.5	3.6	No.impr
		0.5 1	2	2.1	1.3	1.61	23.33
		1 0.5	2	2.6	5.5	3.81	No.impr

Table 2. Results in minimizing *(f_{eval}=FP+ FN)*

In a first scenario (formula 6), we did not penalize for the unclassifiable cases (Uc), and penalized by one unit the FP (False Positive) and the FN (False Negative) errors. The results of this scenario are shown in Table 2.This table shows that the average values of f_{eval} obtained from the IHBA on the AP and TR datasets were 17.83, 18.18 respectively. Furthermore, theses values of f_{eval} were optimal than the average values of f_{eval} achieved by the stand-alone algorithms on AP and TR datasets by about 6.65, 22.76 respectively.

In the second scenario (formula 7), we assumed that all three error types would be penalized by an identical value, equal to three units. The results are presented in Table 3. The average values for f_{eval} obtained from IHBA on AP, TR datasets were 147.55, 75.75 respectively. These values for f_{eval} were less than the average values of f_{eval}

Datasets	Alg	α1	α2	Original-Alg TC_{Test}	Original-Alg f_{eval}	IHBA TC_{Test}	IHBA f_{eval}	Improv (%)
AP	ANFIS	1	1	233.33	193.1	288.9	221.0	No.impro
		0.5	**1**	233.33	208.0	300.0	252.6	No.impro
		1	**0.5**	233.33	189.6	300.0	211.9	No.impro
	LVQ	**1**	**1**	233.33	144.2	166.6	110.6	23.3
		0.5	**1**	233.33	171.8	144.4	112.1	34.7
		1	0.5	233.33	108.7	144.4	78.88	27.4
	PMC	1	1	233.33	160.9	155.5	119.9	25.5
		0.5	1	233.33	181.2	155.5	126.5	30.2
		1	0.5	233.33	121.8	155.5	94.47	22.4
TR	ANFIS	1	1	220.8	129.3	91.66	64.64	50.0
		0.5	1	220.8	162.6	95.83	79.26	51.5
		1	0.5	220.8	109.9	95.83	68.21	37.9
	LVQ	1	1	283.3	174.9	83.33	74.43	57.4
		0.5	1	220.8	218.2	83.33	126.0	No.impro
		1	0.5	283.3	138.3	83.33	71.33	48.4
	PMC	1	1	212.5	113.6	108.3	58.90	48.1
		0.5	1	216.7	153.8	133.3	95.41	37.9
		1	0.5	212.5	80.04	108.3	43.57	45.5

Table 3. Results in minimizing (f_{eval}= 3*FP+ 3FN+3Uc)*

achieved by original algorithms by about 18.16 and 41.8 on the AP, and TR datasets respectively. In the last scenario (formula 8), we assumed that the FN would be more penalized than the other two types of errors (FP, FN). In particular, table 4 shows that the average values for f_{eval} obtained from IHBA on the AP and TR datasets were 236.63, 94.29 respectively .This table, shows that the f_{eval} were less than the original algorithms (ANFIS, LVQ, PMC) by about 0.43, 63.23 when applied on the AP and TR datasets respectively.

When comparing the tables 2, 3, 4, it appears that PMC and ANFIS models, usually obtain better results. However, ANFIS is more practical due to its transparency. Additionally, in some cases, the f_{eval} value of a standalone approach yielded better values than the one achieved by the IHBA metaheuristic. A reason for that is that the standalone algorithm may have reached the global optimal value (or close to that) for f_{eval} .Note that the number of membership functions and hidden layers affect the structural complexity of the neuro-fuzzy system and the artificial neural network models respectively, in this work, we proposed to use two membership function for ANFIS system and one hidden layer for LVQ and PMC classification models .

The best architecture model found for ANFIS, LVQ and PMC models were (128,20,20) for AP and (32,20,20) for TR dataset respectively.

Datasets	**Alg**	**α1 α2**	**Original-Alg** TC_{Test}	f_{eval}	**IHBA** TC_{Test}	f_{eval}	**Improv (%)**
AP	ANFIS	1 1	296.29	174.0	300.00	175.9	No.impr
		0.5 1	296.29	215.5	292.59	213.0	1.16
		1 0.5	296.29	136.3	288.88	133.9	1.76
	LVQ	1 1	211.11	187.8	281.48	223.3	No.impr
		0.5 1	225.92	229.4	281.48	266.7	No.impr
		1 0.5	211.11	194.0	281.48	217.7	No.impr
	PMC	1 1	288.88	367.6	281.48	363.8	1.03
		0.5 1	225.92	265.9	281.48	304.9	No.impr
		1 0.5	203.7	203.2	281.48	230.5	No.impr
TR	ANFIS	1 1	1327.6	670.8	30.55	21.6	96.8
		0.5 1	1327.6	891.0	30.55	25.52	97.1
		1 0.5	1327.6	455.1	40.27	25.56	94.4
	LVQ	1 1	221.05	121.9	30.55	26.24	78.4
		0.5 1	1327.6	1358	30.55	488.9	63.9
		1 0.5	221.05	88.57	30.55	24.8	71.9
	PMC	1 1	222.36	12.47	97.22	51.8	No.impr
		0.5 1	225.00	158.2	155.5	109.6	30.7
		1 0.5	221.05	116.5	101.3	74.64	35.9

Table 4. Results in minimizing (f_{eval}=*FP+ 20FN+3Uc)*

In order to shed some light upon the second contribution, Table 5 provides an overview of the results obtained throughout the empirical comparison of different meta-optimization based solution (PSO, SA and GA) to the problem of tuning SHB's parameters. The colon improvement 1 shows any improvement of f_{eval} achieved by IHBA enhanced by means of metaheuristics approaches to find optimal thresholds values (α^+, α^-, β^+, β^-), when compared with the standalone algorithms under the first consideration (where α_1=0.5, α_2=1) and by using ANFIS model. The colon improvement 2 shows any improvement of f_{eval} achieved by IHBA enhanced by means of meta-optimization parameters tuning, when compared with best results obtained with IHBA under the first consideration, where α_1,=0.5; α_2=1. We have simulated this scenario (α_1=0.5, α_2=1), because it seems to be more realistic that the ability to learn the model is less relevant than the ability to generalize (i.e. find a correct output value for an unknown data sample).

The colon parameters setting specify different parameters configuration for considered metaheuristics (PSO, SA and GA). In particular, PSO algorithm has been applied

with different values of number of iterations (50, 100), population size (20, 40), social attraction and cognitive attraction(0.25, 0.7). In case of SA metaheuristic, we optimized SHB's factors (α^+, α^-, β^+, β^-) by setting different values of iteration number (500, 1000) and the perturbation function. The initial temperature was set either to 50 or 100. In the GA, each chromosome encodes the two expansion thresholds values (α^+, α^-) and the two breaking thresholds values (β^+, β^-). The population evolves in search for the optimal values of these parameters. We have applied the GA with different values of: number of generation (200, 1000), population size (15, 35) and crossover fraction (0.5, 0.7). Mutation fraction equaled 0.01.

Data sets	**Meta-Heuristic**	**Parameters Setting**	**Results** TC_{Test} f_{eval}	**Improv1 Rate**	**Improv2 Rate**
AP	PSO	100 ; 20 ; 0.7 ; 0.25	3.7 37.1	No.impr	No.impro
		50 ; 20 ; 0.25 ; 0.7	7.4 39.6	No.impr	No.impro
		50 ; 40 ; 0.7 ; 0.25	3.7 37.1	No.Impr	No.Impr
		100 ; 40 ; 0.25 ; 0.7	0.00 34.6	No.impr	No.impro
	SA	500;50 ;Fast	3.7 37.1	No.impr	No.impro
		1000;100. Fast	3.7 37.1	No.impr	No.impro
		1000 ;50 ; Bolz	14.8 44.5	No.impr	No.impro
		500 ;100 ;Bolz	3.7 37.1	No.impr	No.impro
	GA	15; 200; 0.5; 0.01	0.00 34.6	No.impr	No.impro
		35;1000; 0.7 ;0.01	0.00 34.6	No.impr	No.impro
		35; 200; 0.5; 0.01	3.7 37.1	No.impr	No.impro
		15;1000; 0.7 ;0.01	0.00 34.6	No.impr	No.impro
TR	PSO	100 ; 20 ; 0.7 ; 0.25	2.77 12.0	76.4	48.9
		50 ; 20 ; 0.25 ; 0.7	26.4 27.8	45.6	No.impro
		50 ; 40 ; 0.7 ; 0.25	9.60 16.7	67.3	29.23
		100 ; 40 ; 0.25 ; 0.7	25.0 26.9	51.0	No.impro
	SA	500;50 ;Fast	2.77 12.0	76.4	48.9
		1000;100. Fast	2.77 12.0	76.4	48.9
		1000 ;50 ; Bolz	2.77 12.0	76.4	48.9
		500 ;100 ;Bolz	2.77 12.0	76.4	48.9
	GA	15; 200; 0.5; 0.01	12.5 18.5	63.7	21.4
		35;1000; 0.7 ;0.01	11.1 17.6	65.5	25.4
		35; 200; 0.5; 0.01	2.77 12.0	76.4	48.9
		15;1000; 0.7 ;0.01	11.1 17.6	65.5	25.4

Table 5. Results of IHBA improved by means of parametes tuning (f_{eval}=*FP+ FN)*

It is clearly visible, that the PSO metaheuristic achieved better results (in minimizing f_{eval}), for big number of iterations, population size and cognitive attraction. Simulated annealing algorithm was slightly worse than GA metaheuristic.

Table 5 shows that the average values of f_{eval} obtained from IHBA approach improved by means of meta-optimization approaches on AP and TR datasets were 37.09 and 16.45 respectively. In addition, these values of f_{eval} were less than those achieved by standalone methods and IHBA approach depicted in Table 2 on TR dataset by about 68.08 and 32.9 respectively. Note that, The proposed IHBA approach improved by means of parameters tuning based on (PSO, SA and GA) metaheuristics, when applied on AP dataset, found no improvement of f_{eval} , compared to original results depicted in Table 2. A reason for that, is that the standalone approaches or IHBA may have achieved optimal (or near-optimal) values of f_{eval} .

4 Conclusion

Considering importance of parameters tuning of a given metaheuristic algorithm, in this paper, we proposed an Improved Homogeneity Based-Algorithm which uses computational complexity of a classifier model as a modified objective function. Additionally, we employed several metaheuristics approaches (Simulated annealing, Genetic Algorithm and Particle Swarm Optimization) to find optimally thresholds values, used to refine the inferred models regions obtained by applying a classification method. The proposed method IHBA (Improved Homogeneity-Based Algorithm) tested on some benchmarks data sets from the UCI repository indicated the increased performance of the proposed algorithm in comparison with the standalone algorithms (ANFIS, LVQ and PMC). Future works will extend the SHB metaheuristic with feature subset selection aiming to reduce classification time and making HBA applicable to higher data dimensionality.

References

1. J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).
2. Kirkpatrick, S., Gelatt, C.D., Jr., and Vecchi,. Optimization by simulated annealing. Science 220 : 671-680, M. P. 1983 .
3. E.-G. Talbi. Metaheuristics: From Design to Implementation. Wiley, June 2009.
4. Pham HNA, Triantaphyllou E. The impact of overfitting and overgeneralization on the classification accuracy in data mining. In: Maimon O, Rokach L, editors. Soft computing for knowledge discovery and data mining, pages 391–431, part 4, chapter 5,New York, NY, USA: Springer; 2007.
5. Pham HNA, Triantaphyllou E. Prediction of diabetes by employing a new data mining approach which balances fitting and generalization. In: Yin Lee R, editor. Studies in computation intelligence, vol. 131. pages 11–26, chapter 2, Berlin, Germany: Springer; 2008.
6. Pham HNA, Triantaphyllou E. An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets. Expert Systems with Applications, vol 36,number 5, pages 9240–9249; 2009.
7. Adenilson R. Carvalho, Fernando M. Ramos, Antonio A. Chaves. Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem. Neural Computing and Applications,Vol 20, Issue 8, pages 1273-1284, November 2011.

8. S.M. Weiss, I. Kapouleas, An empirical comparison of pattern recognition, neural nets and machine learning classification methods, in: J.W. Shavlik and T.G. Dietterich, Readings in Machine Learning, Morgan Kauffman Publ, CA 1990.
9. UCI repository of machine learning databases, University of California at Irvine, Departmentof Computer Science, http://archive.ics.uci.edu/ml/datasets/Thyroid+Disease, last accessed (2015).
10. R.Jang, Anfis : adaptative network-based fuzzy inference système ;IEEE Trans. on Systems, Man and Cybernetics,J.S.,1993.
11. Kohonen, T, The Self-Organizing Map. *Proceedings of the IEEE*, Vol 78, Issue 9, pages 1464-1480. 1990.