A robust learning algorithm for evolving first-order Takagi-Sugeno fuzzy classifiers

Abdullah Almaksour, Eric Anquetil

A robust learning algorithm for evolving first-order Takagi-Sugeno fuzzy classifiers

Abdullah Almaksour, Eric Anquetil

INSA de Rennes, Avenue des Buttes de Coesmes, F-35043 Rennes
CNRS, UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{Abdullah.Almaksour, Eric.Anquetil}@irisa.fr

Abstract : We present in this paper a new method for the design of customizable self-evolving fuzzy rule-based classifiers. The presented approach is based on a first-order Takagi-Sugeno fuzzy inference system. This approach involves first an incremental clustering and adaptation of the premise part of the system, and secondly, an incremental learning of the linear consequents parameters of the system using a modified version of the Recursive Least Square method. We use this method to build an evolving handwritten gesture recognition system. The self-adaptive nature of this system allows starting the learning process by few learning data, to continuously adapt and evolve according to any new data, and to keep robust when introducing a new unseen class at any moment in the life-long learning process.

Keywords : Incremental learning, fuzzy inference system, handwriting recognition.

1 Introduction

Classification techniques appear frequently in many application areas, and become the basic tool for almost any pattern recognition task. The main problem in classification is to induce a classifier from a set of data samples. A large amount of samples is needed to set up and evaluate a classification system that can achieve a high accuracy, and it is practically very difficult to have such number of samples covering the whole feature space. Therefore, life-long classifier adaptation becomes more and more an essential point. Moreover, in many application contexts, the classifier needs to take into account new unseen classes and to integrate them in the classification process, which increases the need for "evolving" classifiers. One good example of such applications is the use of online handwriting gesture classifiers which aim at facilitating interactions with computers using pen-based interfaces like whiteboards, tablet PCs, PDA...Etc. The main drawback in the current existing systems is that they are trained "offline" on a specific group of gestures and then implemented to operate without changing their structure during the use. This fixed structure does not allow the user to choose his

own set of gestures or to add new ones to assign them to new interactive commands (according to his special needs), for example.

In our work, we aim at building a handwriting classifier, on-the-fly, from scratch and using only few data. Thus, the classifier will be incrementally adapted to achieve high recognition rates as soon as possible and to keep the system robust when introducing new unseen classes at any moment in the life-long learning process.

In order to clarify the meaning of the terms "incremental" and "online" learning, one can compare them to the "batch" and "offline" learning as follows Kasabov (2007):

- **Batch versus incremental learning:** In a batch mode of learning a predefined training dataset is learned by the system through propagating this dataset several times through the system. Each time the system optimizes its structure based on the average value of the goal function over the whole dataset. The incremental mode of learning is concerned with learning each data sample separately and the data might exist only for a short time. After introducing each data sample, the system makes changes in its structure to optimize the goal function. An incremental learning algorithm must learn new data without fully destroying the knowledge learned from old data and without the need to retrain the system on the old and the new data.

- **Offline versus online learning:** In an offline learning mode, the system is trained on a specific dataset and then implemented to operate in a real environment, without changing its structure during the use. While in an online learning mode, the model learns from new data during its use. It must learn rapidly and be operational and ready to be used in any moment during the lifelong incremental learning process.

An incremental learning algorithm is defined in Polikar *et al.* (2001) as being one that meets the following criteria: it should be able to learn additional information from new data; it should not require access to the original data (i.e. data used to train the existing classifier); it should preserve previously acquired knowledge (it should not suffer from *catastrophic forgetting*, significant loss of original learned knowledge); and it should be able to accommodate new classes that may be introduced with new data. Many of the existing "incremental learning" algorithms are not truly incremental because at least one of the mentioned criteria is violated. These criteria can be briefly expressed by the so called "plasticity-stability dilemma"Zwickel & Wills (2005), which is that a system must be able to learn in order to adapt to a changing environment but that constant change can lead to an unstable system that can learn new information only by forgetting everything it has so far learned.

We can distinguish two main types of incremental learning algorithms: algorithms for parameter incremental learning and algorithms for structure incremental learning. The incremental learning of parameters can be considered as "adaptation" algorithm. The structure in such systems is fixed and initialized at the beginning of the learning process, and the system parameters are learned incrementally according to newly available data. Some examples of these systems are presented in Jr. & Zeleznik (2007), Jang (1993) and Mouchere *et al.* (2007).

Most of structure incremental learning algorithms are based on the principle of the ART clustering algorithm Carpenter & Grossberg (1988), such as Carpenter *et al.* (1992), Sadri *et al.* (2006), Gary G. Yen (2001) and Lughofer (2008). The main problem of these systems is that they are sensitive to the selection of the vigilance parameter, to noise level in the training data and to the order in which training data are presented.

In an online incremental learning algorithm, the training set is not available a priori, since the learning examples come over time. Although online learning systems can continuously update and improve their models, not all of them are necessarily based on a real incremental approach. For some systems the model is completely rebuilt at each step of learning using all available data, they are systems with "full instance memory" Reinke (1988). On the other hand, if the learning algorithm modifies the model using only the last available learning example, it is called a system with "no instance memory" Littlestone & Warmuth (1994) Littlestone (1991). A third category is that of systems with "partial instance memory", which select and maintain a reduced subset of learning examples to use them in the next learning step Aha *et al.* (1991).

By these previous works, we found out that prototype-based models are generally better suited to incremental learning problems, where the model must be modified for each new example without forgetting the old knowledge, nor repeating a retraining process using an infinite memory of the previous examples. It is worth to mention that our leaning problem is stationary, and the data samples do not change their class with the time).

In this paper, we present an online incremental algorithm for structure and parameters learning with a partial instance memory. Our system is used in our context for the recognition of online handwritten gestures, and is able to learn new class of gestures and to evolve, sample after sample, without using all the old data.

The rest of the paper is organized as follows. In Section 2, we present one of our previous work and other related work from which we inspired the proposed model in this paper. Then, we describe in Section 3 the architecture of the used system. The different elements of our online incremental learning algorithm are detailed in section 4. Section 5 studies the experimental evaluation results.

2 Related work

We presented in previous work Almaksour *et al.* (2008) Almaksour & Anquetil (2009) an incremental learning model based on Fuzzy Inference System (FIS). We chose this classification system for its flexible nature as prototype-based system, which makes it able to adapt and to evolve according to the new data. We used a zero-order Takagi-Sugeno FIS Takagi & Sugeno (1985). The fuzzy rules make a link between intrinsic models (antecedents) and the system outputs by consequent functions. For a K classes problem, a rule R_i is built for each fuzzy model P_i:

$$\textbf{IF } \vec{x} \; is \; P_i \textbf{ THEN } y_i^1 = a_i^1 \textbf{ and ... and } y_i^k = a_i^k \tag{1}$$

where $\vec{x}$ is the n-dimensional feature vector. The a_i^c is a weight that expresses the participation of the model P_i in the description of the class c. Each Fuzzy models P_i is defined by its membership function.

In non-incremental learning systems, where the system is built using learning dataset with sufficient number of examples, supervised or unsupervised clustering methods are used to create the models (centers and covariance matrices) for each class.

In Almaksour & Anquetil (2009), we proposed an incremental clustering strategy for FIS based on the detection of confusion zones. We used the confusion rejection of a given sample to trigger a new prototype (model) creation. We aim to limit prototypes number in the system by avoiding creating prototypes in places where there is no likelihood of confusion. This eliminates adding "unnecessary" prototypes for a specific class where this class was already dominant. Although this method had a good performance in many incremental learning problems, it was difficult to find a universal and optimal reject threshold value. The other main disadvantage of this method was the big number of creating prototypes (models) for some problems, which leads to "heavy" and over-sophisticated systems.

An alternative incremental clustering approach had been presented in Angelov & Filev (2004) called *eClustering*. The main idea of the proposed approach is that of a *potential* of a given point in the data space, which is a value that represents the density at that point. Data samples with high potential are considered to be candidates to form a cluster. The potential of a sample, which was originally introduced in Yager & Fileu (1993), can be defined as inverse of the sum of the distances between that sample and all the other data samples. An interesting on-line (one-pass, non-iterative, recursive) procedures for calculation of the potential of the new data were introduced in Angelov & Filev (2004). The main advantage of this incremental clustering method is its parameter-free nature, and that it generally generates less number of prototypes (models) than the confusion-driven clustering strategy.

Although the eClustring method can generate a compact rule-base when used to incrementally learn an FIS, the intrinsic part of the FIS is less efficient comparing to that generated by the confusion-driven strategy (too many rules). To cope with this problem, the consequent functions of the FIS need to be enhanced by increasing the local discrimination function of each fuzzy rule. For this reason, we propose to use a first-order Takagi-Sugeno FIS by replacing the constant value in the consequent part of the fuzzy rule in [1] by a linear consequent function (a linear hyper-plane). More details about the system architecture are presented in the next section.

3 System architecture

As aforementioned, our system is based on first-order Takagi-Sugeno fuzzy inference system. It consists of a set of fuzzy rules of the following form:

$$\textbf{Rule}_i : \quad \textbf{IF } \vec{x} \; is \; P_i \textbf{THEN } y_i^1 = l_i^1(\vec{x}) \textbf{ and ... and } y_i^k = l_i^k(\vec{x}) \tag{2}$$

where $l_i^m(\vec{x})$ is the linear consequent function of the rule i for the class m:

$$l_i^m(\vec{x}) = \vec{\pi}_i^m \vec{x} = a_{i0}^m + a_{i1}^m x_1 + a_{i2}^m x_2 + ... + a_{in}^m x_n \tag{3}$$

To find the class of $\vec{x}$, its membership degree $\beta_i(\vec{x})$ to each fuzzy prototype is computed. This one is represented in our system by a hyper-ellipsoidal Radial Basis Function (RBF) characterized by a center $\vec{\mu_i}$ and a covariance matrix Q_i . The activation function influence decreases according to the Mahalanobis distance from the center $d_{Q_i}(\vec{x}, \vec{\mu_i})$:

$$\beta_i(\vec{x}) \quad = \quad 1/(1 + d_{Q_i}(\vec{x}, \vec{\mu_i})) \tag{4}$$

The sum-product inference is then used to compute the system output for each class:

$$y^m(\vec{x}) = \sum_{i=1}^{R} \beta_i(\vec{x})\ l_i^m(\vec{x}) \tag{5}$$

The winning class label is given by finding the maximal output and taking the corresponding class label as response:

$$class(\vec{x}) = y = argmax\ y^m(\vec{x}) \qquad m = 1, .., k \tag{6}$$

4 Proposed on-line incremental learning model

4.1 Incremental local clustering

In an online incremental learning problem, training data become available continuously, and the system must be learned using the first-arrived data, and then continue to evolve in a transparent manner from the user viewpoint. Classic clustering algorithms need to know all the objects in order to perform the clustering and each time we modify the set of objects, it is necessary to cluster the whole collection again. As mentioned in the introduction, a very important criterion of an efficient incremental learning solution is avoiding (or minimizing) access to previous learning data. Thus, we need algorithm able to update the clusters each time a new data become available, without neither rebuilding the whole set of clusters, nor requiring access to previous data. We use the term "local" because the whole learning process is supervised, but we use an incremental clustering algorithm within the group of samples of each class to incrementally build the fuzzy prototypes.

When introducing a new training sample in an online learning mode, it will either reinforce the information contained in the previous data and represented by the current clustering, or bring enough information to form a new cluster or modify an existing one. The importance of a given sample in the clustering process can be evaluated by its *potential* value. The potential of a sample is defined as inverse of the sum of distances between a data sample and all the other data samples Yager & Fileu (1993):

$$Pot_k(x(k)) = \frac{1}{1 + \sum_{i=1}^{k-1} \|x(k) - x(i)\|^2} \tag{7}$$

A recursive method for the calculation of the potential of a new sample was introduced in Angelov & Filev (2004), which made this technique a promised solution for

any incremental clustering problem. The recursive formula avoids memorizing the whole previous data but keeps - using few variables - the density distribution in the feature space based on the previous data:

$$Pot_k(x(k)) = \frac{k-1}{(k-1)\alpha(k) + \gamma(k) - 2\zeta(k) + k - 1} \tag{8}$$

where

$$\alpha(k) = \sum_{j=1}^{n} x_j^2(k) \tag{9}$$

$$\gamma(k) = \gamma(k-1) + \alpha(k-1), \quad \gamma(1) = 0 \tag{10}$$

$$\zeta(k) = \sum_{j=1}^{n} x_j(k)\eta_j(k), \quad \eta_j(k) = \eta_j(k-1) + x_j(k-1), \quad \eta_j(1) = 0 \tag{11}$$

Introducing a new sample affects the potential values of the centers of the existing clusters, which can be recursively updated by:

$$Pot_k(\mu_i) = \frac{(k-1)Pot_{k-1}(\mu_i)}{k - 2 + Pot_{k-1}(\mu_i) + Pot_{k-1}(\mu_i)\sum_{j=1}^{n} \|\mu_i - x(k-1)\|_j^2} \tag{12}$$

If the potential of the new sample is higher than the potential of the existing centers then this sample will be a center of a new cluster (and a new fuzzy rule will be formed in the case of our FIS). If the sample with high potential is close to an existing center μ_i, then this sample will replace μ_i and no new cluster will be created.

4.2 Incremental learning of linear consequent parameters

The consequents parameters learning problem in a first-order FIS can be solved by the weighted Recursive Least Square method (wRLS) Angelov *et al.* (2008). Let Π be the matrix of all the linear consequents parameters in first-order FIS :

$$\Pi = \begin{bmatrix} \vec{\pi}_1^1 & \vec{\pi}_1^2 & ... & \vec{\pi}_1^m \\ \vec{\pi}_2^1 & \vec{\pi}_2^2 & ... & \vec{\pi}_2^m \\ ... & ... & ... & ... \\ \vec{\pi}_r^1 & \vec{\pi}_r^2 & ... & \vec{\pi}_r^m \end{bmatrix} \tag{13}$$

where m is the number of classes, and r is the number of fuzzy rules; It can be globally and incrementally estimated by:

$$\Pi_k = \Pi_{k-1} + C_k\psi_k(Y_k - \psi_k^T\Pi_{k-1}) \; ; \quad \Pi_1 = 0 \tag{14}$$

$$C_k = C_{k-1} - \frac{C_{k-1}\psi_k\psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1}\psi_k} \; ; \quad C_1 = \Omega I \tag{15}$$

where $\psi_k = [\beta_1(\vec{x}_k)\vec{x}_k, \beta_2(\vec{x}_k)\vec{x}_k, ..., \beta_r(\vec{x}_k)\vec{x}_k]$ is a vector of the inputs that are

weighted by the activation levels of the prototypes, Ω is a large positive number, and I is the identity matrix.

When a new rule is added to the system, its consequents parameters are initialized by the weighted average of the parameters of the other rules, while the parameters of the other rules do not change:

$$\Pi_k = \begin{bmatrix} \vec{\pi}^1_{1(k-1)} & \vec{\pi}^2_{1(k-1)} & \cdots & \vec{\pi}^m_{1(k-1)} \\ \vec{\pi}^1_{2(k-1)} & \vec{\pi}^2_{2(k-1)} & \cdots & \vec{\pi}^m_{2(k-1)} \\ \cdots & \cdots & \cdots & \cdots \\ \vec{\pi}^1_{r(k-1)} & \vec{\pi}^2_{r(k-1)} & \cdots & \vec{\pi}^m_{r(k-1)} \\ \vec{\pi}^1_{(r+1)k} & \vec{\pi}^2_{(r+1)k} & \cdots & \vec{\pi}^m_{(r+1)k} \end{bmatrix} \tag{16}$$

where

$$\vec{\pi}^c_{(r+1)k} = \sum_{i=1}^{r} \beta_i(\vec{x}_k)\vec{\pi}^c_{i(k-1)} \tag{17}$$

In addition, the covariance matrix C is extended as follows:

$$C_k = \begin{bmatrix} \rho \begin{bmatrix} C_{k-1} \end{bmatrix} & \begin{bmatrix} 0 \end{bmatrix} \\ \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} \Omega & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \Omega \end{bmatrix} \end{bmatrix} \tag{18}$$

where $\rho = (r^2+1)/r^2$.

4.3 Premise adaptation

As it can be noted in section 4.1, the condition to have a high potential is a very hard one, and it is inversely proportional to the growing number of data. In this way, we can imagine a cluster center $\vec{\mu_i}$ which is not really in the optimal center position according to the data history, but it keeps to be the center because it still have the highest potential value. Therefore, incremental clustering process of the premise part of the FIS will not be able to take advantage of the data points that do not have a very high potential to move (or reshape) the existing clusters.

We enhance the incremental clustering process (described in section 4.1) by an adaptation algorithm which allows modifying all the prototypes of the FIS by re-centering and re-shaping them for each new data point. Our algorithm is inspired by the Fuzzy Learning Vector Quantization (FLVQ) used for a supervised fuzzy competitive prototype-based learning Chung & Lee (1994). In this method, the farther the activation β_i of the premise of the rule i is away from its objective score β^*_i, the more it must be moved:

$$\Delta\vec{\mu}_i = \lambda * (\beta^*_i - \frac{\beta_i(\vec{x})}{\sum_{q=1}^{r}\beta_q(\vec{x})}) * (\vec{x} - \vec{\mu_i}) \tag{19}$$

where the adaptation parameter λ lies between 0 and 1. The objective score β^*_i is 1 if the

prototype P_r and $\vec{x}$ belong to the same class and 0 otherwise. The main disadvantage of directly using this method in our FIS is that it only takes into consideration the objective score of the prototype and not its participation in the final output score. Therefore, our adaptation algorithm allows modifying all the prototypes of the FIS for each new data sample according to their participation in the recognition process (not to their objective activations). Thus, the update of a prototype must improve the score of each class, so the displacement $\Delta\vec{\mu}_i$ must be significant if the class score y^c is different from the objective class score $\hat{y}^c$, the participation of the prototype to the final class score y_i^c is high and the rule premise β_i is highly activated:

$$\begin{aligned}\Delta\vec{\mu}_i &= \lambda * \left(\beta_i(\vec{x}) \sum_{c=1}^{m} \left((\hat{y}^c(\vec{x}) - y^c(\vec{x}))\ y_i^c(\vec{x})\right)\right) * (\vec{x} - \vec{\mu_i}) \\ &= \lambda * \left(\beta_i^2(\vec{x}) \sum_{c=1}^{m} \left((\hat{y}^c(\vec{x}) - y^c(\vec{x}))\ \vec{l}_i^c(\vec{x})\right)\right) * (\vec{x} - \vec{\mu_i}) \qquad (20)\end{aligned}$$

where $\hat{y}^c$ is 1 if c is the class of $\vec{x}$ and 0 otherwise.

As mentioned in section 4.1, when the potential of the new sample is higher than the potential of the existing centers then this sample will be a center of a new prototype P_{r+1}, and the covariance matrix $Q_(r+1)$ that defines the influence zone (according to the Mahalanobis distance) will be initialized by a matrix proportional to identity matrix (which results is a hyper-spherical shape).

In order to better represent the repartition of the data cloud that incrementally formed the prototype, the shape of the prototype which is given by its covariance matrix must be adapted and updated according to each new data sample. A recursive formula to update the inverse of the covariance matrix in an unsupervised context is given in De Backer & Scheunders (2001) and based on Elliptical Fuzzy Competitive Learning (EFCL). EFCL uses the activation of the prototypes in the covariance matrix calculation. In Mouchere *et al.* (2007), a new version of this formula is presented by using, in addition to the activation of the prototypes, the participation of the prototype on the recognition of each class and the error made on each class. Based on the formula presented in Mouchere *et al.* (2007), and by adapting it to our architecture (first-order FIS), we get the following recursive formula to update the inverse of the covariance matrix Q_i:

$$Q_i^{-1} \Leftarrow \frac{Q_i^{-1}}{1-\alpha\delta_i} - \frac{\alpha\delta_i}{1-\alpha\delta_i} \cdot \frac{(Q_i^{-1}\vec{d}) \cdot (Q_i^{-1}\vec{d})^T}{1+\alpha\delta_i(\vec{d}^T Q_i^{-1}\vec{d})} \qquad (21)$$

$$\delta_i = \beta_i^2(\vec{x}) \sum_{c=1}^{m} \left((\hat{y}^c(\vec{x}) - y^c(\vec{x}))\ \vec{l}_i^c(\vec{x})\right) \qquad (22)$$

with $\vec{d} = \vec{x} - \vec{\mu_r}$ and α is the learning rate.

4.4 Linear consequent parameters stability

One of the favorable properties of the weighted recursive least squares method is that it converges to the optimal solution for each learning step. But, this property is only

true when the weights (premise activations in our case) which were associated to the previous data points keep unchanged. In our system, the fuzzy prototypes that form the antecedent part are dynamic, they can be re-centered, shifted or reshaped according to the new data sample by the incremental clustering method (section 4.1) and the adaptation algorithm (section 4.3). Therefore, the older activations that have participated in the learning of the consequent parameters for the previous data are no longer the same. This makes the prior estimated linear consequent parameters non-optimal for the current premise model.

In Lughofer (2008), the author points out this stability problem. He theoretically proposes to introduce, after each premise modification, a correction vector for the linear parameters and a correction matrix for their covariance matrix in order to balance out the current non-optimal solution toward the optimal one according to the degree of change in the premise part. To the best of our knowledge, there is not yet any proposed solution to estimate these correction vectors and matrices, it is still an open and sophisticated issue.

To cope with the linear parameters stability problem in our FIS, we apply the incremental clustering and the adaptation algorithm on the premise part in a batch-wise mode instead of a sample-wise mode. In this way, we keep on applying the consequent parameters updating algorithm in a sample-wise mode with fixed premise structure. In addition, we keep the data samples in a memory buffer with size F. Thus, after introducing F data samples, the incremental clustering and the adaptation algorithm are applied for each sample in the buffer. Then, a readjustment of the consequent parameters is done by applying the wRLS method on the samples in the buffer (with the modified premise structure). After this readjustment process, the buffer is emptied out and the premise structure "freezes" while the consequent parameters updating process continues in a sample-wise mode.

The complete learning algorithm can be summarized by Algorithm 1.

5 Evaluation

We evaluate the performance of our learning model on two different problems. We will particularly focus in our experiments on three points: (a) the rapidity of the performance improvement in the beginning of the incremental learning process (for the first few samples), (b) the overall performance of our learning model in the long term compared to well-known incremental and non-incremental algorithms and (c) the stability and the recovery speed of the performance when introducing a new unseen class.

The following parameters values were used in the next two experiments: $\lambda = 0.05$, $\alpha = 0.005$ and the data buffer has a length of 30 samples.

5.1 Experiments on real data: Online handwritten symbols

We led this experiment on datasets of nine online handwritten electrical symbols (figure 1). Each symbol is described by a set of 21 features. We run 12 different tests on 12 writer-specific datasets drawn by 12 different persons on Tablet PCs. Each writer has drawn 50 samples of each symbol, i.e. 450 symbols in each writer-specific dataset.

Algorithm 1: First-order FIS online incremental learning algorithm

foreach *new sample* $\vec{x}$ **do**
  if $\vec{x}$ *is the first sample of a new class* **then**
    create a new fuzzy prototype based at $\vec{x}$;
    initialize its potential by 1;
    add a new fuzzy rule to the system;
    extend the consequent parameters matrix as in [16] and [17];
    update and extend the covariance matrix as in [18];
  else
    calculate the activations of the fuzzy rules by [4];
    determine the winning class label by [6];
    get the true class label;
    calculate the potential of $\vec{x}$ by [8];
    update the potentials of the existing prototypes centers using [12];
    if $Pot(\vec{x}) > Pot_k(\vec{\mu}_i) \ \ \forall i \in [1, R]$ **then**
      if $\vec{x}$ *is close to an existing center* $\vec{\mu}_i$ **then**
        let $\vec{x}$ be the new center of the prototype P_i and keep the same consequents of the $rule_i$;
      else
        create a new fuzzy prototype based at $\vec{x}$;
        initialize its potential by 1;
        add a new fuzzy rule to the system;
        extend the consequent parameters matrix as in [16] and [17];
        update and extend the covariance matrix as in [18];
      end
    end
    update the consequents parameters using [14] and [15];
    add $\vec{x}$ to the memory buffer;
    if *memory buffer is full* **then**
      foreach *sample* $\vec{x}_b$ *in the buffer* **do**
        Apply premise adaptation according to $\vec{x}_b$ by [20] and [21];
      end
      foreach *sample* $\vec{x}_b$ *in the buffer* **do**
        update the consequents parameters according to $\vec{x}_b$ by [14];
      end
      dump the memory buffer;
    end
  end
end

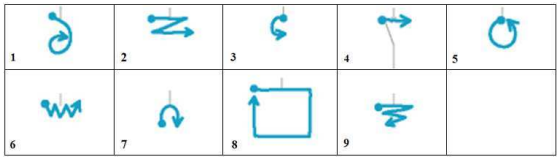


Figure 1: The used handwritten electrical symbols.

We used three quarters of the dataset for the incremental learning process and the last quarter is used to estimate the evolution of the system's performance during the learning process. Four fuzzy incremental learning models (with different architectures or different learning algorithms) are compared in this experiment:

- Model I: presented in Almaksour & Anquetil (2009), and briefly described in section 2.
- Model II: first-order FIS with incremental "potential-based" clustering and wRLS method (Angelov *et al.* (2008)).
- Model III: model II + premise adaptation in sample-wise mode.
- Model IV: model II + premise adaptation in batch-wise mode + consequents parameters readjustments.

In the first experimental protocol, we introduce the nine classes in the beginning of the learning process. A new sample from each class is presented to the system between each two consecutive evaluation points. Figure 2 shows the evolution of the recognition rate in the beginning of the incremental learning process using the four models. The presented results are the average of 12 tests for the 12 writers. In order to have referential values in the evaluation of the recognition rate on the used dataset, we trained two well-known non-incremental classifiers using the whole training set in batch mode, and we measured their performance on the test dataset. We choose a Multi-Layer Perceptron (MLP) classifier with one hidden layer and a K-Nearest Neighbor (K-NN, with K=5) classifier.

We note from the figure 2 that using the model IV the performance improves rapidly with very few data. For instance, a recognition rate greater than 91% is reached after only 5 samples per class, and it rapidly improves and exceeds 94% after 10 data sample per class. By comparing the performance of the model IV with that of the model II in the figure 2, we note that integrating the premise adaptation in the model helps to boost and to accelerate the incremental learning process especially at the beginning when only few learning data were introduced. This property is very important in our application context because the user needs to be able to use the new gesture as soon as possible, and that the performance of the system becomes satisfactory as fast as possible. It is not surprising that the curves of the two models meet after introducing a significant number of samples per class. This is because the proposed classification problem is relatively easy, and the model II, even without an optimal premise structure, can finally meet the performance of the model IV. We note also from the same figure the instability

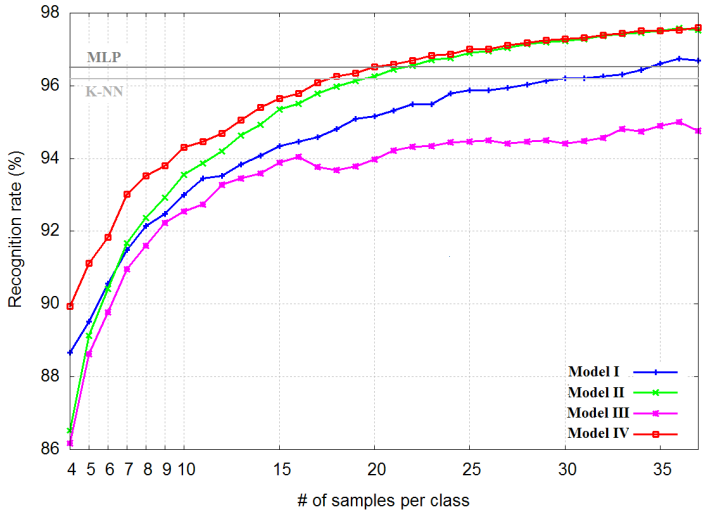
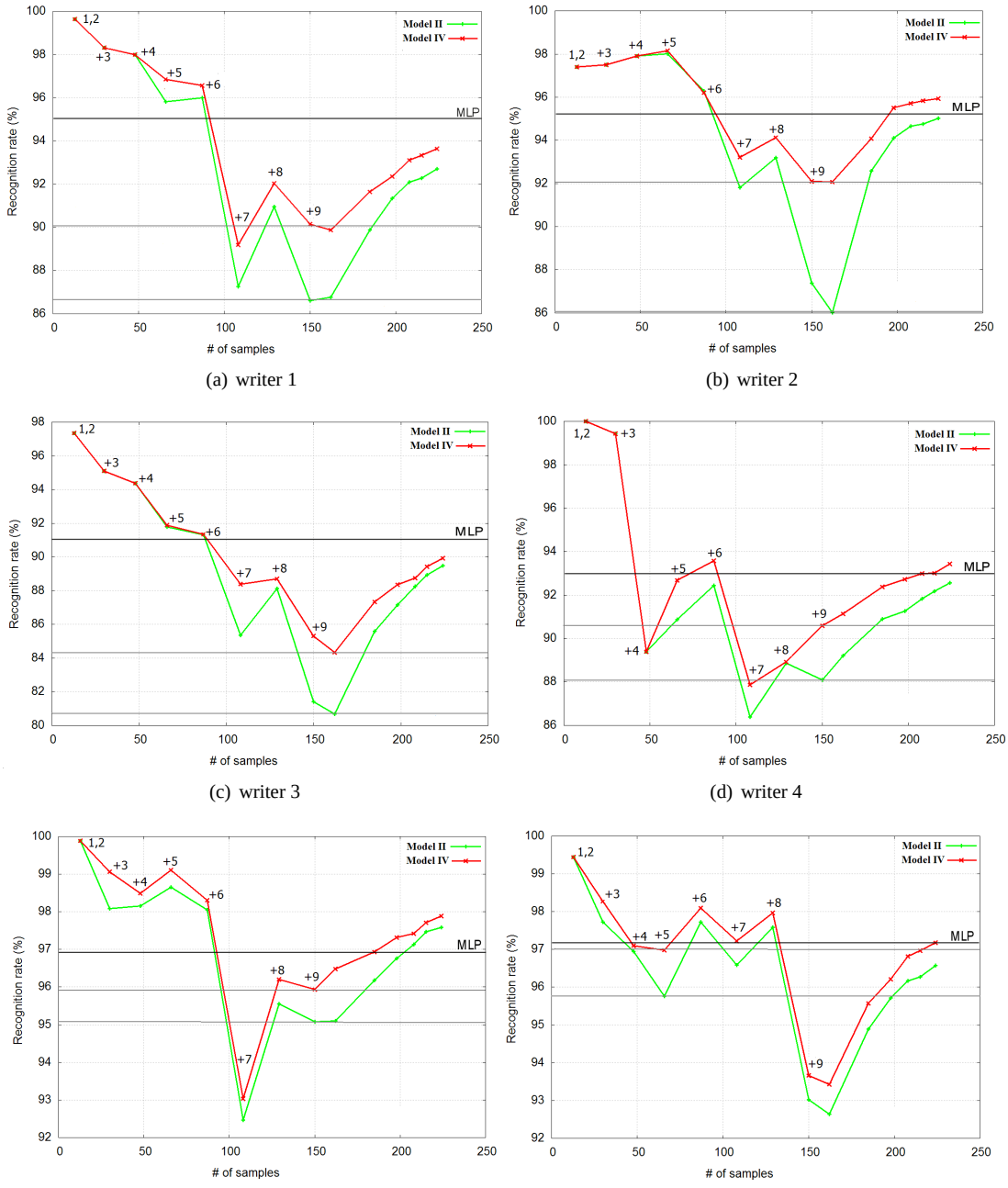


Figure 2: Evolution of the recognition rate at the beginning of the incremental learning process.

of the performance of the model III, which is due to the continuous disruption of the consequent parameters because of the sample-wise premise adaptation. The model I has a good performance at the beginning of the learning, but as we mentioned in section 2, at the price of creating so many prototypes in the system, for instance, there are 90 fuzzy prototypes (so 90 fuzzy rules) in the model I after introducing 315 samples (9x35), while for the other three models there are in average less than 11 prototypes. A high number of prototypes can accelerate the performance at the beginning, but the system becomes rapidly too sophisticated and less capable to learn and to be adapted. In the second experimental protocol, we emulate the real application context in which the model starts with few classes of gestures, and the user adds progressively new gestures according to his needs. We aim to study the behavior of the model and its ability to learn new class of data without fully destroying the knowledge learned from old data. Thus, the learning process starts with only two classes of gestures, and then, few samples of a new unseen gesture and some samples of the already learned classes are introduced between each two consecutive evaluation points. The Results for six different writers are shown in figure 3. We note that thanks to the premise adaptation techniques that enhance significantly the intrinsic part of the fuzzy system, the system can resist better when introducing a new class which sharply perturbs the consequents parameters.

5.2 Experiments on benchmark data

We used in this experiment the benchmark pen-based dataset from the UCI Machine Learning Repository Asuncion & Newman (2007). The dataset contains 7494 training samples and 3498 test samples. Each sample is represented by 16 features. The set contains 10 classes for the 10 digits. It was created by collecting 250 samples from 44

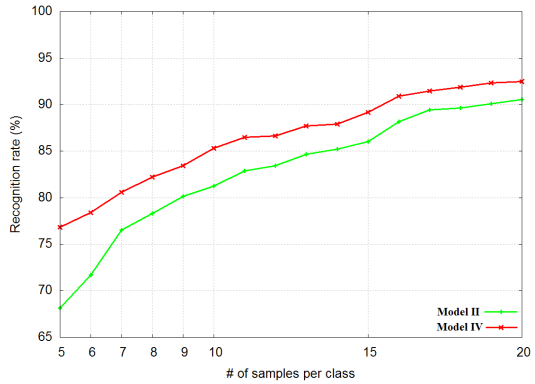


(a) writer 1 (b) writer 2

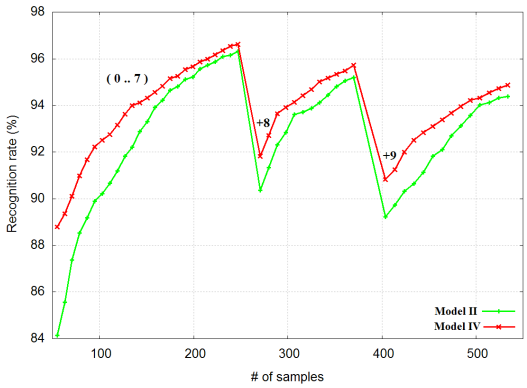
(c) writer 3 (d) writer 4

(e) writer 5 (f) writer 6

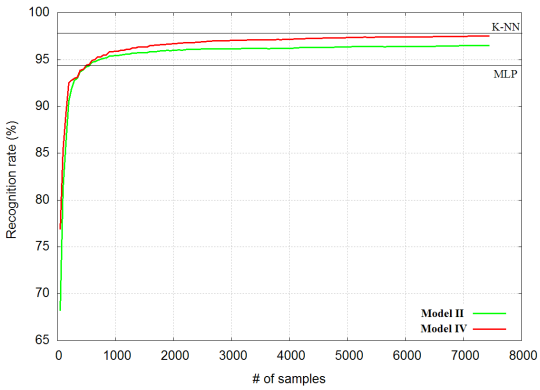
Figure 3: Performance stability and recovery when introducing new unseen classes for six different writers.



(a)



(b)



(c)

Figure 4: (a) Evolution of the recognition rate at the beginning of the incremental learning process. (b) Performance stability and recovery when introducing new unseen classes. (c) Evolution of the recognition rate in long-term incremental learning process.

writers and using a WACOM PL-100V pressure sensitive tablet with an integrated LCD display and a cordless stylus. We can note from the figure 4(a) that the recognition rate that can be achieved by the system in the beginning of the online learning process improves better when using model IV rather than model II. It is proved by figure 4(b) that a better stability and a faster performance recovery after introducing new classes are achieved using model IV when compared to model II. Furthermore, we can note from the figure 4(c) that the proposed incremental one-pass learning model can achieve or exceed the performance of some well-known iterative learning algorithms. We notice also the continuous performance progress in the long term and during the whole incremental process.

6 Conclusion

In the context of handwritten gestures recognition system, we presented in this paper a new fast online incremental learning algorithm based on first-order fuzzy inference system. Thanks to this algorithm, the recognition system can start to learn from scratch and with few learning data. It can also improve and adapt to each newly available data. The good performance of the presented model is validated in two different classification problem and with different incremental learning scenarios. As a future work, special emphasis will be laid on finding an estimation of a correct vector for the linear consequents after each premise adaptation; this can enhance the system performance and eliminate the need for the memory buffer. Another prospect related to the application context is to find techniques to generate artificial handwritten gestures starting from original one, in order to boost the learning process when only few data are available.

References

AHA D. W., KIBLER D. & ALBERT M. K. (1991). Instance-based learning algorithms. *Mach. Learn.*, **6**(1), 37–66.

ALMAKSOUR A. & ANQUETIL E. (2009). Fast incremental learning strategy driven by confusion reject for online handwriting recognition. In *Tenth International Conference on Document Analysis and Recognition (ICDAR 2009)*, p. 81–85.

ALMAKSOUR A., MOUCHÈRE H. & ANQUETIL E. (2008). Fast online incremental learning with few examples for online handwritten character recognition. In *Proceedings of the Eleventh International Conference on Frontiers in Handwriting Recognition (ICFHR'08)*, p. 623–628, Montréal, Québec, Canada.

ANGELOV P. & FILEV D. (2004). An approach to online identification of takagi-sugeno fuzzy models. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **34**(1), 484–498.

ANGELOV P., LUGHOFER E. & ZHOU X. (2008). Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets Syst.*, **159**(23), 3160–3182.

ASUNCION A. & NEWMAN D. (2007). UCI machine learning repository.

CARPENTER G., GROSSBERG S., MARKUZON N., REYNOLDS J. & ROSEN D. (1992). Fuzzy artmap: A neural network architecture for incremental supervised

learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, **3**.

CARPENTER G. A. & GROSSBERG S. (1988). The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, **21**(3), 77–88.

CHUNG F. & LEE T. (1994). Fuzzy competitive learning. *IEEE Transaction on Neural Network*, **7**(3), 539–551.

DE BACKER S. & SCHEUNDERS P. (2001). Texture segmentation by frequency-sensitive elliptical competitive learning. *Image and Vision Computing*, **19**(9–10), 639–648.

GARY G. YEN P. M. (2001). An effective neuro-fuzzy paradigm for machinery condition health monitoring. *IEEE Transactions on Systems, Man, and Cybernetics*, **31-4**, 523 – 536.

JANG J.-S. (1993). Anfis: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, **23**, 665–685.

JR. J. J. L. & ZELEZNIK R. C. (2007). A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(11), 1917–1926.

KASABOV N. (2007). *Evolving connectionist systems: The knowledge engineering approach (2nd Ed.)*. Springer.

LITTLESTONE N. (1991). Redundant noisy attributes, attribute errors, and linear-threshold learning using winnow. In *COLT '91: Proceedings of the fourth annual workshop on Computational learning theory*, p. 147–156, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

LITTLESTONE N. & WARMUTH M. K. (1994). The weighted majority algorithm. *Inf. Comput.*, **108**(2), 212–261.

LUGHOFER E. (2008). Flexfis: A robust incremental learning approach for evolving takagi-sugeno fuzzy models. *IEEE T. Fuzzy Systems*, **16**(6), 1393–1410.

MOUCHERE H., ANQUETIL E. & RAGOT N. (2007). On-line writer adaptation for handwriting recognition using fuzzy inference systems. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, **21(1)**, 99–116.

POLIKAR R., UDPA L., UDPA S. & HONAVAR V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **31**, 497–508.

REINKE, R. M. R. (1988). Incremental learning of concept descriptions: A method and experimental results. *Hayes, J., Michie, D., Richards, J., eds.: Machine Intelligence*, **11**, 263288.

SADRI J., SUEN C. Y. & BUI T. D. (2006). A new clustering method for improving plasticity and stability in handwritten character recognition systems. *Pattern Recognition, International Conference on*, **2**, 1130–1133.

TAKAGI T. & SUGENO M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE TSMC*, **15**(1), 116–132.

YAGER R. R. & FILEU D. P. (1993). Learning of fuzzy rules by mountain clustering. volume 2061, p. 246–254: SPIE.

ZWICKEL J. & WILLS A. J. (2005). *New Directions in Human Associative Learning*, chapter Integrating associative models of supervised and unsupervised categorization, p. 118. Psychology Press.