A dynamic classification method for the discrimination of evolving data

Laurent Hartert Moamar Sayed Mouchaweh Patrice Billaudel

Université de Reims Champagne-Ardenne, CReSTIC,
Moulin de la Housse, BP 1039, 51687 Reims Cedex, France
Email: laurent.hartert@univ-reims.fr, moamar.sayed-mouchaweh@univ-reims.fr, patrice.billaudel@univ-reims.fr

Abstract*—Classes issued of evolving systems are dynamic and their characteristics vary over the time. Assigning a pattern to a class is achieved using a classifier. Therefore, the classifier parameters must be adapted online in order to take into account the temporal changes in the classes' characteristics. This adaptation is based only on the recent and useful information carried out by the new incoming classified patterns. In this paper, we propose to develop the classification method Fuzzy Pattern Matching (FPM) to be operant in the case of dynamic classes. This development is based on the use of an incremental algorithm to follow the accumulated gradual temporal changes of classes' characteristics after the classification of each new pattern. When these changes reach a suitable predefined threshold, the classifier parameters are adapted online using the recent and representative patterns.*

Keywords— Classification, dynamic patterns, evolving systems, pattern recognition.

1 Introduction

Dynamic systems evolve between different normal or faulting functioning modes in the course of time. In statistical Pattern Recognition (PR) [12, 9], observations about system functioning modes are divided into groups of similar patterns, called classes, using an unsupervised learning method [10, 4] or human experience. These patterns, with their class assignments, constitute the learning set. They are represented by a set of d features, or attributes, so they can be viewed as d-dimensional features vectors, or points, in the feature space. A supervised learning method [9, 16] uses the learning set to build a classifier that best separates the different classes in order to minimize the misclassification error. The model of each class can be represented by a membership function which determines the membership value of a pattern to a class. Then, new incoming patterns are assigned to the class for which they have the maximum membership value. In supervised learning methods, the membership function can be generated using Probability Density Function (PDF) estimation based methods or heuristic based ones.

Patterns describing the system functioning can be static or dynamic. A static pattern is represented by a point in the feature space while a dynamic pattern is represented by a multidimensional trajectory. In this case, the feature space has an added dimension which is the time [3]. Classes can also be static or dynamic. Static classes are represented by restricted areas formed by similar static patterns in the feature space. Hence, the way in which patterns occur is irrelevant to their membership values. Therefore, the classifier's parameters remain unchanged with the time. However, data issued from evolving processes are non stationary. In this case, classes become dynamic and their characteristics change in the course of time. Thus, the classes' membership functions must be adapted to take into account these temporal changes. This requires an adaptive classifier with a mechanism for adjusting its parameters over the time. Hence, some of the new incoming points reinforce and confirm the information contained in the previous data, but the other ones could bring new information (creation, drift, fusion, splitting of classes, etc.). This new information could concern a change in operating conditions, development of a fault or simply more significant changes in the process dynamic.

The general principle of dynamic PR methods [14, 3, 7, 13] is to observe the change of some statistical properties of classes, in order to decide in which state the system is: unchanged, gradually changed or completely changed. Thus, the classifier parameters, i.e. the membership functions, will be respectively unchanged, slightly adapted, or relearned from scratch.

In the literature, the membership functions are generally adapted using two approaches [2, 3, 14]. The first one acts directly on the classifier parameters, by substituting or adding some recent and representative patterns to the learning set [3, 14], according to the state in which the system is. This adaptation is based only on the most recent batch of patterns selected by one of the two following methods. The first method uses a time window, with a fixed or a variable size, which permits to reduce or limit the growing size of the database by accepting the n most recent patterns [14, 17]. The size of the time window must be well-chosen to obtain a compromise between a fast adaptation and a sufficient number of representative patterns. The second method is based on the use of a template containing a fixed number of selected patterns according to their age and usefulness [11]. Nevertheless, it is subjective and difficult to estimate the utility of patterns. The second approach to adapt the membership functions is based on the use of evolving neural networks [1, 2, 7]. In [2], a potential function based on the distance between data points is defined for the new points. The first data point potential is considered as equal to 1 and it establishes the first neuron (or rule) which is considered as the prototype (or centre) of the first cluster. Then, the next new data points may possess a potential close or greater than the one of the prototype neuron. This point can reinforce or confirm the information contained in the previous ones, or if the point is more informative than the data used as prototype, a new neuron (new rule) is added. In [1], the neural network is based on a multiprototype Gaussian modeling of non convex classes. The activation

ISBN: 978-989-95079-6-8

function of each hidden neuron determines the membership degree of an observation to one prototype of a class. With the first acquisition, the network is initialized there is creation of the first prototype constituting the first class. The prototype is parameterized by its centre and an initial covariance matrix. Then, according to the membership degree of new acquisitions, the prototype (the hidden neuron) can be adapted, eliminated or a new prototype can be created.

Dynamic classification methods are used to solve several real problems. The application of [3] concerns the credit-scoring which aims to decide whether a new customer is a good or a bad risk according to changes in his consumption. In [1], the authors aim to detect and to follow up the progressive evolution of the functioning mode from normal to faulty one of a thermal regulator. This evolution is due to the age of the system's components or to other temporal factors in its environment. In [13], the author looks to estimate a ratio between the number of good parts in a completed batch of microelectronic chips and the number of parts obtained from the same batch if there is no default. Due to high complexity and variability of modern microelectronics manufacturing industry, this ratio is affected daily by hundreds of material-related, equipment-related and human-related factors. In [7], dynamic traffic data streams are treated in order to reduce the waiting time of drivers at the road intersections.

We use the classification method Incremental Fuzzy Pattern Matching (IFPM) [15] which is an incremental version of FPM [6]. The membership functions are based on the estimation of the marginal class probability density functions using histograms. Then, the membership functions are adapted incrementally after the classification of each new pattern. IFPM provides good result in the case of static classes based on static patterns. This method is simple and has a low and constant classification time according to the size of the database. However, IFPM has no mechanism to forget the no more useful patterns.

In this paper, we propose to develop IFPM to be operant in the case of dynamic classes. This development is called Dynamic FPM (DFPM). The paper is structured as follows. The two first parts are devoted respectively to the functioning of IFPM and DFPM. In the third part, the limits of IFPM and results of DFPM are presented in the case of dynamic classes. The last part concludes the paper and presents the perspectives of our future work.

2 Incremental Fuzzy Pattern Matching

2.1 Principle

The functioning of Incremental FPM (IFPM) involves the learning, the classification, and the incremental update phases.

2.1.1 Learning phase

Let X_i be the set of learning points belonging to the class C_i. Each class C_i contains N_i points, or patterns, in the feature space, $\Re^d$, formed by d attributes. X is the learning set, containing N points x belonging to C classes, so $X = \bigcup_{i=1}^{C} X_i$. The learning phase consists in building, based on X, a decision rule characterized by a set of membership functions. These latter are based on histograms allowing to estimate the conditional probability density of each class C_i according to each attribute j. The number h of bins b_k, $k \in \{1,2,..., h\}$ of a histogram is determined experimentally. The choice of h conditions the performances of IFPM. The lower $\min^j$ and upper $\max^j$ borders of each attribute are determined manually by an expert. In the case of dynamic classes, these borders must be well defined to contain all patterns of the classes' evolutions. This constitutes a shortcome of IFPM when the classes are dynamic. The width Δ^j of a bin according to the attribute j is defined by:

$$\Delta^j = \frac{(\max^j - \min^j)}{h} \tag{1}$$

The histogram or the distribution of probability $\{p_i^j(b_{ik}^j), i \in \{1,2,...,c\}, j \in \{1,2,...,d\}, k \in \{1,2,...,h\}\}$ for the class C_i according to the attribute j is determined by calculating the probability $p_i^j(b_{ik}^j)$ of each bin b_{ik}^j:

$$p_i^j(b_{ik}^j) = \frac{n_{ik}^j}{N_i} \tag{2}$$

where n_{ik}^j is the number of learning points of the class C_i which are located in the bin b_{ik}^j and N_i is the total number of points of the class C_i. Then, the resulting distribution of probability is transformed into a distribution of possibility $\{\pi_i^j(b_{ik}^j), i \in \{1,2,...,c\}, j \in \{1,2,..., d\}, k \in \{1,2,..., h\}\}$ by using the transformation of Dubois and Prade [8]:

$$\pi_i^j(b_{ik}^j) = \sum_{z=1}^{h} \min(p_i^j(b_{iz}^j), p_i^j(b_{ik}^j)) \tag{3}$$

The possibility measure has the advantage to take into account the imprecision and the uncertainty contained in the data [8]. Finally, the density of possibility Π_i^j of the class C_i according to the attribute j is obtained by a linear interpolation of the bins centers of the histogram of possibility.

2.1.2 Classification phase

A pattern x is assigned to a known class C_i using the following three steps:

- determination of the membership possibility value π_i^j of x^j to each class C_i according to each attribute j. This possibility is obtained by projection of x^j on the density Π_i^j of each class C_i according to each attribute j (Fig. 1).

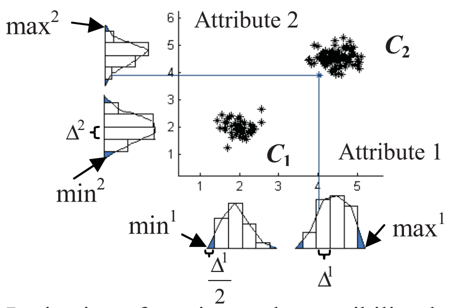

Figure 1: Projection of a point on the possibility densities. Δ^1 and Δ^2 are the bins' widths according respectively to attributes 1 and 2.

ISBN: 978-989-95079-6-8

- fusion, for each class C_i, of all membership possibility values $\pi_i^1, \pi_i^2, ..., \pi_i^d$ by the aggregation operator "minimum". The result of this fusion represents the membership possibility π_i of x to each class C_i,
- classification of the point x to the class for which it has the highest membership value.

2.1.3 Incremental update phase

Each new classified pattern carries information about the temporal changes of the class' characteristics. In order to take into account this new information, the membership functions must be relearned after the classification of each new pattern. An incremental learning algorithm is proposed in [15] to relearn the membership functions based on the most recently classified pattern in such a way that the previous learned information is reused. When a new point is classified in the class C_i, the number of points belonging to this class becomes $N_i + 1$, so the probability of each bin changes. If the point is located in the bin b_{ik}^j, the new probability $p_i'^j(b_{ik}^j)$ of this bin is [15]:

$$p_i'^j(b_{ik}^j) = \frac{n_{ik}^j}{N_i} \times \frac{N_i}{N_i+1} + \frac{1}{N_i+1} = p_i^j(b_{ik}^j) \times \frac{N_i}{N_i+1} + \frac{1}{N_i+1} \quad (4)$$

For each other bin, the new probability is:

$$p_i'^j(b_{iz}^j) = \frac{n_{iz}^j}{N_i} \times \frac{N_i}{N_i+1} = p_i^j(b_{iz}^j) \times \frac{N_i}{N_i+1}, \; z = 1 \ldots h \;\&\; z \neq k \quad (5)$$

Using (4) and (5), the probability histograms can be updated after the classification of a new point without the need to recalculate them. We just need to know in which bin the new point is located. The histogram of possibility is then calculated by (3). IFPM has been developed to deal with data represented in a feature space of correlated attributes as the case of XOR database [5]. In the next part, the improvements realized on IFPM are presented.

3 Dynamic Fuzzy Pattern Matching

We have called Dynamic Fuzzy Pattern Matching (DFPM) the improved version of IFPM. DFPM integrates a mechanism to adjust the classifier's parameters when serious changes in the classes' characteristics are detected during a growing time window. The size of the time window depends of the application's dynamic. The next subsections detail the two phases of DFPM's functioning.

3.1 Detection phase

In the detection phase, three indicators are used to monitor the temporal changes of a system. The first indicator is a measure evaluating the bins' usefulness. This indicator is based on the accumulated temporal changes which have occurred in the probability distributions during a variable time sliding window. It considers the number of new points located in each bin of the probability histograms, as well as its initial probability before the assignment of the new pattern(s). Let $\Delta T = t_w - t_s$ be the growing size of the time window at the present time t_w. This time window starts at the instant t_s. Let $\{p_i^j(b_{ik}^j), k \in \{1,2,..., h\}\}$ be the probability histogram for the class C_i according to the attribute j at the instant t_s. The probability of each bin is calculated by (2). Let ΔN_i be the number of new classified points in the class C_i and let Δn_{ik}^j be the number of new points located in the bin b_{ik}^j at the present time t_w. The new probability of this bin is calculated by:

$$p_i'^j(b_{ik}^j) = \frac{n_{ik}^j + \Delta n_{ik}^j}{N_i + \Delta N_i} \quad (6)$$

where n_{ik}^j and N_i are respectively the number of points of the class C_i located in the bin b_{ik}^j and the total number of points of the class C_i before the classification of the new pattern(s). The accumulated temporal changes for each bin are evaluated as the difference between the bin's probabilities at the present time $p_i'^j(b_{ik}^j)$ and at the beginning $p_i^j(b_{ik}^j)$ of the growing time window:

$$\Delta p_i^j(b_{ik}^j) = p_i'^j(b_{ik}^j) - p_i^j(b_{ik}^j) = \frac{N_i.\Delta n_{ik}^j - n_{ik}^j.\Delta N_i}{N_i.(N_i + \Delta N_i)} \quad (7)$$

We define the usefulness measure as a gradual measure, $I_i^j(b_{ik}^j) : \{1,2,...,h\} \to [-1, 1]$ for each bin b_{ik}^j of each class C_i according to each attribute j. This measure evaluates the usefulness of each bin based on the accumulated temporal changes in the probability distributions defined by (7):

$$I_i^j(b_{ik}^j) = \begin{cases} \dfrac{p_i'^j(b_{ik}^j) - p_i^j(b_{ik}^j)}{p_i^j(b_{ik}^j)} \in [-1, 0] \; if \; \Delta p_i^j(b_{ik}^j) < 0 \\ \dfrac{p_i'^j(b_{ik}^j) - p_i^j(b_{ik}^j)}{p_i'^j(b_{ik}^j)} \in [0, 1] \; if \; \Delta p_i^j(b_{ik}^j) > 0 \end{cases} \quad (8)$$

This measure is calculated after the classification of each new classified point. When the value of this indicator is negative for one bin, it indicates a decrease of its usefulness. This decrease is due to the fact that no new classified points, or just a few ones, are located in this bin. On the contrary, the positive values indicate an increase of the bin's usefulness due to the assignment of many new points in the bin, according to its initial value. It is interesting to take into account the old or initial probabilities. Indeed, if there is no important change in the probability distribution, it is normal that the bin with the highest probability receives more new points than the others. If most of the new classified points are located in the bins which possess a small initial probability, this indicates a serious change in the probability distribution and thus their usefulness indicators must have a more important positive value.

To detect an evolution, a second indicator, which is a residual, is calculated based on the usefulness' measure of all histograms' bins according to all attributes. The goal of this residual is to provide one indicator for each class, which evaluates the temporal accumulative changes in the probability distributions of this class. This residual is sensitive for these changes. Two thresholds $th1$ and $th2$ predefined by an expert according to the system dynamic are used to determine the size ΔT of the time window. The value of this residual is updated and compared after each pattern assignment to $th1$ and $th2$. If the residual value is

ISBN: 978-989-95079-6-8

inferior to $th1$, then no serious change in the class' characteristics is yet started. As long as the residual is inferior to $th1$, the measure of usefulness is based on the difference between the current bin probability and the one of the previous classified pattern. If the residual value is superior to $th1$ and inferior to $th2$, a serious change has begun to occur. The time window starts at this moment, $t=t_s$, and patterns are stored in a block, called the evolution block. During this window, the measure of usefulness is calculated using (8). Patterns are stored in the evolution block until the residual value reaches $th2$. At this moment the growing time window reaches its final size $t_w=t_f$. This residual is defined for each class C_i as follows:

$$0 \le R_i = \max\left(\frac{\sum_{k=1}^{h}\left|I_i^j(b_{ik}^j)\right|}{h}\right) \le 1,\ j \in \{1,..,d\} \tag{9}$$

When the residual value reaches $th2$, a serious change can be decided in a class so the classifier parameters (densities of possibility) must be adapted using only the patterns stored in the evolution block. The old patterns which have become obsolete are forgotten and an online update of the classifier's parameters can then be performed.

3.2 Adaptation phase

This phase consists in the adaptation of the probability histograms and the update of the lower and upper attributes' borders as well as the number of bins h. Indeed, in IFPM, these latter are defined initially using (1) and they remain unchanged. For dynamic classes, it is not realistic to define static attributes' borders. Thus, in DFPM these borders will be updated online with the classification of new points. If a new classified point involves the creation of a bin for a histogram, then a bin is added to the histogram according to this attribute so that h is incremented and the histograms' borders are updated:

$$\left[\begin{array}{l} x^j < \min^j - \dfrac{\Delta^j}{2} \Rightarrow \min'^j = \min^j - \Delta^j \\ x^j > \max^j + \dfrac{\Delta^j}{2} \Rightarrow \max'^j = \max^j + \Delta^j \end{array}\right. \tag{10}$$

$$h' = h + 1 \tag{11}$$

The value of Δ^j is defined initially by (1). With this update h can be different according to each attribute. The adaptation of probability histograms is achieved using an incremental approach. Let $\{p_i^j(b_{ik}^j),\ k \in \{1,2,..., h\}\}$ be the probability histogram for the class C_i according to the attribute j. When the residual value of a class reaches $th2$, old patterns are deleted and only the new patterns of the evolution block are considered. So, the value of probability for each bin is calculated incrementally by:

$$p_i''^j(b_{ik}^j) = \frac{\Delta n_{ik}^j}{\Delta N_i} = \left(\frac{n_{ik}^j + \Delta n_{ik}^j}{N_i + \Delta N_i} - \frac{n_{ik}^j}{N_i} \times \frac{N_i}{N_i + \Delta N_i}\right) \times \frac{N_i + \Delta N_i}{\Delta N_i}$$

$$p_i''^j(b_{ik}^j) = \left(p_i'^j(b_{ik}^j) - \left(p_i^j(b_{ik}^j) \times \frac{N_i}{N_i + \Delta N_i}\right)\right) \times \frac{N_i + \Delta N_i}{\Delta N_i},\ k \in \{1,..,h\} \tag{12}$$

This incremental adaptation permits to follow the evolution of classes online with a constant and low classification time. Then, the histogram of possibility is calculated using (3). Fig. 2 illustrates the functioning phases of DFPM.

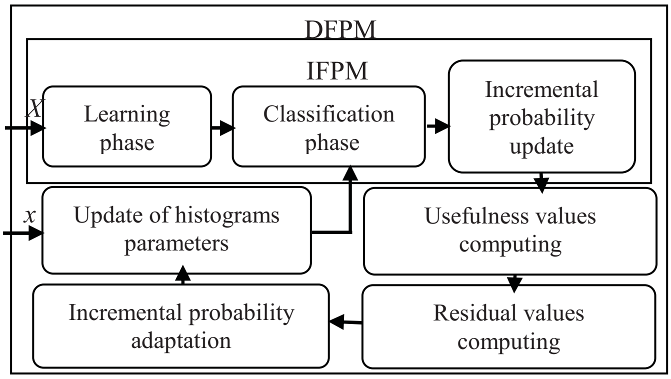

Figure 2: Functioning phases of DFPM.

4 Results with dynamic classes

Classes become dynamic if at least one of their characteristics changes with the time. For the sake of simplicity we have used a set of simulations described in a two dimension space. The obtained results can be extended for higher dimensions. The generated datasets follow a normal distribution of two independent variables (mean and standard deviation). We suppose that one unit of time corresponds to the classification of one point. This hypothesis is verified since DFPM has a constant and low classification time.

4.1 Class drift

Figure 3 shows a drift of a class in the course of time. The drift database has been created as follows:

- $t=0$: the possibility densities are already learned using 100 labeled patterns. The mean values of these data points are $\mu^1=3$ and $\mu^2=3$ (Fig. 3.a).
- $t=1$ to 100: 100 new incoming patterns, possessing the same initial class' characteristics (mean and standard deviation) occur in the class; thus there is no evolution or drift.
- $t=101$ to 200: a sudden change takes place in the class' mean values μ^j according to each attribute j, $j \in \{1,2\}$. This sudden change is followed by a slow drift of the mean value according to each attribute (Fig. 3.b):

$$\left.\begin{array}{l} \mu^{1'}(t) = \mu^1 + 2 + \dfrac{4 \times (t-100)}{100} \\ \mu^{2'}(t) = \mu^2 + 2 + \dfrac{2 \times (t-100)}{100} \end{array}\right] \quad 101 \le t \le 200 \tag{13}$$

- $t=201$ to 300: 100 new patterns with the same characteristics as the ones of the drifted class occur. Fig 3.c presents the 0.1 membership level curve expected for the final drifted class. This curve contains all the points which have a membership possibility value to the class greater or equal to 0.1.

ISBN: 978-989-95079-6-8

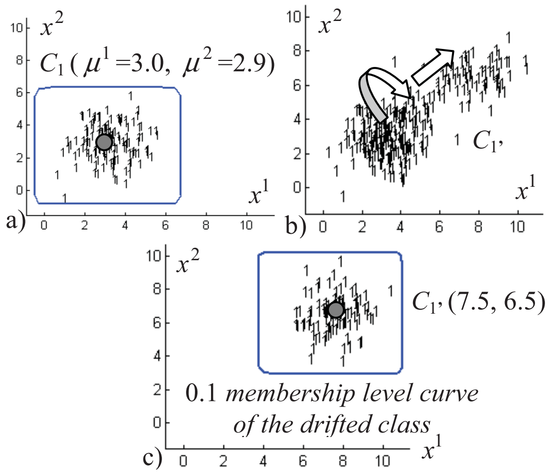

Figure 3: Evolution of the drift for the class C_1. The mean values of the class are presented before and after evolution between parentheses.

Fig. 4 presents the 0.1 membership level curve obtained by IFPM at the end of the class' drift ($t = 200$). We suppose here that each attribute borders were correctly determined in order to include the class drift.

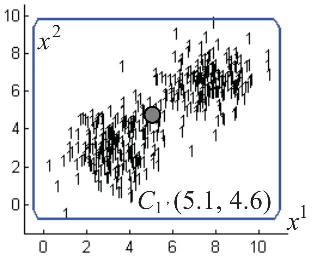

Figure 4: Classification of all patterns of the drifted class by IFPM.

All the patterns classified by IFPM are considered representative. Thus, the membership function, represented by the 0.1 membership level curve in Fig. 4, is not adapted to the drifted class with its new means values (see Fig. 3.c). This is due to the fact that all patterns were considered without eliminating any old and useless pattern.

The DFPM method is evaluated on the same drift of a class generated by (14). The residual values obtained progressively after the classification of each pattern in the class are represented in Fig. 5.

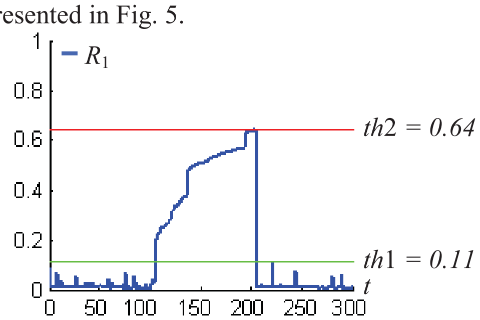

Figure 5: Residual values obtained by DFPM for the class drift.

Based on Fig. 5, we can conclude:

- between t=1 and t=100, the class' evolution has not yet begun.
- at t=105, the evolution of the class is detected. To detect this evolution the $th1$ must be equal to 0.11.
- at t=204, the class drift is finished. To detect the end of this class drift the second threshold $th2$ must be equal to 0.64.
- finally, between t=205 and t=300 no evolution is observed so the residual values are small for the latest patterns and the class is stable. These patterns are classified in the class as usual.

Fig. 6 permits to see the membership level curve obtained by DFPM just before the evolution (Fig. 6.a) and the final result of classification after evolution when the adaptation of the histograms was realized (Fig. 6.b).

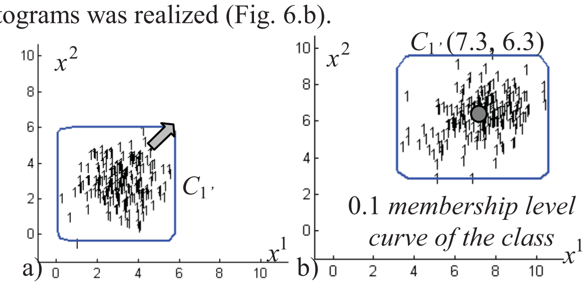

Figure 6: 0.1 membership level curve obtained by DFPM before evolution and after classification of all patterns.

The drift of the class has been well detected for both attributes. The classifier has been adapted with the representative patterns. However, the thresholds $th1$ and $th2$ must be well determined. A too big or too small $th1$ value leads to lose some representative patterns or to keep some obsolete ones, while a too big or a too small $th2$ value leads to detect too late or too early the class evolution.

The mean values of the class obtained by DFPM are close to those expected, contrary to those obtained by IFPM (Fig. 4). However, some patterns which represent the connection between the old class and the evolved one are still in the final class obtained by DFPM.

4.2 Rotation of class

The rotation of a class is presented in Fig. 7. The standard deviations of the class according to the two attributes are initiated at σ_1=4 and σ_2=0.25. The possibility densities are already learned using 150 learning patterns of the class (Fig. 7.a). Then, new incoming points evolve, between t=1 and t=150, to achieve a rotation of the class (Fig. 7.b).

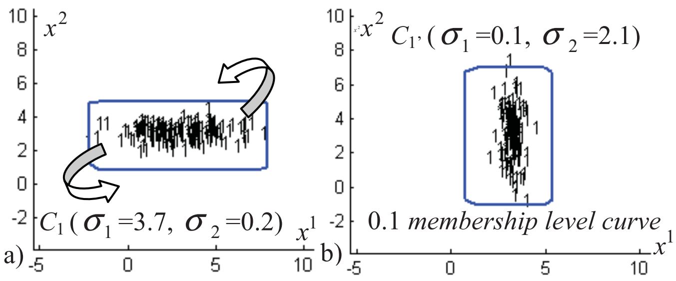

Figure 7: Rotation of the class C_1.

These changes are generated for the class according to each attribute j, $j \in \{1,2\}$, by:

$$\left.\begin{aligned} \mu_1^{1'} &= \mu_1^1 \times \sqrt{\frac{\sigma_2 \times t}{150}} \\ \mu_1^{2'} &= \mu_1^2 \times \sqrt{\frac{\sigma_1 \times t}{150}} \end{aligned}\right] \; 1 \le t \le 150 \qquad (14)$$

Fig. 8 presents the 0.1 membership level curve obtained by IFPM after the rotation of the class (t=150).

ISBN: 978-989-95079-6-8

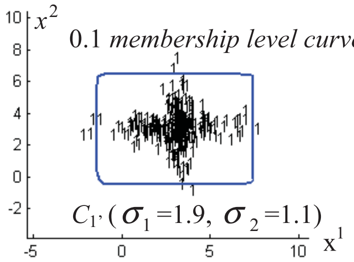

Figure 8: Final classification result obtained by IFPM for the patterns of the rotation's case.

All patterns have been classified in the class, but IFPM has not detected the rotation of the class. The membership level function is not adapted and useless patterns are still in the class. With DFPM, the residual values obtained after the classification of each pattern in the class are represented in Fig. 9.

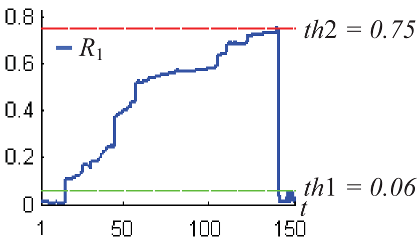

Figure 9: Residual values obtained by DFPM for the rotation of the class.

Based on Fig. 9, we can conclude that the use of $th1$=0.06 and $th2$=0.75 leads to well detect the class rotation.

Fig. 10 presents the membership level curve obtained by DFPM after classification of the evolving patterns when the adaptation of the histograms was realized.

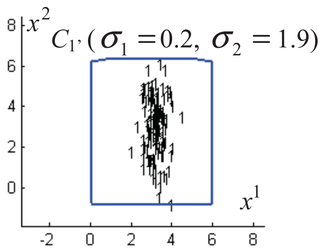

Figure 10: Classification result obtained by DFPM after adaptation of the class.

The classifier has been adapted with all representative patterns detected since the start of the class rotation. The new conserved useful patterns are the only patterns used to calculate the membership function. The resulting class corresponds well to the expected one, contrary to the class obtained by IFPM (Fig. 8).

5 Conclusions

In this paper, the Incremental version of the classification method Fuzzy Pattern Matching (IFPM) has been developed to discriminate dynamic classes. This development, called Dynamic FPM (DFPM), follows online the accumulated gradual changes in classes' probability distributions after the classification of each new pattern. Bins' usefulness values are calculated for each class and attribute. Based on these usefulness values, a residual value is calculated for each class in order to follow the evolution of each class. Then, the classes histograms are adapted in an incremental manner using the recent and useful patterns. This adaptation is achieved when the residual value reaches a suitable threshold according to the system dynamic. The DFPM method will be applied on two real applications. The first one concerns the following of a treatment's evolution for hemiplegic's patient according to its response to a medical treatment. The second application is a folding metal system which evolves from a normal mode to a faulty one according to temporal changes in the characteristics of the system and to the wear of its tools.

References

[1] H. Amadou Boubacar, *et al.,* AUDyC Neural Network using a new Gaussian Densities Merge Mechanism. *ICANNGA05*, pp. 155-158. Coimbra, Portugal, 2005.

[2] P.P. Angelov, A fuzzy controller with evolving structure. *Information Sciences* 161 (1-2), pp.21-35, 2004.

[3] L. Angstenberger, Dynamic Fuzzy Pattern Recognition. Dissertation, *Fakultät für Wirtschaftswissenschaften der Rheinisch-Westfälischen Technischen Hochschule*, Aachen, Germany, 2000.

[4] J.C. Bezdek, Pattern Recognition with fuzzy objective function algorithms, *Plenum Press,* 1981.

[5] M. S. Bouguelid, *et al.*, New results on diagnosis by pattern recognition, 4th International Conference on Informatics in Control, Automation and Robotic, pp. 167-172, Angers, France, 2007.

[6] M. Cayrol, *et al.,* Fuzzy Pattern Matching, Kybernetes, 11, pp. 103-116, 1982.

[7] L. Cohen, *et al.,* Incremental Info-Fuzzy Algorithm for Real Time Data Mining of Non-Stationary Data Streams, *TDM Workshop*, Brighton UK, 2004.

[8] D. Dubois, H. Prade, On possibility/probability transformations. *Fuzzy Logic*, pp. 103-112, 1993.

[9] R.O. Duda, *et al.*, Pattern Classification 2nd edition. *Wiley-Interscience*, 2001.

[10] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration, *Pattern Recognition 30* (7)*,* pp. 1109-1119, 1997.

[11] W.J. Gibb, *et al.*, Adaptive classification of myocardial electrogram waveforms, *IEEE Transactions on Biomedical Engineering*, N° 41, pp. 804-808, 1994.

[12] A.K. Jain, *et al.*, Statistical Pattern Recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22 (1), pp. 4-37, 2000.

[13] M. Last, Online classification on nonstationary data streams, *Intelligent Data Analysis*, vol. 6 (2), pp. 129-147, 2002.

[14] G. Nakhaeizadeh, *et al.,* Dynamic Supervised Learning. Some Basic Issues and Application Aspects. *Classification and Knowledge Organization*. Springer Verlag, Berlin, pp. 123-135, 1997.

[15] M. Sayed Mouchaweh, *et al.*, Incremental Learning in fuzzy pattern matching. *Fuzzy Sets and Systems* 132 (1), pp. 49-62, 2002.

[16] C.W. Therrien, Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics. *John Wiley & Sons*, New York, 1989.

[17] G. Widmer, M. Kubat, Learning in the Presence of Context Drift and Hidden Contexts, *Machine Learning 23*, pp. 69-101, 1996.

ISBN: 978-989-95079-6-8