

Chapter 5

Semisupervised Dynamic Fuzzy K -Nearest Neighbors

Laurent Hartert and Moamar Sayed-Mouchaweh

Abstract This chapter presents a semi-supervised dynamic classification method to deal with the problem of diagnosis of industrial evolving systems. Indeed, when a functioning mode evolves, the system characteristics change and the observations, i.e. the patterns representing observations in the feature space, obtained on the system change too. Thus, each class membership function must be adapted to take into account these temporal changes and to keep representative patterns only. This requires an adaptive method with a mechanism for adjusting its parameters over time. The developed approach is named Semi-Supervised Dynamic Fuzzy K -Nearest Neighbors (SS-DFKNN) and comprises three phases: a detection phase to detect and confirm classes evolutions, an adaptation phase realized incrementally to update the evolved classes parameters and to create new classes if necessary and a validation phase to keep useful classes only. To illustrate this approach, the diagnosis of a welding system is realized to detect the weldings quality (good or bad), based on acoustic noises issued of weldings operations.

5.1 Introduction

Evolving systems are functioning in a dynamic environment. With the occurrence of new events, evolving systems change, and their corresponding classes and patterns characteristics evolved in the feature space. Indeed, the functioning mode of an

L. Hartert (✉)

University of Reims Champagne-Ardenne, CReSTIC, Moulin de la Housse, BP 1039,
51687 Reims Cedex, France

e-mail: laurent.hartert@univ-reims.fr

M. Sayed-Mouchaweh

Ecole des Mines de Douai, Computer Science and Automatic Control Lab, EMDouai-IA,
F-59500 Douai, France

e-mail: moamar.sayed-mouchaweh@mines-douai.fr

evolving system can evolve from normal to faulty in response to the occurrence of a fault, such as a leak, to the wear of a tool or to a bad setting. To realize the diagnosis of these systems, Pattern Recognition (PR) methods need to adjust their parameters by doing automatic corrections or by warning an operator that will adjust himself the classifier parameters. When a self-adaptation of the classifier parameters is wanted, the method has to monitor the evolution of a system over time. In this case, dynamic learning is necessary to update the feature space characteristics. Then, the PR method has to use the informative patterns only to adjust the class structure. In the literature, several PR methods [9, 10, 17] are used to monitor the functioning modes evolutions of dynamic systems, to realize the fault diagnosis of complex systems or to accomplish the fault prognosis. Indeed, these methods are particularly adapted when the prior knowledge about the system behavior is not sufficient to construct an analytical model of the process.

5.1.1 Pattern Recognition

PR methods use exclusively a set of measurements, i.e., quantitative observations, about process operating modes to build a mapping from the observation space into a decision space, called the feature space. In PR, historical patterns or observations about system functioning modes are divided into groups of similar patterns, called classes. Each class is associated to a functioning mode (normal or faulty). Classes and patterns are represented by a set of d attributes, so they can be viewed as d -dimensional vectors, or points, in the feature space. The PR principle consists in classifying the new patterns by using a classifier. According to the a priori information available on the system, three types of PR methods can be used: supervised PR methods, unsupervised PR methods and semi-supervised PR methods. When labeled patterns, i.e., patterns with their class assignment, can be obtained the PR is supervised [28]. These methods use the known labeled patterns, i.e., the learning set, to build a classifier that best separates the different known 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. On the contrary when no information is available on the classes of a system, PR is unsupervised [6, 11, 12, 29]. The unsupervised PR methods, or clustering methods, are based on similarity functions, so that when patterns with the same characteristics occur they are classified in the same class, and when patterns with different characteristics occur a new class is created to classify them. Once the classifier has learned the classes membership functions, new incoming patterns are assigned to the class for which they have the maximum membership value. The third type of PR methods, the semisupervised one [8, 13] uses the supervised information, the known labeled patterns and classes, to estimate the classes characteristics and the unsupervised learning is used to detect new classes and to learn their membership functions.

5.1.2 Evolving Systems

In the case of evolving systems [2, 3, 18, 22, 23], classes are dynamic and their characteristics change in the course of time. Classes can evolve slowly or abruptly to a new position in the feature space, according to the system parameters which evolve over time. Thus, each class membership function must be adapted to take into account these temporal changes. This requires an adaptive classifier with a mechanism for adjusting its parameters over time. Hence, some of the new incoming patterns reinforce and confirm the information contained in the previous ones, but the other ones can bring new information (creation, drift, fusion, splitting of classes, etc.). This new information can concern a change in operating conditions, the development of a fault or simply more significant changes in the system's dynamic. Angstenberger [4] and Nakhaeizadeh et al. [25] act on the classifier parameters, by substituting or adding some recent and representatives patterns to the learning set according to the state (stable, warning, action) in which the system is. This adaptation is based only on the most recent batch of patterns selected by a time window [25] or by an estimation of the patterns usefulness [15]. Other approaches providing a global model rather than a local model on demand are based on the use of evolving neural networks [1, 4, 7]. In [2], a potential function based on the distance between data points is defined for the new points. According to the potential obtained for new data points, the point can reinforce or confirm the information contained in the previous ones, or a new rule can be added. In [1], the neural network is based on a multi-prototype Gaussian modeling of nonconvex classes. The activation function of each hidden neuron determines the membership degree of an observation to a prototype of a class. According to the membership degree of new acquisitions, the prototype, i.e., the hidden neuron can be adapted, deleted or a new prototype can be created. Data analysis can be realized on data coming from evolving systems in order to obtain the most informative parameters of a system that will be necessary to discriminate classes using a PR method. In this chapter, we use the statistical characteristics to supply spatial information like the number of peaks present in a signal, the standard deviation value, the root mean square value, the maximum value, the kurtosis value, etc. Some information can be computed on different parts of each signal or on entire signals. The set of characteristics, i.e. parameters, found by these methods represents the attributes which permit to characterize each signal obtained on a system. Using these informative parameters, signals are transformed into patterns in the feature space. If the parameters are well determined, classes are well discriminated and they are represented in different regions of the feature space.

5.1.3 *Dynamic Learning and Classification*

In this chapter, a semi-supervised dynamic method based on Fuzzy K -Nearest Neighbors (FKNN) [19] is developed. It was interesting to develop this method for the case of evolving systems since FKNN is a simple but efficient well known classification method. However, FKNN becomes inefficient when the size of the learning set is too important or when k is not well chosen. k is generally determined by experimentation, but it is still a parameter difficult to determine. A criterion often used is [9], where N is the number of patterns in the learning set. Several other versions of KNN exist in the literature (KNN with prototype, Adaptive KNN [26], etc.). In [21], a version of KNN pre-assigns a class to several subregions of the feature space in order to classify more rapidly the new patterns. In [30], a hierarchical search algorithm is developed to find the k -nearest neighbors using a nonmetric measure in a binary feature space. This measure is a similarity measure computed between the binary values representing the patterns. In [14, 20], respectively high dimensional and k -dimensional trees are used to find the most interesting parts of the feature space where to find the k -nearest neighbors. Only some branches of the trees have to be browsed to find the k neighbors, but the trees branches can be fast unbalanced. Another version of FKNN, called Instance-Based Learning on Data Streams (IBL-DS) [5], detects changes in the data streams by using a prediction error and the standard deviation of the 100 first patterns. If a change is detected, the 20 latest classified patterns are used to estimate the evolution realized. Based on the used indicators, a percentage of patterns initially defined is deleted from the reference base according to their spatial location and to their temporal behavior. Song et al. [27] uses two informative measures to find patterns susceptible to be the k most informative neighbors. These measures are based on probability measures calculated locally and globally. Another version of KNN [24] uses kernel-based dimensionality reduction methods to improve the classification results. These methods are used to solve some challenging problems like the application of [4] which concerns the credit scoring. The authors aims to decide whether a new customer is a good or a bad risk according to changes in his consumption. Guedalia et al. [16] deals with the problem of classification of the quality of fruits according to the damage resulting from bad weather or other external events. In [1], the authors aim to detect and to follow up the progressive evolution of the functioning modes of a thermal regulator due to the age of its components or to other temporal factors in its environment. Cohen et al. [7] treats dynamic traffic data streams in order to reduce the waiting time of drivers at the road intersections.

In this chapter, we have chosen to develop the FKNN approach since it is well known and often used in machine learning (ML) applications. The developed Semisupervised Dynamic Fuzzy K -Nearest Neighbors method is semi-supervised in order to consider the known information of a system, even when only a few observations are available, and in order to detect unknown classes and to estimate their characteristics. Semisupervised methods are particularly well adapted to evolving systems for which all classes can not be known in advance. The developed

method is presented to realize the monitoring of evolving systems. The method is applied on a real industrial system in order to detect the weldings quality and to monitor their progressive evolutions. The chapter is organized as follows. In Sect. 5.2, the functioning of the proposed approach is detailed and illustrated. Then, in Sect. 5.3, the approach is applied and evaluated using the application. Finally, conclusions and perspectives end this chapter.

5.2 Semisupervised Dynamic Fuzzy K -Nearest Neighbors (SS-DFKNN)

The selection of a PR method has to be realized according to the system on which the method is applied. Indeed, according to the application several parameters change as the number of patterns available for the learning set, the number of classes, the system dynamic, the number of dimensions, i.e., attributes, of the feature space, etc. In this section, we develop the PR method Fuzzy K -Nearest Neighbors (FKNN) in order to detect classes evolutions and to adapt these latter according to the dynamic of their evolutions. The proposed version is semisupervised, in order to:

- take into account an initial learning set X representative of the known information of a system.
- improve the classes characteristics estimation by using the new patterns
- detect new classes or subclasses according to the evolutions of the system characteristics

In this chapter, the SemiSupervised Dynamic Fuzzy KNN (SS-DFKNN) developed method permits to consider patterns evolutions even in the area of the feature space where no pattern was learned. The objectives of this approach are to follow classes evolutions by taking into account the patterns usefulness, and well estimate the new functioning modes of a system according to the estimated adapted classes characteristics. SS-DFKNN is composed of several phases which are presented in the following parts, and the method is illustrated with an example.

5.2.1 *Learning and Classification Phases*

In the learning phase of SS-DFKNN, all labeled patterns and classes are learned. The learning set X must contain a minimum of two patterns in order to calculate the initial center of gravity and standard deviation of each class, which are used in the indicators of evolution computed by SS-DFKNN. These values are calculated as follows:

- the current center of gravity $CG_{A_{curr}}$ of each class C according to each attribute A .
- the initial standard deviation $\sigma_{A_{init}}$ of each class C according to each attribute A .

These values permit to consider the dispersion of a class and its drift in the feature space. The center of gravity and the standard deviation values can be calculated for all types of classes. However, in the case of complex classes, we consider that these latter can be estimated using Gaussian subclasses. In the classification phase of SS-DFKNN, each new pattern is classified sequentially according to the class of its k -nearest neighbors. So, as for FKNN, the parameter k has to be defined initially. Once a new pattern is classified in one of the known classes, the detection of classes evolutions can be realized based on two indicators.

5.2.2 Detection of a Class Evolution

The classification of a new pattern x in one of the known classes determines the class, which can be evolving. Indeed, after the classification of x in the class C , only the class C has to be updated. In this phase of SS-DFKNN, the detection phase, the new characteristics of the class C are calculated to detect a class evolution. The current value of the standard deviation $\sigma_{A_{\text{curr}}}$ and of the current center of gravity $CG_{A_{\text{curr}}}$ of C are incrementally updated by:

$$\sigma_{A_{\text{curr}}} = \sqrt{\frac{N_C - 1}{N_C} \times \sigma_{A_{\text{curr}-1}}^2 + \frac{(x - CG_{A_{\text{curr}-1}})^2}{N_C + 1}}, \quad (5.1)$$

$$CG_{A_{\text{curr}}} = \frac{CG_{A_{\text{curr}-1}} \times N_C}{N_C + 1} + \frac{x}{N_C + 1}, \quad (5.2)$$

where N_C is the number of patterns in C before the classification of x . $\sigma_{A_{\text{curr}-1}}$ and $CG_{A_{\text{curr}-1}}$ are respectively the variance and the center of gravity of the class, according to the attribute A , before the classification of x . Based on the computed values of $CG_{A_{\text{curr}}}$, $\sigma_{A_{\text{curr}}}$, and $\sigma_{A_{\text{init}}}$ two drift indicators are used to monitor the temporal changes of a system.

- the first indicator i_{1A} represents the change of compactness of the class for each attribute A of the feature space:

$$i_{1A} = \frac{\sigma_{A_{\text{curr}}} \times 100}{\sigma_{A_{\text{init}}}} - 100. \quad (5.3)$$

i_{1A} is given in percentage. If at least one attribute A has a value of i_{1A} greater than a threshold th_1 then the class C has begun to change its characteristics according to this attribute. th_1 can be fixed to a small value when it is interesting to follow small evolutions of the class. For example, fixed to 5, it represents an evolution of 5% of the class characteristics. On the contrary when only important evolutions have to be detected, a greater value of th_1 can be necessary.

- the second indicator i_{2A} represents the distance between x_A and $CG_{A_{curr}}$ according to the current standard deviation $\sigma_{A_{curr}}$ for each attribute A of the feature space:

$$i_{2A} = \frac{|(x_A - CG_{A_{curr}})| \times 100}{\sigma_{A_{curr}}} - 100. \quad (5.4)$$

i_{2A} is given in percentage. If at least one attribute A has a value of i_{2A} greater than th_1 , then the point is not situated in the same area of the feature space than the other patterns of the class C .

However, one single pattern can involve changes for the center of gravity and for the standard deviation of a class. In some cases, this pattern can be a noise instead of a class evolution so a minimum number of successive evolved patterns $NbMin$ has to be detected in order to confirm the evolution. If $NbMin$ is fixed to a high number, then the delay detection of the class evolution can be important. This number has to be defined according to a ratio between the noise present in the patterns and the delay detection of a class evolution. The class evolution is confirmed when $NbMin$ successive values of the two indicators i_{1A} and i_{2A} are greater than th_1 . The adaptation phase which permits to adapt classes based on their evolution, is explained in the next section.

5.2.3 *Adaptation of an Evolving Class after Validation of its Evolution*

SS-DFKNN integrates a mechanism to adjust the evolved class parameters in the adaptation phase, when serious changes in a class' characteristics are detected during the detection phase. When a class evolution is confirmed, a new class or subclass is created based on useful patterns only. This adaptation is realized in several parts:

- a new class or subclass C' is created and the most representative patterns of the evolution are selected. Since the last classified pattern x of the class represents one of the evolved patterns of the class, x is selected. x also represents the most recent change in the class evolution. The other informative selected patterns are the $k - 1$ nearest neighbors of x . No distance has to be calculated to find these patterns since they were already determined during the classification of x . Indeed, to classify x , the classifier has computed several distances between the known patterns to find its nearest neighbors. These k selected patterns represent the only new patterns of a new class C' . Indeed, only k patterns are selected to represent an evolved class since the parameter k corresponds to the number of patterns judged as sufficiently representative to classify a new pattern.
- the k selected patterns are deleted from the class C .

- the new center of gravity $CG_{A_{curr}}$ of the class C is calculated and the current standard deviation $Std_{A_{curr}}$ of the class is computed.
- $CG_{A_{curr}}$ and $\sigma_{A_{init}}$ are computed for the class C' . These values are computed rapidly since the number of patterns in the evolved class is equal to k .
- the number of classes is updated.

This adaptation permits to online follow the evolution of classes with a constant and low adaptation time. Then, new patterns are classified in their corresponding class. Using this approach, all patterns and classes are kept in the feature space and an evolving class C generates at least one new class or subclass C' . If C is considered as useless, i.e., not anymore representative of a class, it can be interesting to delete this class C in order to avoid the problem of growing size of the data set. This approach permits to update and reinforce the known classes using new patterns. This is for classes for which no evolution has occurred but they are still informative and useful classes. The approach also permits to create new classes when an evolution of the system characteristics occurs. The solution presented in this chapter to deal with useless classes is presented in the next section.

5.2.4 Validation of the Existing Classes

The noise is taken into account by SS-DFKNN since a sufficient number $NbMin$ of evolving patterns is needed to consider a class evolution. However, in some cases, the noise or other events can lead to delete useless classes:

- when a short time living class is created based on few patterns, it represents only a transitory functioning mode. This temporary functioning mode can appear during the evolution of a system characteristics, changing this latter from a normal functioning mode to an abnormal functioning mode. Transitory classes are not representatives of any system functioning mode, so they can be deleted.
- when a class considered as noisy is created.
- when a class containing very few information is kept.

SS-DFKNN deletes classes corresponding to these cases, when:

- an insufficient number n_1 of patterns is contained in the class ($n_1 > k$),
- and when no pattern has been classified in the class while a sufficient number n_2 of patterns has been classified in the others classes.

However, it is not an obligation to define these two parameters if the suppression of classes is not necessary. For example, for the application considering critical data, it can be better to keep all characteristics patterns of all classes. Sometimes, classes do not need to be deleted but to be merged. Indeed, according to classes which can be created and to the evolutions of classes, it is necessary to measure over time the overlapping of classes. If several classes are created and if they drift or grow toward a common direction, then these classes have to be merged. To decide if two

classes have become sufficiently close to be merged, a similarity measure has been used [11]. This measure considers the overlapping or the closeness between classes based on the membership values of the classified patterns.

$$\delta_{iz} = 1 - \frac{\sum_{x \in C_i \vee x \in C_z} |\pi_i(x) - \pi_z(x)|}{\sum_{x \in C_i} \pi_i(x) + \sum_{x \in C_z} \pi_z(x)}, \quad (5.5)$$

where $\pi_i(x)$ and $\pi_z(x)$ are respectively the membership values of x according to C_i and C_z . δ_{iz} is the similarity measure between two classes. More the similarity value is close to 1, more the two classes are similar and have to be merged. The maximal value represents two classes completely overlapped so it is not needed to wait until the similarity value is equal to 1 to merge two classes. After each new classifier pattern, this measure is calculated, if it is greater than a threshold $rmt h_{\text{Fusion}}$ between two classes then they must be merged.

5.2.5 SS-DFKNN Algorithm

In Fig. 5.1, the algorithm describing all parts of SS-DFKNN is presented.

5.2.6 Hints for the Definition of SS-DFKNN Parameters

SS-DFKNN needs several parameters which can be defined according to each application characteristics. The defined parameters influence the classifier performances, however we can propose some default values which are generally adapted to dynamic systems:

- k corresponds to the number of neighbors considered by the k -NN methods to realize the classification of a pattern. It is the most common parameter of the k -NN methods. It should be defined according to the size of the data set, to the noise of a system and to the closeness between classes.
- th_1 is one of the most important parameter of the method. It permits to detect the evolution of a class. A class which does not evolve will have almost always the same characteristics, even if noise occurs. So, if an evolution is realized, abruptly or gradually, its characteristics will change. To allow small changes of class without waiting for an important evolution, a value equal of th_1 equal to 5 is a good compromise.
- $NbMin$ permits to validate an evolution. It should be defined at least equal to k , ($k \geq 1$) in order to wait for a sufficient number of representative patterns permitting to well estimate the characteristics of a new class. Moreover it must not be too high to delay the evolution detection. $NbMin$ should be defined

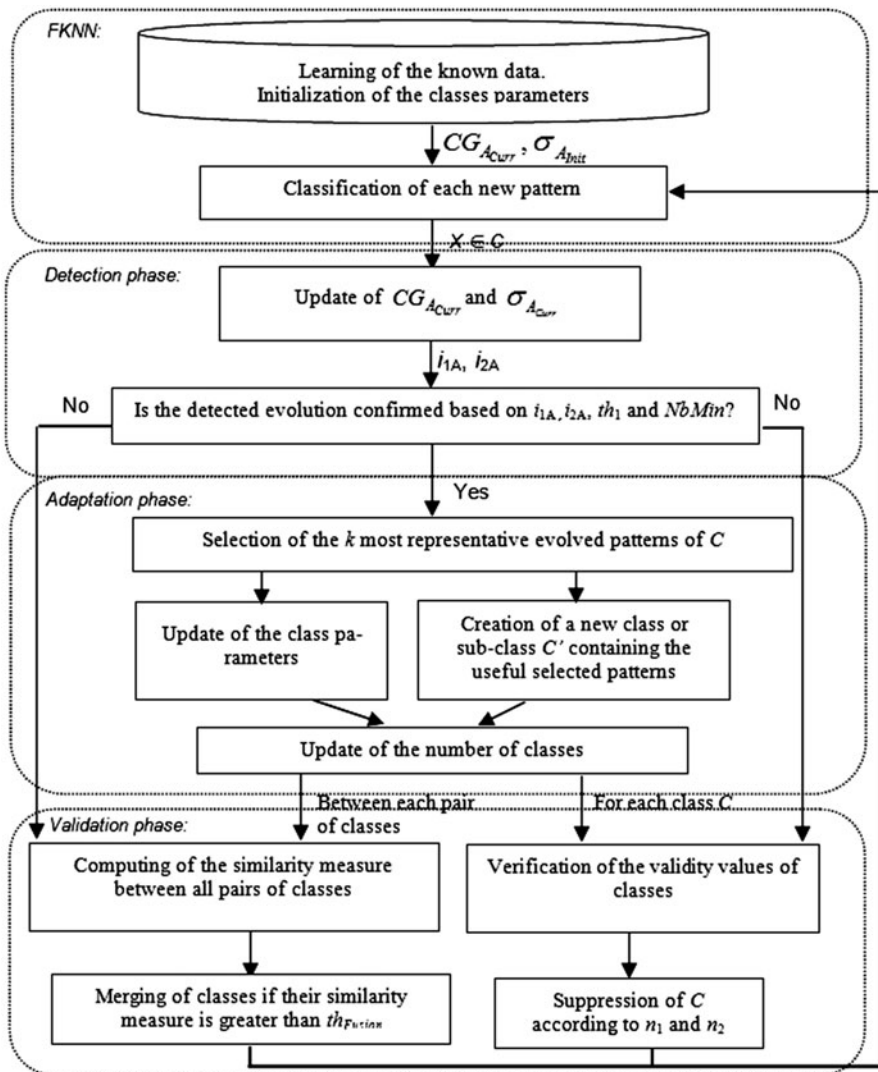


Fig. 5.1 Description of the semisupervised dynamic fuzzy k -NN method

between k and $k + 5$, respectively if k is high or small. If k and $NbMin$ are small, the risk to obtain false alarms becomes bigger.

- th_{Fusion} is an optimization parameter. Indeed, even if no fusion occurs, the simple occurrence of a class means an evolution of the system has been realized. In that case, an alarm should be raised on the system to call a human operator which will verify the system state. A th_{Fusion} value between 0.05 and 0.2 permits to merge classes which begin to have the same characteristics.

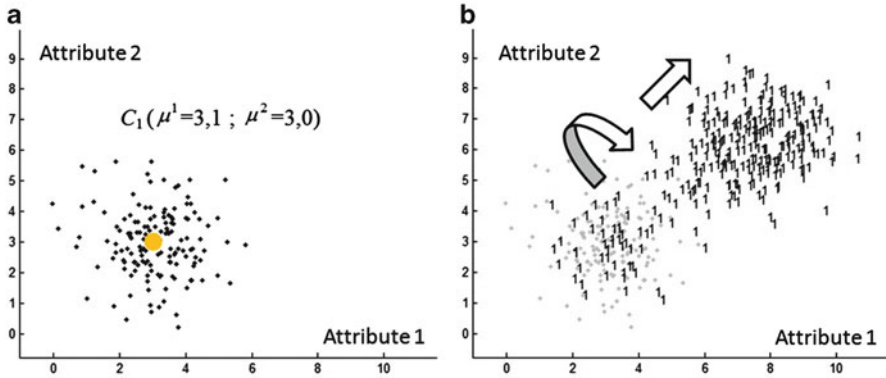


Fig. 5.2 (a) Learning set; (b) class evolution

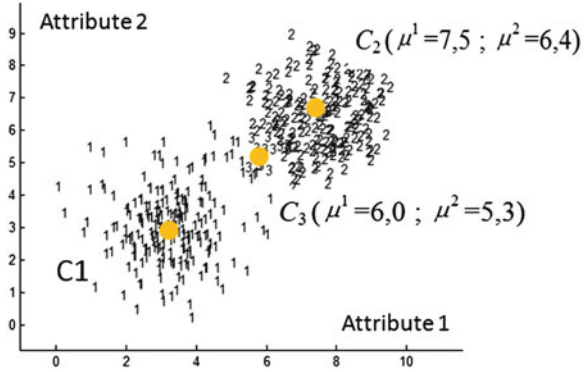
- n_1 should be defined greater than k , ($n_1 > k$) since a class will contain at least k patterns (at its creation). A default value of n_1 should be $k * 2$.
- n_2 should not be defined too small since after the creation of a class, it can be necessary to wait in order to classify more patterns in the created class. On the contrary, if no new pattern is classified in a new class after a large number of classified patterns, then the class is not useful. It is probably a noisy class or an ephemeral problem has occurred on the system. Then, the value of n_2 should be defined around 20. It means that 1 pattern on 20 should be classified in a new class, in order to confirm progressively its usefulness. For the others classes, even if they received no more patterns for a long time they will not be deleted since they have already confirmed their usefulness by having a sufficient number of patterns.

5.2.7 Illustrative Example

This example presents the dynamic evolution of a class. A progressive drift is generated according to the following equations:

- $t = 0$: One hundred and fifty patterns are used as a learning set. Only the initial class is known. The values of mean and standard deviation of the class are for the attribute 1, $\mu^1 = 3$ and $\sigma^1 = 1$, and for the attribute 2, $\mu^2 = 3$ and $\sigma^2 = 1$ (Fig. 5.2a).
- $t = 1-50$: Fifty new patterns appear with the same characteristics than the initial ones; so, there is still no evolution or drift.
- $t = 51-200$: A sudden change appears in the mean values of the class according to each attribute j , $j \in \{1, 2\}$. This change is followed by a progressive drift of the class mean according to each attribute (Fig. 5.2b):

Fig. 5.3 Classification result obtained by SS-DFKNN after classification of all patterns



$$\mu^{1'}(t) = \mu^1 + 2 + \frac{4 \times (t - 50)}{150}, \quad (5.6)$$

$$\mu^{2'}(t) = \mu^2 + 2 + \frac{2 \times (t - 50)}{150}, \quad (5.7)$$

whenever $51 \leq t \leq 200$.

- $t = 201-300$: One hundred new patterns appear. They have the same characteristics than the ones of the final class.

During the classification of the evolving patterns, several classes have been created. Then, some of them have been merged and others have been deleted. The final classification result obtained by SS-DFKNN is presented in Fig. 5.3. The method has finally obtained 3 classes: one corresponds to the initial class, one corresponds to the final location of the class, and one corresponds to a transition class which could have been deleted. Then, the method has succeeded in detecting the class evolution. The initial class C_1 has kept its characteristics and the class C_2 well corresponds to the expected class. The classification results of SS-DFKNN were obtained using the following parameters ($k = 5$; $th_1 = 5$; $NbMin = 5$; $th_{Fusion} = 0.2$; $n_1 = 10$; $n_2 = 20$) and a delay detection of 4 patterns has occurred in order to detect the class evolution. The maximum classification time obtained was equal to 5×10^{-2} s and the mean classification time was equal to 5×10^{-3} s. In the next part, SS-DFKNN is applied to a welding system in order to realize the diagnosis of the system and to follow the classes evolutions.

5.3 Application Results

In this section, we use SS-DFKNN to deal with the problem of weldings quality monitoring on an industrial welding system (Fig. 5.4) used by the company Turquais (Raucourt-et-Flaba, France).

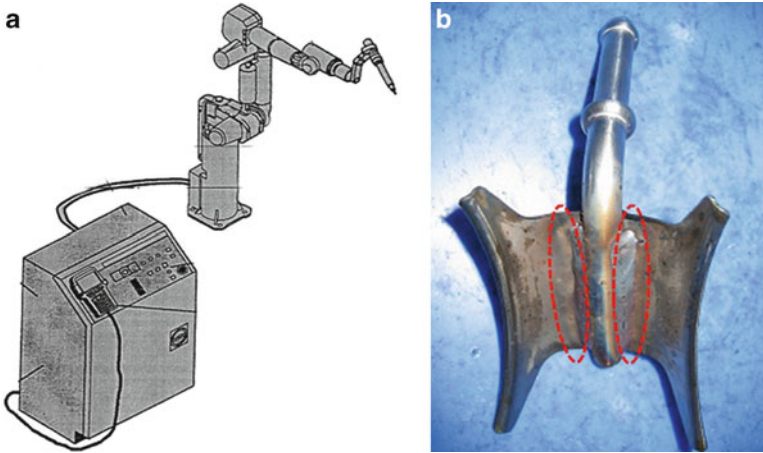


Fig. 5.4 The monitored welding system with its control system (a), and the two weldings realized between two metal pieces (b)

5.3.1 Application and Acquisition of Acoustic Noises

The welding system is able to realize the weldings of different types of metals in few seconds in order to obtain several welded pieces in a row. In this chapter, we monitor the weldings quality obtained between two metal pieces (Fig. 5.4b). The interest to monitor this system is to online detect all bad welded pieces in order to correct as soon as possible the system parameters or to change one of its welding tools. The proposed SS-DFKNN method has to detect every change of welding quality and it has to warn the human operator if a welded piece is considered as bad quality. The approach is based on the analysis, on the interpretation and on the classification of the acoustic signals issued of the weldings between two metal pieces. Currently, the human expert operator in charge of the welding machine detects weldings qualities according to the welding noise he hears. Based on this observation, we have installed an acquisition system using a microphone which is sensitive for the audible sound range that a human ear can hear. This sound represents the noises issued of the welding operation. The microphone is placed near the welding system, there is approximately 50 cm between the microphone and the metal pieces being welded. This permit to obtain more accurate sounds and to reduce significantly the welding system environment noises. The sampling frequency was fixed initially to 15 KHz for the set of measures. This frequency has been fixed in order to contain all sounds that a human can hear and to respect the Shannon's law which imposes a sampling frequency at least twice higher than the frequency of the event to study. A signal is obtained for each welding realized by the system. Two examples of noisy weldings obtained on the system are presented in Fig. 5.5. In Fig. 5.5a, a good quality welding is presented, its shape is almost constant and even if a lot of noises is present, no discontinuity is observed. On the contrary, in

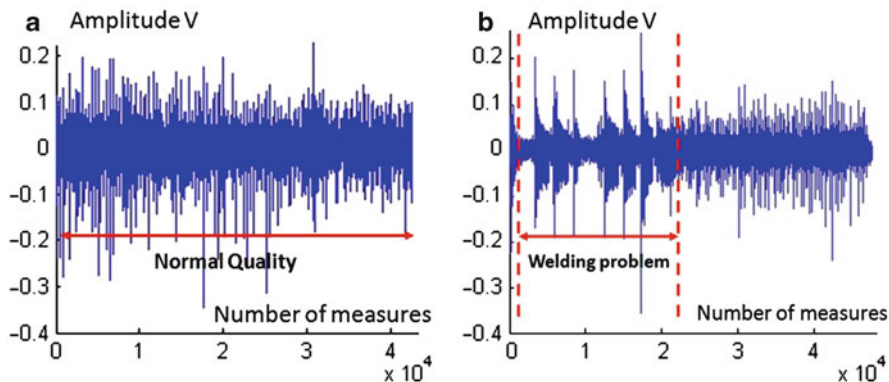


Fig. 5.5 Examples of noisy signals obtained for a good quality welding (a) and for a bad quality one (b)

Fig. 5.5b a bad quality welding is presented. The quality of this welding is initially bad, and then the welding becomes good. So, the evolution of welding quality can be distinguished by observing changes in some characteristics of the emitted acoustic signals. The quality of a welding can evolve so quickly that even when only a part of a welding is a bad quality, the global welding quality is considered as a bad quality. The acquisition of multiple acoustic signals was realized during the functioning of the welding system in order to construct a learning data set and a test set of the good and bad welded pieces. In the next part, the data analysis of these signals is realized to find informative parameters which can be used to discriminate the weldings qualities.

5.3.2 Signal Analysis and Feature Space

The ratio signal to noise is poor on signals issue of this industrial system in the Turquais company. To be able to select the interesting frequencies of signals, we have begun to search the main informative frequencies used during the realization of a welding. To do this, we have calculated the Energy Spectral Density (ESD) of each signal Fig. 5.6.

In Fig. 5.6, we can see some frequencies which are particularly present, for example the ones from 2,000 Hz to 4,000 Hz and from 6,000 Hz to 7,000 Hz. From a global point of view, the set of informative frequencies seems to be situated below 7,000 Hz. In order to follow the evolution of each welding over time, we have used a sliding window. It is important that this window contains enough observations in order to obtain representative patterns. We have studied experimentally different sizes, containing between 20 and 500 patterns. A window too small did not permit to characterize the functioning modes since it did not contain enough observations,

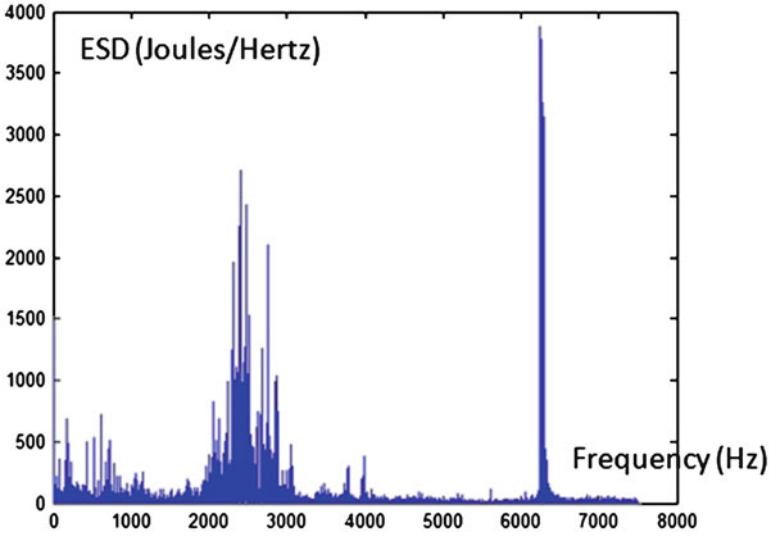


Fig. 5.6 Energy spectral density of an acoustic welding signal

while a too large window creates a delay to detect evolutions and the classification result was lower. This is a time window including 200 observations which has been selected. A window with this size was sufficiently informative and did not generate a delay in the computing of the parameters of the feature space. The window shifts with 200 new patterns. For each one of these windows, we have calculated the energy spectral density and several statistical parameters (mean, maximum, RMS, Kurtosis, dissymmetry coefficient, standard deviation, etc.). We have selected the statistical parameters which permitted to discriminate classes of good quality weldings from the ones of bad quality weldings. Two parameters were kept to establish the feature space:

- the value of dissymmetry coefficient (skewness), noted p_1 , calculated for the first derivative of each time window.
- the RMS value of the spectral density for the frequencies between 6,000 Hz and 7,000 Hz, noted p_2 , calculated for each time window. Parameters were only selected for these frequencies they were the most discriminative frequencies to characterize the welding quality.

Each observation window corresponds to a pattern in the feature space. In Fig. 5.7, a good quality welding is represented with its corresponding patterns in the feature space.

In Fig. 5.8a, welding signal of bad quality is represented. On that figure, the beginning of the window of each pattern corresponding to a bad quality welding is represented by “*.” Patterns corresponding to this signal are presented in Fig. 5.8b. In Fig. 5.8, we can see two classes which should be estimated by the classifier.

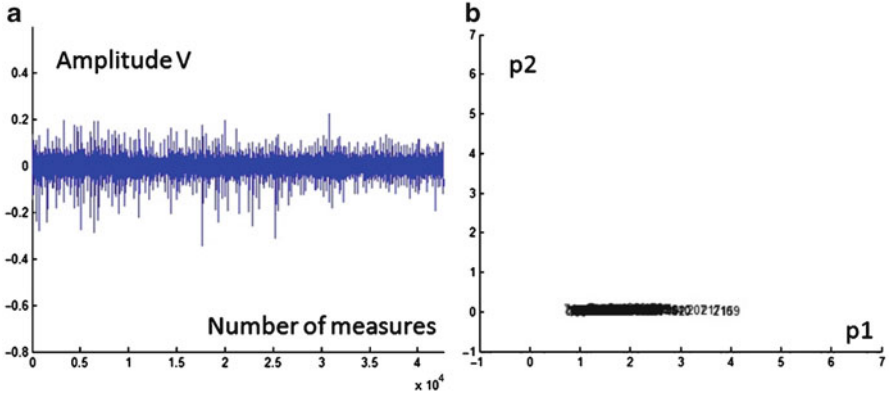


Fig. 5.7 Good quality welding (a) and its corresponding patterns in the feature space (b). Each pattern is represented by its number of window

We can also see that some patterns realize a transition between the windows corresponding to good and bad weldings qualities. For example, the pattern of window 18 leads the system toward the faulty class, while pattern 26 brings back the system toward the normal functioning mode. Then, when a bad quality appears, several round trips occur between the two classes. The functioning mode evolves according to the temporal and frequential characteristics of the system. In order to show with more precision the round trip of patterns, a zoom is realized on a part of the signal and its corresponding patterns are presented in Fig. 5.9.

5.3.3 Classification Results

In order to be in the same position that a human operator that will use our method, we only consider a single class as known; the class C_1 that contains patterns which correspond to the windows of good quality welding. For the classification of all acquired weldings, we have used a learning set such as the one of Fig. 5.10. From this learning set, we have realized the classification of each welding, i.e., the classification of each signal acquired on the system, one after the other. After the classification of a first welding of bad quality, SS-DFKNN ($k = 5$; $th_1 = 5$; $NbMin = 5$; $th_{Fusion} = 0.1$; $n_1 = 10$; $n_2 = 400$) permits to obtain classes of the Fig. 5.11. After the classification of this first welding, two classes have been estimated by the method. The evolution of the class has been validated at $t = 299$ while the evolution has really started at $t = 295$. A delay detection of $NbMin = 5$ windows has occurred. This delay corresponds to the patterns which can be classified in a transition class. It permits to confirm the evolution of the class C_1 with a small delay while avoiding some false alarms which can occur with the noise present in this system. Then the others weldings of bad quality, coming from the others acquired signals, have

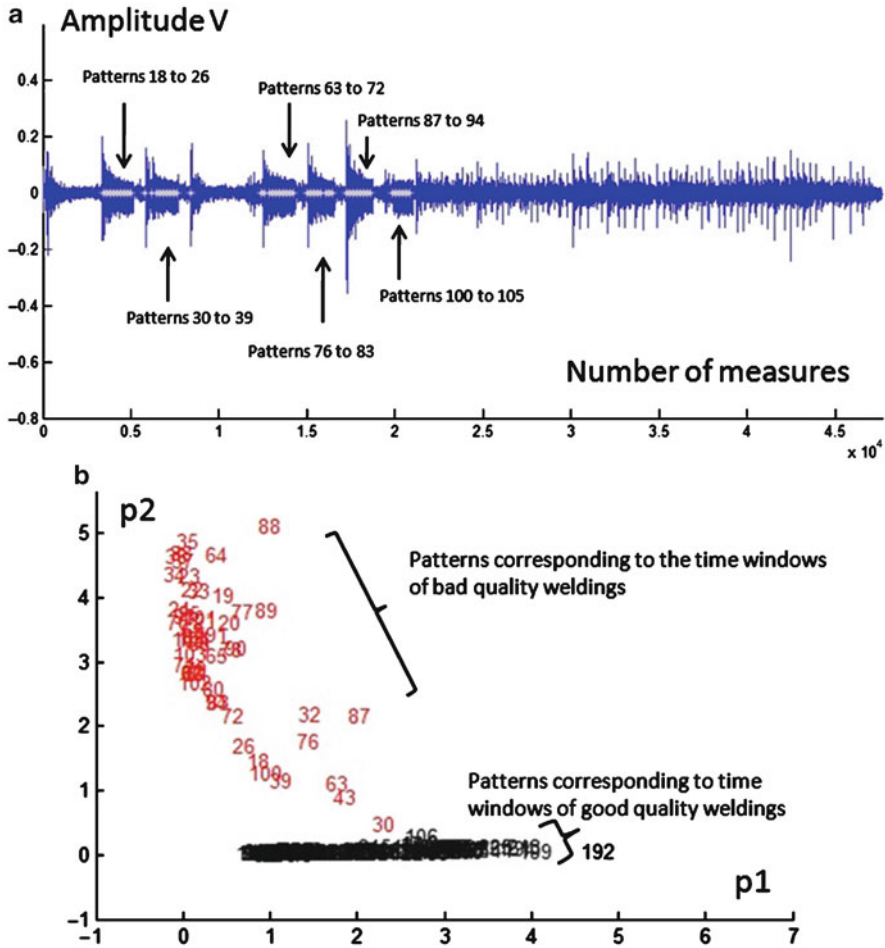


Fig. 5.8 Bad quality welding (a) and its corresponding patterns in the feature space (b)

also been classified. The classification result of all these weldings is presented in Fig. 5.12. A new class C_3 has been created, it corresponds to a transitive area of average welding quality present between the good welding quality class and the bad quality class. Only patterns which had a sufficient change in their characteristics are classified in class C_2 . Then, no false alarm was raised during classification of these patterns. The set of others weldings of good quality has then be classified. All weldings of good quality are classified in C_1 (Fig. 5.13).

After the classification of all acquired patterns, some conclusions can be presented:

- 100% of the good quality weldings are classified in C_1 .
- 100% of the bad quality weldings are detected.

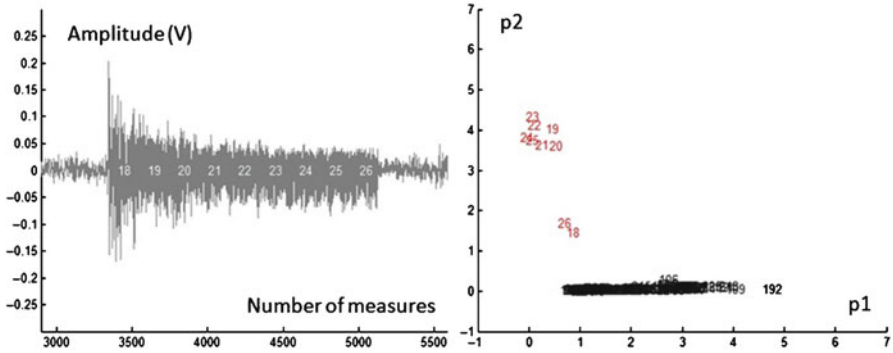
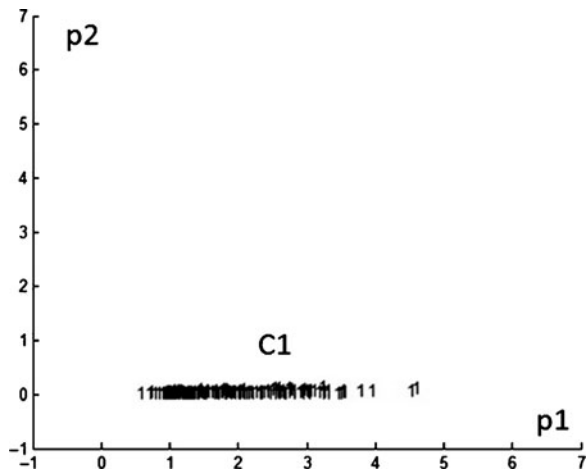


Fig. 5.9 Zoom on the patterns corresponding to a part of the welding problem and its corresponding patterns are presented in the feature space. Patterns of the class of good quality are also presented to have a global view of the two classes and to understand the evolution realized

Fig. 5.10 Example of learning set used for the classification of the new weldings. One hundred and seventy five patterns are part of C_1



- misclassified patterns (0.2% of all patterns) correspond to transitive patterns. They only influence the delay detection of some weldings which have a bad welding quality,
- very few information (2 patterns of C_1 at a minimum) are necessary to use SS-DFKNN.
- the delay detection of a bad welding is small (8 ms).

All classification results obtained by SS-DFKNN for this application are presented in Table 5.1.

The set of bad quality weldings has had several patterns classified in C_2 and C_3 . Only some bad quality weldings patterns were misclassified in C_1 . It concerns only the patterns which have generated delay evolution detection. These patterns were

Fig. 5.11 Classification of the first welding of bad quality

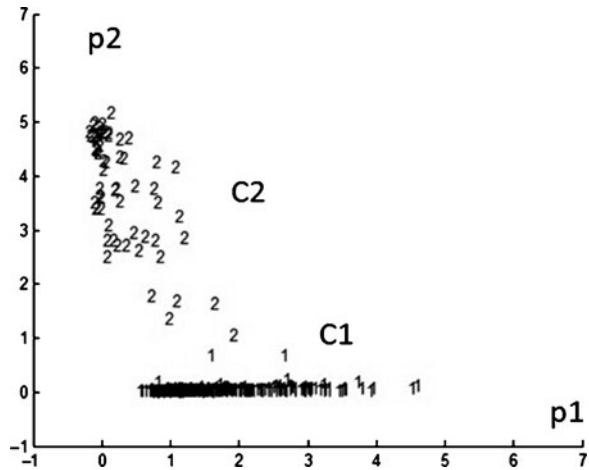
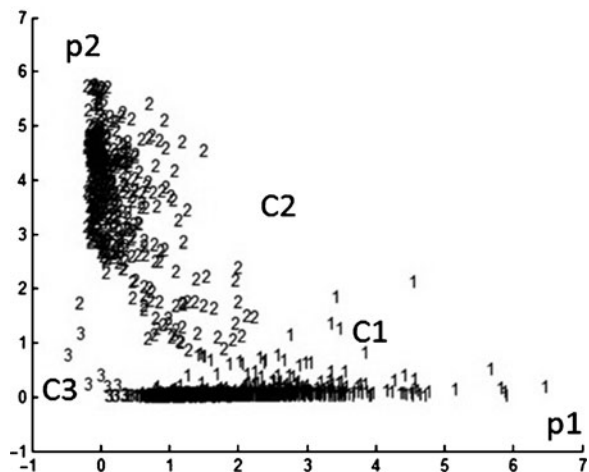


Fig. 5.12 Classification of all weldings of bad quality



then not misclassified consecutively but they correspond to few first patterns of bad quality weldings. No welding pattern of good quality was misclassified. Then, the classification result obtained by SS-DFKNN permits to perfectly distinguish good quality weldings from those of bad quality. Moreover, only small delay detection occurs so that the dynamic PR proposed can be online applied to this system. An acoustic or visible alarm system will be set up in order to warn human operators if a welding problem occurs.

Fig. 5.13 Classification result of all weldings of the database

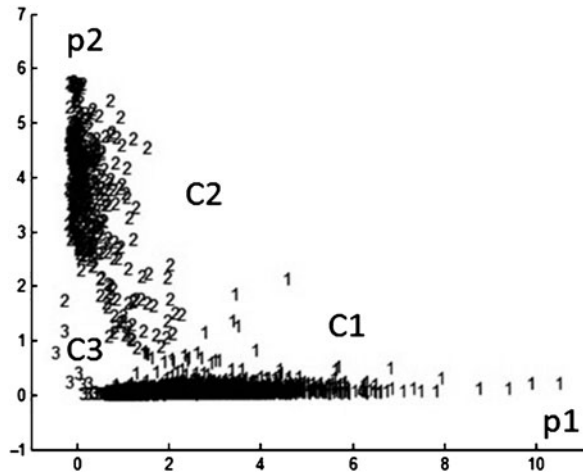


Table 5.1 Classification results obtained by SS-DFKNN

	Semi-supervised dynamic FKNN
Detection of weldings with a bad quality	100%
Detection of weldings with a good quality	100%
Mean t_{delay} to detect a bad quality welding	5 patterns
Mean t_{delay} to detect a good quality welding	No delay

5.4 Conclusion

The dynamic PR method named SemiSupervised Dynamic Fuzzy K -Nearest Neighbors (SS-DFKNN) has been developed in this chapter in order to demonstrate its capacities to realize the diagnosis and monitoring of industrial evolving systems. SS-DFKNN integrates two indicators of patterns usefulness which permit to follow classes evolutions by adapting these latter if an evolution is confirmed. When an evolution is realized, classes or subclasses are created to represent the current functioning mode of a system. These evolved classes can permit to better estimate the current functioning mode of an evolving system according to the time, to well monitor the evolutions of complex classes (defined by several subclasses) and to progressively find which functioning mode a class may reach after its evolution. Indeed without adapting classes, an evolution of the classes characteristics will be detected much later than when classes are adapted. SS-DFKNN can use only a few patterns to initiate the method. However, more the learning set is representative of the classes characteristics, better the detection of evolutions is. The classes characteristics of all classes are refined sequentially with the classification of the new patterns. The update of the evolved classes parameters is realized in a low time so that this method can be applied online. In this chapter, SS-DFKNN has been illustrated by a drift example and applied on an industrial welding system. For each welding operation, an acoustic signal was acquired and used by SS-DFKNN.

SS-DFKNN has well classified these signals which permitted to detect all bad quality weldings and it also detected each one of the welding quality evolution realized by the welding system.

SS-DFKNN uses several parameters to monitor evolving systems. Among these parameters we can particularly estimate that k , th_1 , and $NbMin$ have a major importance in the results the method can obtain. According to their values, a delay detection can occur, noisy patterns can be considered as a class evolution and patterns can be misclassified. A new version of this method is being developed in order to progressively adapt the classes parameters and the classifier parameters, in order to obtain better results and to simplify the initial definition of these latter.

References

1. Amadou-Boubacar, H., Lecoche, S., Maouche, S.: Self-adaptive kernel machine: Online clustering in RKHS. In: Proceedings of the IEEE IJCNN05. Montreal, Canada (2005)
2. Angelov, P.: A fuzzy controller with evolving structure. *Information Sciences* **161**(1–2), 21–35 (2004)
3. Angelov, P., Filev, D., Kasabov, N.: *Evolving Intelligent Systems—Methodology and Applications*. John Wiley & Sons, New York (2010)
4. Angstenberger, L.: Dynamic fuzzy pattern recognition. Ph.D. thesis, Fakultät für Wirtschaftswissenschaften der Rheinisch-Westfälischen Technischen Hochschule (2000). Aachen, Germany
5. Beringer, J., Hüllermeier, E.: Efficient instance-based learning on data streams. *Intelligent Data Analysis* **11**(6), 627–650 (2007)
6. Bezdek, J.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic/Plenum Publishers, USA (1981)
7. Cohen, L., Avrahami, G., Last, M.: Incremental info-fuzzy algorithm for real time data mining of non-stationary data streams. In: Proceedings of the TDM Workshop. Brighton, UK (2004)
8. Cozman, F., Cohen, I., Cirelo, M.: Semi-supervised learning of mixture models. In: Proceedings of the 20th International Conference on Machine Learning (ICML). Washington DC, USA (2003)
9. Dubuisson, B.: Diagnostic et reconnaissance des formes. Tech. rep., Trait des Nouvelles Technologies, srie Diagnostic et Maintenance, HERMES (1990)
10. Duda, R., Hart, P., Stork, D.: *Pattern Classification—Second Edition*. Wiley-Interscience (John Wiley & Sons), Southern Gate, Chichester, West Sussex, England (2000)
11. Frigui, H., Krishnapuram, R.: A robust algorithm for automatic extraction of an unknown number of clusters from noisy data. *Pattern Recognition Letters* **17**, 1223–1232 (1996)
12. Frigui, H., Krishnapuram, R.: Clustering by competitive agglomeration. *Pattern Recognition* **307**, 1109–1119 (1997)
13. Gabrys, B., Bargiela, A.: General fuzzy min–max neural network for clustering and classification. *IEEE Transactions on Neural Networks* **11**(3), 769–783 (2000)
14. Garcia, V.: Suivi d’objets d’intrt dans une sequence d’images: des points saillants aux mesures statistiques. Tech. rep., University of Nice (2009)
15. Gibb, W., Auslander, D., Griffin, J.: Adaptive classification of myocardial electrogram waveforms. *IEEE Transactions on Biomedical Engineering* **41**, 804–808 (1994)
16. Guedalia, I., London, M., Werman, M.: An on-line agglomerative clustering method for non-stationary data. *Neural Computation* **11**(2), 521–540 (1999)
17. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1), 4–37 (2000)

18. Kasabov, N.: *Evolving Connectionist Systems: The Knowledge Engineering Approach—Second Edition*. Springer Verlag, London (2007)
19. Keller, J., Gray, M., Givens, J.: A fuzzy k -nn neighbor algorithm. *IEEE Transactions on Systems, Man and Cybernetics* **15**(4), 580–585 (1985)
20. Kybic, J.: Incremental updating of nearest neighbor-based high-dimensional entropy estimation. In: *Proceedings of the ICASSP 2006*, pp. 804–807 (2006)
21. Law, Y., Zaniolo, C.: An adaptive nearest neighbor classification algorithm for data streams. In: *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, pp. 108–120. Porto, Portugal (2005)
22. Lughofer, E.: *Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications*. Springer, Berlin Heidelberg (2011)
23. Lughofer, E., Angelov, P.: Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing* **11**(2), 2057–2068 (2011)
24. Min, R.: A non-linear dimensionality reduction method for improving nearest neighbour classification. Ph.D. thesis, University of Toronto (2005). Toronto, Canada
25. Nakhaeizadeh, G., Taylor, C., Kunsch, G.: Dynamic supervised learning. Some basic issues and application aspects. classification and knowledge organization, pp. 123–135. Springer Verlag, Berlin Heidelberg (1997)
26. Roncaglia, A., Elmi, I., Dori, L.: Adaptive K -NN for the detection of air pollutants with a sensor array. *IEEE Sensor Journal* **4**(2), 248–256 (2004)
27. Song, Y., Huang, J., Zhou, D.: lk -NN: Informative k -nearest neighbor pattern classification. In: *Proceedings of the PKDD 2007 conference*, pp. 248–264 (2007)
28. Therrien, C.: *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. John Wiley & Sons, New York (1989)
29. Vachkov, G.: Online classification of machine operation modes based on information compression and fuzzy similarity analysis. In: *Proceedings of the IFSA-EUSFLAT 2009 conference*, pp. 1456–1461. Lisbon, Portugal (2009)
30. Zhang, B., Srihari, S.: A fast algorithm for finding k -nearest neighbors with non-metric dissimilarity. In: *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, pp. 13–19 (2002)