International Journal of Computational Methods
Vol. 12, No. 2 (2015) 1550008 (19 pages)

DOI: 10.1142/S0219876215500085

World Scientific
www.worldscientific.com

Bayes Classification for Nonstationary Patterns

Piotr Kulczycki* and Piotr Andrzej Kowalski†

Polish Academy of Sciences, Systems Research Institute
Centre of Information Technology for Data Analysis Methods
AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
Division for Information Technology and Biometrics
**kulczycki@ibspan.waw.pl*
†*pakowal@ibspan.waw.pl*

Received 16 November 2013
Accepted 11 September 2014
Published 4 November 2014

The paper's subject is classification with nonstationary patterns. The attribute space is finite-dimensional, while its coordinates in particular may be continuous, binary, discrete, categorical in character, or also a combination of these. The number of patterns is not methodologically limited. Use of the Bayes approach minimizes the expected value of misclassifications, allowing additionally for an influence in the proportions of probability of errors when assigning to specific classes. In turn, the statistical kernel estimators method makes the algorithm independent of patterns' shapes. The investigated procedure also eliminates elements of patterns which have insignificant or even negative influence on the results' accuracy. Appropriate modifications follow the classifier parameters, which increases the effectiveness of procedure adaptation for nonstationary patterns. The algorithm concept is based on the sensitivity method, used with artificial neural networks.

Keywords: Data analysis; classification; pattern nonstationarity; pattern size reduction; Bayes approach; classifier adaptation; statistical kernel estimators; artificial neural networks.

1. Introduction

Classification [Duda *et al.* (2001)] is one of the basic procedures of data analysis and exploration [Han and Kamber, 2001]. It consists in assigning a tested element to earlier established (either in a fundamental way or "automatically" by clustering [Everitt *et al.* (2011); Kulczycki and Charytanowicz (2010)]) groups, named here as classes. They are most often represented by patterns, which are sets of elements typical for particular classes. In most of the methods used today, one assumes the stationarity (unchanged by time) of these patterns. However, nowadays — as models have become more accurate, and investigated phenomena have become more

complex [Kulczycki *et al.* (2007)] — this assumption is successfully ignored. It particularly concerns those practical tasks where new elements, with the most current naturally being the most valuable, are continuously added to patterns.

Generally the methods of classification with nonstationary patterns can be divided into three groups, by and large consisting of:

(1) the supplementing or elimination of selected elements of patterns (in the trivial case, the most recent and the oldest), e.g., [Salganicoff (1997); Widmer and Kubat (1996)];
(2) the mapping of appropriate weights to particular elements of patterns (in the simplest case, in proportion to how current they are), e.g., [Klikenberg (2004)];
(3) successive changes and modifications of the classification procedure itself or its parameters, e.g., [Harries *et al.* (1998); Muhlbaier and Polikar (2007)].

The majority of these methods are heuristic in character and have been investigated with specific conditions, so are dedicated solely to a particular research task [Kenyon (1991); Krasotkina *et al.* (2011)]. The problem of the changeability over time of various aspects of subjects under research, and resulting nonstationarity, is formulated in literature using a significantly diverse terminology, from the intuitive "changing environments" [Kuncheva (2004)] or "evolving data stream" [Aggarwal *et al.* (2006)], to "concept drift" [Zlobaite (2009)] specific for the machine learning area, or "adaptation" [Bouchachia (2009)] coming from the methodology for solving such tasks, and others, frequently equally inadequate. It is worth remembering that this leads to generalizations as well as bibliographic and comparative research in this field, often very specific and rarely coherent.

The concept proposed in this paper belongs firmly in the first of the above groups 1–3, although elements of the other two are also used. It was conceived on the basis of the sensitivity method used in artificial neural networks. As a result of its operation, particular elements of patterns receive weights proportional to their significance for correct classification. Elements of the smallest weights are eliminated. For the sake of the patterns' nonstationarity, their elements whose weights are currently small but increase successively are kept. In consequence a procedure is proposed ensuring that an adaptation to changing conditions is obtained by correcting classifier parameters values. The concept of the investigated method is based on the Bayes approach, providing a minimum of potential losses arising from incorrect classification. It is also possible to introduce preferences for those classes whose elements — due to potential nonsymmetrical conditioning of the task — especially should not be mistakenly assigned to others. The classifier was constructed applying the statistical kernel estimators methodology, thus freeing the above procedure from arbitrary assumptions regarding patterns' shapes — their identification is an integral part of the algorithm presented here. The correct functioning and effectiveness of the investigated method have been verified by experimental and comparative analysis.

The first sections of this paper, i.e., 2–7, briefly describe mathematical apparatus and component procedures used in the main part — Sec. 8 — to synthesize the classification algorithm for the nonstationary case investigated here. The numerical verification and comparison with the similarly conditioned support vector machine method is the subject of Sec. 9, followed by final comments and remarks.

This concept is the generalization of the procedure for the stationary case presented in the paper [Kulczycki and Kowalski (2011)]. This publication can be recommended at this point, since the idea itself — as a basis — is naturally simpler and more straightforward there. A preliminary version of this paper was partially presented as [Kulczycki and Kowalski (2013a, 2013b)].

2. Statistical Kernel Estimators

Let (Ω, Σ, P) denote a probability space. First, the continuous random variable case will be considered. This provides the basis for the investigation both of theoretical and practical applications of kernel estimators. Thus, suppose the n-dimensional random variable X, with a distribution characterized by the density f. Its kernel estimator $\hat{f} : \mathbb{R}^n \to [0, \infty)$ is calculated on the basis of the random sample

$$x_1, x_2, \ldots, x_m, \tag{1}$$

and defined — in the basic form — by the formula

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} K\left(\frac{x - x_i}{h}\right), \tag{2}$$

where the positive coefficient h is known as a smoothing parameter, while the measurable function $K : \mathbb{R}^n \to [0, \infty)$ symmetrical with respect to zero, having at this point a weak global maximum and fulfilling the condition $\int_{\mathbb{R}^n} K(x)\mathrm{d}x = 1$ is termed a kernel. For interpretation of the definition, see Fig. 1. The monographs [Kulczycki (2005); Silverman (1986); Wand and Jones (1995)] contain a detailed description of the above methodology, in particular the selection of the shape of the kernel K [Kulczycki (2005, Sec. 3.1.3); Wand and Jones (1995, Secs. 2.7 and 4.5)] and the calculation of the smoothing parameter h value [Kulczycki (2005, Sec. 3.1.5); Wand and Jones (1995, Chap. 3 and Sec. 4.7)], based on mean-square criterion.

In this paper the generalized (one-dimensional) Cauchy kernel is applied:

$$K(x) = \frac{2}{\pi(x^2 + 1)^2}, \tag{3}$$

due to its "heavy tails", which work well in peripheral areas of distributions, themselves potential regions dividing classes in the classification task investigated here. In the multi-dimensional case, the kernel is defined using the concept of a product

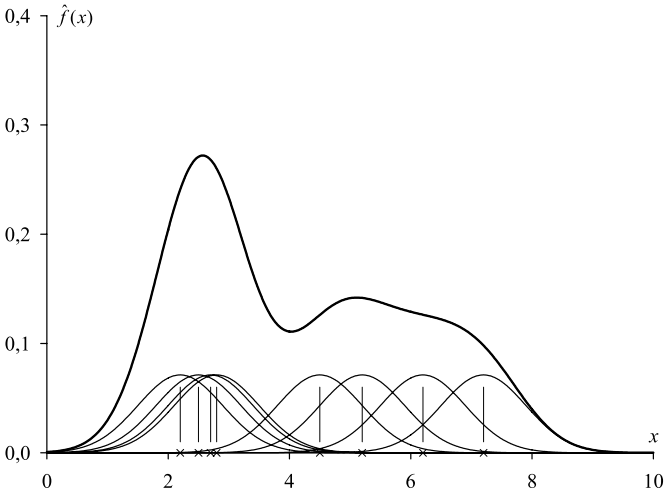

Fig. 1. Kernel estimator (2).

kernel resulting from one-dimensional kernels for particular coordinates:

$$K(x) = K\left(\begin{bmatrix} \mathsf{x}_1 \\ \mathsf{x}_2 \\ \vdots \\ \mathsf{x}_n \end{bmatrix}\right) = \mathsf{K}_1(\mathsf{x}_1)\mathsf{K}_2(\mathsf{x}_2)\cdots\mathsf{K}_n(\mathsf{x}_n), \tag{4}$$

where K_1, $\mathsf{K}_2, \ldots, \mathsf{K}_n$ denote here one-dimensional kernels, for the Cauchy form given by formula (3). (When one uses the product kernel, the expression h^n appearing in definition (2) should be replaced by $h_1 \cdot h_2 \cdot \ldots \cdot h_n$, the product of smoothing parameters for particular coordinates.)

Generally, for calculation of the smoothing parameter h value it is recommended to avail of the effective plug-in method [Kulczycki (2005, Sec. 3.1.5); Wand and Jones (1995, Sec. 3.6.1)], used here both for the one-dimensional and — thanks the application of the product kernel — multidimensional case, separately for particular coordinates. However if the classification method investigated here uses the correction of this parameter presented in Sec. 3.3, the simplified method is enough [Kulczycki (2005, Sec. 3.1.5); Wand and Jones (1995, Sec. 3.2.1)]. The smoothing parameter is then given as:

$$h = \left(\frac{W(K)}{U(K)^2}\frac{8\sqrt{\pi}}{3m}\right)^{1/5}\hat{\sigma}, \tag{5}$$

where

$$W(K) = \int_{\mathbb{R}} K(x)^2 \mathrm{d}x, \tag{6}$$

$$U(K) = \int_{\mathbb{R}} x^2 K(x) \mathrm{d}x, \tag{7}$$

while $\hat{\sigma}$ denotes an estimator of standard deviation obtained from sample (1); (in the multi-dimensional case: for this coordinate of the sample elements, for which the smoothing parameter value has been calculated). For the Cauchy kernel (3) suggested above, coefficients (6) and (7) amount to $W(K) = 1$ and $U(K) = 5/4\pi$.

In practice one employs additional procedures to generally increase the quality of the kernel estimator and fit its features to those of the considered reality. In this paper the modification of the smoothing parameter [Kulczycki (2005, Sec. 3.1.6); Silverman (1986, Sec. 5.3.1)] will be applied. The definition of the kernel estimator then takes the form

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} \frac{1}{s_i} K\left(\frac{x - x_i}{hs_i}\right), \tag{8}$$

where additionally introduced non-negative modifying coefficients are given by

$$s_i = \left(\frac{\hat{f}_*(x_i)}{\bar{s}}\right)^{-c} \quad \text{for } i = 1, 2, \ldots, m, \tag{9}$$

while $\hat{f}_*$ denotes the kernel estimator in its basic form (2), and $\bar{s}$ is the geometric mean of the quantities $\hat{f}_*(x_1)$, $\hat{f}_*(x_2), \ldots, \hat{f}_*(x_m)$. The constant $c \geq 0$ is referred to as modification intensity. The case $c = 0$, implying in consequence $s_i \equiv 1$, determines the lack of smoothing parameter modification, whereas together with an increase in the value c its intensity grows. Corollaries resulting from the mean-square criterion primarily point to the value:

$$c = 0.5. \tag{10}$$

Figure 2 shows an interpretation of the above procedure. In the areas where elements of the random sample are dense, for the elements x_i it is true that $\hat{f}_*(x_i) > \bar{s}$,

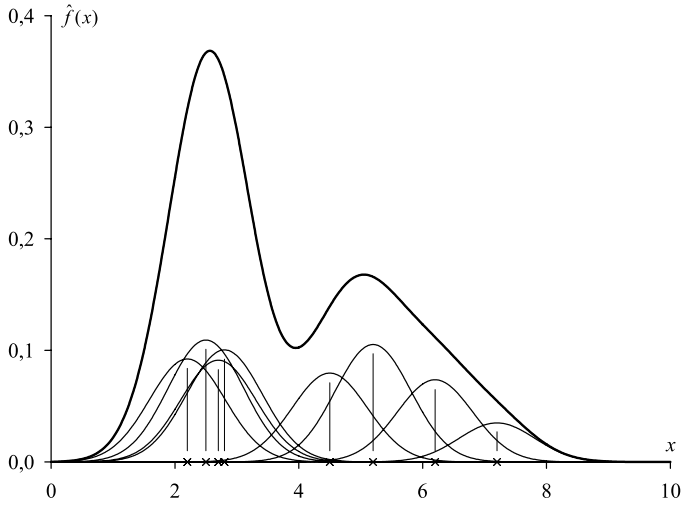

Fig. 2. Kernel estimator with smoothing parameter modification.

and therefore, as a result of formula (6), also $s_i < 1$. This leads to a narrowing of the kernels assigned to them, which in turn allows for better characterization of specific properties of distribution. In contrast, in the areas where the elements of the random sample are sparse, one has $\hat{f}_*(x_i) < \bar{s}$ and consequently $s_i > 1$. This causes "flattening" and thus — advantageous to estimation quality — additional smoothing of the kernel estimator in "peripheral" regions of distribution. As mentioned previously, this is of particular significance in the task of classification, having great influence on "peripheral" areas where classes potentially border.

Similarly to the above for the continuous random variable, kernel estimators can be constructed for binary, discrete and categorical (ordered as well). Moreover, any composition of these variables types is possible. The literature on the subject is quite broad and varied. For the first case, it is worth quoting the classic monographs [Kulczycki (2005, Sec. 3.1.8); Silverman (1986, Sec. 6.1.4)] as well as paper [Aitchison and Aitken (1976)], and for the second [Ahmad and Cerrito (1994); Wang and Ryzin (1981)]. Issues connected with categorical variables can be found in the publications [Gaosheng *et al.* (2009); Li and Racine (2008); Ouyang *et al.* (2006)].

3. Bayes Classification

Consider J sets consisting of elements of the space $\mathbb{R}^n$:

$$x'_1, x'_2, \ldots, x'_{m_1}, \tag{11}$$

$$x''_1, x''_2, \ldots, x''_{m_2}, \tag{12}$$

$$\vdots$$

$$x_1^{\prime\prime\ldots\prime}, x_2^{\prime\prime\ldots\prime}, \ldots, x_{m_J}^{\prime\prime\ldots\prime}, \tag{13}$$

representing assumed classes. The sizes m_1, $m_2, \ldots, m_J$ should be proportional to the "contribution" of particular classes in the population under investigation. The classification task consists of deciding to which of these groups the tested element

$$\tilde{x} \in \mathbb{R}^n, \tag{14}$$

should be assigned. Let now $\hat{f}_1$, $\hat{f}_2, \ldots, \hat{f}_J$ denote kernel estimators of a probability distribution density, calculated successively based on sets (11)–(13) treated as random samples (1) — a description of the methodology used for their construction is contained in Sec. 2. In accordance with the classic Bayes approach [Duda *et al.* (2001)], ensuring a minimum of expected value of losses, the classified element (14) should then be given to the class for which the value

$$m_1\hat{f}_1(\tilde{x}), m_2\hat{f}_2(\tilde{x}), \ldots, m_J\hat{f}_J(\tilde{x}) \tag{15}$$

is the greatest. The above can be generalized by introducing the positive coefficients z_1, $z_2, \ldots, z_J$:

$$z_1 m_1\hat{f}_1(\tilde{x}), z_2 m_2\hat{f}_2(\tilde{x}), \ldots, z_J m_J\hat{f}_J(\tilde{x}). \tag{16}$$

Taking as standard values $z_1 = z_2 = \cdots = z_J = 1$, formula (16) brings us to (15). By appropriately increasing the value z_i, a decrease can be achieved in the probability of erroneously assigning elements of the ith class to other wrong classes, although theoretically the danger does then exist of slightly increasing the general number of misclassifications. This approach is warranted for those tasks where a class is associated with a particular phenomenon (e.g., in technical diagnostics — with some especially dangerous fault type), the oversight of which could have exceptionally negative consequences. Here an increase in the value z_i is inversely proportional to the raising of probability of an erroneously assigned element of the ith class to another. A respective growth in the values of a few coefficients z_i is possible, although theoretically this may additionally cause a slight increase in the general number of classification errors.

4. Discrete Derivative

The task of computing the value of the discrete derivative of the function $g : \mathbb{R} \to \mathbb{R}$ consists in calculating the quantity $g'(t)$ based on values of this function obtained for a finite number of arguments $t_1, t_2, \ldots, t_k$. For the problem under investigation a backward derivative will be used, i.e., where $t = t_k$. As the task considered here does not require the differences between subsequent values $t_1, t_2, \ldots, t_k$ to be equal, it is therefore advantageous to apply interpolation methods. In the procedure worked out here, favorable results were achieved using a classic method based on Newton's interpolation polynomial. Detailed formulas, as well as a treatment of other related concepts are found in the survey paper [Venter (2010)]. For the purposes of the procedure investigated in this paper, $k = 3$ can be taken as a standard value, a useful compromise between stability of results and possibility to react to changes (the derivative has then two degrees of freedom).

5. Sensitivity Analysis for Learning Data

When modeling multi-dimensional problems using artificial neural networks, particular components of an input vector most often are characterized by diverse significance of information, and in consequence influence variously the result of the data processing. In order to eliminate redundant input vector components, a sensitivity analysis of the network with respect to particular learning data is often used. A basic factor for network reduction is sensitivity of the output function with regards to particular input data.

The essence of the sensitivity method [Zurada (1992)] consists in defining — after the network learning phase — the influence of the particular inputs x_j on the output value y, which is characterized by the real coefficients

$$S_i = \frac{\partial y(x_1, x_2, \ldots, x_m)}{\partial x_i} \quad \text{for } i = 1, 2, \ldots, m. \tag{17}$$

Next, one aggregates the particular coefficients $S_i^{(p)}$ originating from successive iterations of the previous phase and corresponding to the sensitivity of subsequent learning data, with $p = 1, 2, \ldots, P$. The result is the final coefficient $\bar{S}_i$ given by the formula

$$\bar{S}_i = \sqrt{\frac{\sum_{p=1}^{P} (S_i^{(p)})^2}{P}} \quad \text{for } i = 1, 2, \ldots, m. \tag{18}$$

After the sorting operation for the vector $\bar{S}_i$ according to decreasing values, an analysis of the relevance of particular components to the result of network operation is performed, and then the least significant inputs are eliminated. (In the general case the above algorithm can be used repeatedly to achieve further reduction. However, during empirical testing of the classification method developed here, such action did not bring positive results and so was forsaken.)

The application of the above method led to reducing the input dimension of the neural network by removing information of little significance or even elimination of data (input vector components) having an unfavorable influence on the obtained result's correctness. This resulted in an increase in speed as well as a reduction of errors of learning and generalization.

Detailed considerations concerning the above procedure are found in the publications [Engelbrecht *et al.* (1995); Zurada (1992)].

6. Reducing Patterns' Size

In practice, some elements of sets (11)–(13), constituting patterns of particular classes, may have insignificant or even negative — in the sense of classification correctness — influence on quality of obtained results. Their elimination should therefore imply a reduction in the number of erroneous assignments, as well as decreasing calculation time. To this aim the sensitivity method for learning data, used in artificial neural networks, briefly presented in the previous section, will be applied.

To meet the requirements of this procedure, the definition of the kernel estimator will be generalized below with the introduction of the non-negative coefficients $w_1, w_2, \ldots, w_m$, normed by the condition

$$\sum_{i=1}^{m} w_i = m, \tag{19}$$

and mapped to particular elements of random sample (1). The basic form of kernel estimator (2) then takes the form

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} w_i K\left(\frac{x - x_i}{h}\right). \tag{20}$$

Formula (8) undergoes analogous generalization. The coefficient w_i value may be interpreted as indicating the significance (weight) of the ith element of the pattern

to classification correctness. Note that if $w_i \equiv 1$, then definition (20) is regressed to initial form (2). Generalization (8) and (20) can be joined naturally, which gives

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} \frac{w_i}{s_i} K\left(\frac{x - x_i}{hs_i}\right). \tag{21}$$

The procedure for reducing patterns sets (11)–(13) consists — in its basic form — of two phases: of calculating the weight w_i, and then removing those elements of random sample (1), for which the respective weights have the lowest values. These tasks will subsequently be presented in the next two sections.

6.1. *Calculation of weights w_i*

In the method designed here, for the purpose of reduction of sets (11)–(13), separate neural networks are built for each investigated class. In order to ensure coherence of the notation below, let now the index $j = 1, 2, \ldots, J$ characterizing particular classes, be arbitrarily fixed.

The constructed network has three layers and is unidirectional, with m inputs (corresponding to particular elements of a pattern), a hidden layer whose size is equal to the integral part of the number $\sqrt{m}$, and also one output neuron. This network is submitted to a learning process using a data set comprising of the values of particular kernels for subsequent pattern elements, while the given output constitutes the value of the kernel estimator calculated for the pattern element under consideration. The network's learning is carried out using backward propagation of errors with momentum factor. On finishing this process, the thus obtained network undergoes sensitivity analysis on learning data, in accordance with the method presented in the previous section. The resulting coefficients $\bar{S}_i$ describing sensitivity, obtained on the basis of formula (18), constitute the fundament for calculating preliminary values

$$\tilde{w}_i = \left(1 - \frac{\bar{S}_i}{\sum_{j=1}^{m} \bar{S}_j}\right), \tag{22}$$

after which they are normed to

$$w_i = m \frac{\tilde{w}_i}{\sum_{i=1}^{m} \tilde{w}_i}, \tag{23}$$

with the aim of guaranteeing condition (19). It is worth noting that the form of formulas (17) and (18) accounting in practice for all coefficients $\bar{S}_i$ cannot be equal to zero, which guarantees feasibility of the above operation. The shape of formula (22) results from the fact that the network created here is the most sensitive to atypical and redundant elements, which — taking into account the form of kernel estimator (20) — implies a necessity to map the appropriately smaller values $\tilde{w}_i$, and in consequence w_i, to them. Coefficients (23) represent — as per the idea presented while introducing the general form (20) — the significance of particular elements of the pattern to accuracy of the classification process.

6.2. *Removal of pattern elements*

Empirical research confirmed the natural assumption that the pattern set should be relieved of those elements for which $w_i < 1$. (Note that, thanks to normalization made by formula (23), the mean value of coefficients w_i equals 1.) An increase in this value caused a sharp fall in classification quality, due to a loss of valuable and nonredundant information included in the pattern. In turn, decreasing such an assumed threshold value resulted in a significant drop in the degree of a pattern size reduction, while in the vicinity of the value 1 its influence on classification quality was practically unnoticeable. However, considerable diminishing implied a sizable rise in number of errors to the level obtained without reduction.

7. Correcting the Smoothing Parameter and Modification Intensity Values

Subject literature often presents the opinion that the classic universal methods of calculating the smoothing parameter value — most often based on a quadratic criterion — are not proper for the classification task [Ghosh *et al.* (2006)]. Available literature does not suggest a definitive solution for such a problem, especially in the multidimensional case and with more than two classes. This paper will propose a procedure suited to the conditioning of the investigated method of classification for nonstationary patterns, in particular those enabling successive adaptation with regard to the occurring changes.

Thus, it can be proposed to introduce $n+1$ multiplicative correcting coefficients for the values of the parameter defining the intensity of modification procedure c and smoothing parameters for particular coordinates $h_1, h_2, \ldots, h_n$, with respect to optimal ones calculated using the integrated square error criterion. Denote them as $b_0 \geq 0$, $b_1, b_2, \ldots, b_n > 0$, respectively. It is worth noticing that $b_0 = b_1 = \cdots = b_n = 1$ means in practice no correction. Next through a comprehensive search using a grid with a relatively large discretization value, one finds the most advantageous points regarding minimal incorrect classification sense. The final phase is a static optimization procedure in the $(n+1)$-dimensional space, where the initial conditions are the points chosen above, while the performance index is given as:

$$J(b_0, b_1, \ldots, b_n) = \#\,\{\text{incorrect classifications}\}, \tag{24}$$

when # denotes the size of a set, that is here the number of its elements. The value of the above functional for a fixed argument is calculated with the help of the classic leave-one-out method. As this value is an integer, to find the minimum a modified Hook–Jeeves algorithm [Kelley (1999)] was applied. Alternative concepts can be found in the survey paper [Venter (2010)].

Following experimental research it was assumed that the grid used for primary searches has intersections at the points $0.25, 0.5, \ldots, 1.75$ for every coordinate. For such intersections the value of functional (24) is calculated, after which the obtained results are sorted, and the 5 best become subsequent initial conditions for the

Hook–Jeeves method, where the value of the initial step is taken as 0.2. After finishing every one of the above 5 "runs" of this method, the functional (24) value for the end point is calculated, and finally among them the one with the smallest value is shown.

Apart from the first step, the above procedure can be used in the simplified version, to successively specify current values for the correcting coefficients $b_0, b_1, \ldots, b_n$ as part of adaptation to changes in nonstationary conditions. To this end the Hook–Jeeves algorithm is used only once, taking the coefficients' previous values as initial conditions.

Finally it is worth noting that the correction of classification parameters is not necessary in this procedure. It does, however, increase classification accuracy and furthermore enables the use of a simplified method for calculating smoothing parameters values (5).

8. Classification Method for Nonstationary Patterns

This section, the most essential in this publication, presents the classification method for the nonstationary case, that is when all or some patterns of classes undergo significant — considering the investigated task — changes. Here, material worked out and described in Secs. 2–7 will be used. A block diagram of the calculation procedure is presented in Fig. 3. Blocks symbolizing operations performed on all elements of patterns (11)–(13) jointly are drawn with a continuous line; a dashed line denotes operations on particular classes, while a dotted line is used for separate operations for each element of those patterns.

First one should fix the reference sizes of patterns (11)–(13), hereinafter denoted by $m_1^*, m_2^*, \ldots, m_J^*$. The patterns of these sizes will be the subject of a basic reduction procedure, described in Sec. 6. The sizes of patterns available at the beginning of the algorithm must not be smaller than the above referential values. These values can however be modified during the procedure's operation, with the natural condition that their potential growth does not increase the number of elements newly provided for the patterns. For preliminary research, $m_1^* = m_2^* = \cdots = m_J^* = 25 \cdot 2^n$ can be proposed. Lowering these values may worsen the classification quality, whereas an increase results in an excessive calculation time.

The elements of initial patterns (11)–(13) are provided as introductory data. Based on these — according to the procedures presented in Sec. 2 — the value of the parameter h is calculated (for the parameter c it is given by formula (10)). Figure 3 shows this action in block A. Next corrections in the parameters h and c values are made by taking the coefficients b_0, $b_1, \ldots, b_n$, as described in Sec. 7 (block B in Fig. 3).

The next procedure, shown by block C, is the calculation of the parameters w_i values mapped to particular patterns' elements, separately for each class, as in Sec. 6.1. Following this, within each class, the values of the parameter w_i are sorted (block D), and then — in block E — the appropriate m_1^*, $m_2^*, \ldots, m_J^*$ elements of

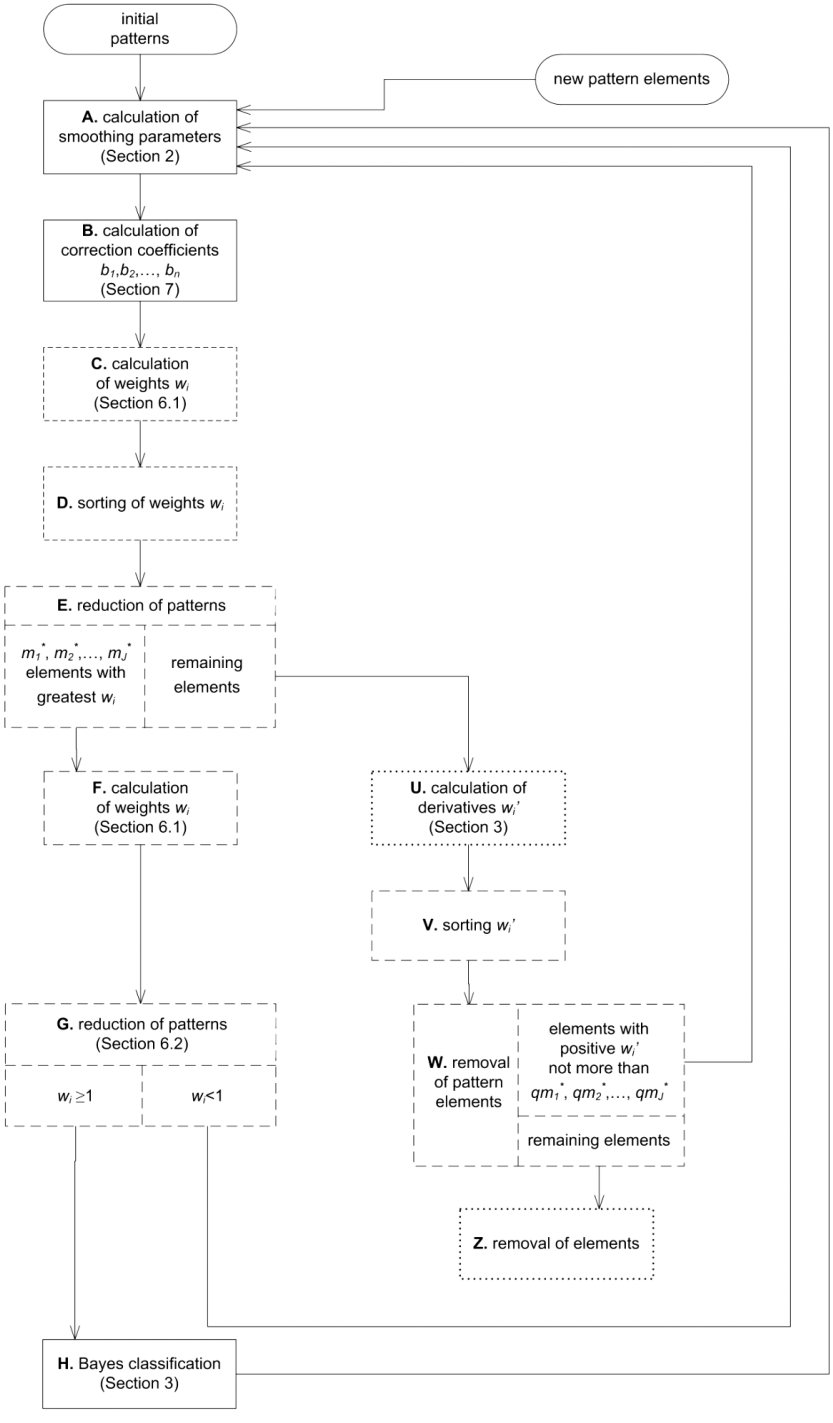

Fig. 3. Block diagram for classification algorithm.

the largest values w_i are designated to the classification phase itself. The remaining ones undergo further treatment, denoted in block U, which will be presented below, after Bayes classification has been dealt with.

The reduced patterns separately go through a procedure newly calculating the values of parameters w_i, presented in Sec. 6.1 and depicted in block F. According to Sec. 6.2, as block G in Fig. 3 denotes, these patterns' elements for which $w_i \geq 1$ are submitted to further stages of the classification procedure, while those with $w_i < 1$ are sent to block A for further processing in the next steps of the algorithm, after adding new elements of patterns. The final, and also the principal part of the procedure worked out here is Bayes classification, presented in Sec. 3 and marked by block H. Obviously many tested elements (14) can be subjected to classification separately. After the procedure has been finished, elements of patterns which have undergone classification are sent to the beginning of the algorithm to block A, to further avail of the next steps, following the addition of new elements of patterns.

Now — as mentioned two paragraphs earlier, in the last sentence — it remains to consider those patterns' elements, whose values w_i were not counted among the m_1^*, $m_2^*, \ldots, m_J^*$ largest for particular patterns. Thus, within block U, for each of them the derivative w_i' is calculated. If the element is "too new" and does not possess the $k-1$ previous values w_i, then the gaps are filled with zeros (because the values w_i generally oscillate around unity, such behavior significantly increases the derivative value, and in consequence ensures against premature elimination of this element). Next for each separate class, the elements w_i' are sorted (block V). As marked in block W, the respective

$$qm_1^*, qm_2^*, \ldots, qm_J^*, \tag{25}$$

elements of each pattern with the largest derivative values, on the additional requirement that the value is positive, go back to block A for further calculations carried out after the addition of new elements. If the number of elements with positive derivative is less than qm_1^*, $qm_2^*, \ldots, qm_J^*$, then the number of elements going back may be smaller (including even zero). The remaining elements are permanently eliminated from the procedure, as shown in block Z. In the above notation q is a positive constant influencing the proportion of patterns' elements with little, but successively increasing meaning. As a standard value $q = 0.2$ is proposed, or more generally $q \in [0.1, 0.25]$ depending on the size/speed of changes. An increase in this parameter value allows more effective conforming to pattern changes, although this potentially increases the calculation time, while lowering it may significantly worsen adaptation. In the general case this parameter can be different for particular patterns — then formula (25) takes the form $q_1 m_1^*$, $q_2 m_2^*, \ldots, q_J m_J^*$, where q_1, $q_2, \ldots, q_J$ are positive.

The above procedure is repeated following the addition of new elements (block A in Fig. 3). Besides these elements — as has been mentioned earlier — for particular patterns respectively m_1^*, $m_2^*, \ldots, m_J^*$ elements of the greatest values w_i are taken, as well as up to $qm_{_1}^*$, $qm_{_2}^*, \ldots, qm_{_J}^*$ (or in the generalized case $q_1 m_{_1}^*$, $q_2 m_{_2}^*, \ldots, q_J m_{_J}^*$)

elements of the greatest derivative w_i', so successively increasing its significance, most often due to the nonstationarity of patterns.

9. Empirical Verification

The correct functioning and properties of the concept under investigation have been comprehensively verified numerically, and also compared with results obtained using a related procedure based on a support vector machine method [Krasotkina *et al.* (2011)]. Research was carried out for data sets in various configurations and with different properties, particularly with nonseparated classes, complex patterns, multimodal and consisting of detached subsets located alternately. Nonstationarity was successive either in steps or periodical. The standard values of the parameters previously proposed in this paper were obtained through research carried out for verification purposes.

The following are the results obtained for a simple but representative case, enabling a telling illustration and interpretation of the procedure summaries in Sec. 8. For visual purposes the two dimensional space ($n = 2$) and the two classes ($J = 2$) will be used. The first class is invariable, while the second is also invariable at the beginning, but then moves, describing a circle around the first, before stopping at its initial location.

For both classes, the patterns begin with 100 elements ($m_1 = m_2 = 100$), obtained using a generator with normal distribution, respectively

$$E_s = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathrm{Cov}_s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{26}$$

$$E_{\mathrm{ns}} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad \mathrm{Cov}_{\mathrm{ns}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{27}$$

Next 10 elements are added at every step, whereas when the second class describes the circle, generator (27) is generalized to

$$E_{\mathrm{ns}} = \begin{bmatrix} 3\cos(k) \\ 3\sin(k) \end{bmatrix}, \quad \mathrm{Cov}_{\mathrm{ns}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{for } k = 0^\circ, 10^\circ, 20^\circ, \ldots, 360^\circ. \tag{28}$$

So, the first — stationary — class is located permanently in the origin of the space $\mathbb{R}^2$, while the second — nonstationary — following an initial period of no movement, encircles it with the radius 3, adding 10 new elements every 10° before coming to a stop in its original location. According to the suggestions formulated earlier, it was also assumed $m_1^* = m_2^* = 100$ and $q = 0.2$.

Figure 4 illustrates the number of misclassifications in a typical course of a procedure created in this paper. From the beginning, up to step 18, the second class is invariable. First a slight increase in the number of erroneous classifications occurs — in every step new elements (around 10%) are added to patterns, which worsens the working conditions for the neural network. Finally, however, once the patterns are stabilized, the number of misclassifications settles at the level 0.08.

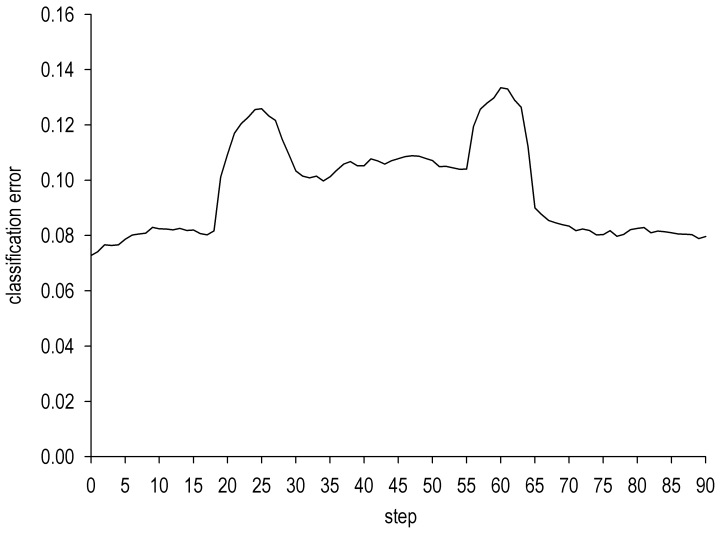

Fig. 4. Typical course using the investigated procedure ($z_1 = z_2 = 1, q = 0.2$).

In step 18 the aforementioned orbital movement of the second class begins. First the number of erroneous classifications rises to around 0.12, and then — after the kernels, which were not previously removed due solely to a positive derivative w_i', have received the appropriate meaning — the number of misclassifications drops and levels off at 0.105. In step 54, where the second class stops, occurrences similar to the above take place, when the number of classification errors returns to its initial level of 0.08.

Let us analyze the influence of the coefficients $z_1, z_2, \ldots, z_J$, introduced by formula (16). The research showed that a number of misclassifications to a large degree result from the assignation of elements from a nonstationary class to a stationary one, as opposed to the contrary. This is easy to interpret, as the pattern of a nonstationary class contains elements which are current to varying degrees and therefore has a naturally smaller power of "attraction" than a stationary pattern. Moreover, the pattern of a stationary class contains not only current elements (*nota bene*: in the nonstationary case such elements may not even exist in large enough numbers to infer), but also, as the procedure progresses, elements of this pattern are successively improved by substituting them for elements with ever greater values w_i, which increases its "attraction" even more. Both factors lead to a significant asymmetry of classification errors. This can be dealt with by increasing the coefficients z_i values in proportion to the size/speed of changes of particular classes. As an initial value, 1.25, which has been fixed during the following research, is suggested. Figure 5 shows the course similar to Fig. 4, and in addition the number of classification errors of elements from a stationary to a nonstationary class and *vice versa*, with $z_1 = 1$ and $z_2 = 1.25$, respectively. During the first 18 steps and the last ones,

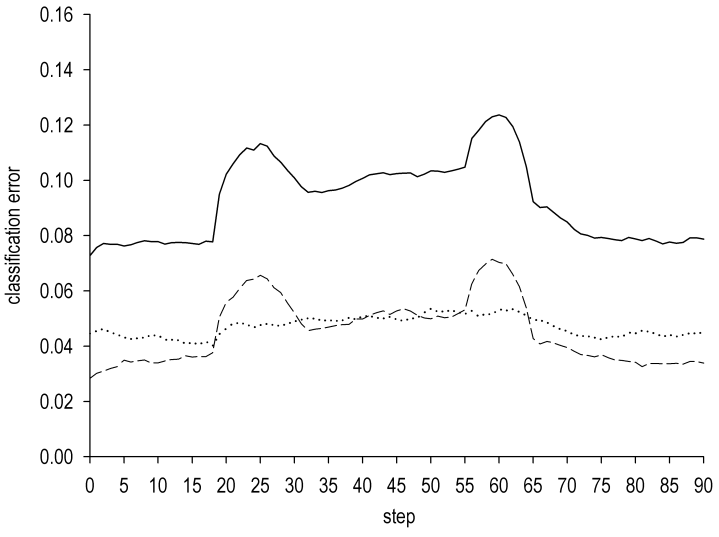

Fig. 5. Course with differing values of the coefficients $z_i (z_1 = 1, z_2 = 1.25)$; the dotted line denotes errors in assigning elements of the first pattern to the second, the broken line shows the inverse.

following the 54th, when the second class once more does not change its position, the errors in assigning elements from the first to the second dominate, which results directly from assuming $z_1 < z_2$. The opposite situation occurs between steps 18 and 54, when the second class moves. If however the standard $z_2 = 1$ remains, the disproportion would be significantly greater here. Conversely the number of errors would be leveled here for $z_2 = 1.5$, but this would in turn increase the difference in the period of no change of the second class (steps to 18 and after 54). It is worth noting that the total number of misclassifications lowered with respect to that obtained for the basic case in Fig. 4. This especially concerns local maximums existing after the second class starts and stops moving (steps 18 and 54).

The procedure worked out and described here was compared with a method based on the support vector concept (SVM), presented in the publication [Krasotkina *et al.* (2011)], taken as the closest regarding the conditioning considered in this paper research task. The obtained results are shown in Fig. 6 — they were achieved in conditions identical to Fig. 5, with which they will be compared. Although in conditions of stationarity of the second pattern (steps before 18 and after 54), the number of misclassifications leveled off at 0.08, the case using the SVM, however, starts at 0.10, instead of 0.07 as in the procedure investigated here (compare Figs. 5 and 6). When the second pattern changes (steps 18–54) the amount of errors generated by the SVM settles at the level 0.12, or even slightly higher than that of local maximums appearing in the method presented in this paper after steps 18 and 45 (compare again Figs. 5 and 6). It should be underlined, though, that when the second class is moving, the number of misclassifications does not fall to level 0.10

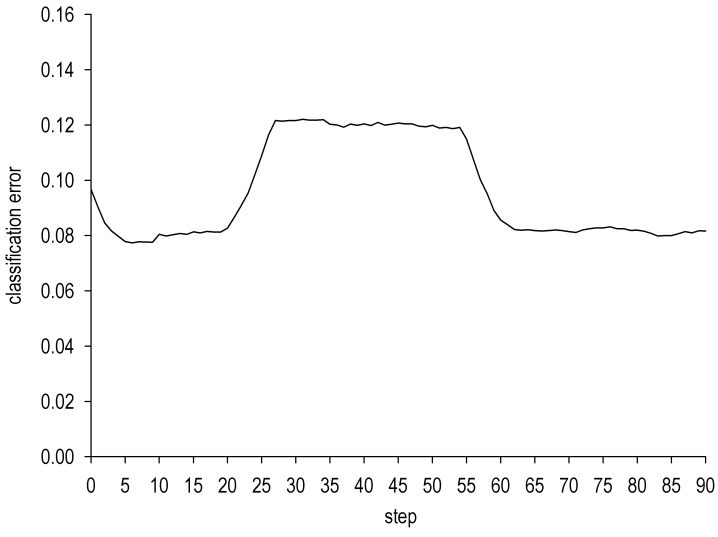

Fig. 6. Course using the SVM (compare with Fig. 5).

(Fig. 6), as is the case with the procedure worked out here (Fig. 5). Thus one can see that the concept method in this paper has an advantage over the SVM procedure, especially in conditions of gradual change. Taking into account the fact that its idea is based on derivatives of a predictive nature, this observation is completely understandable.

10. Final Remarks

This paper presented a classification procedure which allows for nonstationarity of patterns and successive supply of new elements to them. Neither the number of classes itself, nor the number of nonstationary ones are methodologically limited. The attribute space is a finite-dimensional, and particular coordinates can be continuous, binary, discrete or categorical in character; they may also be any combination of these types.

The concept is based on the Bayes approach, which allows for the minimization of expected loss value arising from erroneous classifications, as well as actively influencing the proportion of probabilities of classification errors between particular classes. The use of kernel estimators frees the algorithm from patterns' shapes. The procedure operation is based on the sensitivity method used with artificial neural networks. It enables the removal of those elements of patterns, which are of insignificant or even negative influence on accuracy of results. However, it retains for further calculation some of these elements which due to nonstationarity successively increase their positive impact. Appropriate adaptation is also performed on classifier parameters.

Numerical testing wholly confirmed the positive features of the method worked out. In particular, the results show that the classifying algorithm can be used successfully for inseparable classes of complex multimodal patterns as well as for those consisting of incoherent subsets at alternate locations. The examined nonstationarity increased successively, and was periodical as well as occurring in steps. For the former type, the procedure investigated and presented in this paper proved to be particularly advantageous and useful.

The investigated method has been described in this paper in its fundamental form. In concrete applications however its various modifications may prove useful. For example, one can apply other concepts for calculating the value of the derivative or static optimization besides Newton's and Hooke–Jeeves methods proposed in Secs. 4 and 7. It is worth persuading designers to try modifying the proposed algorithm — an illustrative interpretation of its particular procedures would allow it to be suited to the specific conditions of elaborated problems as well as individual preferences and customs.

References

Aggarwal, C. C., Han, J., Wang, J. and Yu, P. S. [2006] "A framework for on-demand classification of evolving data streams," *IEEE Transactions on Knowledge and Data Engineering* **18**, 577–589.

Ahmad, I. A. and Cerrito, P. B. [1994] "Nonparametric estimation of joint discrete-continuous probability densities with applications," *Journal of Statistical Planning and Inference* **41**, 349–364.

Aitchison, J. and Aitken, C. [1976] "Multivariate binary discrimination by the Kernel method," *Biometrika* **63**, 413–420.

Bouchachia, A. [2009] "Adaptation in classification systems," in *Foundations of Computational Intelligence*, Vol. 2, eds. Hassanien, A. E., Abraham, A. and Herrera, F. (Springer, Berlin), pp. 237–258.

Duda, R. O., Hart, P. E. and Storck, D. G. [2001] *Pattern Classification* (Wiley, New York).

Engelbrecht, A. P., Cloete, I. and Zurada, J. [1995] "Determining the significance of input parameters using sensitivity analysis," in *From Natural to Artificial Neural Computation*, eds. Mira, J. and Sandoval, F., Lecture Notes in Computer Science (Springer-Verlag, Berlin-Heidelberg), pp. 382–388.

Everitt, B. S., Landau, S., Leese, M. and Stahl, D. [2011] *Cluster Analysis* (Wiley, New York).

Gaosheng, J., Rui, L. and Zhongwen, L. [2009] "Nonparametric estimation of multivariate CDF with categorical and continuous data," *Advances in Econometrics* **25**, 291–318.

Ghosh, A. K., Chaudhuri, P. and Sengupta, D. [2006] "Classification using Kernel density estimation: Multiscale analysis and visualization," *Technometrics* **48**, 120–132.

Han, J. and Kamber, M. [2001] *Data Mining*: *Concepts and Techniques* (Wiley, New York).

Harries, M., Sammut, C. and Horn, K. [1998] "Extracting hidden context," *Machine Learning* **32**, 101–126.

Kelley, C. T. [1999] *Iterative Methods for Optimization* (SIAM, Philadelphia).

Kenyon, S. C. [1991] "Hyperspace organization for classification of non-stationary patterns," *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, Charlottesville, 13–16 October 1991, pp. 185–190.

Krasotkina, O. V., Mottl, V. V. and Turkov, P. A. [2011] "Bayesian approach to the pattern recognition problem in nonstationary environment," in *Pattern Recognition and Machine Intelligence*, eds. Kuznetzov, S. O., Mandal, D. P., Kundu, M. K. and Pal, S. K., Lecture Notes in Computer Science, Vol. 6744 (Springer, Berlin), pp. 24–29.

Kulczycki, P. [2005] *Estymatory jadrowe w analizie systemowej* (WNT, Warsaw).

Kulczycki, P. and Charytanowicz, M. [2010] "A complete gradient clustering algorithm formed with kernel estimators," *International Journal of Applied Mathematics and Computer Science* **20**, 123–134.

Kulczycki, P., Hryniewicz, O. and Kacprzyk, J., eds. [2007] *Techniki informacyjne w badaniach systemowych* (WNT, Warsaw).

Kulczycki, P. and Kowalski, P. A. [2011] "Bayes classification of imprecise information of interval type," *Control and Cybernetics* **40**, 101–123.

Kulczycki, P. and Kowalski, P. A. [2013a] "Klasyfikacja bayesowska przy niestacjonarnych wzorcach," in *11th International Conference on Diagnostics of Processes and Systems*, Lagow Lubuski, Poland, 8–11 September 2013, pendrive: paper_4.

Kulczycki, P. and Kowalski, P. A. [2013b] "An algorithm of classification for nonstationary case," in *12th Mexican International Conference on Artificial Intelligence*, Mexico City, 24–30 November 2013; *Advances in Artificial Intelligence and Its Applications*, eds. Castro, F., Gelbukh, A. and Mendoza, M. G., Lecture Notes in Computer Science, Vol. II (Springer, Berlin), pp. 301–313.

Kuncheva, L. I. [2004] "Classifier ensembles for changing environments," in *Multiple Classifier Systems*, eds. Roli, F., Kittler, J. and Windeatt, T., Lecture Notes in Computer Science (Springer, Berlin), pp. 1–15.

Li, Q. and Racine, J. S. [2008] "Nonparametric estimation of conditional CDF and quantile functions with mixed categorical and continuous data," *Journal of Business and Economic Statistics* **26**, 423–434.

Muhlbaier, M. D. and Polikar, R. [2007] "An ensemble approach for incremental learning in nonstationary environments," in *Multiple Classifier Systems*, eds. Haindl, M., Kittler, J. and Roli, F., Lecture Notes in Computer Science (Springer, Berlin), pp. 490–500.

Ouyang, D., Li, Q. and Racine, J. [2006] "Cross-validation and the estimation of probability distributions with categorical data," *Journal of Nonparametric Statistics* **18**, 69–100.

Salganicoff, M. [1997] "Tolerating concept and sampling shift in lazy learning using prediction error context switching," *AI Review* **11**, 133–155.

Silverman, B. W. [1986] *Density Estimation for Statistics and Data Analysis* (Chapman and Hall, London).

Venter, G. [2010] "Review of optimization techniques," in *Encyclopedia of Aerospace Engineering* (Wiley, New York), pp. 5229–5238.

Wand, M. P. and Jones, M. C. [1995] *Kernel Smoothing* (Chapman and Hall, London).

Wang, M. and van Ryzin, J. [1981] "A class of smooth estimators for discrete distributions," *Biometrika*, **68**, 301–309.

Widmer, G. and Kubat, M. [1996] "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, **23**, 69–101.

Zlobaite, I. [2009] *Learning Under Concept Drift*: *An Overview*, Technical Report, Faculty of Mathematics and Informatics, Vilnius University.

Zurada, J. [1992] *Introduction to Artificial Neural Systems* (West Publishing, St. Paul).