Int. J. Appl. Math. Comput. Sci., 2003, Vol. 13, No. 2, 215–223

A FUZZY *IF-THEN* RULE-BASED NONLINEAR CLASSIFIER

JACEK ŁĘSKI*

* Institute of Electronics, Silesian University of Technology
Akademicka 16, 44–100 Gliwice, Poland
e-mail: jl@boss.iele.polsl.gliwice.pl

This paper introduces a new classifier design method that is based on a modification of the classical Ho-Kashyap procedure. The proposed method uses the absolute error, rather than the squared error, to design a linear classifier. Additionally, easy control of the generalization ability and robustness to outliers are obtained. Next, an extension to a nonlinear classifier by the mixture-of-experts technique is presented. Each expert is represented by a fuzzy *if-then* rule in the Takagi-Sugeno-Kang form. Finally, examples are given to demonstrate the validity of the introduced method.

Keywords: classifier design, fuzzy *if-then* rules, generalization control, mixture of experts

1. Introduction

Pattern recognition is concerned with the classification of patterns into categories. This field of study was developed in the early 1960s, and it plays an important role in many engineering fields, such as medical diagnosis, computer vision, character recognition, data mining, communication, etc. Two of the main textbooks on pattern recognition are those written by Duda and Hart (1973), and Tou and Gonzalez (1974).

There are two main categories of classification methods: supervised (discrimination) and unsupervised (clustering) ones. In supervised classification we have a set of data, called the training set, with class labels associated with each datum. In the literature there are many classifiers, including statistical, linear discriminant, k-nearest neighbour, kernel, neural network, classification tree, and many more (Duda and Hart, 1973; Ripley, 1996; Tou and Gonzalez, 1974; Webb, 1999). But linear classifiers are of special interest, due to their simplicity and easy expansibility to nonlinear classifiers. One of the most powerful classical methods of linear classifiers is the least mean-squared error procedure with the Ho-Kashyap modification (Ho and Kashyap, 1965; 1966). Two main disadvantages of this approach are: (i) the use of the quadratic loss function, which leads to a non-robust method, (ii) the impossibility of minimizing the Vapnik-Chervonenkis (VC) dimension of the designed classifier.

The most important feature of the classifier is its generalization ability, which refers to producing a reasonable decision for data previously unseen during the process of classifier design (training). The easiest way to measure the generalization ability is to use a test set that contains data that do not belong to the training set.

From statistical learning theory, we know that in order to achieve good generalization capability, we should select a classifier with the smallest VC dimension (complexity) and the smallest misclassification error on the training set. This principle is called the principle of Structural Risk Minimization (SRM) (Vapnik, 1998; 1999).

In real applications, data from the training set are corrupted by noise and outliers. It follows that classifier design methods need to be robust. According to Huber (Huber, 1981), a robust method should have the following properties: (i) reasonably good accuracy at the assumed model, (ii) small deviations from the model assumptions should impair the performance only by a small amount, (iii) larger deviations from the model assumptions should not cause a catastrophe. In the literature there are many robust loss functions (Huber, 1981). In this work, due to its simplicity, the absolute error loss function is of special interest.

The paper by Bellman *et al.* (Bellman *et al.*, 1966) was the starting point in the application of fuzzy set theory to pattern classification. Since then, researchers have found several ways to apply this theory to generalize the existing pattern classification methods, as well as to develop new algorithms (Abe S. and Lan, 1995; Bezdek and Pal, 1992; Ishibuchi *et al.*, 1999; Kuncheva, 2000a; Malek *et al.*, 2002; Marín-Blázquez and Shen, 2002; Nath and Lee, 1982; Setnes and Babuška, 1999). There are two main categories of fuzzy classifiers (Kuncheva, 2000b): fuzzy *if-then* rule-based and non *if-then* rule fuzzy classifiers. The second group may be divided into fuzzy k-nearest neighbours (Keller *et al.*, 1985) and generalized nearest prototype classifiers (GNPC) (Kuncheva and Bezdek, 1999). Several approaches have been proposed for automatically generating fuzzy *if-then* rules

and tuning parameters of membership functions from numerical data. These methods fall into three categories: neural-network-based methods, with high learning abilities, genetic (evolution)-based methods, with the Michigan and Pittsburg approaches, and clustering-based methods. There are several methods that combine the above-enumerated categories that have proved effective in improving classification performance (Czogała and Łęski, 2000; Rutkowska, 2002). Recently, a new direction in the fuzzy classifier design field has emerged: a combination of multiple classifiers using fuzzy sets (Bezdek *et al.*, 1998; Kuncheva, 2001; 2002), which may be included into the non *if-then* fuzzy classifier category. There are generally two types of the combination: classifier selection and classifier fusion. In the first approach each classifier is an 'expert' in some local area of the feature space. In the second approach all classifiers are trained over the whole feature space. Thus, in this case, we have competition, rather than complementing, among the fuzzy classifiers. Various methods have been proposed for fuzzy classifier design; however, in contrast to statistical and neural pattern classifiers, both theoretical and experimental studies concerning fuzzy classifiers do not deal with the analysis of the influence of the classifier complexity on the generalization error. Therefore, in this paper, the generalization ability of a fuzzy classifier will also be discussed.

The goal of this work is twofold. First, we wish to introduce a modification to the classical Ho-Kashyap procedure. Next, a chief aim is to propose an extension of this method to the nonlinear case, using the mixture-of-experts technique. Each expert is represented by a fuzzy *if-then* rule in the Takagi-Sugeno-Kang form. The regions of the experts' work are obtained by the fuzzy c-means clustering method. The proposed method uses the absolute loss function, resulting in robustness to outliers and a better approximation of the misclassification error. Additionally, this method minimizes the VC dimension of the designed classifier. The reminder of this work is concerned with two-class problems. The proposed method can be easily generalized to a multi-class problem using the class-rest and class-class methodologies (Tou and Gonzalez, 1974).

According to the characteristics of the fuzzy classifiers presented above, the classifier discussed in this paper falls into the fuzzy *if-then* rule classifier category. However, a new subcategory is proposed, where fuzzy *if-then* rules are extracted automatically using a combination of the fuzzy clustering method and a weighted support vector machine, which may be called the weighted-support-vector-based fuzzy classifier. The weighted support vector machine leads to a quadratic-programming problem which is characterized by a high computational burden (Łęski, 2002). Thus, a computationally effective method based on a modification of the Ho-Kashyap algorithm (Ho and Kashyap, 1965) will be proposed. A nonlinear *if-then* rule-based classifier may be also included into the combination of multiple classifiers using the fuzzy sets methodology with competition.

This paper is organized as follows: Section 2 describes design procedures for linear and nonlinear classifiers with generalization control. Section 3 presents simulation results and discusses the classification of simple synthetic two-dimensional data and real-world high-dimensional data. Finally, conclusions are drawn in Section 4.

2. Classifier Design

2.1. Linear Case

The classifier is designed on the basis of a data set, called the training set, $Tr^{(N)} = \{(\mathbf{x}_1, \varphi_1), (\mathbf{x}_2, \varphi_2), \ldots, (\mathbf{x}_N, \varphi_N)\}$, where N is the data cardinality, and each independent datum (pattern) $\mathbf{x}_i \in \mathbb{R}^t$ has a corresponding dependent datum $\varphi_i \in \{+1, -1\}$, which indicates the assignment to one of two classes, ω_1 or ω_2:

$$\varphi_i = \begin{cases} +1, & \mathbf{x}_i \in \omega_1, \\ -1, & \mathbf{x}_i \in \omega_2. \end{cases} \tag{1}$$

Defining the augmented pattern vector $\mathbf{x}'_i = \left[\mathbf{x}_i^T, 1\right]^T$, we seek a weight vector $\mathbf{w} \in \mathbb{R}^{t+1}$ such that

$$g(\mathbf{x}_i) \triangleq \mathbf{w}^T \mathbf{x}'_i \begin{cases} > 0, & \mathbf{x}'_i \in \omega_1, \\ < 0, & \mathbf{x}'_i \in \omega_2, \end{cases} \tag{2}$$

where $g(\mathbf{x}_i)$ is called the linear discrimination (or decision) function.

If the condition (2) is satisfied for all members of the training set, then the data are said to be linearly separable. For overlapping classes it is impossible to find a weight vector $\mathbf{w}$ such that (2) is satisfied for all data from the training set. If we multiply by -1 all patterns of the training set which are members of the class ω_2, then (2) can be rewritten in the form $\varphi_i \mathbf{w}^T \mathbf{x}'_i > 0$ for $i = 1, 2, \ldots, N$. Let $\mathbf{X}$ be the $N \times (t+1)$ matrix

$$\mathbf{X} \triangleq \begin{bmatrix} \varphi_1 \mathbf{x}_1'^T \\ \varphi_2 \mathbf{x}_2'^T \\ \vdots \\ \varphi_N \mathbf{x}_N'^T \end{bmatrix}. \tag{3}$$

Then (2) can be written down in the matrix form $\mathbf{Xw} > \mathbf{0}$. To obtain a solution, the above system of linear inequalities is replaced by the system of linear equalities $\mathbf{Xw} = \mathbf{b}$, where $\mathbf{b} > \mathbf{0}$ is an arbitrary vector. We define the error

vector as $\mathbf{e} = \mathbf{X}\mathbf{w} - \mathbf{b}$. If the p-th component of $\mathbf{e}$ is positive, i.e. $e_p \geq 0$, then the p-th pattern falls on the right side of the separation hyperplane, and by increasing the respective component of $\mathbf{b}$ (b_p), e_p can be set to zero. If the p-th component of $\mathbf{e}$ is negative, then the p-th pattern falls on the wrong side of the separation hyperplane, and it is impossible to retain the condition $b_p > 0$ while decreasing b_p. Thus, the misclassification error can be written in the form

$$I(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} \mathbb{U}(-e_i), \tag{4}$$

where $\mathbb{U}(\cdot)$ denotes the unit step pseudo-function, $\mathbb{U}(e_i) = 1$ for $e_i > 0$, and $\mathbb{U}(e_i) = 0$ otherwise. We should minimize the criterion (4), but due to its nonconvexity this optimization problem is NP-complete. To make this optimization problem tractable, we approximate the criterion (5) by a convex one

$$I(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} |e_i| \quad \text{or} \quad I(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} (e_i)^2. \tag{5}$$

The above approximations are possible due to the fact that positive error values can be set to zero by increasing the respective components of $\mathbf{b}$. The first criterion in (5) is a better approximation of (4), but due to the simplicity of the solution, we start from the second criterion (5).

Now, we seek vectors $\mathbf{w}$ and $\mathbf{b}$ by minimizing the criterion function

$$\min_{\mathbf{w}\in\mathbb{R}^{t+1}, \mathbf{b}>\mathbf{0}} I(\mathbf{w}, \mathbf{b}) \triangleq (\mathbf{X}\mathbf{w} - \mathbf{b})^T \mathbf{D} (\mathbf{X}\mathbf{w} - \mathbf{b}) + \tau \mathbf{w}_n^T \mathbf{w}_n, \tag{6}$$

where $\mathbf{w}_n$ is formed from $\mathbf{w}$, by removing its last component. The matrix $\mathbf{D} = \mathrm{diag}(d_1, d_2, \ldots, d_N)$, where d_i is the weight corresponding to the i-th pattern, can be interpreted as reliability attached to this pattern. The criterion function (6) is the squared error weighted by coefficients d_i with the second term related to the minimization of the Vapnik-Chervonenkis dimension (complexity) of the classifier. The parameter $\tau > 0$ controls the trade-off between the classifier complexity and the amount up to which the errors are tolerated.

The most important idea in statistical learning theory is the Structural Risk Minimization (SRM) induction principle. It implies a trade-off between the quality of approximation and the complexity of the approximation function (Vapnik, 1998). The measure of the approximation function complexity (or capacity) is called the VC-dimension. It is a purely theoretical quantity which measures the capacity of a learning machine. This capacity is a determining factor in bounding the difference between the training and generalization (testing) errors of the learning machine. An analytic calculation of the VC-dimension can only be performed for very few and simple learning machines. Thus, the parameter values of the learning machine (τ in our case) were chosen as the values for which the machine has the best generalization ability measured by cross-validation on the test set.

Optimality conditions are obtained by differentiating (6) with respect to $\mathbf{w}$ and $\mathbf{b}$, and setting the results to zero:

$$\begin{cases} \mathbf{w} = \left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \tau\widetilde{\mathbf{I}}\right)^{-1} \mathbf{X}^T\mathbf{D}\mathbf{b}, \\ \mathbf{e} \triangleq \mathbf{X}\mathbf{w} - \mathbf{b} = \mathbf{0}, \end{cases} \tag{7}$$

where $\widetilde{\mathbf{I}}$ is the identity matrix with the last element on the main diagonal set to zero.

From the first equation of (7), we see that the vector $\mathbf{w}$ depends on the vector $\mathbf{b}$. The vector $\mathbf{b}$ is called the margin vector, because its components determine the distance from the patterns to the separating hyperplane. For a fixed $\mathbf{w}$, if a pattern lies on the right side of the hyperplane, the corresponding margin can be increased to obtain the zero error. However, if a pattern lies on the wrong side of the hyperplane, then the error is negative, and we may decrease the error only by decreasing the corresponding margin value. But one way to prevent $\mathbf{b}$ from converging to zero is to start with $\mathbf{b} > \mathbf{0}$ and to refuse to decrease any of its components. Ho and Kashyap (1965; 1966) proposed an iterative algorithm for alternately determining $\mathbf{w}$ and $\mathbf{b}$, where the components of $\mathbf{b}$ cannot decrease. Now, this algorithm can be extended to our weighted squared error criterion with regularization. The vector $\mathbf{w}$ is determined based on the first equation of (7), i.e. $\mathbf{w}^{[k]} = (\mathbf{X}^T\mathbf{D}\mathbf{X} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{D}\mathbf{b}^{[k]}$, where the superscript $[k]$ denotes the iteration index. The components of $\mathbf{b}$ are modified by the components of the error vector $\mathbf{e}$, but only in the case when it results in an increase in the components of $\mathbf{b}$. Otherwise, the components of $\mathbf{b}$ remain unmodified:

$$\mathbf{b}^{[k+1]} = \mathbf{b}^{[k]} + \rho\left(\mathbf{e}^{[k]} + \left|\mathbf{e}^{[k]}\right|\right), \tag{8}$$

where $\rho > 0$ is a parameter.

Note that for $\mathbf{D} = \mathbf{I}$ ($d_i = 1$) and $\tau = 0$ the original Ho-Kashyap algorithm is obtained. Now, another method for the selection of the parameters d_i will be proposed. Real data have noise and outliers. It follows that classifier design methods need to be robust. It is well known from the literature (Huber, 1981) that the minimum squared error procedure does not lead to robust methods. One of the simplest techniques to obtain robust methods is to use the minimum absolute error procedure. The absolute error criterion is easy to obtain by taking $d_i = 1/|e_i|$ for all

$i = 1, 2, \ldots, N$, where e_i is the i-th component of the error vector. But the error vector depends on $\mathbf{w}$. So, we use the vector $\mathbf{w}$ from the previous iteration. This procedure is based on the hypothesis that near the optimum, solution sequential vectors $\mathbf{w}^{[k]}$ differ imperceptibly. The absolute error minimization procedure for classifier design can be summarized in the following steps:

1. Fix $\tau > 0$, $\rho > 0$ and $\mathbf{D}^{[1]} = \mathbf{I}$. Initialize $\mathbf{b}^{[1]} > \mathbf{0}$. Set iteration index $k = 1$.
2. Set $\mathbf{w}^{[k]} = (\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{X} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{b}^{[k]}$.
3. Determine $\mathbf{e}^{[k]} = \mathbf{X}\mathbf{w}^{[k]} - \mathbf{b}^{[k]}$.
4. Set $d_i = 1/|e_i|$, for $i = 1, 2, \ldots, N$, $\mathbf{D}^{[k+1]} = \operatorname{diag}(d_1, \ldots, d_N)$.
5. Calculate $\mathbf{b}^{[k+1]} = \mathbf{b}^{[k]} + \rho(\mathbf{e}^{[k]} + |\mathbf{e}^{[k]}|)$.
6. If $\|\mathbf{b}^{[k+1]} - \mathbf{b}^{[k]}\| > \xi$, then set $k = k + 1$ and go to Step 2. Otherwise, stop.

Remark 1. Appendix shows that for $0 < \rho < 1$ and any diagonal matrix $\mathbf{D}$, the above algorithm is convergent. If Step 4 of this algorithm is omitted, then a procedure for the minimization of the squared error is obtained. In practice, a divide-by-zero error in Step 4 does not occur. This results from the fact that some components of the vector $\mathbf{e}$ tend to zero as $[k] \to \infty$. But in this case convergence is slow and the condition from Step 6 stops the algorithm.

2.2. Nonlinear Extension

In the previous subsection the linear discriminant function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}'$ that minimizes the absolute (or squared) error as well as the classifier complexity has been described. Now, we propose an extension of this classifier using c linear discriminant functions $g_i(\mathbf{x}) = \mathbf{w}^{(i)T}\mathbf{x}'$, $i = 1, 2, \ldots, c$. The input space $\mathbb{R}^t$ is softly partitioned into c regions. If we denote by u_{ik} the membership of the k-th datum from the training set to the i-th region, then the criterion (6) takes the form

$$I\left(\left\{\mathbf{w}^{(i)}\right\}, \left\{\mathbf{b}^{(i)}\right\}\right) = \sum_{i=1}^{c}\left(\mathbf{X}\mathbf{w}^{(i)T} - \mathbf{b}^{(i)}\right)^T\mathbf{D}^{(i)}\left(\mathbf{X}\mathbf{w}^{(i)T} - \mathbf{b}^{(i)}\right) + \tau\mathbf{w}_n^{(i)T}\mathbf{w}_n^{(i)}, \tag{9}$$

where

$$\mathbf{D}^{(i)} = \operatorname{diag}\left(u_{i1}/\left|e_1^{(i)}\right|, u_{i2}/\left|e_2^{(i)}\right|, \ldots, u_{iN}/\left|e_N^{(i)}\right|\right)$$

and

$$e_k^{(i)} = \mathbf{w}^{(i)T}\mathbf{x}_k' - b_k^{(i)}.$$

The vector $\mathbf{b}^{(i)}$ denotes the margin vector for the i-th classifier.

Now, for a fixed partition of the input space, represented by u_{ik}, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, N$, the minimization of the criterion (9) can be decomposed into c minimization processes of the criterion (6) for $\mathbf{D}^{(i)}$ of the above-mentioned form. To obtain a partition of the input space, each of the two classes (ω_1 and ω_2) from the training set is first clustered by the fuzzy c-means algorithm (Bezdek, 1982). There are two approaches to represent the obtained clusters: (i) the use of the original fuzzy c-means membership functions (Setnes and Babuška, 1999), (ii) the use of a parameterized approximation of clusters obtained by fuzzy c-means (Runkler and Bezdek, 1999). Both the approaches have advantages and disadvantages. The original fuzzy c-means membership function decreases monotonically around the cluster centre, but increases in regions distant from other cluster centres. This effect comes from the use of the probabilistic constraint that the memberships of a datum across clusters must sum up to one (Krishnapuram and Keller, 1993). The fuzzy c-means membership functions are also non-symmetric due to a non-uniform distribution of cluster centres. Usually, in the second approach to represent data clusters, symmetric Gaussian membership functions are used (Czogała and Łęski, 2000; Kim *et al.*, 1997; Łęski and Henzel, 2001; Rutkowska, 2002).

The use of Gaussian membership functions leads to simplicity in further calculations and a possibility to interpret the obtained system as a radial-basis neural network. It is an open problem which approach leads to better accuracy in fuzzy modelling. However, for simplicity, in further deliberations, each cluster is represented parametrically by a Gaussian membership function with centre $\mathbf{v}^{(i)(j)}$ and dispersion $\mathbf{s}^{(i)(j)}$, where $j \in \{1, 2\}$ is a class index, and $i \in \{1, 2, \ldots, c\}$ is a cluster index. The p-th component of $\mathbf{s}^{(i)(j)}$ represents the dispersion of the data which belong to the i-th cluster of the j-th class, along the p-th axis in the input space. If we denote the elements of fuzzy partition matrices by $u_{ik}^{(1)}$ and $u_{ik}^{(2)}$ for the ω_1 and ω_2 classes, respectively, the parameters of Gaussian membership functions can be obtained as

$$\mathbf{v}^{(i)(j)} = \frac{\sum_{k=1}^{N_j} u_{ik}^{(j)}\mathbf{x}_k}{\sum_{k=1}^{N_j} u_{ik}^{(j)}} \tag{10}$$

and

$$\mathbf{s}^{(i)(j)} = \frac{\sum_{k=1}^{N_j} u_{ik}^{(j)}\left[\mathbf{x}_k - \mathbf{v}^{(i)(j)}\right]^{(\cdot 2)}}{\sum_{k=1}^{N_j} u_{ik}^{(j)}}, \tag{11}$$

where the superscript '$(\cdot 2)$' denotes the component-by-component squaring.

After clustering, we search for c nearest pairs of clusters which belong to different classes. Each cluster belongs only to one pair. In searching for the nearest centre (prototype) of clusters, the norm $\|\mathbf{v}^{(i)(1)} - \mathbf{v}^{(j)(2)}\|_1$, $i, j = 1, 2, \ldots, c$ is used. Let $\Im^{(1)}$ and $\Im^{(2)}$ denote the sets of prototypes not used in searching for the nearest pairs for the ω_1 and ω_2 classes, respectively. $\Im$ denotes the set of ordered pairs of cluster centres from different classes. An algorithm for determining the nearest pairs of clusters can be summarized in the following steps:

1. Set $\Im = \emptyset$, $\Im^{(1)} = \Im^{(2)} = \{1, 2, \ldots, c\}$, $k = 1$.
2. Determine $\min\limits_{i \in \Im^{(1)},\, j \in \Im^{(2)}} \|\mathbf{v}^{(i)(1)} - \mathbf{v}^{(j)(2)}\|_1$
   $= \left\|\mathbf{v}^{(\eta_1(k))(1)} - \mathbf{v}^{(\eta_2(k))(2)}\right\|_1$.
3. Set $\Im = \Im \cup \{(\eta_1(k), \eta_2(k))\}$,
   $\Im^{(1)} = \Im^{(1)} \setminus \{\eta_1(k)\}$,
   $\Im^{(2)} = \Im^{(2)} \setminus \{\eta_2(k)\}$, $k = k + 1$.
4. If $k < c$, then go to Step 2, otherwise stop.

The symbol '$\setminus$' denotes the set-theoretic subtraction and $\eta_j(k)$ denotes the permutation function for the j-th class.

The fuzzy set-theoretic union of the nearest pairs of clusters defines c fuzzy sets. These sets form a fuzzy partition of the input space, and for the i-th set we have

$$A^{(i)} = \left(A_1^{(\eta_1(k))(1)} \cap A_2^{(\eta_1(k))(1)} \cap \ldots \cap A_t^{(\eta_1(k))(1)}\right) \cup \left(A_1^{(\eta_2(k))(2)} \cap A_2^{(\eta_2(k))(2)} \cap \ldots \cap A_t^{(\eta_2(k))(2)}\right), \quad (12)$$

where $A_p^{(\eta_j(k))(j)}$ is the fuzzy set representing the p-th component of the $\eta_j(k)$-th cluster for the j-th class. Using the algebraic product as the t-norm and the maximum operator as the s-norm (Czogała and Łęski, 2000) yields the membership function

$$A^{(i)}(\mathbf{x}) = \max\left\{\exp\left[-\frac{1}{2}\sum_{p=1}^{t}\left(\frac{x_p - v_p^{(\eta_1(k))(1)}}{s_p^{(\eta_1(k))(1)}}\right)^2\right], \exp\left[-\frac{1}{2}\sum_{p=1}^{t}\left(\frac{x_p - v_p^{(\eta_2(k))(2)}}{s_p^{(\eta_2(k))(2)}}\right)^2\right]\right\}. \quad (13)$$

Finally, the memberships needed in (9) are obtained as $u_{ik} = A^{(i)}(\mathbf{x}_k)$. For all $i = 1, 2, \ldots, c$ the algorithm described in Subsection 2.1 leads to a linear classifier with parameters $\mathbf{w}^{(i)}$. These classifiers can be represented as a set of fuzzy *if-then* rules in the Takagi-Sugeno-Kang form (Czogała and Łęski, 2000):

$$\text{IF } \mathbf{x} \text{ is } A^{(i)}, \text{ THEN } y = g_i(\mathbf{x}) = \mathbf{w}^{(i)T}\mathbf{x}', \quad i = 1, 2, \ldots, c. \quad (14)$$

The overall output for each datum $\mathbf{x}_k$ is obtained by the weighted average (Czogała and Łęski, 2000):

$$y_k = \frac{\sum\limits_{i=1}^{c} A^{(i)}(\mathbf{x}_k)\ \mathbf{w}^{(i)T}\mathbf{x}'_k}{\sum\limits_{i=1}^{c} A^{(i)}(\mathbf{x}_k)} \begin{cases} > 0, & \mathbf{x}_k \in \omega_1, \\ < 0, & \mathbf{x}_k \in \omega_2. \end{cases} \quad (15)$$

The above classifier can be also named a mixture-of-experts classifier. It is assumed that different experts (classifiers, *if-then* rules) work best in different regions of the input space. The integrating unit, described by (15), called the gating network, acts as a mediator among the experts.

3. Numerical Experiments and Discussion

In all experiments the values of $\mathbf{b}^{[1]} = 10^{-6}$ and $\rho = 0.98$ were used. The iterations were stopped as soon as the Euclidean norm in a successive pair of $\mathbf{b}$ vectors was less than 10^{-4}. The fuzzy c-means clustering (FCM) algorithm was applied with the weighted exponent equal to 2. For initialization a random partition matrix was used, and the iterations were stopped as soon as the Frobenius norm in successive pairs of partition matrices was less than 10^{-6}. All experiments were run in the MATLAB environment. Benchmark databases were obtained via the Internet — `ftp://markov.stats.ox.ac.uk/pub/PRNN` and `http://www.stats.ox.ac.uk/pub/PRNN/`.

3.1. Simple Synthetic Two-Dimensional Data

The purpose of this experiment was to compare the classical and proposed methods of classifier design. The simulations were performed for data generated by Ripley (Ripley, 1996). These data consist of patterns having two features and assigned to two classes. Each class has a bimodal distribution obtained as a mixture of two normal distributions. The class distribution was chosen to allow the best-possible error rate of about 8%. The training set consists of 250 patterns (125 patterns belong to each class), and the testing set consists of 1000 patterns (500 patterns belong to each class).

The parameter τ was in the range from 0 to 10 (step 0.1), and the number of *if-then* rules (experts) was changed from 2 to 10. After the training stage (the classifier design on the training set), the generalization ability of the classifier was determined as the error rate on the test set. For each combination of the above parameter values, the training stage was repeated 25 times for different random initializations of the FCM method. Table 1 shows the lowest error rate for each number of *if-then* rules.

Table 1. Minimal error rate obtained for the testing part of databases.

c	Synthetic two-class problem		Pima Indians diabetes	
	Error rate	τ	Error rate	τ
2	9.0%	4.6	19.57%	5.0
3	8.6%	4.6	19.27%	5.3
4	8.5%	1.6	19.57%	7.3
5	**8.2%**	**1.4**	19.57%	3.0
6	8.6%	0.5	**17.77%**	**5.6**
7	8.7%	0.5	18.97%	2.4
8	8.6%	5.0	18.67%	2.8
9	8.5%	2.7	19.57%	8.1
10	8.7%	0.7	19.57%	5.6

The best generalization (the lowest error rate on the testing set), equal to 8.2%, is obtained for 5 *if-then* rules and τ equal to 1.4. For other numbers of *if-then* rules, we also have an optimum of the generalization ability, but it is worse than that obtained for 5 *if-then* rules. It is very interesting that for increased numbers of *if-then* rules the generalization ability slightly decreases. This provides evidence that the classifier is not overtrained. The discrimination curve (the continuous line) of the classifier with the best generalization ability for 2 and 5 *if-then* rules are shown in Figs. 1 and 2, respectively. In these figures the linear classifiers (experts) are plotted as dotted lines, and the prototypes of classes ω_1 and ω_2 are marked with triangles and squares, respectively. The parameter values of the classifier, i.e., the number of *if-then* rules and parameter τ, were chosen as the values for which the machine has the best generalization ability, measured by the cross-validation on the test set. Thus, in this example $c = 5$ and $\tau = 1.4$ were chosen for the final classifier. For the number of *if-then* rules greater than 5, the learning problem is underdetermined because the classifier complexity is too large. We also see that for the number of *if-then* rules less than 5, the learning problem is overderdetermined because the classifier complexity is too small.

For one *if-then* rule (the linear case), the error rate 10.2% was obtained for $\tau = 5.1$. For comparison, the nearest-prototype classifier with optimization based on deterministic annealing (Miller *et al.*, 1996) leads to an error rate equal to 8.6% for 12 prototypes, and the neuro-fuzzy classifier (ANNBFIS) (Czogała and Łęski, 2000) leads to an error rate of 8.8% for 2 *if-the* rules. In (Tipping, 2001) it is reported that the 'state-of-the-art' support vector machine classifier has the error rate 10.6% and the relevance vector machine classifier leads to the error rate equal to 9.3%. Table 2 shows the generalization ability of the classifiers for the synthetic two-class problem.

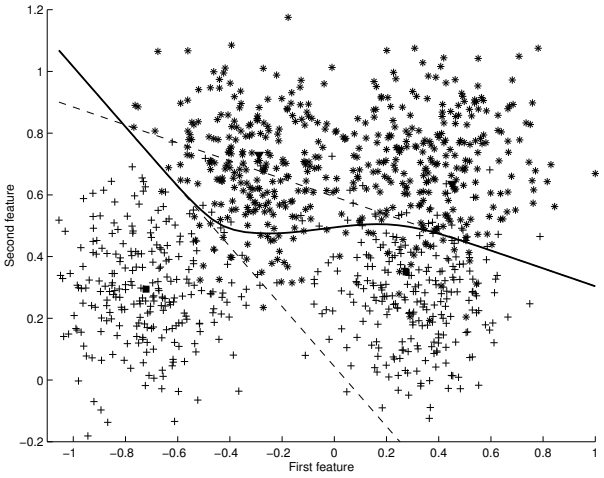

Fig. 1. Testing set for Ripley's two-class problem with the classification curve for 2 *if-then* rules.

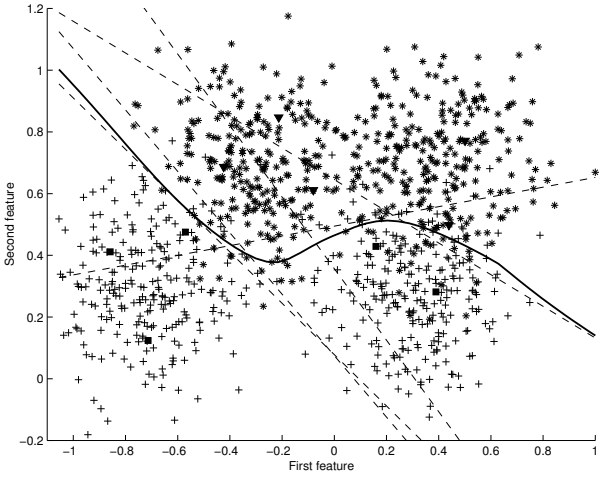

Fig. 2. Testing set for Ripley's two-class problem with the classification curve for 5 *if-then* rules.

3.2. Real High-Dimensional Data

The main goal of the experiments here was to examine the usefulness of the proposed method in constructing a classifier for real-world high-dimensional data. The data were collected by the US National Institute of Diabetes and Kidney Diseases. According to the criteria of the Expert Committee on the Diagnosis and Classification of Diabetes Mellitus, a population of women who were at least 21 years old was tested for diabetes. The women are Pima Indians (living near Phoenix, Arizona). For each woman the following personal data were collected: the number of pregnancies, plasma glucose concentrations in the fasting plasma glucose test, diastolic blood pressure (mm Hg), tricep skin fold thickness (mm), body mass index (weight in kg/(height in m)^2), diabetes pedigree func-

Table 2. Comparison of the generalization ability of classifiers for Ripley's synthetic two-class problem.

Classifier	Reference	Generalization ability
if-then rule based on the Ho-Kashyap method	this paper	91.8%
nearest prototype with deterministic annealing	(Miller *et al.*, 1996)	91.4%
ANNBFIS	(Czogała and Łęski, 2000)	91.2%
relevance vector machine	(Tipping, 2001)	90.7%
modified Ho-Kashyap classifier	this paper	89.8%
support vector machine	(Tipping, 2001)	89.4%

Table 3. Comparison of the generalization ability of classifiers for the Pima Indians diabetes dataset.

Classifier	Reference	Generalization ability
if-then rule based on the Ho-Kashyap method	this paper	82.23%
relevance vector machine	(Tipping, 2001)	80.4%
linear logistic discrimination	(Ripley, 1996)	80.2%
support vector machine	(Tipping, 2001)	79.9%
linear discrimination	(Ripley, 1996)	79.8%
ANNBFIS	(Czogała and Łęski, 2000)	78.0%
backpropagation neural network	(Ripley, 1996)	77.9%
learning vector quantization	(Ripley, 1996)	77.9%
Lagrangian support vector machine	(Mangasarian and Musicant, 2000)	78.12%
combined classifiers using fuzzy sets	(Kuncheva, 2002)	77.5%
Bayes point machine	(Herbrich *et al.*, 2001)	68.0%

tion (the function of the number and location in the pedigree tree of common ancestors up to the second degree relatives suffering from diabetes mellitus), and the age in years. Out of 768 collected records 376 were incomplete. Ripley divided randomly the complete records into a training set of the size 200 and a test set of the size 332 (Ripley, 1996). The performance of several classical pattern recognition methods was then tested. The obtained error rate (in percent) was as follows: linear discrimination – 20.2%, projection pursuit regression – 22.6%, linear logistic discrimination – 19.8%, backpropagation neural network – 21.1%, learning vector quantization – 21.1%. The Lagrangian support vector machine (Mangasarian and Musicant, 2000) leads to an error rate equal to 21.88%.

For the proposed algorithm, the parameter τ was in the range from 0 to 10 with step 0.1, and the number of *if-then* rules (experts) was changed from 2 to 10. Table 1 shows the lowest error rate for each number of *if-then* rules. From this table we see that the best generalization, equal to 17.77%, is obtained for 6 *if-then* rules and $\tau = 5.6$. It is also seen that the worst result for the proposed classifier is better then the best result obtained for the classical method, i.e. linear logistic discrimination. As in the previous subsection, for increased numbers of *if-then* rules, the generalization ability slightly decreases.

For comparison, the support vector machine classifier (Tipping, 2001) leads to the error rate equal to 20.1% for 109 support vectors and the relevance vector machine classifier leads to an error rate of 19.6% for 4 relevance vectors. The Bayesian point machine classifier has the error rate equal to 32.0% (Herbrich *et al.*, 2001). The technique based on a combination of multiple classifiers using fuzzy sets leads to the error rate equal to 22.5% using the average-and-majority vote fusion method (Kuncheva, 2002). The neuro-fuzzy classifier (ANNBFIS) (Czogała and Łęski, 2000) leads to the error rate 21.0% for 2 *if-the* rules. Table 3 shows the generalization ability of the classifiers for the Pima Indians diabetes dataset.

4. Conclusions

In this work, a new nonlinear *if-then* rules-based classifier design method has been introduced. This method constitutes a modification of the classical Ho-Kashyap methodology, which uses an absolute loss function, rather than a quadratic one. This results in a better approximation of the misclassification error and robustness against outliers. Additionally, the proposed method minimizes the Vapnik-Chervonenkis dimension, which results in easy control of

the generalization ability of the classifier. An extension to nonlinear classifier design using a mixture of experts was also shown. This method establishes a new subcategory in the partition of fuzzy classifier design methods, i.e. fuzzy *if-then* rules-based with rules extracted automatically by using a connection of a fuzzy clustering method and a weighted support vector machine. This method can also be viewed as the competition-combination of multiple classifiers using the fuzzy set methodology.

Two numerical examples were given to illustrate the validity of the presented method. These examples show the usefulness of the new method in the classification of both synthetic and real-world high-dimensional data. For these databases the results obtained by the proposed method are better compared with the methods reported in the literature. The new classifier consistently outperforms the state-of-the-art classifier "support vector machine" on both synthetic and real-world benchmark datasets.

References

Abe S. and Lan M.-S. (1995): *A method for fuzzy rules extraction directly from numerical data and its application to pattern classification*. — IEEE Trans. Fuzzy Syst., Vol. 3, No. 1, pp. 18–28.

Bellman R., Kalaba K. and Zadeh L.A. (1966): *Abstraction and pattern classification*. — J. Math. Anal. Appl., Vol. 13, No. 1, pp. 1–7.

Bezdek J.C. (1982): *Pattern Recognition with Fuzzy Objective Function Algorithms*. — New York: Plenum Press.

Bezdek J.C. and Pal S.K. (Eds.) (1992): *Fuzzy Models for Pattern Recognition*. — New York: IEEE Press.

Bezdek J.C., Reichherzer T.R., Lim G.S. and Attikiouzel Y. (1998): *Multiple-prototype classifier design*. — IEEE Trans. Syst. Man Cybern., Part C, Vol. 28, No. 1, pp. 67–78.

Czogała E. and Łęski J.M. (2000): *Fuzzy and Neuro-Fuzzy Intelligent Systems*. — Heidelberg: Physica-Verlag.

Duda R.O. and Hart P.E. (1973): *Pattern Classification and Scene Analysis*. — New York: Wiley.

Herbrich R., Graepel T. and Campbell C. (2001): *Bayes point machines*. — J. Mach. Res., Vol. 1, No. 2, pp. 245–279.

Ho Y.-C. and Kashyap R.L. (1965): *An algorithm for linear inequalities and its applications*. — IEEE Trans. Elec. Comp., Vol. 14, No. 5, pp. 683–688.

Ho Y.-C. and Kashyap R.L. (1966): *A class of iterative procedures for linear inequalities*. — J. SIAM Contr., Vol. 4, No. 2, pp. 112–115.

Ishibuchi H., Nakashima T. and Murata T. (1999): *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems*. — IEEE Trans. Syst. Man Cybern., Part B, Vol. 29, No. 5, pp. 601–618.

Huber P.J. (1981): *Robust Statistics*. — New York: Wiley.

Keller J.M., Gray M.R. and Givens J.A. (1985): *A fuzzy k-nearest neighbors algorithm*. — IEEE Trans. Syst. Man Cybern., Vol. 15, No. 3, pp. 580–585.

Krishnapuram R. and Keller J.M. (1993): *A possibilistic approach to clustering*. — IEEE Trans. Fuzzy Syst., Vol. 1, No. 2, pp. 98–110.

Kim E., Park M., Ji S. and Park M. (1997): *A new approach to fuzzy modeling*. — IEEE Trans. Fuzzy Syst., Vol. 5, No. 3, pp. 328–337.

Kuncheva L.I. and Bezdek J.C. (1999): *Presupervised and postsupervised prototype classifier design*. — IEEE Trans. Neural Netw., Vol. 10, No. 5, pp. 1142–1152.

Kuncheva L.I. (2000a): *How good are fuzzy if-then classifiers?* — IEEE Trans. Syst. Man Cybern., Part B, Vol. 30, No. 4, pp. 501–509.

Kuncheva L.I. (2000b): *Fuzzy Classifier Design*. — Heidelberg: Physica-Verlag.

Kuncheva L.I. (2001): *Using measures of similarity and inclusion for multiple classifier fusion by decision templates*. — Fuzzy Sets Syst., Vol. 122, No. 3, pp. 401–407.

Kuncheva L.I. (2002): *Switching between selection and fusion in combining classifiers: An experiment*. — IEEE Trans. Syst. Man Cybern.. Part B, Vol. 32, No. 2, pp. 146–156.

Łęski J. and Henzel N. (2001): *A neuro-fuzzy system based on logical interpretation of if-then rules*, In: Fuzzy Learning and Applications (Russo M. and Jain L.C., Eds.). — New York: CRC Press, pp. 359–388.

Łęski J. (2002): *Robust weighted averaging*. — IEEE Trans. Biomed. Eng., Vol. 49, No. 8, pp. 796–804.

Malek J.E., Alimi A.M. and Tourki R. (2002): *Problems in pattern classification in high dimensional spaces: Behavior of a class of combined neuro-fuzzy classifiers*. — Fuzzy Sets Syst., Vol. 128, No. 1, pp. 15–33.

Mangasarian O.L. and Musicant D.R. (2000): *Lagrangian support vector machines*. — Technical Report 00-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, available at `ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.ps`

Marín-Blázquez J. and Shen Q. (2002): *From approximative to descriptive fuzzy classifiers*. — IEEE Trans. Fuzzy Syst., Vol. 10, No. 4, pp. 484–497.

Miller D., Rao A.V., Rose K. and Gersho A. (1996): *A global optimization technique for statistical classifier design*. — IEEE Trans. Signal Process., Vol. 44, No. 12, pp. 3108–3121.

Nath A.K. and Lee T.T. (1982): *On the design of a classifier with linguistic variables as inputs*. — Fuzzy Sets Syst., Vol. 11, No. 2, pp. 265–286.

Ripley B.D. (1996): *Pattern Recognition and Neural Networks*. — Cambridge: Cambridge University Press.

Runkler T.A. and Bezdek J.C. (1999): *Alternating cluster estimation: A new tool for clustering and function approximation*. — IEEE Trans. Fuzzy Syst., Vol. 7, No. 4, pp. 377–393.

Rutkowska D. (2002): *Neuro-Fuzzy Architectures and Hybrid Learning*. — Heidelberg: Physica-Verlag.

Setnes M. and Babuška R. (1999): *Fuzzy relational classifier trained by fuzzy clustering*. — IEEE Trans. Syst. Man Cybern., Part B, Vol. 29, No. 5, pp. 619–625.

Tipping M.E. (2001): *Sparse Bayesian learning and the relevance vector machine*. — J. Mach. Res., Vol. 1, No. 2, pp. 211–244.

Tou J.T. and Gonzalez R.C. (1974): *Pattern Recognition Principles*. — London: Addison-Wesley.

Vapnik V. (1998): *Statistical Learning Theory*. — New York: Wiley.

Vapnik V. (1999): *An overview of statistical learning theory*. — IEEE Trans. Neural Netw., Vol. 10, No. 5, pp. 988–999.

Webb A. (1999): *Statistical Pattern Recognition*. — London: Arnold.

Appendix

The first equation of (7) can be rewritten in the form $\mathbf{X}^T\mathbf{De} = -\tau\widetilde{\mathbf{I}}\ \mathbf{w}$. Thus, for $\tau > 0$, all elements of the error vector cannot be zero. This is true in either linearly separable or nonseparable cases. If we define $\mathbf{X}^\dagger \triangleq (\mathbf{X}^T\mathbf{DX} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{D}$ and $\mathbf{e}_+^{[k]} \triangleq \mathbf{e}^{[k]} + |\mathbf{e}^{[k]}|$, then using (7) and (8) yields: $\mathbf{e}^{[k+1]} = \mathbf{e}^{[k]} + \rho(\mathbf{XX}^\dagger - \mathbf{I})\mathbf{e}_+^{[k]}$ and $\mathbf{w}_n^{[k+1]} = \widetilde{\mathbf{I}}\mathbf{X}^\dagger(\mathbf{b}^{[k]} + \rho\mathbf{e}_+^{[k]}) = \mathbf{w}_n^{[k]} + \rho\widetilde{\mathbf{I}}\mathbf{X}^\dagger\mathbf{e}_+^{[k]}$. Substituting the above results into (6) gives $I^{[k+1]} = I^{[k]} + 2\rho\mathbf{e}^{[k]T}\mathbf{D}(\mathbf{XX}^\dagger - \mathbf{I})\mathbf{e}_+^{[k]} + \rho^2\mathbf{e}_+^{[k]T}(\mathbf{XX}^\dagger - \mathbf{I})^T\mathbf{D}(\mathbf{XX}^\dagger - \mathbf{I})\mathbf{e}_+^{[k]} + 2\tau\rho\mathbf{w}_n^{[k]T}\widetilde{\mathbf{I}}\mathbf{X}^\dagger\mathbf{e}_+^{[k]} + \tau\rho^2\mathbf{e}_+^{[k]T}\mathbf{X}^{\dagger T}\widetilde{\mathbf{I}}\mathbf{X}^\dagger\mathbf{e}_+^{[k]}$. From the first equation of (7) we have $\widetilde{\mathbf{I}}\mathbf{X}^T\mathbf{De}^{[k]} = -\tau\mathbf{w}_n^{[k]}$. From this and the equality $2\rho\mathbf{e}^{[k]T}\mathbf{D}(\mathbf{XX}^\dagger - \mathbf{I})\mathbf{e}_+^{[k]} = \rho\mathbf{e}_+^{[k]T}\mathbf{D}(\mathbf{XX}^\dagger - \mathbf{I})\mathbf{e}_+^{[k]}$, after some simple algebra, we obtain $I^{[k+1]} - I^{[k]} = \rho(\rho - 1)\mathbf{e}^{[k]T}\mathbf{De}_+^{[k]} + \rho^2\mathbf{e}_+^{[k]T}\mathbf{X}^{\dagger T}(\mathbf{X}^T\mathbf{DX} + \tau\widetilde{\mathbf{I}})\mathbf{X}^\dagger\mathbf{e}_+^{[k]} - 2\rho^2\mathbf{e}_+^{[k]T}\mathbf{DXX}^\dagger\mathbf{e}_+^{[k]}$. Since $\mathbf{X}^{\dagger T}(\mathbf{X}^T\mathbf{DX} + \tau\widetilde{\mathbf{I}})\mathbf{X}^\dagger = \mathbf{DXX}^\dagger$, the second and third terms simplify to $-\rho^2\mathbf{e}_+^{[k]T}\mathbf{DXX}^\dagger\mathbf{e}_+^{[k]}$.

Thus $I^{[k+1]} - I^{[k]} = \rho(\rho - 1)\mathbf{e}^{[k]T}\mathbf{De}_+^{[k]} - \rho^2\mathbf{e}_+^{[k]T}\mathbf{DXX}^\dagger\mathbf{e}_+^{[k]}$. The matrix $\mathbf{DXX}^\dagger$ is symmetric and positive semidefinite. It follows that the second term is negative or zero. For $0 < \rho < 1$ the first term is negative or zero. Thus the sequence $I^{[1]}, I^{[2]}, \ldots$ is monotonically decreasing. For both linearly separable and nonseparable cases, convergence requires that $\mathbf{e}_+^{[k]}$ tend to zero (no modification in (7)), while $\mathbf{e}^{[k]}$ is bounded away from zero, since $\mathbf{X}^T\mathbf{De} = -\tau\widetilde{\mathbf{I}}\mathbf{w}$.

Received: 28 May 2002
Revised: 29 August 2002