pattern recognition, feature selection, k-NN rules, pair-wise classifier, artificial features, linear classifier, reference set size reduction

Adam JÓŹWIK[1]

NONPARAMETRIC METHODS OF SUPERVISED CLASSIFICATION

Selected nonparametric methods of statistical pattern recognition are described. A part of them form modifications of the well known k-NN rule. To this group of the presented methods belong: a fuzzy k-NN rule, a pair-wise k-NN rule and a corrected k-NN rule. They can improve classification quality as compared with the standard k-NN rule. For the cases when these modifications would offer to large error rates an approach based on class areas determination is proposed. The idea of class areas can be also used for construction of the multistage classifier. A separate feature selection can be performed in each stage.

The modifications of the k-NN rule and the methods based on determination class areas can be too slow in some applications, therefore algorithms for reference set reduction and condensation, for simple NN rule, are proposed. To construct fast classifiers it is worth to consider also a pair-wise linear classifiers. The presented idea can be used as in the case when the class pairs are linearly separable as well as in the contrary case.

1. INTRODUCTION

Pattern recognition deals with methods of object classification, where objects are understood in a very general sense. It is assumed that each object is described by a set of features that forms a vector or a point in the feature space, usually Euclidean one. The classification task consists in assigning a class to an object. However, the decision rule is not known. The classes are not defined by their descriptions, but in statistical manner, i.e. by a set of objects with known class membership called a training set. The decision rule must be derived from the information contained in this set. The class membership of the object can be crisp, when it can belong to one class only or fuzzy, when it's membership is distributed between all considered classes, represented in the training set.

The probability of misclassification is usually used as a classification quality criterion. It can be estimated by an error rate calculated with a use of a separate testing set or on a basis of the training set using, for instance, the leave one out method. This method consists in classification of each object $\underline{u}$ from the training set U by the decision rule derived from the set $U - \{\underline{u}\}$ [11].

Any object, described in the feature space by a vector $\underline{x}$, belongs to each of the considered classes j, $j = 1, 2, \ldots, nc$, with a certain unknown probability $p(j/\underline{x})$. The ideal classifier ought to assign to an object $\underline{x}$ a class i that corresponds to the highest value of $p(j/\underline{x})$, so $p(i/\underline{x}) = max_j p(j/\underline{x})$. The probabilities $p(j/\underline{x})$ are unknown but they can be estimated by the well known k-NN rule [2], i.e. $p(j/\underline{x}) = k_j/k$, where k_j is a number of objects from the class j among k nearest neighbors of $\underline{x}$. The standard k-NN rule assigns to the object $\underline{x}$ the class i such that $k_i/k = max_j(k_j/k)$. As it was presented in the work [1], the k-NN rule outperforms other known classifiers like Fuzzy ArtMap, RBF or neural classifiers trained by back propagation methods.

[1]Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science, Warsaw, Poland and Łódź University, Faculty of Physics and Applied Informatics, Łódź, Poland.

2. FUZZY k-NN RULE

The probabilities $p(j/\underline{x})$, $j = 1, 2, \ldots, nc$, can be written in the form of a fuzzy membership vector $\underline{v}_f = [p(1/\underline{x}), p(2/\underline{x}), \ldots, p(nc/\underline{x})]$. Also the crisp (nonfuzzy) membership of the object $\underline{x}$ to the class j can be written as a vector $\underline{v}_c = [0_1, 0_2, \ldots, 1_j, \ldots, 0_{nc}]$. Such notation is more general because comprises the both membership types. It is very convenient too. Applying the k-NN rule first the fuzzy membership vector $\underline{v}_f = [k_1/k, k_2/k, \ldots, k_{nc}/k]$ can be computed as a mean of the membership vectors of nearest neighbors. For instance, let $\underline{v}_{1,NN} = [1, 0], \underline{v}_{2,NN} = [0, 1], \underline{v}_{3,NN} = [0, 1]$ be the class membership vectors, in a two class task, of the first, second and the third nearest neighbor respectively. Then their mean vector $\underline{v}_f = [1/3, 2/3]$ forms the fuzzy decision that is converted into the crisp one $\underline{v}_c = [0, 1]$, where the value of 1 corresponds to the largest ratio k_j/k, $j = 1, 2$, and the remaining components equal to zero.

Each point $\underline{x}_i$ in the feature space, usually Euclidean one, occupied by an object from the training set U, belongs to the class j with unknown probability $p(j/\underline{x}_i)$. It concerns also to the nearest neighbors of the classified object. Thus, during voting, each nearest neighbor ought to distribute its voice between all classes in accordance with the probabilities $p(j/\underline{x}_i)$. Any object in the training set can be used as a nearest neighbor, so the original crisp membership vectors of the form $\underline{v}_i = [0_1, 0_2, \ldots, 1_j, \ldots, 0_{nc}]$, $i = 1, 2, \ldots, m$, where m is a training set numerical force, ought to be replaced by the fuzzy ones. It can be done by the use of the following formula:

$$\underline{v}_i^{r+1} = [p_{1,i}^{r+1}, p_{2,i}^{r+1}, \ldots, p_{nc,i}^{r+1}] = (\sum_{h=1}^{k} \underline{v}_{i,hNN}^{r} + \underline{v}_i^r)/(k+1). \tag{1}$$

The vector $\underline{v}_{i,hNN}^r$ is a membership vector of h-th nearest neighbor of the i-th object and the upper index r is a number of sequential $p(j/\underline{x}_i)$ approximation. It assumes at the beginning the value of 0. The vector $\underline{v}_i^r$ denotes a membership of the i-th object in the r-th approximation. It can be noticed that if $r = 0$ then

$$\underline{v}_i^{r+1} = \underline{v}_i^1 = [k_1/(k+1), k_2/(k+1), \ldots, (k_l+1)/(k+1), \ldots, k_{nc}/(k+1)], \tag{2}$$

where l is a class of the object i. In the approximation of $p(j/\underline{x}_i)$ take part, by voting, as k nearest neighbors of $\underline{x}_i$ as well as the object $\underline{x}_i$ itself. It is obvious that the object will give its voice in favor of the class l. Components $\underline{p}_{j,i}^1$ of the vector $\underline{v}_i^1$ can be treated as a first approximation of the unknown probabilities $p(j/\underline{x}_i)$.

The estimation of the probabilities $p(j/\underline{x}_i)$ can be improved by applying the learning scheme proposed by the author in [3]. It consists in generating an infinite sequence:

$$(W_0, k_0, e_0), (W_1, k_1, e_1), (W_2, k_2, e_2), \ldots, (W_r, k_r, e_r), (W_{r+1}, k_{r+1}, e_{r+1}) \ldots. \tag{3}$$

If the subject of interest is crisp classification then W_0 is a binary membership matrix consisted of crisp membership vectors $\underline{v}_i^0 = [0_1, 0_2, \ldots, 1_j, \ldots, 0_{nc}]$, $i = 1, 2, \ldots, m$ and $1 \leq j \leq nc$. It has m rows and nc columns. Using the leave one out method and reviewing all possible numbers of nearest neighbors one can find the value k_0 of nearest neighbors offering the lowest error rate e_0. Then, applying the formula (1) with $k = k_0$, the matrix W_1 can be found. The rows of this matrix could be also found in accordance with the formula (2). Next, the values k_r and e_r for $r \geq 1$, are determined by the leave one out method. A number k_r of nearest neighbors is chosen in such a way that k_r-NN rule operating with W_r offers the lowest error e_r. The matrix W_{r+1} is determined on the basis of W_r and k_r. Generation of the sequence (3) is stopped when $e_{r+1} > e_r$. Finally, the k-NN rule is used with the fuzzy membership matrix W_r and $k = k_r$. That is why it is called a fuzzy k-NN rule. It produces, similarly as the standard version of k-NN rule, the fuzzy decision, which is next converted into a crisp one. Till now, it was assumed that the subject of interest is a crisp classification.

However, there are some applications [13], where the components of fuzzy decision vector cannot be interpreted as estimations of the probabilities $p(j/\underline{x})$. The classification task consisted in recognition of

contribution of different metals in alloys. The error e of a single classification is then calculated by the formula:

$$e = \sum_{j=1}^{nc} |v_j - w_j|/2, \tag{4}$$

where $\underline{v} = [v_1, v_2, \ldots, v_{nc}]$ and $\underline{w} = [w_1, w_2, \ldots, w_{nc}]$ are the true and assigned fuzzy membership vectors respectively. The fuzzy error rate er can be calculated as a mean value of a single fuzzy errors of the type (4). In the case of applying the leave one out method it is calculated according with the following relation:

$$er = (1/m) \circ \sum_{i=1}^{m} (\sum_{j=1}^{nc} |v_{i,j} - w_{i,j}|/2), \tag{5}$$

where m is the numerical force of the training set.

3. PAIR-WISE k-NN RULE

A multi-decision classifier can be constructed with some two-decision classifiers. One of the solutions may be a construction of a parallel net of two-decision classifiers, a separate classifier for each pair of classes, and then forming the final decision by voting of these two-decision classifiers as it is illustrated in Fig. 1.

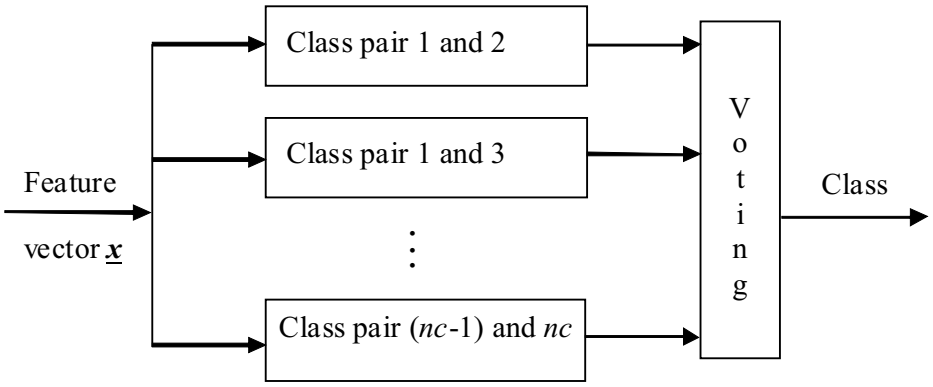

Fig. 1. The structure of the pair-wise classifier.

This parallel network of two-decision classifiers should offer better performance than the standard k-NN. It derives from the geometrical interpretation of both the discussed types of classifiers. In the case of the standard classifier, the boundary separating any pair of classes i and j depends also on the samples from the remaining classes. They have an influence on the value of k and on the selected features if feature selection is being performed. These samples may act as a noise. The parallel net may reduce this noise effect. By using the error rate estimated by the leaving-one-out method as a criterion, we can find an optimum number of k for the k-NN rules and perform the feature selection separately for each of the component classifiers. The pair-wise k-NN rule was proposed in [10] and then experimentally verified in [4] using artificially generated data. In the experiments the classes occupied areas as they are shown in the Fig. 2.

One thousand points, according with uniform distribution and precision 0.01, were generated in each square. The standard and the parallel k-NN classifiers were applied. Ten such experiments were performed. Error rates were calculated for all possible values of k and the best results were finally chosen. The mean error rate for the standard k-NN rule was 0.400% while the pair-wise classifier offered the error rate that was equal to 0.014%. After feature selection, performed separately for each of the component classifiers, the error rate for the pair-wise classifier equaled 0.006%.

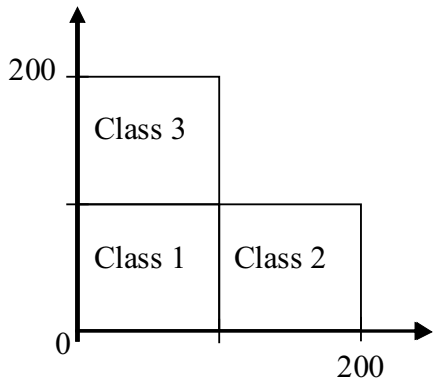

Fig. 2. Areas of classes used in experiments.

4. CORRECTED k-NN RULE

The class numerical force proportions in the training set ought to be approximately the same as in the reality. However, in biomedical investigations very often appear data with missing feature values. The training set has usually a form of a matrix contained m rows and n columns, where m is a training set numerical force and n denotes a number of features. Pattern recognition methods require complete data sets, so it can be necessary to reject some rows (objects) to receive data set without missing feature values. In this manner the original class proportion in the training set can be destroyed. Such object rejections influence on data standardization, error rate and confusion matrices calculation and also on the k-NN rule, if this rule is being used. It especially complicates forward and backward feature selection procedures since different feature combinations are reviewed and the set of rejected objects depends on currently reviewed feature combinations. However, if the training set was gathered according with real proportion of the class frequencies then class a priori probabilities $p(j)$, $j = 1, 2, \ldots, nc$, can be determined before rejection of any object. These probabilities enable correction of the process of k-NN classifier construction [5].

Data standardization

The global feature mean values and standard deviations can be calculated as functions of these statistics determined for each class separately:

$$mv_j = \sum_{k=1}^{nc} p_k \circ mv_{k,j} \quad \text{and} \quad sd_j = (\sum_{k=1}^{nc} [p_k \circ (sd_{k,j}^2 + mv_{k,j}^2)] - mv_j^2)^{1/2}, \tag{6}$$

where mv_j and sd_j are global mean value and standard deviation respectively of the feature j, $mv_{k,j}$ and $sd_{k,j}$ are also mean values and standard deviations of the feature j but calculated for the class k. The values of mv_j and sd_j, found by formulas (6), are used for data standardization.

Misclassification rate

Let b_k be an error rate calculated for the class k. Then the global error rate can be computed by the formula:

$$b = \sum_{k=1}^{nc} p_k \circ b_k. \tag{7}$$

Confusion matrices

The matrix $R = \{r_{k,j}\}_{k,j=1}^{nc}$, where $r_{k,j}$ is a number of objects from the class k assigned to the class j, does not require any correction.

The matrix $P = \{p_{k,j}\}_{k,j=1}^{nc}$, where $p_{k,j}$ is a probability that object from the class k will be assigned to the class j, can be calculated as

$$p_{k,j} = r_{k,j}/m_j, \tag{8}$$

where m_j denotes number of object in the class j in the training set (after rejection of rows with missing feature values).

The matrix $Q = \{q_{k,j}\}_{k,j=1}^{nc}$, where $q_{k,j}$ is a probability that object assigned to the class k comes in fact from the class j, can be calculated as

$$q_{k,j} = (r_{j,k}/m_j) \circ p_j / \sum_{j=1}^{nc} [(r_{j,k}/m_j) \circ p_j]. \tag{9}$$

Decision rule

The classified object ought to be assign to the class i, which satisfies following formula:

$$q_i = max_j(q_j), \tag{10}$$

where $q_j = (k_j/m_j) \circ p_j$.

The relations (6)-(10) define completely the corrected k-NN rule. It can be noticed that the training set with missing some feature values can be more effectively explored if the pair-wise structure will be applied. Then each of the component two-decision classifiers can operate according to the corrected k-NN rule.

5. CLASSIFICATION BASED ON CLASS AREAS DETERMINATION

The classifiers based on k-NN rule may offer too large values of an error rate to accept it. A reasonable solution in such a case can be determination of class areas in the feature space. If the classified object falls to area of one class only then classification can be performed with satisfactory confidence. Let $U_1, U_2, \ldots, U_{nc}$ be the subsets of the training set U representing different classes. The class areas A_i can be defined by the following formulas:

$$\begin{aligned} e_i = max \quad & d(U_i - \{u_j\}, \{u_j\}), \\ & u_j \in U_i \end{aligned} \tag{11}$$

$$A_i = \{x : d(U_i, \{x\}) \leq e_i\}, \tag{12}$$

where $d(\circ, \circ)$ is a distance function. The shapes of the areas A_i, for two dimensional case, two classes and Euclidean distance measure are presented in the Fig. 3.

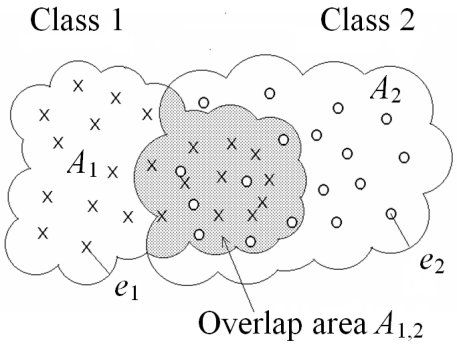

Fig. 3. Class areas A_1, A_2 and an overlap areas $A_{1,2}$.

The object which fall only in one class area A_i is being assigned to the class i and this decision can be treated as a confident one. If the classified object falls in a class overlap area then the decision can be refused or a majority class in the overlap area is assigned or, for instance, the k-NN rule or its fuzzy version can be applied. In this case the decision is less confident. The classifier refuse classification if objects fall outside of all class area.

The idea of the class overlap areas can be used for construction multistage classification. Objects which belong to the class overlap area can form a training set for the next stage. To illustrate this approach the Iris Data (http://archive.ics.uci.edu/ml/datasets.html) has been used and the results of computations are given in Table 1. They were obtained by the use of the leave one out method [9]. In the first stage the training set contained 150 objects, 46 were in class overlap area, 3 objects were outside each area, so the decisions in these cases were refused.

The 46 objects from class overlap area formed a training set of the next, i.e. second stage and for this stage similar computations were performed. Continuing such approach one can finally reach the stage V. Finally, the 141 object were correctly classified, 3 were misclassified and for 6 objects this multistage classifier refused decision making.

Table 1. Illustration of multistage classification on the example of Iris Data.

Situations - rows/stage number - columns	I	II	III	IV	V	Together
Training set numerical force for indicated stage	150	46	22	12	6	-
Number of objects in class overlap area	46	22	12	6	0	-
Number of object with refused decision	3	1	0	2	0	6
Number of misclassified objects	0	0	1	1	1	3
Number of correctly classified objects	101	23	9	3	5	141

The Iris Data is very small and it would be more reasonable to stop this procedure, for instance at stage II, and to apply k-NN rule for set of 22 objects treating it as a training one.

The ideas of class areas and the pair-wise classifier structure can be combined, what was presented, using real remote sensing data, in the work [8].

6. REFERENCE SET REDUCTION ALGORITHMS

The error rate may be not the only criterion that ought to be taken into account. Very often the classifier must be applied to very large training sets and any type of k-NN rule cannot be accepted because classification would be not sufficiently fast. Slight acceleration can be obtained by an approximation of the k-NN, $k > 1$, by a simple 1-NN rule, but still distances between the classified object and all objects in the training set must be computed. The whole training set is commonly used as a reference set, i.e. as set that must be stored in the memory during classification. Further acceleration is possible by reference set size reduction.

6.1. TOMEK'S ALGORITHM

There are numerous algorithms for the reference set reduction in the literature [6]. One very simple algorithm, concerned two classes only, was proposed by Tomek in [15] and it will be described below for an example.

At the beginning the reduced reference set is empty. For each pair of objects $\underline{x}$ and $\underline{y}$, coming from different classes, a hyperball with the centre in the point $(\underline{x}+\underline{y})/2$ and the radius $r = d(\underline{x},\underline{y})/2$, where $d(\circ,\circ)$ is Euclidean distance measure, is constructed. If the interior of this hyperball does not contain any object from the training set then both objects are qualified to the reduced reference set. The Fig. 4 illustrates how this algorithm operates. All objects, except of object $\underline{x}_1$ and $\underline{x}_4$ are qualified to the reduced reference set by the Tomek's method. The nearest neighbor classifier operating with the reduced set can be defined by the line (1) in the Fig. 4.

6.2. REFERENCE SET REDUCTION AS A SELECTION OF ARTIFICIAL FEATURES

For each Tomek's pair $\underline{x}_i$ and $\underline{x}_j$ of objects one can construct a hyperplane passing by the point $(\underline{x}_i + \underline{x}_j)/2$, orthogonal to the vector $(\underline{x}_i - \underline{x}_j)$ and oriented in such a way that objects from the class

1 lie on its positive side and objects from the class 2 on its negative side. These hyperplanes can be used to create artificial features, i.e. each hyperplane generates a feature, that assumes the value of $+1$ if an object lies on its positive side or on this hyperplane and assumes the value of -1 when it lies on its negative side.

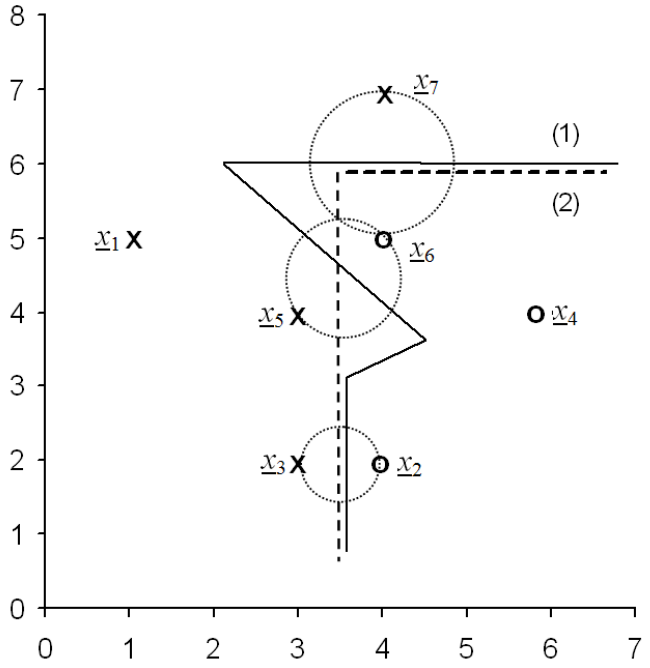

Fig. 4. Reference set reduction using Tomek's method and artificial features.

The whole original training set can be converted in the new training set of objects described by artificial features, as it was shown in the Table 2.

Table 2. The values of original and artificial features.

Object	Class	Feature c_1	Feature c_2	Feature $c_{3,2}$	Feature $c_{5,6}$	Feature $c_{7,6}$
1	1	1,0	5,0	1	1	-1
2	2	4,0	2,0	-1	1	-1
3	1	3,0	2,0	1	1	-1
4	2	6,0	4,0	-1	-1	-1
5	1	3,0	4,0	1	1	-1
6	2	4,0	5,0	-1	-1	-1
7	1	4,0	7,0	-1	-1	1

Thus, two original features c_1 and c_2 were replaced by three artificial features $c_{3,2}$, $c_{5,6}$ and $c_{7,6}$. Using the leave one out method and city distance measure one can check that all three artificial features offer the error rate equal $2/7$ since two objects were misclassified ($\underline{x}_2$ and $\underline{x}_7$). After applying selection of artificial features, using error rate calculating by the leave one out method and city distance measure, it is easy to verify that the error rate for the two artificial features $c_{3,2}$ and $c_{7,6}$ is the lowest and equals $1/7$ (only the object $\underline{x}_7$ is misclassified).

The classified object $\underline{x}$ must be converted into the artificial feature space and it will be assigned to the class 1 if $\underline{x} = [1, -1]$ or $\underline{x} = [-1, 1]$ and to the class 2 if $\underline{x} = [-1, -1]$. In the classification phase only two hyperplanes or four objects ($\underline{x}_2, \underline{x}_3, \underline{x}_6$ and $\underline{x}_7$) of the new training set must be stored in the computer memory. The objects $\underline{x}_2, \underline{x}_3, \underline{x}_6$ and $\underline{x}_7$ can form the reduced reference set in the new feature space. One hyperplane is determined by the pair $\underline{x}_3$ and $\underline{x}_2$ while the second hyperplane is defined by the pair $\underline{x}_7$ and $\underline{x}_6$. Finally, the classifier operating in the artificial feature space can be defined in the original space by the dashed line (2) shown in the Fig. 4. It can be noticed that NN rule operates in the artificial space faster than in the original one. So, it is expected that the costs of conversion the classified objects into the new feature space will be worth to be paid.

6.3. HYPERBALL DECISION RULE

The reference set reduction algorithm, presented in this section, consists in covering each class in the training set by a certain number of hyperballs. Each hyperball contains objects from one class only and hyperballs covering different classes do not intersect. If the classified object falls in any of these hyperballs then it is assigned to the class associated with this hyperball otherwise it is qualified to the class of a nearest hyperball. A distance between an object and a hyperball is understood as a distance to its center decreased by its radius.

For each object $\underline{x}$ in the training set U, the distance d to the nearest object $\underline{y}$ from the opposite class and the distance q to the farthest object from the same class as $\underline{x}$ but closer than the object $\underline{y}$ are determined. The hyperball $K(\underline{x}, r)$ with the center in $\underline{x}$ and the radius $r = (d + q)/2$ contains some objects from the training set, which create a certain set Z. The Fig. 5 illustrates how the hyperballs $K(\underline{x}, r)$ are determined.

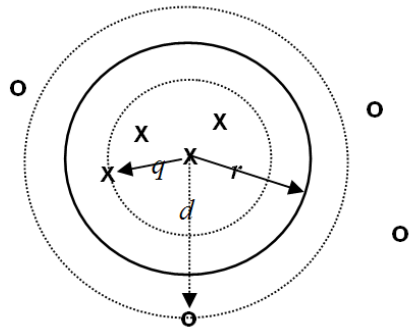

Fig. 5. The way of determination hyperball radius.

These hyperballs are then ordered in a sequence in the following way. As the first, a hyperball $K(\underline{x}_1, r_1)$ that contains the greatest set Z_1 of objects from the training set is selected. Finding the next elements of this sequence can be defined in the recurrent manner. Let $K(\underline{x}_j, r_j)$, $j = 1, 2, \ldots, i$, be already created part of the above mentioned sequence and let the set $S_i = Z_1 \cup Z_2 \cup \ldots \cup Z_i$. As the $K(\underline{x}_{i+1}, r_{i+1})$ a hyperball with the set Z_{i+1} that contains the largest set of objects from the set $U - S_i$ is selected.

Starting with a certain i_0 it will be impossible to find $K(\underline{x}, r)$ with the set Z contained any object from outside the set S_k, where $k = i_0 - 1$. It is expected that the number k of hyperballs $K(\underline{x}_j, r_j)$, $j = 1, 2, \ldots, k$, will be significantly smaller than the numerical force m of the training set.

7. EFERENCE SET CONDENSATION ALGORITHM

The reference set can contain artificial objects. For instance, each class can be represented by its gravity center, as it is commonly assumed in case of the minimum distance classifier. This classifier is very fast but may offer not satisfactory value of an error rate. Another solution can consist in division of the primary reference set U, i.e. the training set, by cutting hyperplanes into some subsets [7]. The gravity centers of these subsets, with a class memberships as a majorities of their objects, will then form the condensed reference set S. The set S is not a subset of the set R that is why it is not called the reduced reference set. The original reference set can be divided into desired number of subsets. Another author suggested to perform division in such a way that each subset will contain objects from one class only, i.e. the number of subsets will be determined automatically.

Below, the version with the desired size of the condensed reference set R will be presented. The algorithm starts with division of the set R into two subsets R_1 and R_2. Let at certain stage the set R be divided into nc subsets R_j, $j = 1, 2, \ldots, nc$. Furthermore, let $D = R_i$ be the set that contains two objects $\underline{p}_1$ and $\underline{p}_2$ with the largest distance between them, as compared to other sets R_j, and contains

objects from two classes at least. The subset D is divided into the sets D_1 and D_2. Then $R_i = D_1$ and $R_{nc+1} = D_2$. In this way the number of subsets R_j increased form nc to $nc+1$. If the subsets R_j will be exhausted then the subsets contained objects from one class only are further divided. More formally, the algorithm may be described in the following way.

Definition of the algorithm

1. Put $R_1 := R$, $i := 1$, $nc := 1$ and give the desired size nd of the condensed set;
2. In the set $D = R_i$ find two objects $\underline{p}_1$ and $\underline{p}_2$ separated by the largest distance;
3. Divide the set D into two subsets D_1 and D_2, where $d(\circ, \circ)$ is a distance measure: $D_1 := \{\underline{p} \in D : d(\underline{p}, \underline{p}_1) \leq d(\underline{p}, \underline{p}_2)\}, D_2 := \{\underline{p} \in D : d(\underline{p}, \underline{p}_1) > d(\underline{p}, \underline{p}_2)\}$;
4. $nc := nc + 1, R_i := D_1, R_{nc} := D_2$;
5. If $nc = nd$, then go to 11;
6. Put $J_1 := \{j : j \leq nc\}$ and R_j contains objects at least from two classes;
7. $J_2 := \{j : j \leq nc\} - J_1$;
8. Put $J := J_1$ if J_1 is not empty, otherwise $J := J_2$;
9. Find pairs of objects $\underline{p}_1$ and $\underline{p}_2$ separated by the largest distance in each subset R_j for $j \in J$ and determine an index $i \in J$ of the set R_i with the largest distance between $\underline{p}_1$ and $\underline{p}_2$;
10. Go to 3;
11. Find gravity centers of all obtained subsets R_j, $j = 1, 2, \ldots, nc$, and put them to S.

The determination of the pairs of objects $\underline{p}_1$ and $\underline{p}_2$ separated by the largest distance can be replaced by the pairs of mutually farthest objects [6]. The way of they determination is explained in the Fig. 6.

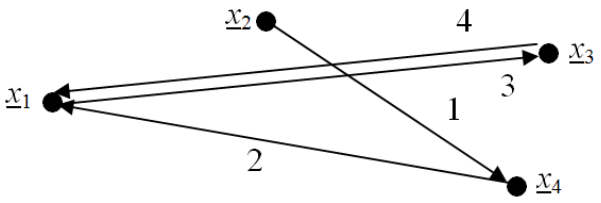

Fig. 6. The way of finding the mutually farthest objects.

In the case of situation shown in the Fig. 6, the algorithm starts with the objects $\underline{x}_2$. The farthest object in relation to $\underline{x}_2$ is $\underline{x}_4$. Then the maximally distanced object from $\underline{x}_4$ is $\underline{x}_1$. Finally, a loop will be obtained since $\underline{x}_3$ is it the farthest in relation to $\underline{x}_1$ and vice versa.

8. TRAINING SET EDITING FOR LINEAR SEPARABILITY

The most condensed reference set, as it was already mentioned above, consist of class gravity centers. Such a condensed reference set is used in the case of the minimum distance classifier. The classification task can concern two classes only. In this case the minimum distance classifier can be defined by a hyperplane H_0 passing in the middle between two gravity centers $\underline{a}$ and $\underline{b}$ and orthogonal to the vector $\underline{a} - \underline{b}$, as it is illustrated in the Fig. 7.

The hyperplane H_0 does not separate the classes 1 (crosses) and 2 (wheels) correctly. From among hyperplanes parallel to H_0 it is possible to find a hyperplane H_1, which perfectly separates the sets X_1 and X_2. This idea can be used also in the case when the sets X_1 and X_2 would be not linearly separable [5]. It is easy to find the hyperplane H_1, parallel to H_0, separating correctly maximum number of objects from the sets X_1 and X_2. Another approach consists in removing the objects which disturb to separate the investigated sets by a hyperplane.

One dimensional case is sufficient to explain how the proposed procedure can operate. The illustrating example is given in the Fig. 8.

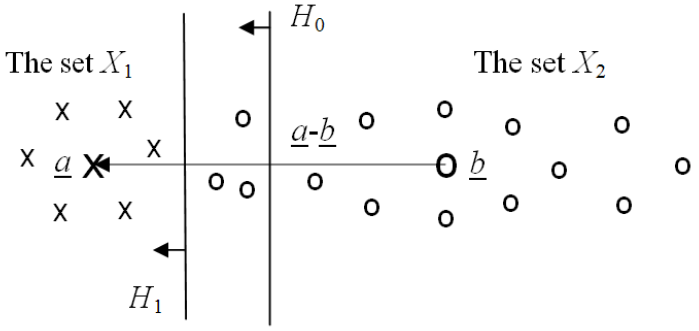

Fig. 7. The disadvantage of the minimum distance classifier.

Feature value	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Class symbol	x	x	x	-o	x x	x o	o	-x	o		o	o		o
2. Class symbol	x	x	x		x x	x -o	o	-x	o		o	o		o
3. Class symbol	x	x	x		x x	x	o		o		o	o		o

Fig. 8. The training set editing procedure for obtaining the linear class separability.

The extreme objects are $\underline{d}_1 = [4]$ and $\underline{c}_1 = [8]$, what was marked by the symbol "-", see row number 1. There are 4 crosses and 3 wheels (including $\underline{c}_1$ and $\underline{d}_1$) between $\underline{c}_1$ and $\underline{d}_1$, thus the crosses are in majority. The area between $\underline{c}_1$ and $\underline{d}_1$ ought to belong rather to the class 1, so the object $\underline{d}_1 = [4]$ is rejected and the training set contains now objects listed in the row 2. In the obtained situation the extreme objects are the wheel $\underline{d}_2 = [6]$ and the cross $\underline{c}_2 = \underline{c}_1 = [8]$. There are 2 crosses and 2 wheels between the objects $\underline{c}_2$ and $\underline{d}_2$, including these objects. In the case of ties appearing, both such objects are removed, i.e. the cross $\underline{c}_2 = [8]$ and the wheel $\underline{d}_2 = [6]$, marked by "-", in the considered example.

Finally, see row 3, the extreme objects are $\underline{c}_3 = [6]$ and $\underline{d}_3 = [7]$ and no objects are marked by "-". There are no objects between $\underline{c}_3$ and $\underline{d}_3$ except they themselves. It is easy to notice that the objects $\underline{c}_i$, $i = 1, 2$ were on the right side in the relation to the objects $\underline{d}_i$. They were in shorter distances to the opposite classes than to the their own classes.

Algorithm definition

1. Find the gravity centers $\underline{a}$ and $\underline{b}$ of the classes 1 and 2.
2. Determine the hyperplane $g_B(\underline{x}) = (\underline{a} - \underline{b}) \circ (\underline{x} - \underline{b})$.
3. From among objects from the class 1 find the object $\underline{c}$ such that $g_B(\underline{c})$ is the minimum value of $g_B(\underline{x})$ for $\underline{x} \in X_1$ and the object $\underline{d}$ from the class 2 such that $g_B(\underline{d})$ is maximum of $g_B(\underline{x})$ for $x \in X_2$. Determine two hyperplanes $g_C(\underline{x}) = (\underline{a} - \underline{b}) \circ (\underline{x} - \underline{c}) = 0$ and $g_D(\underline{x}) = (\underline{a} - \underline{b}) \circ (\underline{x} - \underline{d})$.
4. If $g_B(\underline{c}) > g_B(\underline{d})$ then go to 8.
5. If $g_B(\underline{c}) \leq g_B(\underline{d})$ then find the a number l_1 of objects from the set X_1 and a number l_2 of objects from the set X_2 satisfied the condition: $g_C(\underline{x}) \geq 0$ and $g_D(\underline{x}) \leq 0$.
6. If $l_1 > l_2$ then remove the object $\underline{d}$. When $l_2 > l_1$ then remove the object $\underline{c}$. Remove $\underline{c}$ and $\underline{d}$ in the case of $l_1 = l_2$.
7. Go to 3.
8. Determine the discriminant function $h(\underline{x}) = g_C(\underline{x}) + g_D(\underline{x})$.

The function $h(\underline{x})$ defines the classifier for the class pair 1 and 2. If $h(\underline{x}) \geq 0$, then the object $\underline{x}$ is qualified to the class 1 otherwise it is assigned to the class 2. The algorithm can be improved if in the step 8 the discriminant function $h(\underline{x})$ will be determined in a such way that the hyperplane $h(\underline{x}) = 0$

is an optimum one, i.e. this hyperplane is maximally distanced from the nearest objects from the set $Y = Y_1 \cup Y_2$, where Y_1 and Y_2 are the sets obtained from the sets X_1 and X_2 by application the above presented algorithm. An advantage of this modification is explained on the example shown in the Fig. 9. The hyperplane H_{AB} is more desired than the hyperplane H_{CD}, received in the step 8 of the above presented algorithm. The simple procedure for finding the optimum hyperplane was proposed in the paper [11].

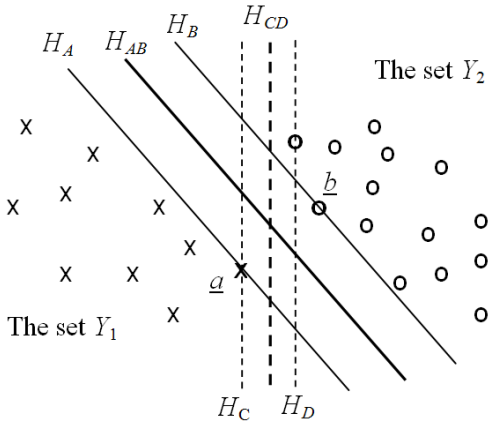

Fig. 9. The algorithm improvement by determining the optimum separating hyperplane.

The algorithm for finding the optimum separating hyperplane consists in determination of two nearest points $\underline{a}$ and $\underline{b}$ in the convex hulls of the sets Y_1 and Y_2 respectivelly, what was illustrated in the Fig. 10.

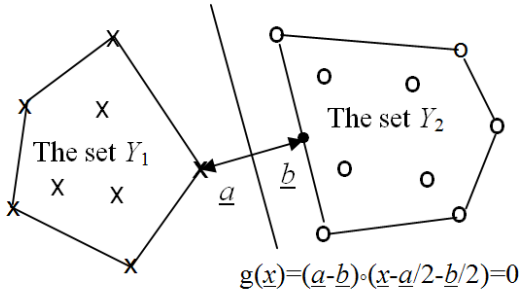

Fig. 10. The idea of algorithm for finding the optimum separating hyperplane.

BIBLIOGRAPHY

[1] CARPENTER G. A., GROSSBERG S., Learning, categorization, rule formation, and prediction by fuzzy neural networks, in the book "Fuzzy logic and neural network handbook", edited by C.H. Chen, McGraw-Hill Series on Computer Engineering, New York, 1996, pp. 1.3-1.45.

[2] FIX E., HODGES J. L., Discriminatory analysis: nonparametric discrimination small sample performance, Project 21-49-004, report number 11, USAF school of aviation medicine, Randolph Field, Texas, 2001, pp. 280-322.

[3] JÓŹWIK A., A learning scheme for a fuzzy k-NN rule, Pattern Recognition Letters 1, 1983, pp. 287-289.

[4] JÓŹWIK A., Pattern recognition method based on k nearest neighbor rule, Journal of Communications, 1994, Vol. XLV, pp. 27-29.

[5] JÓŹWIK A., Nieparametryczne metody klasyfikacji nadzorowanej, Wydawnictwo Instytutu Biocybernetyki i Inżynierii Biomedycznej PAN, 2013.

[6] JÓŹWIK A., KIEŚ P., Reference set reduction for 1-NN rule based on finding mutually nearest and mutually furthest pairs of points, Advances in Soft Computing, Computer Recognition Systems, Springer-Verlag, Berlin-Heidelberg, 2005, pp. 195-202.

[7] JÓŹWIK A., SERPICO S. B., ROLI F., Condensed Version of the k-NN rule remote sensing image classification, Image and Signal Processing for Remote Sensing II, EUROPTO Proceedings, SPIE, 1995, Vol. 2579, pp. 196-198.

[8] JÓŹWIK A., SERPICO S., ROLI F., A parallel network of modified 1-NN and k-NN classifiers-application to remote-sensing image classification, Pattern Recognition Letters 19, 1998, pp. 57-62.

[9] JÓŹWIK A., STAWSKA Z., Wielostopniowy klasyfikator typu najbliższy sąsiad z każdej klasy, Materiały VIII Konferencji "Sieci i Systemy Informatyczne", Łódź, 2000, str. 339-346 (in Polish).
[10] JÓŹWIK A., VERNAZZA G., Recognition of leucocytes by a parallel k-NN classifier, Lecture Notes of the 'ICB Seminar, 1988, pp. 138-153.
[11] KOZINIEC B. N., Recurent algorithm separating convex hulls of two sets (V. N. Vapnik, Ed.), Soviet radio, Moscow, 1973, pp. 43-50 (in Russian).
[12] LACHENBRUCH P. A., Estimation of Error Rates in Discriminant Analysis, Ph.D. dissertation, University of California, Los Angeles, 1965, Chapter 5.
[13] LESIAK B., JÓŹWIK A., Quantitative analysis of AuPd alloys from the shape of XPS spectra by the fuzzy rule, Surface and Interface Analysis, 2004, Vol. 36, pp. 793-797.
[14] SANCHEZ J. S., High training set size reduction by space partitioning and prototype abstraction, Pattern Recognition 37, 2004, pp. 1561-1564.
[15] TOMEK I., Two modifications of CNN, IEEE Trans. Systems, Man, and Cybernetics, 1977, Vol. 7, No. 2, pp. 92-94.