# Heterogeneous distance functions for prototype rules: influence of parameters on probability estimation.

Marcin Blachnik[1], Włodzisław Duch[2,3], Tadeusz Wieczorek[1]

[1] Division of Engineering Informatics, Department of Electrotechnology, Faculty of Materials Engineering and Metallurgy, Silesian University of Technology
Krasińskiego 8, 40-019 Katowice, Poland
[2] Department of Informatics, Nicolaus Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland
[3] School of Computer Engineering, Nanyang Technological University
Singapore

**Abstract.** An interesting and little explored way to understand data is based on prototype rules (P-rules). The goal of this approach is to find optimal similarity (or distance) functions and position of prototypes to which unknown vectors are compared. In real applications similarity functions frequently involve different types of attributes, such as continuous, discrete, binary or nominal. Heterogeneous distance functions that may handle such diverse information are usually based on probability distance measure, such as the Value Difference Metrics (VDM). For continuous attributes calculation of probabilities requires estimations of probability density functions. This process requires careful selection of several parameters that may have important impact on the overall classification accuracy.

In this paper various heterogeneous distance function based on VDM measure are presented, among them some new heterogeneous distance functions based on different types of probability estimation. Results of many numerical experiments with such distance functions are presented on artificial and real datasets, and quite simple P-rules for several heterogeneous databases extracted.

**Keywords:** Prototype rules, probability estimation, heterogeneous distance functions, similarity-based methods, classification, data mining.

## 1 Introduction

Many important problems in artificial intelligence may be reduced to classification problems. Although the problem this is still a very active research area. Many different approaches to classification have been proposed, including such popular methods as artificial neural networks or support vector machines. Unfortunately their possible applications are limited, because they usually act as black boxes, and it is impossible to understand their decisions in logical terms (however, it is possible to visualize them [1,2]). For some inputs their decisions may be unpredictable, leading to disastrous consequences. This constitutes the reason why they may be dangerous to use in decision support systems that require transparency of decisions. If some understanding of data is demanded machine learning algorithms for logical rule extraction are used [3]. The challenge is to generate set (or multiple sets) of rules that will be reliable, accurate and easy to understand by humans. Although the need for rule-based descriptions is generally acknowledged many methods create too many rules with too complex conditions, in effect creating incomprehensible descriptions of data. In such a case it may be better to use reliable pattern recognition classifiers rather than rules.

Statistical approaches based on the "divide-and-conquer" idea lead to univariate decision trees that generate crisp logic rules operating on each attribute separately. The most popular examples are Quinlan's C4.5, Breiman's CART [4] or SSV tree algorithms [5]. Expressive power of crisp rules (C-rules) is rather limited, therefore fuzzy rules (F-rules), generated by neurofuzzy systems [6] are frequently used, although they may sometimes generate rather complex description of the data that is hard to understand, even though a simple crisp set of rules exist [3]. An alternative is offered by recently introduced prototype-based rules (P-rules) [7,8]. The F-rules and P-rules may be transformed into one another and allow to express more interesting concepts then crisp logic rules. Experiments showed that both approaches are usually capable of generating highly accurate, yet small and understandable sets of rules.

To specify F-rules the shape of membership functions has to be defined and parameters determining rule properties estimated from the data. For the P-rules the type of distance measure has to be defined. The usual choice is the Euclidean distance function or more generally the Minkovsky's family of distance functions. However, in practical applications it is not always the best solution. In real world most of datasets have mixed type of attributes, with some real-valued, some discrete, and some symbolic or nominal. For such data Euclidian distance functions are not directly applicable. In case of symbolic features results will depend on the method of conversion from nominal to numeric values. This problem also plays a role in fuzzy rules, where it is sometimes not clear what type of a specific membership function is appropriate.

Mixed types of attributes may be used in heterogeneous distance functions that use different types of measures for different types of attributes, combining information contained in their differences. This type of functions are usually based on probability distance measures, such as the Value Difference Metric (VDM) [9], adopted for continuous attributes by estimating probability density function for real feature values. The process of estimation requires selection of several parameters that may have an important influence on overall classification accuracy. This is the subject of this paper.

Section 2 describes relations between F-rules and P-rules and advantages which may be derived from this relation. In section 3 different heterogeneous distance function based on VDM measure are presented. Section 4 presents some new heterogeneous distance functions based on different types of probability estimation. Numerical experiments on artificial and real data are presented in section 5, and in section 6 the summary of results is given and some conclusions are drawn.

## 2 F-rules versus P-rules

Classical crisp logical rules are quit easy to understand, seem to be natural to most people, are easy to apply, and therefore are in wide use in decision support and similar applications. However, if the goal is to understand real-world data collected from some experiments crisp rules may often fail, because a large number of these rules with many conditions may be needed, making it impossible to understand structure of the data. F-rules and P-rules are much more flexible and a smaller number of such rules may be sufficient, therefore also more robust and easier to understand.

The process of learning F-rules starts with selecting shape of membership function (MF), their initial position separately for each feature, selecting appropriate fuzzy operators like T-norms, aggregation operators and inference scheme [6]. In the second step of the learning process algorithms tune the spread and position of each membership function and try to select appropriate combination of these functions to extract fuzzy if-then rules. Such methods are often based on neural adaptation algorithms, therefore are called neuro-fuzzy systems.

In P-rules the goal is to optimize position of prototypes to which unknown vectors will be compared using previously chosen distance function or similarity measure. Two different types of P-rules exist. First, the nearest neighbour rules (NNR), where distance is calculated between unknown case and all prototypes, and a prototype that is most similar is used in the condition part of the rule, claiming that the output class is the same as the class of the nearest prototype:

$$\text{IF } \mathbf{P}' = \arg\min_{a=1:L} D\left(\mathbf{X}, \mathbf{P}_a\right) \text{THEN C}(\mathbf{X}) = \text{C}(\mathbf{P}')$$

where $\mathbf{X}$ is the input vector, $\mathbf{P}_a$ is one of the L prototypes, and C(**P**) is a function returning the label of the vector **P**. These rules are discriminative and for two prototypes from different classes define a decision hyperplane between them.

P-rules of the second type are threshold rules (TR) where each prototype has associated threshold value defining subspace of activation and all vectors that fall into this subspace have output label that is the same as this prototype label.

$$\text{IF D}\left(\mathbf{X}, \mathbf{P}\right) < \text{Tr THEN C}(\mathbf{X}) = \text{C}(\mathbf{P})$$

where Tr is threshold value. These rules provide coverings of parts of the feature space.

F-rules, and more precisely their MF have four major interpretations [9]. One of the most popular and natural interpretation tries to explain MF as a degree to which elements of universe *X* are similar or typical to a fuzzy set *F* defined by its MF. From this perspective F-rules seems to be special case of similarity base learning (SBL) [11]. This observation leads to a new perspectives in both fields: SBL and Fuzzy Modelling. New distance functions may be derived form membership functions, and vice versa [7,8]. The simplest example is transformation between additive distance functions, which is equivalent to a product of MF using exponential transformation function:

$$D(\mathbf{x},\mathbf{y})^2 = \sum_{i=1}^{m} w_i (x_i - y_i)^2$$

$$F_\mu = \exp\left(D(\mathbf{x},\mathbf{y})^2\right) = \exp\left(\sum_{i=1}^{m} w_i (x_i - y_i)^2\right) = \prod_{i=1}^{m} \exp\left(w_i (x_i - y_i)^2\right)$$

$$F_\mu = \prod_{i=1}^{m} \mu_i(x_i)\,; \qquad \mu_i(x_i) = \exp\left(w_i (x_i - y_i)^2\right)$$

A main weakness of F-rules is the difficulty of their application to heterogeneous datasets. Theoretically fuzzy sets can also be defined for arbitrary features, however most of the neurofuzzy systems (ex. NEFCLASS, ANFIS) don't support symbolic or non ordered attributes, and are defined only for real inputs values [6]. The problem with automatic construction of MFs for nominal attributes may be solved using the SBL approach. Probability-based distance functions, such as VDM metrics, may be defined for any type of features, and they may be combined with real-valued features in heterogeneous distance functions (HDF) [12]. They are described in the next two sections.

## 3 Heterogeneous Distance Functions

In most similarity based systems, such as the nearest neighbor [4], Radial Basis Function networks, or self-organizing maps [13], Minkovsky's distance function are used, sometimes in rotated coordinate systems, or example the Mahalanobis distance function. Unfortunately this type of distance functions does not support symbolic and nominal features that are often found in real applications. On the other hand the Value Difference Metrics (VDM) and similar metrics [9] gives very good results for symbolic attributes, but using it directly with continuous attributes is impossible. Constructing prototype-based rules for datasets with different types of attributes both types of similarity functions should be combined in a heterogeneous distance function [12].

VDM distance measure is based on calculation the differences between posterior probabilities:

$$VDM\left(\mathbf{X},\mathbf{Y}\right)=\sum_{a=1}^{m}vdm\left(x_a,y_a\right) \tag{1}$$

$$vdm\left(x_a,y_a\right)^2=\sum_{i=1}^{n}\left(p\left(C_i\mid x_a\right)-p\left(C_i\mid y_a\right)\right)^2 \tag{2}$$

where probabilities are estimated by frequencies:

$$p\left(C_i|x_a\right)=\frac{Nx_{ai}}{Nx_a} \tag{3}$$

**X** and **Y** are input vectors, $Nx_a$ is a number of instances in a training set with value of $x_a$ for the attribute $a$, $Nx_{ai}$ is the same as $N_a$ but for the class $C_i$, $n$ is number of classes and $m$ is the number of attributes. P-rules use heterogeneous distance functions (HDF) for features of mixed types. One of the simplest ways leading to the HDF is combination of the Euclidean and VDM metrics, called the Heterogeneous Value Difference Metric (HVDM) [11]:

$$\mathrm{HVDM}(\mathbf{x},\mathbf{y})=\sqrt{\sum_{a=1}^{m}\mathrm{d}_\mathrm{a}^2(\mathrm{x}_\mathrm{a},\mathrm{y}_\mathrm{a})} \tag{4}$$

where

$$\mathrm{d_a}(\mathrm{x},\mathrm{y})=\begin{cases}1, & \text{x or y are unknown}\\ \mathrm{n\_vdm_a}(\mathrm{x},\mathrm{y}) & \text{a is discrete or nominal}\\ \mathrm{n\_dif_a}(\mathrm{x},\mathrm{y}) & \text{a is continues}\end{cases} \tag{5}$$

For the nominal data, $\mathrm{d_a}(\mathrm{x,y})$ assumes one of the forms:

$$\begin{aligned}&\mathrm{N1}:\\ &\mathrm{n\_vdm_a}(\mathrm{x},\mathrm{y})=\sum_{i=1}^{n}\left|\frac{Nx_{ai}}{Nx_a}-\frac{Ny_{ai}}{Ny_a}\right|\\ &\mathrm{N2}:\\ &\mathrm{n\_vdm_a}(\mathrm{x},\mathrm{y})=\sqrt{\sum_{i=1}^{n}\left|\frac{Nx_{ai}}{Nx_a}-\frac{Ny_{ai}}{Ny_a}\right|^2}\\ &\mathrm{N3}:\\ &\mathrm{n\_vdm_a}(\mathrm{x},\mathrm{y})=\sqrt{n\cdot\sum_{i=1}^{n}\left|\frac{Nx_{ai}}{Nx_a}-\frac{Ny_{ai}}{Ny_a}\right|^2}\end{aligned} \tag{6}$$

and for continuous data

$$n\_vdm_a(\mathrm{x},\mathrm{y})=\frac{|x-y|}{4\sigma_a} \tag{7}$$

where $\sigma$ is the standard deviation for the attribute $a$. The Euclidean distance used by HVDM for continuous features is normalized by standard deviation of the attribute to reduce the influence of the outliers.

Three different forms of VDM distance with different normalization technique are used, and the decision which one should be chosen depends on a designer of the system. The main problem using HVDM is normalization, because it is very difficult to balance different terms in the overall distance metric. This problem does not occur for the Value Difference Metric when posterior probabilities are estimated for both discrete and continuous features. However, in such case the estimation of probability density for continuous features is a big problem. Martinez and Willson [11] described also Discretized Value Difference Metric (DVDM) and Interpolated Value Difference Metric (IVDM ).

DVDM is based on discretization process and for continuous attributes a simple constant width discretization method is used. DVDM is defined by the equation:

$$\mathrm{DVDM}(\mathbf{x},\mathbf{y})^2 = \sum_{a=1}^{m} vdm_a\left(disc_a(x_a), disc_a(y_a)\right)^2 \tag{8}$$

Where *disc* is a discretization function defined as:

$$disc_a(x_a) = \begin{cases} \left\lfloor \dfrac{x - \min_a}{w_a} \right\rfloor + 1 & \text{if x is continous} \\ x & \text{if x is discrete} \end{cases} \tag{9}$$

*min*$_a$ is the minimum of attribute *a* and *w*$_a$ is a parameter describing the number of discretization bins. However, the upper part of equation (9) can be replaced by a different form of discretization algorithm.

IVDM is very similar to DVDM, but to improve the accuracy of posterior probability estimates a simple linear interpolation is used. IVDM is defined by:

$$IVDM(\mathbf{x},\mathbf{y}) = \sum_{a=1}^{m} indm_a(x_a, y_a) \tag{10}$$

$$ivdm_a(x_a, y_a) = \begin{cases} vdm_a(x_a, y_a) & \text{a is discrete} \\ \sum_{i=1}^{n} \left| p_{ai}(x) - p_{ai}(y) \right|^2 & \text{a is continous} \end{cases} \tag{11}$$

where

$$p_{ai}(x) = p_{aiu} + \frac{x - mid_{au}}{mid_{a,u+1} - mid_{a,u}} \cdot \left(p_{ai,u+1} - p_{aiu}\right) \tag{12}$$

Here $p_{aiu}$ and $p_{ai,u+1}$ are posterior probabilities calculated in the middle of the discretized range *u* and *u*+1, *u=disc(x)* and $mid_{au}$ and $mid_{a,u+1}$ are middles of discretized ranges *u* and next *u*+1, for which actual $x_a$ fulfill inequality.

## 4 NEW HETEROGENEOUS DISTANCE FUNCTIONS

The main problem in application of VDM distance measure to continuous attributes is to obtain appropriate shape of posterior probabilities. For discrete or symbolic features they can be easily computed using frequencies, Eq. (3), but for continuous attributes it will not work. Two simple techniques were presented in the previous section, but a better algorithms used for determining posterior probabilities may lead to a better overall results. These new methods are based on equation (11), but with a different density estimation technique.

### 4.1 Gaussian Value Difference Metric

Kernel smoothing techniques, for example Gaussian smoothing kernels, allow to calculate the posterior probability as:

$$p\left(C_i \mid x_a\right) = \left( \sum_{j=1}^{M_i} \exp\left[ -\left( \frac{x_{aj}}{\sigma} \right)^2 \right] \right) \cdot norm \qquad (13)$$

where $M_i$ is the number of all vectors from the same class $C_i$, $\sigma$ is the width of Gaussian functions, and *norm* is the normalization factor calculated by:

$$norm = 1 \Big/ \sum_{k=1}^{n} \left( \sum_{j=1}^{M_i} \exp\left[ -\left( \frac{x_{aj}}{\sigma} \right) \right]^2 \right) \qquad (14)$$

### 4.2 Local Value Difference Metric (LVDM)

Very simple and very fast technique for estimating probability is based on the Local Value Difference Metric (LVDM). This method uses local calculation of probability density surrounding the query data point. In this method probability is calculated by the equation (3), but the value of $Nx_{ai}$ is the number of points in class $C_i$ in the range limited to $\left[ x_a - \frac{width_a}{2}, x_a + \frac{width_a}{2} \right]$, and $Nx_a$ is similarly calculated in this range for all classes. $Width_a$ is a parameter defining range of width for attribute $a$.

### 4.3 Parzen Value Difference Metric (PVDM)

Another solution for density estimation is based on the Parzen Window technique [4] where rectangle window is moved by the step through the whole range of attribute $a$, and probability is calculated as a mean value of all window probabilities where $x$ occurs:

$$p\left(C_i \mid x_a\right) = \frac{1}{Z} \sum_{z=b+1}^{b+Z} \frac{N_{iz}(x_a)}{N_z(x_a)} \qquad (15)$$

where $Z$ is number of windows, $Z = \frac{width_a}{step_a}$, $b$ is the index of first window where $x$ occurs, $N_{iz}(x_a)$ is the number of data points in $z$-th window that belong to class $C_i$, $N_z(x_a)$ is summed over all classes, $width_a$ is window width for attribute $a$, and $step_a$ is the size of the window movement.

# 5 NUMERICAL EXPERIMENTS

Experiments were performed on artificial and real data. The artificial data were generated to check the quality of probability estimations, and the influence of the parameters used on the probability estimation accuracy. Two artificial datasets were generated for this purpose. First dataset has two dimensions, 3 classes, with vectors for each class generated from a normal distribution. The second dataset has also two dimensios and 3 classes, but data vectors were generated using uniform distribution. In both datasets classes were partially overlapping.

In the second set of experiments P-rules were generated using probabilistic distance functions for several datasets taken from the UCI repository of machine learning datasets [14]. For these experiments different datasets were selected with different types of attributes: continuous, discrete, symbolic and nominal.

All tasks were carried out with SBPS software system especially developed for that purpose. SBPS is a similarity based rule generating system that allows fro defining different type of distance function for different attributes and combining the results obtained with each feature into a final value. SBPS system includes several prototype selection and optimization algorithms which are used to simplify and improve initial rules. For comparison of results obtained in different experiments only Fuzzy C-means algorithm for prototype selection and LVQ algorithm for their optimization have been used [13].

## 5.1 Artificial datasets

Artificial datasets were created to verify the quality of five methods for probability estimation, compare them with a heterogeneous distance function, and evaluate the influence of parameters of these methods on the final classification results. For the first artificial dataset with normal distribution of samples in each class optimal border shape can be obtained using Euclidean distance function. These results determine a basis to judge and compare quality of probability estimation and classification for other functions. In this test only one prototype per class was generated. To reduce the influence of randomness and verify generalization 10-fold crossvalidation test was performed. Results presented in Tab. 1 show balanced accuracy for each method.

**Table 1.** Balanced accuracy for different methods of probability estimation obtained on artificial datasets

|  |  | HVDM | GVDM |  |  | LVDM |  |  |  | IVDM |  | DVDM |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  | sig 0. | sig 0. | sig 0. | width | width | width | width | CW 1 | CW 5 | CW 1 | CW 5 |
| **Dataset 1** | Bal. Acc | 96,83 | 95,67 | 96,50 | 96,17 | 95,00 | 95,33 | 95,50 | 95,33 | 95,17 | 94,33 | 96,50 | 90,50 |
| **Dataset 2** | Bal. Acc | 90,50 | 88,33 | 90,67 | 90,33 | 86,00 | 88,17 | 88,33 | 89,00 | 86,83 | 87,50 | 85,17 | 81,33 |

|  |  | PVDM |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | Step 0.1 |  |  |  | Step 0.01 |  |  |  | Step 0.05 |  |  |  |
|  |  | W0.2 | W0.4 | W0.6 | W0.7 | W0.2 | W0.4 | W0.6 | W0.7 | W0.2 | W0.4 | W0.6 | W0.7 |
| **Dataset 1** | Bal. Acc | 94,67 | 94,83 | 95,83 | 96,17 | 95,00 | 95,50 | 96,00 | 96,50 | 94,67 | 94,00 | 96,00 | 96,17 |
| **Dataset 2** | Bal. Acc | 86,33 | 88,17 | 90,00 | 90,00 | 86,67 | 87,17 | 88,33 | 89,00 | 86,50 | 87,00 | 88,83 | 88,67 |

### 5.2 Real datasets

The 6 types of VDM distance functions have also been tested on real datasets to verify theoretical considerations. Several datasets with different types of attributes were selected from the UCI repository: Flag, Glass, Iris, Lancet and the Pima Indians. Because the aim was to obtain maximum balanced accuracy for all these distance measures we have used the algorithm for constructive rule generation to maximize classifier abilities.

The constructive algorithm for generation of P-rules does not favor any distance function because it adds new prototype to the class with lowest accuracy, maximizing overall balanced accuracy calculated as a mean value of individual accuracies. In all cases the algorithm was stopped after 10 iterations, generating at most 10 prototypes per class.

**Table 2.** Balanced accuracy for different methods of probability estimation obtained on real datasets

| | **flag** | **glass** | **iris** | **lancet** | **pima** |
|---|---|---|---|---|---|
| HVDM | Bal. Acc | Bal. Acc | Bal. Acc | Bal. Acc | Bal. Acc |
| | 18,958 | 37,772 | 96,000 | 90,228 | **73,740** |
| GVDM | | | | | |
| sig 0.2 | 23,229 | 48,948 | 96,000 | 89,994 | 71,815 |
| sig 0.5 | **30,208** | 55,367 | 96,667 | 89,777 | 71,401 |
| sig 0.7 | 28,438 | 46,865 | 96,667 | 89,777 | 71,386 |
| mean | 27,292 | 50,394 | 96,444 | 89,849 | 71,534 |
| std | 3,628 | 4,431 | 0,385 | 0,126 | 0,244 |
| LVDM | | | | | |
| width 0.2 | 25,625 | 47,778 | 96,000 | 90,103 | 72,886 |
| width 0.4 | 27,708 | 44,147 | 96,667 | 89,994 | 72,049 |
| width 0.6 | 26,563 | 48,978 | 95,333 | 89,994 | 71,490 |
| width 0.7 | 26,875 | 42,054 | 94,000 | 89,777 | 71,676 |
| mean | 26,693 | 45,739 | 95,500 | 89,967 | 72,025 |
| std | 0,861 | 3,202 | 1,139 | 0,137 | 0,619 |
| PVDM | | | | | |
| W0.2 St0.1 | 30,104 | 39,722 | 96,667 | 90,103 | 71,613 |
| W0.4 St0.1 | 26,563 | 42,639 | 96,667 | 89,994 | 71,504 |
| W0.6 St0.1 | 24,375 | 49,702 | 95,333 | 89,777 | 70,531 |
| W0.7 St0.1 | 27,396 | 49,206 | 96,667 | 89,876 | 71,034 |
| W0.2 St0.01 | 29,479 | 46,359 | 96,000 | 90,005 | 71,820 |
| W0.4 St0.01 | 25,625 | 45,694 | 96,000 | 89,994 | 71,468 |
| W0.6 St0.01 | 24,375 | **58,046** | 96,667 | 89,777 | 71,234 |
| W0.7 St0.01 | 27,083 | 48,075 | 96,667 | 89,777 | 71,041 |
| W0.2 St0.05 | 28,542 | 46,319 | 96,000 | 90,103 | 71,386 |
| W0.4 St0.05 | 26,250 | 44,345 | 96,000 | 89,994 | 71,482 |
| W0.6 St0.05 | 24,375 | 56,141 | 96,000 | 89,777 | 70,970 |
| W0.7St0.05 | 27,813 | 56,379 | 96,667 | 89,777 | 71,555 |
| mean | 26,832 | 48,552 | 96,278 | 89,913 | 71,303 |
| std | 1,953 | 5,717 | 0,446 | 0,133 | 0,355 |
| IVDM | | | | | |
| CW 10 | 26,563 | 46,984 | 96,000 | 90,225 | 70,818 |
| CW 5 | 26,042 | 48,651 | 96,667 | 90,117 | 72,375 |
| mean | 26,302 | 47,817 | 96,333 | 90,171 | 71,597 |
| std | 0,368 | 1,179 | 0,471 | 0,077 | 1,101 |
| DVDM | | | | | |
| CW 10 | 26,979 | 43,810 | **97,333** | 90,325 | 71,081 |
| CW 5 | 27,083 | 50,635 | 94,667 | **90,330** | 70,142 |
| mean | 27,031 | 47,222 | 96,000 | 90,327 | 70,612 |
| std | 0,074 | 4,826 | 1,886 | 0,003 | 0,664 |

Because of the normalization problem of different distance functions, all continuous features in all datasets were standardized and then normalized to the interval [0,1]. The highest balanced accuracy for each combination of parameters for all datasets is presented in Table 2.

## 6 Discussion of results and conclusions

The “no free lunch” theorem [4] says that no single algorithm for data analysis may always be the winner, and the results presented in Tab. 2 certainly confirm it. For artificial data the GVDM algorithm seems to be better than other estimation methods. It could be expected that this algorithm should give a very good result for this type of artificial data, with high density of points, generating the smoothest estimated probability distributions, but selection of appropriate parameters has significant influence on the estimations.

Results on real datasets show that choosing correct algorithm parameters is now very important and selection of the single best distance function is impossible. In Table 2 the highest accuracies, marked as bold, appear for different methods for each dataset. The GVDM distance does not work so well now, sometimes giving large variance of results for different parameters of probability estimation algorithms. These results unfortunately do not lead to any definite conclusion about what type of distance should be used or which values of parameters are the best. If some values of estimation parameters are wrongly chosen contours of probability distribution may be very jagged and important information about data may be lost.

Some general conclusions about appropriate values of estimation parameters may be reached. For LVDM distance it is impossible to select accurate window size, for example for the Flag dataset most appropriate value is 0.4, but on the Glass dataset for the same value almost worst results have been obtained; still the standard deviation of the accuracy for LVDM is rather small for all dataset, making this method rather insensitive to the choice of its parameters.

Much better but less stable results were obtained with GVDM algorithm. Although for all datasets $\sigma$=0.5 leads to the best or nearly the best results, the variance of these results is larger. DVDM and IVDM methods were tested only with two significantly different parameter values, but the differences in accuracy is for these methods rather small. The Parzen window PVDM algorithms tends to prefer small step sizes, with the best results achieved with step 0.05 and 0.01, while the step size of 0.1 led to the worst results; also a wider window is preferred, about 0.6-0.7.

Comparison between different methods isn’t so clear, calculations performed on some datasets show that even the simplest DVDM measure may sometimes give good results. This situation occurs when a gap between different classes is very small, and the more advanced techniques that use smoothing usually lead to an increased number of errors; it is especially important for datasets with small number of training vectors.

An interesting extension of the work described here may be done by replacing VDM metric function with another probability distance metrics, such as the Minimum Risk Metric (MRM) or Short and Fukunga metric (SFM) [9]. Also other kernel smoothing techniques should be analyzed and compared. A significant influence of more advanced discretization algorithms may be expected. These methods will be analyzed in the near future. The final goal is to create simplest and most accurate P-rules for any kind of data.

Acknowledgement: WD is grateful for the support by the Polish Committee for Scientific Research, research grant 2005-2007.

## References

1. Duch W, Visualization of hidden node activity in neural networks: I. Visualization methods. Lecture Notes in AI Vol. 3070 (2004) 38-43; II. Application to RBF networks. Lecture Notes in AI Vol. 3070 (2004) 44-49
2. Duch, W, Coloring black boxes: visualization of neural network decisions. International Joint Conference on Neural Networks, Portland, Oregon, 2003, IEEE Press, Vol. I, pp. 1735-1740
3. Duch W, Setiono R, Żurada J. (2004) Computational intelligence methods for rule-based data understanding, *Proceedings of the IEEE*, 92(5): 771- 805
4. Hastie T, Tibshirani R. and Friedman J. The Elements of Statistical Learning. Springer, 2001.
5. Grąbczewski K, Duch W. (2000) The separability of split value criterion, *5'th Conf. on Neural Network and Soft Computing*, Zakopane, Poland, pp. 201-208
6. D. Nauck, F. Klawonn and R. Kruse, Foundations on Neuro-Fuzzy Systems. Wiley, Chichester, 1997
7. Duch W, Grudziński K. (2001) *Prototype based rules - a new way to understand the data.* IEEE International Joint Conference on Neural Networks, Washington D.C., IEEE Press, pp. 1858-1863
8. Duch W., Blachnik M. (2004) Fuzzy rule-based system derived from similarity to prototypes, *Lecture Notes in Computer Science,* 3316, 912-917
9. Blanzieri E, Ricci F. (1999) Probability Based Metrics for Nearest Neighbor Classification and Case-Based reasoning. In: Case-Based reasoning and development. Althoff K., Bergmann R., Branting K. (eds), Springer, pp. 14-28
10. Dubois D, Prade H. (eds.) (2000) Measurements of Membership Functions: Theoretical and Empirical Work, *Fundamentals of Fuzzy Sets.* Kluwer
11. Duch W. (2000) Similarity based methods: a general framework for classification, approximation and association. *Control and Cybernetics* 29(4), 937-968
12. Wilson R.T, Martinez T.R. (1997) Improved Heterogeneous Distance Function, *Journal of Artificial Intelligence Research,* vol. 6, 1-34
13. Haykin S, Neural networks: a comprehensive foundations. New York: MacMillian Publishing, 1994
14. Mertz C.J, Murphy P.M, UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html