PROBABILISTIC DISTANCE MEASURES FOR PROTOTYPE-BASED RULES

Włodzisław Duch
Nicolaus Copernicus University, Department of Informatics, Grudziądzka 5, Toruń, Poland
and Nanyang Technological University, School of Computer Engineering, Singapore
Google: Duch

Marcin Blachnik and Tadeusz Wieczorek
Silesian University of Technology, Department of Electrotechnology, Krasińskiego 8, 40-019 Katowice, Poland
e-mail: marcin.blachnik@polsl.pl
e-mail: tadeusz.wieczorek@polsl.pl

ABSTRACT

Probabilistic distance functions, including several variants of value difference metrics, minimum risk metric and Short-Fukunaga metrics, are used with prototype-based rules (P-rules) to provide a very concise and comprehensible classification model. Application of probabilistic metrics to nominal or discrete features is straightforward. Heterogeneous metrics that handle continuous attributes with discretized or interpolated probabilistic metrics were combined with several methods of probability density estimation. Numerical experiments on artificial and real data show the usefulness of such approach as an alternative to neurofuzzy models.

1. INTRODUCTION

One of the most important goals of computational intelligence is data understanding. Many popular pattern recognition methods used for classification, such as the artificial neural networks, Support Vector Machines or statistical methods [1],[2] have limited applications because their recommendations cannot be explained in simple terms. As a result there is a danger that "black box" predictions may completely fail for some specific inputs and thus in safety-critical applications (such as autopilot vehicle control, or medical applications) they should not be used. Algorithms based on logical rules are much better in this respect, but generation of sets of rules that are reliable, accurate and sufficiently simple to understand them is a difficult problem [3]. Most popular methods for rule generation are based on univariate decision trees, with rule conditions that operate on each attribute separately. Examples include Quinlan's C4.5 algorithm [4], CART [5] and SSV trees [6]. These types of trees have limited expressive powers; for example, they are not able to discover a simple rule "majority agrees" for *N* binary inputs, producing instead exponentially large number of prepositional rules.

Neurofuzzy systems [7]-[9] have found a very wide use for fuzzy rule construction, combining fuzzy modeling and neural adaptation. Such systems are applicable directly only to numerical inputs. Prototype-based rules (P-rules) [10] provide an alternative way to understand data, generating small and comprehensible sets of rules, frequently with higher accuracy than crisp or fuzzy rules [11]. Generation of P-rules requires optimization (or selection from the training set) of the position of a prototype (reference vector) to which the unknown vectors are compared, together with the optimization of the distance function or similarity measure (including feature scaling or selection). Two types of P-rules may be used: either the threshold-based rules (distance of X to P is smaller than some threshold), or the nearest prototype (neighbor) rules, where the shortest distance between X (unknown case) and all the prototypes is taken as an indication that the class of X is the some as the class of its nearest prototype.

Selection of the type of distance measure is obviously of primary importance. For numerical inputs weighted Euclidean distance function is frequently a good choice. However, in practical applications many datasets have mixed attribute types, some are continuous, some are discrete, and some are symbolic or nominal. In this case result will strongly depend on the method of conversion of nominal to numerical values. Fuzzy modeling and neurofuzzy systems have to face a more difficult problem, how to generate membership function for nominal data.

In general in the similarity-based methods [12]-[14], and in particular in the prototype-based rules, such problems are solved using heterogeneous distance functions based on probabilistic distances [15]. Different types of distance measures for different type of attributes are combined together, adding weighted Euclidean contributions for numerical attribute to contributions from the Value Difference Metrics (VDM), Short and Fukunaga Metrics

(SFM) and Minimum Risk Metrics (MRM) [16]. The use of probabilistic metrics for continuous attributes requires estimation of probability density functions, and the estimation methods may have important influence on overall classification accuracy.

Comparing to the fuzzy rules and neurofuzzy systems prototype-rules and algorithms to generate them are used very rarely. To show that P-rules are an interesting alternative to F-rules in this paper prototype-based rules are generated using heterogeneous distance functions with several methods for evaluation of probabilities for continuous features. In the next section different heterogeneous distance functions based on three types of probability difference metrics are presented, and methods to estimate probabilities based on discretization, Gaussian smoothing and Parzen windows described. Numerical experiments are presented in section 3, and in section 4 summary and discussion of the results is given.

2. HETEROGENEOUS DISTANCE FUNCTIONS

2.1 Probability difference metrics

All similarity based systems compare unknown case with reference cases using some type of distance (or similarity) measures [12]-[14]. Euclidean distance functions are most popular, giving in the nearest-prototype methods the same shape of decision borders as Gaussian membership functions used in similarity measures [11]. This is easily generalized to the Minkovsky distance function parameterized by the value of exponent that has strong influence on the decision borders.

Unfortunately these functions cannot be applied to the symbolic feature values directly, and the results are strongly dependent on the transformation used to convert symbolic into numerical values. The only principled solution to this problem is based on conditional probabilities. There are a few different ways to calculate probability difference metrics. The most popular is the Value Difference Metric [15], based on calculation of the differences between *a posteriori* probabilities:

$$VDM(\mathbf{x},\mathbf{y}) = \sum_{a=1}^{m} vdm\left(x_a, y_a\right) \quad (1)$$

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

where **x** and **y** are input vectors with symbolic feature values, *n* is the number of classes, *m* is the number of symbolic features, and more general Minkovsky form may be used. Probabilities are calculated using frequencies:

$$p\left(C_i \mid x_a\right) = N\left(C_i, x_a\right) / N\left(x_a\right) \quad (3)$$

Here $N(x_a)$ is number of instances in the training set with value x for the attribute a, and among them $N(C_i,x_a)$ is the number of instances from class C_i.

VDM is a heuristic metric. Alternative probability difference metric derived from probabilistic considerations has been proposed by Short and Fukunaga (*SFM*) and is calculated using the following formula [16]:

$$sfm(x_a, y_a) = \sum_{i=1}^{n} p\left(C_i \mid y_a\right) \left| p\left(C_i \mid y_a\right) - p\left(C_i \mid x_a\right) \right| \quad (4)$$

Assuming independence of all features the distance between (**x**,**y**) vectors is a sum over these contributions:

$$SFM\left(\mathbf{x},\mathbf{y}\right) = \sum_{a=1}^{m} sfm\left(x_a, y_a\right) \quad (5)$$

Another well-founded distance measure called Minimum Risk Metric (*MRM*) has been proposed by Blanzieri and Ricci [16]. *MRM* tries to minimize the risk of misclassification directly and is calculated from:

$$mrm(x_a, y_a) = \sum_{i=1}^{n} p\left(C_i \mid x_a\right) \left| 1 - p\left(C_i \mid y_a\right) \right| \quad (6)$$

with *MRM*(**x**,**y**) distance taken as a sum of contributions from all features:

$$MRM\left(\mathbf{x},\mathbf{y}\right) = \sum_{a=1}^{m} mrm\left(x_a, y_a\right)$$

2.2 Heterogeneous distance functions

Many real world datasets include mixture of attribute types – symbolic, linear, discrete and nominal. P-rules and other methods based on similarity need heterogeneous distance functions (HDFs), taking advantage of additive form of the distance functions (this can be justified only for independent features).

2.2.1 Heterogeneous distance functions

Combining the Euclidean and the Value Difference probabilistic metric a Heterogeneous Metric (*HM*) is obtained [15]:

$$HM\left(\mathbf{x},\mathbf{y}\right) = \sqrt{\sum_{a=1}^{m} d_a(x_a, y_a)^2} \quad (7)$$

where contributions from individual features are:

$$d_a\left(x,y\right) = \begin{cases} 1, & x \text{ or } y \text{ are unknown} \\ pm_a(x,y) & a \text{ is discrete or nominal} \\ dif_a(x,y) & a \text{ is continuous} \end{cases} \quad (8)$$

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

$$pm_a(x,y) = \begin{cases} vdm(x_a, y_a) \\ sfm(x_a, y_a) \\ mrm(x_a, y_a) \end{cases} \quad (9)$$

depending on which type of difference metrics is used. For continuous features:

$$dif_a(x,y) = \frac{|x - y|}{4\sigma_a} \quad (10)$$

where σ_a is the standard deviation for the attribute a. Scaling components of Euclidean distance by standard deviation helps to reduce the influence of outliers, but other normalization techniques could also be used.

The main problem of using *HM* metric is the scaling of different components, because contributions from different types of features may not be combined in an optimal way. In [15] authors use three different forms of VDM distance with different normalizations, leaving the decision which one should be chosen to the designer of the system. Because good theoretical arguments are lacking a question which distance function is most reliable and which normalization to use should be answered by empirical calculations.

2.2.2 Probability estimation by discretization and interpolation

The problem of normalization does not occur if all components of the distance functions are based on the probability difference metrics with posterior probabilities estimated for both discrete and continues features. However, in such a case the estimation of probability density for continuous features becomes a problem. Wilson and Martinez [15] advocate here Discretized Value Difference Metrics (DVDM). Discretization is used for continuous attributes to calculate posterior probabilities. In the simplest case equal width discretization method is used, described by the following formula:

$$disc_a(x) = \begin{cases} \left\lfloor \frac{x - \min_a}{w_a} \right\rfloor + 1 & \text{if } x \text{ is continuous} \\ x & \text{if } x \text{ is discrete} \end{cases} \quad (11)$$

min_a is the minimum of the attribute a and w_a is a parameter describing the number of bins. Step-wise discretization is rather inaccurate. In the Interpolated Value Difference Metrics (IVDM) linear interpolation is used for continuous values to improve the calculation of posterior probability:

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

where 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 $mid_{au} \le x \le mid_{a,u+1}$ are the middles of discretized ranges u and $u+1$. In this case the main problem is how to estimate the distribution of the posterior probabilities accurately. For discrete or symbolic features it can be easily computed using frequencies (eq. 3) but for continuous features it will not work. Better algorithms used for determining posterior probabilities may lead to better overall results. Several methods for estimation of the probability densities are presented below.

2.2.3 Gaussian smoothing estimation

A very popular method for density estimation is based on Gaussian smoothing. The posterior probability is calculated as:

$$p(C_i \mid x_a) = norm\left(\sum_{j=1}^{m_i} \exp\left(-x_{aj}^2 / 2\sigma^2\right)\right) \quad (13)$$

where m_i is the number of all vectors from the a given class i, σ is the dispersion of the Gaussian function and *norm* is the normalization factor calculated from:

$$norm = 1 \Big/ \sum_{a=1}^{n}\left(\sum_{j=1}^{m_i} \exp\left(-x_{aj}^2 / 2\sigma^2\right)\right) \quad (14)$$

2.2.4 Estimation with rectangle Parzen window

A very simple and very fast technique for estimating probability we call the Local Probability Matrix (LPM) like LVDM, LSFM and LMRM. This method is based on local calculation of data density surrounding the point of interest. Probability is calculated by the equation (3), with values of N_{xai} taken as the number of vectors from class i with the attribute value falling into the range

$\left[x_a - \frac{width_a}{2}, x_a + \frac{width_a}{2}\right]$, where N_{xa} is the same as N_{xai} but calculated for all classes. $Width_a$ is a user-defined parameter determining the range of a window for attribute a. This works well if the number of points is sufficiently large.

2.2.5 Motion Parzen window probability estimation

An obvious generalization of the LPM approach to density estimation is based on Parzen Windows. A rectangular window is moved by a small step through the whole range of attribute a and the probability is calculated as a mean value of all probabilities for which x occured in the window:

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

where Z is the number of windows, $Z = width_a / step_a$, b is the index of the first window where x occurs, $N_{iz}(x_a)$ number of data points in z-th window which class is i, $N_z(x_a)$ is the same as $N_{iz}(x_a)$ but summed over all classes, $width_a$ is the window width for attribute a, and $step_a$ is the size of window shifts.

3. EXPERIMENTS AND RESULTS

Experiments were performed in two steps. In the first step quality of probability estimation, and influence of the estimation parameters, was verified. For this purpose two artificial datasets were generated. The first dataset was two-dimensional, three class problem where the distribution of cases for each class was sampled from a normal distribution, and the second dataset, also two dimensional three class problem, with data points sampled from a uniform distribution. In both datasets classes were overlapping.

In the second step P-rules for several datasets taken from the UCI repository [17] were generated. The datasets selected had different type of attributes: continuous, discrete and nominal. All tasks were carried out using the Similarity Based P-rule generation System (SBPS) of programs. It allows defining different types of distance functions for different attributes that are combined into a single distance measure. SBPS has several build-in algorithms for prototype selection and optimization that are used to search for the simplest P-rules. To make results obtained for each problem comparable only one method was used in all experiments below, a simple Fuzzy C-means algorithm for prototype selection, and the LVQ algorithm for their optimization [1]. Thus the results do not represent the highest accuracy that may be achieved using SBPS.

3.1 Artificial datasets

Artificial datasets were created to verify quality of probability estimation and to evaluate the influence of discretization and smoothing parameters on the final classification results. For the first artificial dataset with vectors in each class taken from normal distributions optimal Bayesian borders 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 selected and to reduce influence of randomness and verify generalization ten fold cross validation tests were performed. Results presented in Tab. 1 show balanced accuracy for each method.

3.2 Real datasets

The three heterogeneous distance functions (Sec. 2.2) have been tested also on real datasets using different probability density estimators. A number of datasets with different types of attributes were selected from the UCI repository [17]: Flag, Glass, Iris, Pima Indians diabetes, Promoters, the Wisconsin Brest Cancer (WBC), and the Lancet data (obtained from the authors of paper [18]), but due to the lack of space not all results can be presented here. Because our aim was to obtain maximum balanced accuracy for all this distance measures we have used the algorithm for constructive rule generation to control P-rules capabilities.

	Parameters		Gauss distributed classes dataset	VDM	SFM	MRM	Uniform distributed classes dataset	VDM	SFM	MRM
	Euclidean			96,83	96,83	96,83		90,00	90,00	90,00
Local Parzen Window	W0.05			96,50	92,67	96,67		85,17	73,67	86,00
	W0.1			96,50	92,67	96,67		85,17	73,67	86,00
	W0.3			96,50	92,67	96,67		85,17	73,67	86,00
	W0.5			96,50	92,67	96,67		85,17	73,67	86,00
	W0.7			96,50	92,67	96,67		85,17	73,67	86,00
Moveing Parzen Window	W0.05	St0.01		96,50	92,67	96,67		85,17	73,67	86,00
	W0.1	St0.01		96,50	92,67	96,67		85,17	73,67	86,00
	W0.3	St0.01		96,50	92,67	96,67		85,17	73,67	86,00
	W0.5	St0.01		96,50	92,67	96,67		85,17	73,67	86,00
	W0.7	St0.01		96,50	92,67	96,67		85,17	73,67	86,00
	W0.05	St0.05		51,17	47,17	46,67		61,83	51,17	51,50
	W0.1	St0.05		96,50	92,67	96,67		85,17	73,67	86,00
	W0.3	St0.05		96,50	92,67	96,67		85,17	73,67	86,00
	W0.5	St0.05		96,50	92,67	96,67		85,17	73,67	86,00
	W0.7	St0.05		96,50	92,67	96,67		85,17	73,67	86,00
	W0.05	St0.1		33,33	33,33	33,33		33,33	33,33	33,33
	W0.1	St0.1		87,50	87,50	87,50		66,00	66,00	66,00
	W0.3	St0.1		96,50	92,67	96,67		85,17	73,67	86,00
	W0.5	St0.1		96,50	92,67	96,67		85,17	73,67	86,00
	W0.7	St0.1		96,50	92,67	96,67		85,17	73,67	86,00
Gauss	Si0.2			96,50	92,67	96,67		85,17	73,67	86,00
	Si0.5			96,00	91,33	96,83		86,33	73,67	86,33
	Si0.7			95,00	91,33	96,50		86,33	81,67	87,50
Discretization	Discret_10			96,50	92,67	96,67		85,17	73,67	86,00
	Discret_2			95,17	89,33	95,17		86,17	81,83	83,83
	Discret_4			95,67	95,17	95,67		89,33	81,83	89,33
	Discret_6			95,83	95,83	95,83		89,67	74,50	89,33
	Discret_8			96,33	85,50	96,00		86,67	70,33	89,33
Interpolation	Discret_10			95,17	92,50	95,67		86,83	74,83	87,83
	Discret_2			96,17	92,50	96,17		89,00	84,50	88,17
	Discret_4			96,50	95,17	96,17		89,17	83,83	89,00
	Discret_6			96,17	94,83	96,33		85,83	77,67	86,67
	Discret_8			95,17	91,33	95,67		87,33	82,00	87,33

Table 1. Results obtained on artificial datasets for different probability density estimation algorithms and different probability metric; in the top row results with Euclidean distances are given.

The constructive algorithm used in our tests 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 the accuracies for each class. In all cases constructive algorithm was stopped after the maximum of 10 iterations, providing no more than 10 prototypes per class, in form of a simple and understandable set of rules.

All continuous features in all datasets were initially standardized and then normalized to the interval [0,1]. The results obtained from this approach – highest balanced

accuracy for each combination of parameters – are presented in Table 2. Results for a few interesting datasets containing only discrete features and presented in Table 3.

4. DISCUSSION AND CONCLUSIONS

The "no free lunch" theorem [1]-[2] says that no method may beat all the others on all data and the results obtained here confirm it. For the artificial data the Gaussian MRM distance function is usually better than all other methods, achieving for the first dataset results identical with the Euclidean distance (which obviously is optimal for this data set), proving that probabilistic functions may be competitive even on purely numerical data.

Moreover, for the second artificial data set accuracy obtained using all VDM, SFM and MRM metrics was significantly lower then obtained with Euclidean distance. It was predictable that this algorithm should give very good results because for such data distribution with quite high density of points Gaussian smoothing generates the best approximation to the estimated probability. Low accuracy of the SFM measure, which on the real datasets gives quite good results, is also interesting and seems to confirm previous observations [16]. Optimization of parameter values for density estimation has a very significant influence on performance.

		flag						glass						pima-indians-diabetes					
		VDM		SFM		MRM		VDM		SFM		MRM		VDM		SFM		MRM	
Heterogeneous		23,51	11	23,43	16	24,58	16	47,83	10	47,83	10	47,83	10	**73,74**	7	**73,74**	7	73,74	7
Local Estimation																			
W0.05		**34,08**	10	22,56	14	23,91	12	**53,87**	14	30,01	11	38,11	14	**71,27**	9	70,41	5	70,25	5
W0.1		**33,69**	10	23,26	16	24,58	15	44,58	13	40,31	11	**53,57**	14	**72,50**	10	71,96	10	69,95	10
W0.3		**34,18**	13	24,24	14	27,94	14	42,21	12	31,89	14	**54,65**	14	71,91	9	**71,91**	10	69,09	10
W0.5		**37,73**	14	24,01	14	35,48	16	**45,80**	14	39,38	14	34,29	10	71,46	9	**71,70**	9	67,96	9
W0.7		**34,16**	11	29,65	15	27,59	11	**42,39**	10	39,92	11	27,04	7	71,68	10	**71,73**	10	63,16	9
Parzen																			
W0.05	St0.01	**34,08**	10	22,56	14	23,91	12	**53,87**	14	30,01	11	38,11	14	**71,27**	9	70,41	5	70,25	5
W0.1	St0.01	**33,69**	10	23,26	16	24,58	15	44,58	13	40,31	11	**53,57**	14	**72,50**	10	71,96	10	69,95	10
W0.3	St0.01	**34,18**	13	24,24	14	27,94	14	42,21	12	31,89	14	**54,65**	14	71,91	9	**71,91**	10	69,09	10
W0.5	St0.01	**37,73**	14	24,01	14	35,48	16	**45,80**	14	39,38	14	34,29	10	71,46	9	**71,70**	9	67,96	9
W0.05	St0.05	**34,08**	10	22,56	14	23,91	12	**53,87**	14	30,01	11	38,11	14	**71,27**	9	70,41	5	70,25	5
W0.1	St0.05	**33,69**	10	23,26	16	24,58	15	44,58	13	40,31	11	**53,57**	14	**72,50**	10	71,96	10	69,95	10
W0.3	St0.05	**34,18**	13	24,24	14	27,94	14	42,21	12	31,89	14	**54,65**	14	71,91	9	**71,91**	10	69,09	10
W0.5	St0.05	**37,73**	14	24,01	14	35,48	16	**45,80**	14	39,38	14	34,29	10	71,46	9	**71,70**	9	67,96	9
W0.05	St0.1	**34,08**	10	22,56	14	23,91	12	**53,87**	14	30,01	11	38,11	14	**71,27**	9	70,40	5	70,25	5
W0.1	St0.1	**33,69**	10	23,26	16	24,58	15	44,58	13	40,31	11	**53,57**	14	**72,50**	10	71,96	10	69,95	10
W0.3	St0.1	**34,18**	13	24,24	14	27,94	14	42,21	12	31,89	14	**54,65**	14	71,91	9	**71,91**	10	69,09	10
W0.5	St0.1	**37,73**	14	24,01	14	35,48	16	**45,80**	14	39,38	14	34,29	10	71,46	9	**71,70**	9	67,96	9
Gauss																			
Si0.05		26,38	9	25,75	16	**29,81**	16	48,11	13	44,70	13	**59,76**	13	71,83	9	**73,12**	10	69,00	10
Si0.1		31,18	9	24,86	12	**35,42**	15	43,63	10	**58,25**	14	52,2	13	71,40	8	**72,29**	9	68,83	7
Si0.3		**36,83**	14	33,51	16	34,66	16	51,01	14	**60,33**	12	33,64	13	71,20	9	**71,51**	10	56,66	10
Si0.5		**36,41**	14	34,69	16	23,81	12	**56,75**	12	56,73	12	24,28	9	**71,40**	7	71,09	7	50,00	2
Si0.7		**34,50**	15	33,85	16	23,18	12	47,77	9	**55,85**	14	20,33	7	71,39	7	**71,47**	7	50,00	2
Discretization																			
Discret_1		**33,25**	16	18,92	13	24,33	12	**44,40**	14	30,06	8	38,95	14	**71,08**	10	70,98	10	68,37	10
Discret_2		**36,15**	14	24,03	14	31,8	15	43,69	11	45,45	12	**46,39**	13	**65,89**	8	64,96	4	65,38	9
Discret_4		31,47	10	27,51	13	**35,96**	16	**47,42**	14	38,01	13	44,37	9	**68,6**1	9	67,92	6	67,20	9
Discret_6		**30,26**	14	25,5	13	26,98	16	**49,58**	10	42,21	11	40,80	14	72,73	9	**73,85**	9	69,95	9
Discret_8		**32,06**	11	25,5	15	27,3	15	**53,74**	12	38,95	10	54,76	12	**74,11**	10	73,66	10	71,10	10
Interpolation																			
Discret_1		**33,42**	11	22,27	15	27,74	14	**47,48**	10	34,83	11	41,18	12	70,82	7	**71,22**	9	69,92	10
Discret_2		**32,17**	10	25,82	14	29,81	13	**59,04**	14	57,82	14	41,73	14	73,00	2	**73,07**	8	62,48	10
Discret_4		31,74	9	27,76	13	**36,69**	16	41,10	10	53,01	14	**56,34**	14	71,57	9	**71,87**	9	69,92	7
Discret_6		**33,75**	11	26,05	16	29,5	16	49,59	9	42,96	14	**54,71**	14	72,16	7	**72,60**	10	69,22	8
Discret_8		**32,40**	14	19,41	16	32,13	15	**48,49**	7	40,83	13	45,70	7	**72,11**	10	71,97	10	70,39	8

Table 2. Balanced accuracy and the total number of P-rules for different values of parameters of probability estimators and different probability metrics. *W* – window width, *St* – Window step, *Si* – sigma value, Discret – number of discretization intervals. Reference results obtained with heterogeneous distance functions are given at the top.

On the real datasets adjustment of parameters values is also an important problem with heterogeneous VDM functions. Choosing correct value is now much more important and selection of the best method is not so easy, sometimes even impossible. Highest accuracies, marked as bold in Tab 2 and 3, appear for different methods for each dataset. In general even for dataset with numerical values of the attributes, such as the Pima-indian-diabetes, results are not worse than for the Euclidean distance, while for datasets with many symbolic features they are significantly better. VDM metrics is rather stable for wide range of parameter values. MRM seems to give the best balanced accuracy in most cases (Tab. 3).

Results presented above unfortunately do not lead to any simple conclusions about what type of distance should be used or which parameter values are the best. If some values of estimation parameters are wrongly chosen (for example the window width is smaller than the Parzen step size) estimation of probabilities may become quite inaccurate, as it is shown in Tab. 1. Step size in Parzen Windows algorithms has little impact on results, therefore rather high value (0.1) may be taken. For Gaussian smoothing the middle ranges of analyzed parameter values are the best. Even the simplest discretized DVDM measure may give quite good results, which can be slightly improved by interpolation. In situations when vectors from different classes are very close density estimation become quite difficult and discretization may bring some benefits. This becomes especially important for datasets with small number of training vectors. Exploration of more advanced discretization algorithms seems worthwhile.

	Lancet		promoters		promoters_4		wbc	
VDM	90.33	5	**90.83**	4	92.67	8	97.68	7
MRM	**90.77**	5	89.67	8	**94.50**	8	**97.80**	9
SFM	90.22	4	89.67	4	92.67	8	97.59	5

Table 3. Balanced accuracy and the number of prototypes for datasets with discrete or nominal features only, for different probability metrices.

Although comparison of P-rule accuracy with other systems is not the main subject of this paper it is worthwhile to mention that some results are much better than those obtained from decision trees or neurofuzzy systems, providing a simple description of this data. For example, on the Promoters data [17] four P-rules with VDM lead to a balanced accuracy exceeding 90%, and 8 rules with 4 best features and MRM metrics give 94.5%, while optimal results from decision trees (C4.5, SSV) on data with A, C, T, G features replaced by 1-4 numbers lead to 9 rules with balanced accuracy below 75%.

P-rules in combination with probabilistic metrics and discretization techniques seem to be a powerful alternative to a better established decision trees and neurofuzzy methods, especially for nominal features and certainly deserve more attention.

REFERENCES

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, New York: John Wiley & Sons, 2nd ed, 2001.

[2] A. Webb, Statistical Pattern Recognition. John Wiley and Sons 2002.

[3] W. Duch, R. Setiono, J.M. Zurada, Computational intelligence methods for understanding of data. Proc. of the IEEE ,Vol. 92, No. 5, 771- 805, 2004

[4] J.R. Quinlan, C4.5: Programs for machine learning. San Mateo, Morgan Kaufman 1993

[5] L. Breiman, J.H. Friedman, R.A. Oslhen, and C.J. Stone, Classification and Regression Trees. Belmont, CA: Wadsworth International Group, 1984

[6] K. Grąbczewski, W. Duch "The separability of split value criterion" 5'th Conference Neural Network and Soft Computing, Zakopane, Poland 2000

[7] N. Kasabov, Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering. Cambridge, MA: MIT Press, 1996.

[8] D. Nauck, F. Klawonn, and R. Kruse. Foundations of Neuro-Fuzzy Systems. Chichester: Wiley, 1997.

[9] S.K. Pal and S. Mitra, Neuro-Fuzzy Pattern Recognition. New York: J. Wiley, 1999.

[10] W. Duch, K. Grudziński "Prototype based rules - a new way to understand the data". Proc. of IJCNN 2001, Washington D.C. USA, pp. 1858-1863.

[11] W. Duch M. Blachnik "Fuzzy rule-based system derived from similarity to prototypes", Springer Lecture Notes in Computer Science, vol. 3316, pp 912-917, 2004.

[12] W. Duch, Similarity based methods: a general framework for classification, approximation and association. Control and Cybernetics 29 (4), pp. 937-968, 2000.

[13] W. Duch, R. Adamczak, G.H.F. Diercksen, Classification, Association and Pattern Completion using Neural Similarity Based Methods. Applied Mathematics and Computer Science 10:4, pp. 101-120, 2000.

[14] W. Duch, R. Adamczak, G.H.F. Diercksen, Neural Networks from Similarity Based Perspective. In: New Frontiers in Computational Intelligence and its Applications. Ed. M. Mohammadian, IOS Press, Amsterdam 2000, pp. 93-108.

[15] D.R. Wilson, T.R. Martinez, "Improved Heterogeneous Distance Functions", J. of Artificial Intelligence Research 6, pp. 1-34, 1997.

[16] E. Blanzieri, F. Ricci "Probability Based Metrics for Nearest Neighbor Classification and Case Based Reasoning", Proc. 3rd Int. Conf. on Case-Based Reasoning, Munich, August 1999.

[17] C.J. Mertz and P.M. Murphy, UCI repository of machine learning databases, www.ics.uci.edu/pub/machine-learning-databases.

[18] A.J. Walker, S.S. Cross, R.F. Harrison, Visualization of biomedical datasets by use of growing cell structure networks: a novel diagnostic classification technique. Lancet Vol. 354, pp. 1518-1522, 1999.