Fuzzy rule-based systems derived from similarity to prototypes.

Włodzisław Duch

Department of Informatics, Nicholaus Copernicus University,
Grudziądzka 5, Toruń, Poland,
and School of Computer Engineering,
Nanyang Technological University, Singapore.
www.phys.uni.torun.pl/~duch

Marcin Blachnik

Division of Computer Methods,
Department of Electrotechnology,
The Silesian University of Technology,
ul. Krasińskiego 8, 40-019 Katowice, Poland.
marcinblachnik@poczta.onet.pl

Abstract— **Relations between similarity-based systems, evaluating similarity to some prototypes, and fuzzy rule-based systems, aggregating values of membership functions, are investigated. Similarity measures based on information theory and probabilistic distance functions lead to a new type of membership functions applicable to symbolic data. Fuzzy membership functions on the other hand lead to a new type of distance functions. Several such novel functions are presented. This approach opens new ways to generate fuzzy rules based either on individual features or on their combinations used to evaluate similarity. Transition from prototype-based rules using similarity and fuzzy rules is illustrated using artificial data in two dimensions. As an illustration of usefulness of prototype-based rules very simple rules are derived for leukemia gene expression data.**

I. Introduction

Relationships between fuzzy systems and similarity based systems seem to be largely unexplored. Investigation of these relationships is quite fruitful, leading to new methods in both fields. Fuzzy models usually start from membership functions (MFs) defining linguistic variables. While this may be useful if the application domain is close to human experience, or if the goal of modeling is to mimic human reasoning, in most applications MFs should be derived from data together with logical rules, optimizing the performance of the system. Some neurofuzzy systems can do this [1], [2], [3]. Such systems employ a network of nodes (called "neurons"), with each node implementing a separable function, that is calculating products of one-dimensional MFs. A fuzzy rule is an approximation to some prototype elementary decision making process. Training of neurofuzzy networks optimizes these decisions at three levels: membership functions, rules, and the aggregation of conditions that form the conclusion.

In the simplest probabilistic and RBF networks with Gaussian functions all training data are used as a reference, corresponding to a large number of strongly overlapping fuzzy rules. To be comprehensible prototype-based rules should use a minimal number of rules with little overlap. In both network and fuzzy approaches the form of the MFs is fixed, only their parameters, and sometime their number (if constructive networks are used) are optimized. The system designer selects from a few types of elementary functions that determine the type of decision borders that the system will provide, and thus

determine the complexity of the final system [1], [2], [4].

In similarity-based methods [5] the training set provides the reference examples and the similarity of (or distance to) new cases is used for evaluation. These methods may be used in more general situation than neural or fuzzy methods because they do not require numerical representation of inputs with fixed number of components. Similarity between complex objects (such as proteins, text documents, software codes or organization of financial institutions) may be determined using various quantitative and qualitative procedures. If a small number of prototype reference cases that allow for reliable classification using a few features in the distance function can be found they may be used to formulate comprehensible similarity-based rules. Such methods may also be presented in a network form [6], providing an interesting alternative to neurofuzzy approach.

How do the fuzzy rules that provide the same decision borders as those resulting from similarity to prototypes look like? In the simplest case using Euclidean distance metric and a single prototype per class hyperplane decision borders are created, leading to a linear classification machine. How to create a set of fuzzy rules with identical decision borders to that of prototype-based system? What type of similarity measures correspond to the typical fuzzy functions and vice versa? Are prototype-based rules always equivalent to some sets of fuzzy rules? These questions are addressed below.

In the next section relation between similarity and fuzzy systems are presented, and probabilistic, data-dependent similarity measures leading to prototype rules that have no simple equivalents in fuzzy rules introduced. Some examples illustrating the relations between fuzzy rules and prototype-based systems are shown in the third section. Very simple prototype-based rules are found for leukemia gene expression data. The paper concludes with some general remarks on prototype-based systems.

II. Probability, similarity and fuzzy sets

Fuzzy set $\mathcal{F}$ is defined by the universe $\mathcal{X}$ and the membership functions $\chi_{\mathcal{F}}(X)$, specifying the degree to which elements of this universe belong to the set $\mathcal{F}$. This degree should not be interpreted as probability [7] and in fact at

least four major interpretations of the meaning of membership functions may be distinguished [8]. One natural interpretation is based on the degree to which all elements $X \in \mathcal{X}$ are similar to the typical (or $\chi_{\mathcal{F}}(X) \approx 1$) elements of $\mathcal{F}$. From this point of view fuzzy modeling seems to be a special case of similarity modeling, field that have not yet been fully developed. On the other hand fuzzy models are quite successful and may contribute to new similarity measures. Relations between fuzzy rules and similarity to prototypes are worth of exploration.

An analogy with human object recognition is quite fruitful. Perceiving and recognizing an object requires attention to be paid to its most characteristic features. First feature values $\mathbf{X}_i$ are measured by the senses with relatively high precision (for example, physical parameters of sound), and then primary and secondary sensory cortex filters these measured values creating higher-order receptive fields (such as phoneme perception), integrating spatio-temporal patterns. In fuzzy modeling each feature $\mathbf{X}_i$ of an object $\mathbf{X}$ is filtered through a large receptive field $\mathcal{F}_{ij}$, defined by a membership function $\mu_{F_j}(X_i)$. Simple MFs, such as triangular, trapezoidal or Gaussian, are used to model the degree to which some value X_i belongs to the receptive field $\mathcal{F}_{ij}$. Comparing to the sophisticated processing of sensory signals by the brain this is a very crude approach in which larger receptive fields are obtained directly from individual features using membership functions, instead of non-linear combinations of several features. Brain-like information processing may of course be more accurately modeled using hierarchical fuzzy systems.

Several higher-order features (in the simplest case defined by their membership values) are aggregated together to evaluate similarity to known objects or memorize new objects. A general form of prepositional classification rule may be formulated in the following form:

$$\text{IF } \mathbf{X} \sim \mathbf{O}^{(i)} \text{ THEN Class membership is } \chi_{C_j}(\mathbf{X}) \quad (1)$$

The operator $\sim$ represents similarity determining the membership values $\chi_{C_j}(\mathbf{X})$ in classes C_j to which objects $\mathbf{O}^{(i)}$ typical for this class belong, $\chi_{C_j}(\mathbf{O}^{(i)}) \approx 1$. The similarity operator may have different interpretations; in fuzzy logic it means "belongs to" in the fuzzy sense. Rules partition the feature space into areas where $\chi_{C_i}(\mathbf{X}) > \chi_{C_j}(\mathbf{X})$, that is similarity to objects from different classes dominates. In fuzzy logic the overall similarity is calculated as a T-norm (frequently a product) of membership functions $\mu_{ji}(X_j)$ for relevant features [2], [4], [7].

The crisp form of logical rules (L-rules) is obtained when subsets of nominal values are used, or for continuous features rectangular MFs are used, $\mu_{ji}(X_j) = 1$ if $X_j \in [X_{ji-}, X_{ji+}]$ and zero outside. For most T-norms the feature space is then partitioned into hyperrectangles. Fuzzy rules (F-rules) with popular triangular or Gaussian MFs provide more complex decision borders, depending on the T-norm used. An alternative way to partition the feature space and classify the data is to use a set of prototype-based rules (P-rules) defined by minimal distance:

IF $P = \arg\min_{\mathbf{P}'} D(\mathbf{X}, \mathbf{P}')$ THAN Class($\mathbf{X}$) = Class($\mathbf{P}$),

where $D(\mathbf{X}, \mathbf{P})$ is a dissimilarity function (usually a distance function). For many distance functions decisions borders have polyhedral shapes, providing piecewise-linear partitioning of the feature space.

In contrast to pattern recognition methods such as the k-nearest neighbor method (k-NN) that use many reference vectors, the goal here is to find a small number of prototypes and simple similarity functions that can give accurate classification and understanding of the problem. Similarity functions based on Minkovsky's distance are very useful:

$$D(\mathbf{X}, \mathbf{P})^\alpha = \sum_{i=1}^{N} W_i |X_i - P_i|^\alpha \quad (2)$$

where W_i are feature scaling factors, calculated by the standardization procedure, or treated as adaptive parameters that are optimized to achieve the best classification results. For large exponents α contours of constant Minkovsky distance become rectangular. In the limit of the infinite exponent values Chebyshev (or L_∞ norm) distance function is obtained, $D_\infty(\mathbf{X}, \mathbf{P}) = \max_i |X_i - P_i|$, with rectangular contours of constant values. Introducing thresholds d_P, rules of the form:

IF $D_\infty(\mathbf{X}, \mathbf{P}) \leq d_P$ THEN C,

are equivalent to conjunctive crisp rules:

IF $X_1 \in [P_1 - d_{P1}/W_1, P_1 - d_{P1}/W_1] \wedge$
$... \wedge [P_k - d_{Pk}/W_k, P_k - d_{Pk}/W_k]$ Then C

These rules may not cover the whole feature space, while minimal distance rules always partition the whole space. Although systematic relations between various similarity functions and membership functions and the S and T-norms in fuzzy logic remain to be investigated some remarks are given below.

Any T-norm, for example a product or a minimum of the membership functions $\mu(X_i - P_i)$ centered at P_i (triangular, Gaussian and other MFs have additional parameters besides the center), may always be used as a similarity function. For example, using the product norm $S(\mathbf{X}, \mathbf{P}) = \prod_{i=1} \mu(X_i - P_i)$. Similarity functions may be related to distance functions by many transformations, but here only exponential transformations are considered, $S(\mathbf{X}, \mathbf{P}) = \exp(-D(\mathbf{X}, \mathbf{P}))$. Additive distance functions are then converted to the multiplicative similarity factors (membership functions). For example, Euclidean distance function $D_2(\mathbf{X}, \mathbf{P})^2 = \sum_i W_i (X_i - P_i)^2$ is equivalent to a multivariate Gaussian similarity function $S_2(\mathbf{X}, \mathbf{P}) = \exp(-||\mathbf{X} - \mathbf{P}||^2)$ centered at $\mathbf{P}$ with ellipsoidal contours of constant values $||\mathbf{X} - \mathbf{P}|| =$const, equal

to the product of univariate Gaussian membership functions $S_2(\mathbf{X}, \mathbf{P}) = \prod_i G(X_i, P_i)$.

Using exponential transformation fuzzy rules (F-rules) with product norms may always be replaced by prototype-based rules (P-rules) with appropriate similarity functions. On the other hand all additive distance functions may be replaced by product T-norms with membership functions given by exponential one-dimensional distance factors. For example, the Manhattan distance function $D_1(\mathbf{X}, \mathbf{P}) = \sum_{i=1} |X_i - P_i|$ is equivalent to a product of $\exp(-|X_i - P_i|)$ membership functions. Many other distance measures are useful (see [9], such as the Canberra distance:

$$D_{Ca}(\mathbf{X}, \mathbf{Y}) \sum_{i=1} |X_i - Y_i|/|X_i + Y_i|, \qquad (3)$$

used in the next section. These equivalences lead to new types of membership functions that have not been previously considered. For non-additive distance functions such simple equivalence may not exist.

More general form of rules are obtained if more than one prototype is used in the rule condition: IF among k most similar prototypes P_i class C is dominating than $C(\mathbf{X}) = C$. Such rules should be useful in approximation problems, but for classification they are rather difficult to understand and require more prototypes (at least k) per class. Oblique distribution of data may require linear combination, or non-linear transformation, of input features [9]. The meaning of rules build with such features may be difficult to comprehend. Convex, polyhedral shapes obtained from a union of halfspaces defined by hyperplanes also do not lead to comprehensible rules.

An interesting group of distance measures is based on the Value Distance Metrics (VDM) [9], [10]. A value difference for feature X_j in a K-class problem is defined as:

$$d_V(X_j, Y_j)^q = \sum_{i=1}^{K} |p(C_i|X_j) - p(C_i|Y_j)|^q \qquad (4)$$

where $p(C_i|X_j) = N_i(X_j)/N(X_j)$ is the number of times $N_i(X_j)$ the value X_j of feature j occurred in vectors belonging to class C_i, divided by $N(\mathbf{X}_j)$, the number of times the value $\mathbf{X}_j$ occurred for any class. The distance between two vectors $\mathbf{X}, \mathbf{Y}$ with discrete (nominal, symbolic) elements is computed as a sum of value differences $D_V(\mathbf{X}, \mathbf{Y})^q = \sum_{j=1}^{N} d_V(\mathbf{X}_j, \mathbf{Y}_j)^q$. Distance is defined here via a data-dependent matrix with the number of rows equal to the number of classes and the number of columns equal to the number of all attribute values. Since VDM is additive P-rules may be replaced by F-rules with membership functions $\mu_{VDM}(X_i - P_i) = \exp -d_V(\mathbf{X}_i, \mathbf{P}_i)$ obtained by exponential transformation. This opens the possibility for principled derivation of MFs from symbolic-valued data.

The probabilities $p(C_i|X_j)$ needed for VDM calculation may be replaced by mutual information between the value of a feature and the class label. For continuous inputs probabilities are computed either by discretization (Discrete Value Difference Metric, DVDM), or via interpolation (Interpolated Value Difference Metric, IVDM) (see [9], [10] where other types of VDM functions are also presented). P-rules based on VDM distances in continuous case may still be replaced by F-rules by creating MFs that approximate their decision borders. Distance functions will be converted to similarity functions and replaced by products of MFs (only product T-norm is used in this paper). VDM distance functions are useful especially for symbolic features, where typical distance functions may not work so well.

III. PEDAGOGICAL ILLUSTRATION

Two-dimensional models will be used first to visualize decision borders in order to understand P-rule systems. P-rules may be used for symbolic data, including sequential data and data with complex structure, and it would be fairly easy to generate rules for a real-world data. However, it is quite important to understand first how this approach works in simple cases. Therefore in Fig. 1 three clusters, each with 50 vectors generated from a Gaussian distribution, were used. Prototypes for each cluster have been selected; for the 3-class problem one per class, although this could also represent two-class problem with two prototypes assigned for one of the classes.

Similarity functions corresponding to the Minkowski's distance measures $D(\mathbf{X}, \mathbf{P})^\alpha$, for different α, after exponential transformation become products of membership functions centered on selected prototype. Contours of constant distance are displayed, as well as the decision borders partitioning the feature space into areas where the distance to one of the prototypes is closer than to the others. P-rules that are based on thresholds assign to each prototype only those vectors that are inside some contour and thus may leave areas of the feature space that are far from prototypes not covered by any rule. Classification probabilities may be parameterized for each rule using these thresholds. If the winning prototype (most similar) is selected independently of the threshold, partitioning the whole feature space into areas assigned to each prototype. In either case the shape of decision areas is clearly displayed. Such analysis is unfortunately rarely done for neural or neurofuzzy systems, making it difficult to understand what these systems really do and how do their decision borders look like.

For small α (Fig. 1, upper row) MFs have exponential character with sharp peaks, and piece-wise linear decision borders composed of several segments are created. Euclidean distance (or more precisely, squares of Euclidean components) lead to Gaussian membership functions (Fig. 1, middle row) giving linear decision borders. Increasing α leads to more steep MFs (1, bottom row), with $\alpha \to \infty$ giving rectangular contours for crisp logic.

The number of unique membership functions for each dimension is equal to the number of prototypes. Positions of the prototypes are reflected in the centers of the MFs. The width of these functions result from scaling factors W_i that are used for each feature. These factors may be used as adaptive
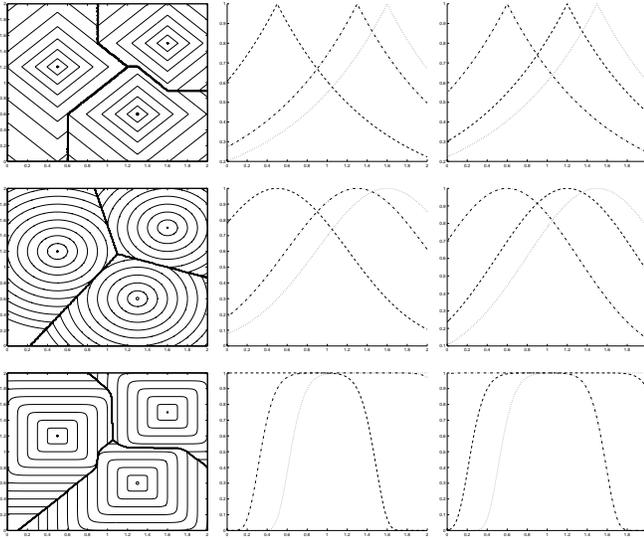
Fig. 1. Contours and decision borders for a 3-class problem using Minkowski distance function with α=1, 2, 10, and the MFs for the two dimensions derived from these distances.
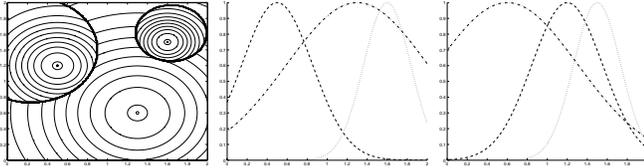

Fig. 2. Contours and decision borders for a 3-class problem using weighted Minkowski distance function with α=2 and $W_i = 1, 2, 3$, and the MFs derived from these distances.
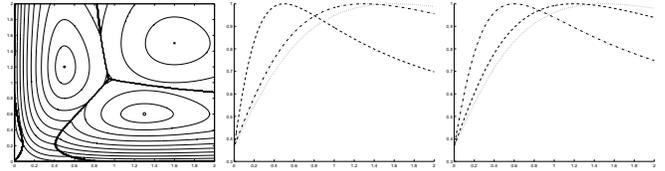

Fig. 3. Contours of a 3 class problem for the square of Canberra distance function and the MFs derived from these distances.

interval and behave like $-\ln(1 + (x - x_m)/\Delta)$ for $x \in [x_m - \Delta x, x_m]$ and $-\ln(1 - (x - x_m)/\Delta)$ for $x \in [x_m, x_m - \Delta x]$. Thus fuzzy logic rules with triangular MFs are equivalent to P-rule system with prototypes centered at the maxima of the products of MFs and $D_3(\mathcal{X} - \mathcal{X}_m; \Delta\mathbf{X})$ distance function (sum of $d_3(x - x_m; \Delta x)$ components). Complex neurofuzzy systems are frequently constructed [1], [2] without checking what type of decision borders such systems provide. Obviously understanding of data without such analysis, just by inspection of MFs and rules, is difficult.

To verify this equivalence in practice 20 points for each of the 3 Gaussian distributions centered at the prototypes shown in Fig. 1 has been generated, and the Adaptive Neuro-Fuzzy Inference System [11] was used to model this data using triangular membership functions. In Fig. 4 decision borders and MFs generated in this way are presented, together with the equivalent distance functions.
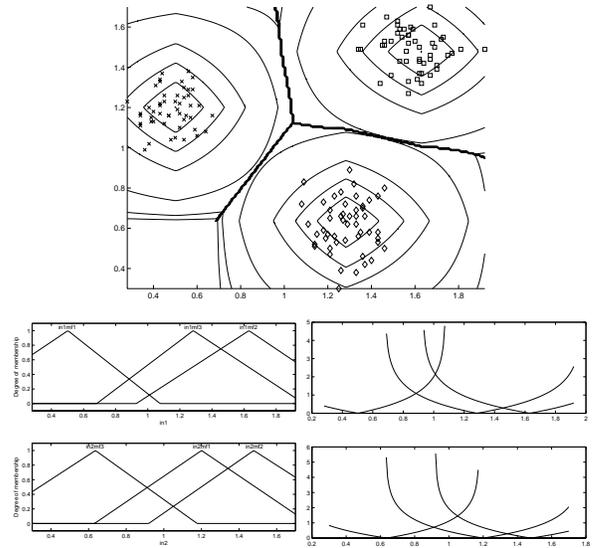

Fig. 4. Decision borders of fuzzy logic rules with triangular MFs, and corresponding prototypes and distance functions for each prototype.

parameters; small value of W_i corresponds to a very flat MF that covers the whole data, and thus provides a rule condition that is always true and therefore may be deleted. Changes of decision borders that result from scaling are quite dramatic, as is evident in Fig. 2. Even for Eculidean distance functions decision borders become non-linear. They are recreated by fuzzy rules with products of MFs if in the defuzzification process the rule with the highest activation is take as a winner.

Most of fuzzy logic systems use a few typical membership functions such as triangular, trapezoidal or Gaussian. Transformation of distance functions into similarity functions leads to new types of MFs of different shapes. For example, using Canberra distance function (popular in the nearest neighbor pattern recognition methods [5]) asymmetric MFs are created (Fig. 3).

The inverse transformation, going from membership functions to distance functions, is also interesting. Gaussian MFs lead to the square of Euclidean distance function, and all bell-shaped MFs will show similar behavior. Membership functions of triangular or trapezoidal shapes do not have an equivalent among the commonly used distance functions. Products of triangular functions $T_3(x - x_m; \Delta x)$ equal to zero outside $x_m \pm \Delta x$ interval and to 1 for $x = x_m$, correspond to sums of distance functions $d_3(x - x_m; \Delta x)$ that are infinite outside this

The same data has been used to find 3 prototypes using the LVQ approach. Fig. 5 presents decision borders for P-rules, together with the membership functions derived from the Euclidean distance measures. They are essentially identical with those found by the ANFIS system.

It is also interesting to see how VDM measure works for such continuous-valued data. The VDM distance measure
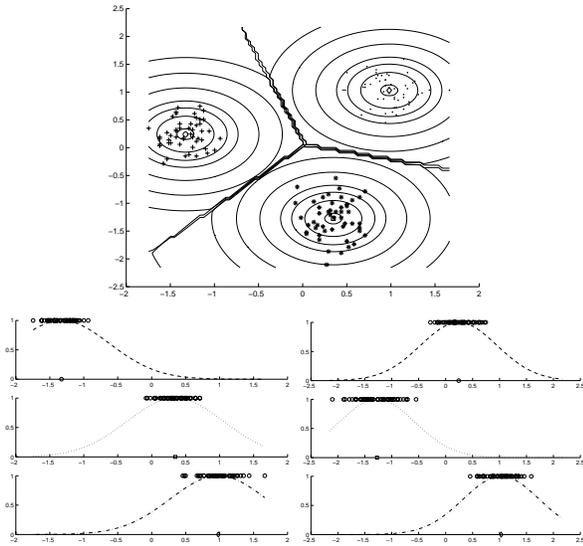
Fig. 5. 3 Gaussian distributions, with Euclidean distances from their centers, and the membership functions derived from these distances.

depends on details of discretization (DVDM method) or interpolation (IVDM method). Naive discretization with 9 bins of equal width leads to the decision contours and membership functions shown in Fig. 6. Numerical artifacts related to the estimation of probabilities required for calculation of VDM distances may manifest themselves in rather complex decision borders, leading to a poor approximation of optimal borders. With sufficient number of points to estimate probability distributions good results may be achieved even with naive discretization, but with small number of points better discretization methods should be used, and various smoothing algorithms or Parzen window techniques will be helpful. Nevertheless even with only 20 points and simple discretization the interpolated IVDM approach generated acceptable decision borders. MFs discovered by IVDM have smooth trapezoidal shapes that may be obtained by a difference of two error functions (erf) or logistic functions [13]. Thus one way of using VDM distances is to generate approximate shapes of membership functions and fit some analytical functional form that may be further optimized.

IV. RULES FOR LEUKEMIA GENE EXPRESSION DATA

The Leukemia gene expression data [12] has been analyzed looking for best prototype for each of the two classes, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Distinguishing between these two leukemias is considered to be challenging because their appearance is highly similar [12]. 7129 gene expression features from microarray experiments are given for each case. The "neighborhood analysis" method developed in the original paper finds 1100 genes that are correlated with ALL-AML class distinction. Prediction is based on a rather complex method that assigns weights to the most useful 50 genes and than calculates "prediction strengths" (PS) as a sum of votes with threshold 0.3. Training was done
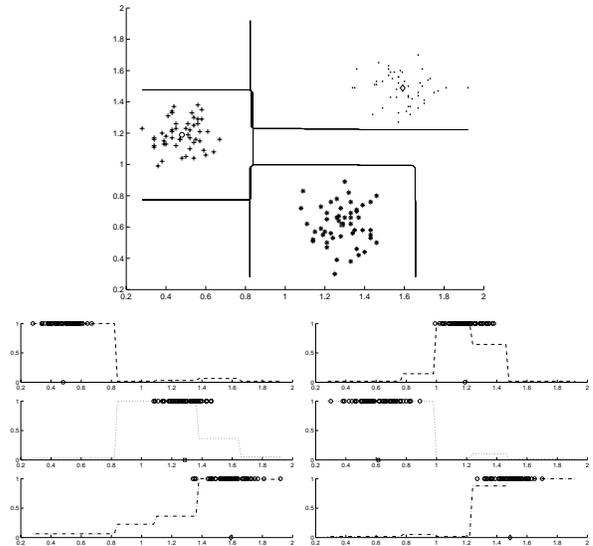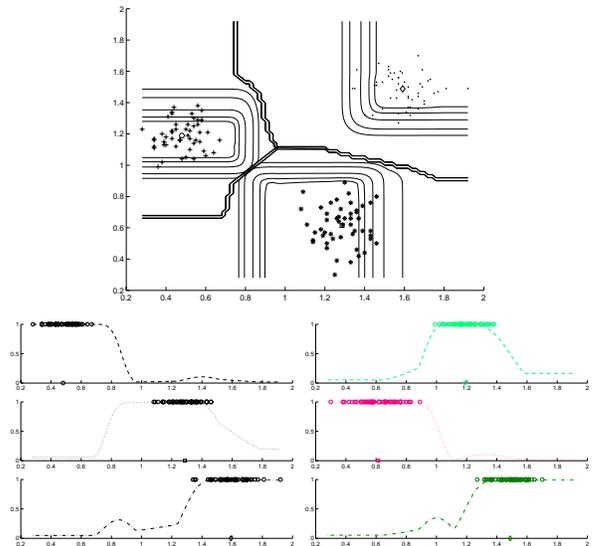

Fig. 6. Contours of a 3 class problem defined by Gaussian distributions. Contours of constant densities calculated with IVDM and DVDM distance functions and the appropriate membership functions derived from this distance

on 38 samples (27 ALL and 11 AML), using the leave-one-out method to set parameters, and testing was done on 34 samples (20 ALL and 14 AML). As a result 36 samples were correctly predicted and for two samples PS was below the critical 0.3 threshold. 29 of 34 test samples had large correct PS (median 0.77).

Using logical rules with a single condition based on threshold value for each of the 7129 features identified one feature for which no errors are made on the training data, two features that make a single error, and 14 features that make 2 errors. Since the best feature has quite narrow gap between the two classes 3 best features were taken to generate prototypes, optimizing them using the LVQ approach. Resulting P-rules use VDM metric and one prototype per class, making no errors on the training data, and only 3 errors on the test data.

Creating membership functions from data-dependent VDM

distance functions allows for a discovery of a "natural" MFs for problems with symbolic or nominal attributes. VDM measure can be used with continuous attributes after discretization of the training data features. Application of this measure to purely symbolic data, such as searching for promoters in the DNA strings [16], gave excellent result that may be reproduced using fuzzy rules derived from a few prototypes. Using 21 prototypes gives only 5 errors in the leave one out test. The 21 prototypes include 9 for the promoters (positive) class and 12 for the non-promoters (negative) class. Each of the prototype's positions was adjusted using LVQ algorithm. Lower number of prototypes leads to an increase in the number of missclassified vectors; two prototypes per class give already 13 errors.

V. CONCLUSIONS

Neurofuzzy systems generate fuzzy rules and optimize membership functions [1]. Selection of prototypes and features together with similarity measures offers new, so far unexplored alternative to neurofuzzy methods. Duality between similarity measures and membership functions allows for generation of prepositional rules based on individual membership functions. Fuzzy rules apply large receptive fields (membership funcitons) to these individual features, combining them later. P-rules in their natural form first create a combination of features (via similarity functions) and apply various membership functions to this combination. Neurofuzzy methods use input vectors defined in fixed-dimensional feature spaces. Similarity may be evaluated between objects with complex structures that are not easy to describe using a common sets of features. In particular the use of probabilistic, data dependent distance functions allows for definition of membership functions for symbolic data (such as the sequential DNA or protein strings) that may be difficult to derive in other way.

Experiments in cognitive psychology show that human categorization is based on exemplars and prototypes; logical rules are rarely used to define natural categories [14]. In the approach presented here similarity functions are used to model the importance of different features in evaluating similarity between the new case in relation to stored prototypes. Multiplicative similarity factors may easily be converted to additive distance factors and vice versa. Although similarity measures provide great flexibility in creating various decision borders this may turn to be a disadvantage if our primary goal is to understand the data (neurofuzzy approaches have of course the same problem). Optimized similarity measures may not agree with human intuition. In such cases larger number of prototype examples with simpler similarity measures may be a better solution. An interesting possibility is to use the prototype-based rules to describe exceptions in the crisp or fuzzy logic systems.

Rule-based classifiers are useful only if the rules they use are reliable, accurate, stable and sufficiently simple to be understood [13]. Prototype-based rules seem to be a useful addition to the traditional ways of data explanation based on crisp or fuzzy logical rules. They may be helpful in cases when logical rules are too complex or difficult to obtain. A

small number of prototype-based rules with specific similarity functions associated with each prototype may provide complex decision borders that are hard to approximate using logical systems. Such simple rules have been recently generated for medical datasets using heterogeneous decision tree [15]. A single rule P-rule for the Wisconsin breast cancer data classifying as malignant cancer all cases that are closer to prototype case (taken as one of the training cases) than a certain threshold achieves 97.4% accuracy (sensitivity 98.8% and specificity 96.8%). The accuracy in this case is at least as good as that of any alternative system tried on this data. Results obtained here for the artificial data, leukemia gene expression and DNA promoters data also confirm the usefulness of P-rules. Combining various feature selection and prototype selection methods with similarity functions leads to many interesting algorithms. Systematic investigation of various membership functions, T-norms and conorms, and their relation to distance functions is under way.

We are confident that algorithms for generation of P-rules will prove to be competitive to the existing neurofuzzy algorithms and will become an important addition to the methods of computational intelligence.

REFERENCES

[1] D. Nauck, F. Klawonn and R. Kruse. *Foundations on Neuro-Fuzzy Systems*. J. Wiley 1997.
[2] S.K. Pal, and S. Mitra *Neuro-Fuzzy Pattern Recognition*. J. Wiley 1999.
[3] W. Duch, G.H.F. Diercksen, "Feature Space Mapping as a universal adaptive system", Computer Physics Comm. vol. 87, pp. 341–371, 1995.
[4] V. Kecman, *Learning and Soft Computing*, MIT Press 2001.
[5] W. Duch, "Similarity based methods: a general framework for classification, approximation and association". Control and Cybernetics, vol. 29, pp. 937–968, 2000.
[6] W. Duch, R. Adamczak, and G.H.F. Diercksen, "Classification, Association and Pattern Completion using Neural Similarity Based Methods." Applied Math. and Computer Science, vol. 10, pp. 101–120, 2000.
[7] B. Kosko, *Neural Networks and Fuzzy Systems*. Prentice Hall 1992.
[8] T. Bilgiç and I.B. Türkşen, "Measurements of Membership Functions: Theoretical and Empirical Work," in D. Dubois and H. Prade (eds.), Vol. 1, *Fundamentals of Fuzzy Sets*. Kluwer, 2000, pp. 195–232.
[9] W. Duch and N. Jankowski, "New neural transfer functions." Neural Computing Surveys, vol. 2, pp. 639–658, 1999.
[10] D.R. Wilson, and T. R. Martinez, "Improved Heterogeneous Distance Functions". J. of Artificial Intelligence Research, vol. 6, pp. 1–34, 1997.
[11] J.-S. Roger Jang, C.-T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997.
[12] T.R. Golub et al. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring". Science, vol. 286, pp. 531–537, 1999.
[13] W. Duch, R. Adamczak, K. Grabczewski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules". IEEE Transactions on Neural Networks, vol. 12: 277–306, 2001.
[14] I. Roth, and V. Bruce, *Perception and Representation*. 2nd ed, Open University Press, London 1995.
[15] K. Grąbczewski and W. Duch, "Heterogenous forests of decision trees". Springer Lecture Notes in Comp. Sci. vol. 2415, pp. 504-509, 2002.
[16] M. Marczak, W. Duch, K. Grudziński and A. Naud. "Transformation Distances, Strings and Identification of DNA Promoters". In: Neural Networks and Soft Computing (eds. L. Rutkowski and J. Kacprzyk), Advances in Soft Computing, pp. 620-625, 2002.