Computational intelligence methods and data understanding

Włodzisław Duch[1] and Yoichi Hayashi[2]

[1] Department of Computer Methods, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
WWW: http://www.phys.uni.torun.pl/kmk
[2] Department of Computer Science, Meiji University,
Kawasaki, Japan.

Abstract. Experts in machine learning and fuzzy system frequently identify understanding the data with the use of logical rules. Reasons for inadequacy of crisp and fuzzy rule-based explanations are presented. An approach based on analysis of probabilities of classification $p(C_i|\mathbf{X};\rho)$ as a function of the size of the neighborhood ρ of the given case $\mathbf{X}$ is presented. Probabilities are evaluated using Monte Carlo sampling or – for some models – using analytical formulas. Coupled with topographically correct visualization of the data in this neighborhood this approach, applicable to any classifiers, gives in many cases a better evaluation of the new data than rule-based systems. Two real life examples of such interpretation are presented.

1 Introduction.

Classification and prediction are the two most common applications of computational intelligence (CI) methods, i.e. methods designed for solving problems that are effectively non-algorithmic. Although sometimes classification by a black box is sufficient – if it is reliable – domain experts use software to help them to understand (or explain) the data. The goal of data mining, or Knowledge Discovery in Databases (KDD), is more ambitious than understanding of new data. Knowledge discovery requires analysis of relations in the whole database, trying to create a symbolic domain theory, while understanding new data may be done locally, for example by pointing that it is typical for a known category. Most methods used for of classification are unable to do that. They rarely provide confidence bounds, so the user is not able to evaluate how reliable is the answer given by the system. Rule-based systems are not much better in this respect if continuos values of attributes are used. Near the rule decision borders reliability of the system should decrease and alternative diagnoses should appear in a smooth way. Statistical methods are oriented towards global characterization of numerical aspects of the data for a whole class of objects, providing means and variances instead of symbolic, understandable descriptions.

Explanation and reliability assessment are in some applications equally important as the accuracy of classification and prediction. Rule-based explanation has been a strong point of the machine learning (ML) inductive systems [1,2]. Although

ML algorithms work well with simple data building models and learning from structured data is still rather difficult. Inductive Logic Programming (IPL) is considered to be the best approach in such cases, but for large databases it may be too slow [3]. Molecular biology and genomics are two very important fields in which theory building, rather than classification models, seem to be necessary. Molecules are not described in a fixed-dimensional feature spaces. By analogy to the database terms one may call typical data analysis problems defined in fixed feature spaces “flat”, while those problems that require construction of a new set of attributes, relating the description to known domain theories, as “relational”.

Leaving relational problems aside for the flat data problems crisp logical rules provide adequate explanation only if all attributes have a few discrete values. Continuous values require discretization, or defining some “receptive fields” that may be used in rule conditions. Many books have been written about the advantages of fuzzy rules, soft versions of neural and other CI algorithms have been published (cf. [4] or [5]), but it still remains to be seen whether these algorithms will have a lasting impact on computational intelligence field. So far no extensive comparison of fuzzy classification methods with other methods on well-known data (such as the Stalog datasets [6]) have been made, and little has been done in applications to difficult, real-world problems such as handwritten character or speech recognition.

An important advantage of the fuzzy and other rule-based approaches to data analysis is their ability to provide explanation for the action of the system in form of understandable rules. The goal of the present paper is to analyze whether such explanations are really satisfactory and to propose a universal approach, applicable in connection with any classifier, that may provide an explanation. In the next section advantages and disadvantages of fuzzy explanations are presented. An alternative approach allowing for explanation of the data, based on calculation of probabilities as a function of the input uncertainty and visualization of the neighborhood of the case analyzed is presented in the third section. Two real-life examples are presented in the fourth section and a short discussion concludes this paper.

2 Fuzzy explanations

Acceptable explanations depend on the particular domain and the type of data. Fuzzy set theory [7] is a method to deal with imprecise information. Since our language is discrete while our senses provide continuous information some form of categorization or discretization must be done to allow for communication. Categorization can be modeled at a high level as clusterization of percepts. An explanation of the sensory data is usually given by an assignment of a percept to some category and the ability to quote the reasons for such assignment. Reasoning refers to a background knowledge, which in the case of our brains is very complex and highly specialized. For example, the ability to recognize faces depends on the activity of neocortex in the inferotemporal gyrus region of the right lobe, containing group of neurons that specialize in high-level object recognition. Explanations referring to features analyzed by lower levels of visual system functions, such as particular shapes, colors, movements, are not only unnatural, but ultimately impossible. All important work enabling the final classification is done at the pre-processing level, defining complex transformations that allow us to recognize a particular pattern as a face. Thus explanation should rely on sophisticated transformation of input features and rarely may be reduced to rules with conditions using large receptive fields defined by some membership functions applied to raw measurements. These receptive fields are usually called linguistic terms or linguistic variables [8].

Although crisp logical rules are the easiest to interpret there are several problems with explanations based on such rules [9]:

- Only one class is predicted as the winner.
- Some cases may be rejected as unclassified without justification.
- Crisp rules are not stable against small perturbations of input values.
- Non-gradient optimization methods should be used because the cost function based on the number of errors is discontinuous.

Fuzzy systems overcome these problems using continuous membership functions, but do they provide an adequate explanation of the data? Fuzzy membership functions allow for a natural description of many features, providing large receptive fields and thus simplifying the classification, so in principle they could provide good explanations. In practice fuzzy rules are rarely successful in explaining the data. For example, in the two recent books on neurofuzzy systems [4,5] many fuzzy rules for different problems have been derived, but would a domain expert find any of these rules useful? Problem of assessing the reliability of classification and of understanding the data are not addressed in these and other books on the subject. It is highly doubtful that fuzzy or even neurofuzzy systems would be able to compete with other classifiers on the ground of their high accuracy only. Reasons for this apparent failure of fuzzy adaptive systems to provide explanation of data are related to the problems of:

- setting up and understanding the linguistic variables;
- complexity of rules the system generate;
- large number of rules generated by some systems;
- problems of finding the simplest set of rules due to overparametrization of adaptive fuzzy systems;
- existence of equivalent sets of rules (or sets of similar accuracy).

Quoting a rule counts as an explanation only if the number of rules is relatively small and their accuracy is sufficiently high. Fuzzy and neurofuzzy systems apply membership functions to the input features, filtering information passed to the classifier by creating large receptive fields. The multilayer perceptron (MLP) networks [10] use their first hidden layer to define combinations of inputs that – treated as new input features – should transform the classification problem to a separable one. The output neurons provide separating hyperplanes in the transformed space. RBF networks [10] perform initial clusterization of the data using all features provided. None of these approaches is really directed at providing explanations at a higher level, with a transformed set of features. For that purpose some restrictions on meaningful transformations of input features should be set first, combinations of these features determined by an MLP-type layer, linguistic variables extracted by a constructive network with transfer function capable of local response (such as the bicentral functions [11]), and finally the outputs should be combined by a rule layer. The C-MLP2LN network [12,13] has an aggregation layer (A-layer), linguistic variables layer (L-layer), a rule layer (R-layer) and an output layer. This network was used previously in data mining tasks [13] but it is also well suited as a general architecture facilitating explanation of data.

Fuzzy rules involve a number of parameters determining position and shape of the membership functions. To avoid overparameterization systems based on fuzzy logic frequently use a fixed set of membership functions, with predetermined shapes. Defining linguistic variables in such context-independent way amounts in effect to a regular partitioning of the whole input space into convex regions. This approach suffers from the curse of dimensionality [10], since with k linguistic variables in d dimensions the number of possible input combinations is k^d. Fuzzy rules simply pick up those areas in the input space that contain vectors from a single class. Without the possibility to adapt membership functions to individual clusters in a single rule fuzzy rules do not allow for optimal description of these clusters. Much better results may be obtained with context-dependent linguistic variables created by some neurofuzzy systems [8,13].

Overparameterization of fuzzy systems and lack of regularization methods [10] commonly used in neural networks is responsible for high complexity of generated rules and difficulties of finding the simplest set of rules. Let us take as an example the Wisconsin breast cancer data [14] (available from the UCI machine learning repository [15]). This dataset describes properties of cancerogenic cells using 9 attributes with integer values in the range 1-10 (for example, feature f_2 is “clump thickness” and f_8 is “bland chromatin”). Fuzzy clustering methods or neurofuzzy systems do not produce membership functions with reasonable linguistic interpretation. The NEFCLASS neurofuzzy system [4], one of the best systems of its kind, combined with fuzzy clustering, produces after simplification 4 rules of the form:

$$\begin{aligned} \text{IF} \quad & f_1 = l \wedge f_2 = l \wedge f_3 = l \wedge f_4 = l \wedge f_4 = l \wedge f_5 = s \wedge \\ & f_6 = l \wedge f_7 = l \wedge f_8 = l \wedge f_9 = s \quad \text{THEN } \textit{malignant} \end{aligned}$$

Most fuzzy systems will produce much more complex sets of rules, but even the rule shown above given as an explanation of observed data, with specific membership functions for each attribute, may be too difficult for medical doctors to understand and use. The simplest rules for this dataset found for the malignant class using the C-MLP2LN are:

$$f_2 \geq 7 \vee f_7 \geq 6 \qquad (95.6\%)$$

These rules cover 215 malignant cases and 10 benign cases, achieving overall accuracy (including the ELSE condition) of 94.9%. They could not be found using decision trees (C4.5 [16] and SSV [17]) and other systems, but they are the easiest to understand to the medical experts.

Understanding the data may require the simplest description possible. This is evident in another medical example of the Ljubliana cancer data [15], containing 286 cases, of which 201 are no-recurrence-events (70.3%) and 85 are recurrence-events (29.7%). Each case is described by 9 attributes, with 2 to 13 different values each. A single logical rule for the recurrence-events has been found by C-MLP2LN system:

$$\text{involved nodes}=\neg[0,2] \wedge \text{Degree-malignant} = 3$$

With ELSE condition for the second class this rule gives over 77% accuracy in crossvalidation tests. The rule is easy to interpret: recurrence is expected if the number of involved nodes is greater than two and the cells are highly malignant. More complex and more accurate sets of rules may be found [13] for these datasets. Unfortunately the more complex the rules are the less likely it is that they represent interesting knowledge. This is true especially for smaller datasets, where accidental correlations easily occur and rules cover small number of cases. An adequate understanding of the data may require generation of several equivalent sets of rules.

In general the more complex the CI system is, the more difficult finding the simplest solution will be, since all adaptive methods require some form of minimization or iterative optimization procedures that are less reliable for larger number of parameters. Neurofuzzy systems frequently introduce additional fuzzification and defuzzification layers, increasing the number of parameters significantly. These layers are sometimes designed by hand so their parameters are fixed, but in effect suboptimal linguistic variables are used, the system is far from being automatic and the hidden layer performing classification has anyway more parameters, because each continuos input is replaced by several quasi-discrete inputs.

Suppose that in a 2-dimensional problem two classes are separated with a decision border (using the data rescaled to [0,1] interval) $x_1 + x_2 = 1$, i.e. the first class is in the lower corner of the unit square and the second outside. A single neuron will handle this problem with 2 weights and 1 bias. It would be difficult, if not impossible, to find a membership functions for the N variables – for example by evolving trapezoidal or other membership functions using genetic algorithms – that could separate approximate correctly a plane $x_1 + x_2 + \ldots x_n = 1$. Thus although in principle neural networks and fuzzy systems are equivalent [18] in practice there will be problems for which an approximation using any fuzzy system will converge very slowly. The same is true for the MLP networks which for some data distributions may suffer from slow convergence. The importance of a proper choice of the transfer functions in neural networks has been realized only quite recently [11].

Undoubtedly there are many industrial applications of fuzzy logic, although they seem to be restricted to control problems [19]. Since global control models are impossible to construct, the space of control parameters is divided into subregions where appropriate actions are fired by rules; in essence fuzzy control is based on local learning in multiple regions and the rules are simply identifying these regions. The best way to solve such problems would be to construct neural systems with weights $W(X)$ parametrized by the input vector X, i.e. to construct neural networks $NN(X;W(X))$ which represent a family of mappings that change in different regions of the input space. For example, in a larger modular network $W(X) = WG(X; P_k, \sigma_{x0})$ may be defined for a number of reference points P_k in the input space. Networks appropriate for this region of space may then perform specific actions. The architecture should be similar to the constructive RBF type of networks (cf. [10,20,21]) with appropriate action conditions performed by the output nodes.

3 Black box systems, assessment of reliability and data explanation

Neural systems usually do not provide explanation or even an assessment of reliability of their decisions. The same is true for other CI methods, such as clusterization, nearest neighbors or discriminant analysis (including Support Vector Machines) methods. Explanation or understanding of new data does not require full data mining. It may be sufficient to relate the new case to known cases, estimate probabilities for different classes, show how these probabilities depend on possible inaccuracies of measurements and visualize the data (this is called "an explanatory data analysis" in the statistical literature, cf. [22]). It is worthwhile to start searching for the simplest description of the data in form of crisp logical rules; it that fails more complex fuzzy rules should be produced and if that fails – i.e. if rules are not sufficiently accurate or they become too complex – more complex, black box classification models should be tried. In all cases given a new vector **X** for evaluation classification probabilities should be calculated and a diagnostic map should be provided, showing this case in relation to others.

Finding the simplest description of data involves the following steps:

- Create good features by combining the existing ones; use methods of feature selection, principal components (PCA), non-linear PCA [23], independent components (ICA), or strong regularization enforcing skeletonization of the input-hidden layer connections in neural network.
- Select linguistic variables [8] as subsets of discrete values or as interval $[X_k, X'_k]$.
- Extract logical rules from the data using neural, machine learning or statistical techniques.
- Optimize linguistic variables ($[X_k, X'_k]$ intervals) using the extracted rules and exploring the reliability/rejection rate tradeoff.
- Repeat the procedure until a stable set of rules is found.

These steps were described in details in [8,9,12,13]. The simplicity/accuracy tradeoff is explored first: rough description of the data is made using appropriate regularization parameters, with a hierarchy of more complex models following, until optimal complexity (from the generalization point of view) of the rule base is found. After selecting satisfactory level of data description another tradeoff is explored, between the reliability of predictions and the rejection rate of the classifier (number of vectors in the "don't know" class). Let $P(C_i, C_j)$ be the confusion matrix for a rule-based classifier M (the model M containing intervals $[X_k, X'_k]$ and subsets defining linguistic variables). The number of wrong predictions $\min_M \left[\sum_{i \neq j} P(C_i, C_j)\right]$

should be minimized simultaneously with maximization of the predictive power $\max_M [\mathrm{Tr}\, P(C_i, C_j)]$ of the classifier over all adaptive parameters of the model M. The following cost function $E(M)$ is minimized without constraints:

$$E(M) = \gamma \sum_{i \neq j} P(C_i, C_j) - \mathrm{Tr}\, P(C_i, C_j) \geq -n \tag{1}$$

where the parameter γ determines a tradeoff between reliability and rejection rate. Sets of rules of lower reliability make larger number of errors but have low rejection rate. These two tradeoffs: simplicity/accuracy and reliability/rejection rate cannot be determined automatically since they depend on the goals of the user.

Crisp rule based system does not allow to evaluate the probabilities of classification and, since $P(C_i, C_j)$ depends in a discontinuous way on the parameters of linguistic variables non-gradient minimization methods are required. Real input values $\mathbf{X}$ are obtained by measurements that are carried with finite precision. The brain uses not only large receptive fields for categorization, but also small receptive fields to extract feature values. Instead of a crisp number X a Gaussian distribution $G_X = G(Y; X, S_X)$ centered around X with dispersion S_X should be used. Probabilities $p(C_i|\mathbf{X}; M)$ may be computed for any classification model M by performing a Monte Carlo sampling from the joint Gaussian distribution for all continuous features $G_{\mathbf{X}} = G(\mathbf{Y}; \mathbf{X}, \mathbf{S}_X)$. Dispersions $\mathbf{S}_X = (s(X_1), s(X_2) \ldots s(X_N))$ define the volume of the input space around $\mathbf{X}$ that has an influence on computed probabilities. One way to “explore the neighborhood” of $\mathbf{X}$ and see the probabilities of alternative classes is to increase the fuzziness $\mathbf{S}_X$ defining $s(X_i) = (X_{i,max} - X_{i,min})\rho$, where the parameter ρ defines a percentage of fuzziness relatively to the range of X_i values.

With increasing ρ values the probabilities $p(C_i|\mathbf{X}; \rho, M)$ change. Even if a crisp rule-based classifier is used non-zero probabilities of classes alternative to the winning class will gradually appear. The way in which these probabilities change shows how reliable is the classification and what are the alternatives worth remembering. If the probability $p(C_i|\mathbf{X}; \rho, M)$ changes rapidly around some value ρ_0 the case $\mathbf{X}$ is near classification border and an analysis of $p(C_i|\mathbf{X}; \rho_0, s_i, M)$ as a function of each $s_i = s(X_i), i = 1 \ldots N$ is needed to see which features have strong influence on classification. Displaying such probabilites allows for a detailed evaluation of the new data also in cases where analysis of rules is too complicated. A more detailed analysis of these probabilities based on *confidence intervals* and *probabilistic confidence intervals* has recently been presented by Jankowski [24,9]. Confidence intervals are calculated individually for a given input vector while logical rules are extracted for the whole *training set*. Confidence intervals measure maximal deviation from the given feature value X_i (assuming that other features of the vector $\mathbf{X}$ are fixed) that do not change the most probable classification of the vector $\mathbf{X}$. If this vector lies near the class border the confidence intervals are narrow, while for vectors that are typical for their class confidence intervals should be wide. These intervals facilitate precise interpretation and allow to analyze the stability of sets of rules.

For some classification models probabilities $p(C_i|\mathbf{X}; \rho, M)$ may be calculated analytically. For the crisp rule classifiers [25] a rule $R_{[a,b]}(X)$, which is true if $X \in [a, b]$ and false otherwise, is fulfilled by a Gaussian number G_X with probability:

$$p(R_{[a,b]}(G_X)=T) \approx \sigma(\beta(X-a)) - \sigma(\beta(X-b)) \tag{2}$$

where the logistic function $\sigma(\beta X) = 1/(1+\exp(-\beta X))$ has $\beta = 2.4/\sqrt{2}s(X)$ slope. For large uncertainty $s(X)$ this probability is significantly different from zero well outside the interval $[a,b]$. Thus crisp logical rules for data with Gaussian distribution of errors are equivalent to fuzzy rules with "soft trapezoid" membership functions defined by the difference of the two sigmoids, used with crisp input value. The slope of these membership functions, determined by the parameter β, is inversely proportional to the uncertainty of the inputs. In the C-MLP2LN neural model [13] such membership functions are computed by the network "linguistic units" $L(X;a,b) = \sigma(\beta(X-a)) - \sigma(\beta(X-b))$. Relating the slope β to the input uncertainty allows to calculate probabilities in agreement with the Monte Carlo sampling. Another way of calculating probabilities, based on the softmax neural outputs $p(C_j|\mathbf{X};M) = O_j(\mathbf{X})/\sum_i O_i(\mathbf{X})$ has been presented in [9].

After uncertainty of inputs has been taken into account probabilities $p(C_i|\mathbf{X};M)$ depend in a continuous way on intervals defining linguistic variables. The error function:

$$E(M,\mathbf{S}) = \frac{1}{2}\sum_{\mathbf{X}}\sum_i \left(p(C_i|\mathbf{X};M) - \delta(C(\mathbf{X}),C_i)\right)^2 \tag{3}$$

depends also on uncertainties of inputs $\mathbf{S}$. Several variants of such models may be considered, with Gaussian or conical (triangular-shaped) assumptions for input distributions, or neural models with bicentral transfer functions in the first hidden layer. Confusion matrix computed using probabilities instead of the number of yes/no errors allows for optimization of Eq. (1) using gradient-based methods. This minimization may be performed directly or may be presented as a neural network problem with a special network architecture. Uncertainties s_i of the values of features may be treated as additional adaptive parameters for optimization. To avoid too many new adaptive parameters optimization of all, or perhaps of a few groups of s_i uncertainties, is replaced by common ρ factors defining the percentage of assumed uncertainty for each group.

This approach leads to the following important improvements for any rule-based system:

- Crisp logical rules provide basic description of the data, giving maximal comprehensibility.
- Instead of 0/1 decisions probabilities of classes $p(C_i|\mathbf{X};M)$ are obtained.
- Inexpensive gradient method are used allowing for optimization of very large sets of rules.
- Uncertainties of inputs s_i provide additional adaptive parameters.
- Rules with wider classification margins are obtained, overcoming the brittleness problem of some rule-based systems.

Wide classification margins are desirable to optimize the placement of decision borders, improving generalization of the system. If the vector $\mathbf{X}$ of an unknown class is quite typical to one of the classes C_k increasing uncertainties s_i of X_i inputs to a reasonable value (several times the real uncertainty, estimated for a given data) should not decrease the $p(C_k|\mathbf{X};M)$ probability significantly. If this is not the case $\mathbf{X}$ may be close to the class border and analysis of $p(C_i|\mathbf{X};\rho,s_i,M)$ as a function of each s_i is needed. These probabilities allow to evaluate the influence of different features on classification. If simple rules are available such explanation may be satisfactory. Otherwise to gain understanding of the whole data a similarity-based approach to classification and explanation is worth trying. Prototype vectors R_i are constructed using a clusterization, dendrogram or a decision tree algorithm and a similarity measure $D(\mathbf{X},\mathbf{R})$ is introduced. Positions of the prototype vectors R_i, parameters of the similarity measures $D(\cdot)$ and other adaptive parameters of the system are then optimized using a general framework for similarity-based methods [26]. This approach includes radial basis function networks, clusterization procedures, vector quantization methods and generalized nearest neighbor methods as special examples. An explanation in this case is given by pointing out to the similarity of the new case $\mathbf{X}$ to one or more of the prototype cases R_i.

The final step for data explanation requires visualization of the data in the neighborhood of $\mathbf{X}$. Although there are many interesting methods of vizualization (cf. [22]) its seems that the most appropriate here are the methods based on multidimensional scaling (MDS, [27]). In essence these methods try to present a map of N-dimensional objects in two (or three) dimensions, trying to preserve all distance relations $D(\mathbf{X}_i,\mathbf{X}_j)$ in the original space. This is achieved by minimization of some form of topographical measure of distortion, for example the Stress $S(\mathbf{x})$ introduced by Kruskal:

$$S(\mathbf{x}) = \sum_{ij}^{N_t} w_{ij} \cdot \left(\hat{D}(\mathbf{X}_i,\mathbf{X}_j) - d(x_i,x_j)\right)^2 \tag{4}$$

where $d(x_i,x_j)$ are distances in the low-dimensional target space and w_{ij} are user-defined weights. Similar formula is used [28] to map a single point relatively to an existing map. An interactive program for such vizualization has been written [28], allowing for creation of maps (cf. Fig. 4) by zooming on the neighborhood of $\mathbf{X}$. The zoomed MDS diagrams may be correlated with estimations of probability $p(C_i|\mathbf{X};\rho,M)$ and with more detailed $p(C_i|\mathbf{X};\rho,s_i,M)$ graphs presenting confidence intervals, giving better evaluation of new data.

4 Real-life examples

Two examples are shown here: psychometric data [29] and the Telugu vovel data [5]. In the first case rule-based approach to data understanding has been successful, in the second a prototype-based approach followed by visualization has been necessary.

The psychometric data [29] has 14 continuous attributes obtained by combining answers to questionnaires. They measure tendencies towards hypochondria, depression, hysteria, psychopathy, paranoia, schizophrenia, etc. The datasets have over

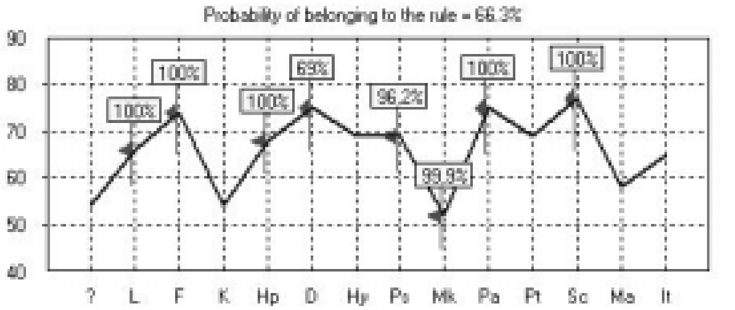

Fig. 1. An example of a rule with Gaussian uncertainties of conditions displayed.

1000 cases belonging to 27 classes (normal, neurotic, drug addicts, schizophrenic, psychopaths, organic problems, malingerers, persons with criminal tendencies etc.) determined by expert psychologists. Crisp logical rules for these data were generated using C4.5 classification tree [16], a very good classification system which may generate logical rules, and the Feature Space Mapping (FSM) neural network [20,21], since these two systems were the easiest to use on such complex data. Statistical estimation of generalization by 10-fold crossvalidation gave for crisp unoptimized FSM rules 82-85% correct answers, for C4.5 decision tree the accuracy was in the 79-84% range. If Gaussian uncertainty is included rule conditions become fuzzy (see Fig. 1) and optimization of probabilities improves the FSM crossvalidation results up to 90-92%. The IncNet model, an ontogenic neural network [30], obtained 93-95% accuracy in crossvalidation tests. Although it is not a rule-based system calculation of probabilities and visualization described below allows to evaluate the new data quite well also with such black-box systems.

Fig. 2 shows the probabilities of a well-behaved case belonging to the “organic problem class” as a function of ρ. The two plots in Fig. 3 show the probability as a function of two feature values for the case in which a sharp change in probabilities with increasing ρ has been observed, essentially mixing psychopathy, paranoia and

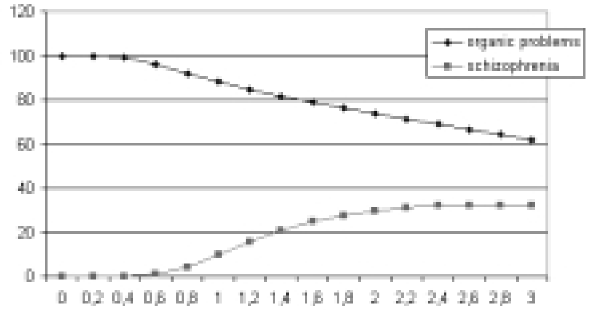

Fig. 2. Probabilities of different classes as a function of assumed uncertainties ρ shown in percents of the input feature range; this case is uniquely classified as “organic problems”.

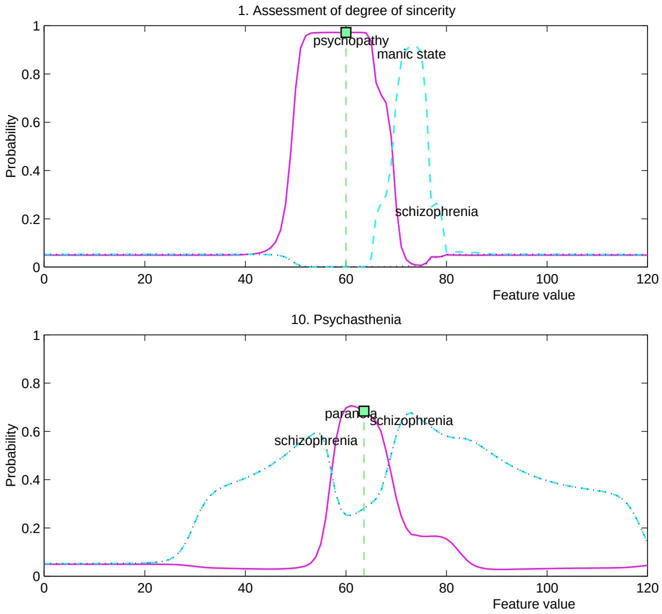

Fig. 3. Psychopathy is quite probable here, but small perturbations of feature 1 change it into a maniac state. In the lower picture paranoia diagnosis lies within the schizophrenia class.

schizophrenia classes. Probabilities for the most probable class at the measured **X** value and for the second most probable class are displayed. Seeing the mixture of schizophrenia and paranoia in Fig. 3 an expert may inferred that it is the case of schizoidal paranoia, which was not present as one of the classes in the data base.

The Telugu vovel data [5] give an example for which no rule-based understanding of the data is possible. Each vovel is described by intensities of 3 formants, therefore one may view the data in 3 dimensions and notice a strong overlap among the 6 classes – in Fig. 4 MDS maps of this data are displayed. The results of some tests are collected in Table 1. Unfortunately the variance has not been given in ref. [5] and since only two-fold crossvalidation was done for some models it may be quite large (even well-behaved models show on this dataset differences exceeding 5% due to the data partitioning). Using fuzzy rules or other sophisticated logical description in this case does not explain anything. A possible explanation of what makes one speech sound different than the other could either invoke a rule – but the number of rules produced by different systems is rather large and they are not too accurate, so no good rules exist here – or evaluate similarity to a prototype case.

FSM [20,21] neurofuzzy system with rectangular functions produces a large (75 or more) number of neurons, giving the 10-fold stratified crossvalidation (10xCV) accuracy of about 80%. The simplest FSM prototype-based solutions with Gaussian weighting of the Euclidean distance has 6 prototypes (one per class) and in the 10xCV test it gives 72.4% accuracy (74% on the whole dataset). Although this ac-

Table 1. Comparison of results on Telugu vovel data. 10xCV means 10-fold stratified cross-validation test, Manhattan is the distance function.

System	Accuracy	Remarks
kNN	88.1± 0.4%	k=3, Manhattan, 10xCV, this paper
FSM	87.4 ± 0.5%	Gauss, about 65 neurons, 10xCV, this paper
SSV	86.0%	22 rules, 10xCV, this paper
kNN	86.1± 0.6%	k=3, Euclidean, 2xCV, this paper
FSM	85.2 ± 1.2%	Gauss, over 40 neurons, 2xCV, this paper
MLP	84.6 %	2xCV, 10 neuronów, Ref. [5]
Fuzzy MLP	84.2 %	2xCV, 10 neuronów, Ref. [5]
SSV	83.3± 0.9 %	2xCV, beam search, this paper
Bayes Classifier	79.2 %	2xCV, Ref. [5]
Fuzzy Kohonen	73.5 %	2xCV, Ref. [5]

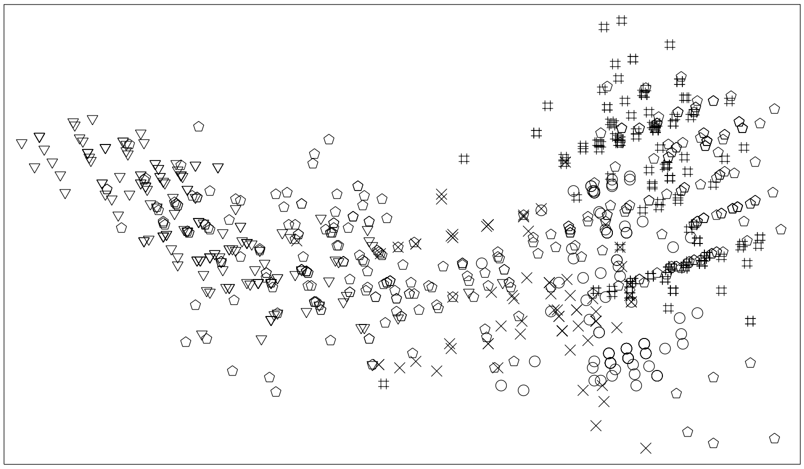

Fig. 4. The MDS map of the Telugu vovel data – classes strongly overlap.

curacy is not the highest 6 prototypes seem to be sufficient to explain main features of this data. Adding more neurons to the FSM network (the algorithm is constructive and the number of neurons is regulated by the desired accuracy) is equivalent to using more than one prototype per class.

5 Discussion

Acceptable explanations depend on the particular domain and the type of data analyzed. Machine Learning community has focused on artificial cases with a few symbolic attributes, for which simple logical rules are sufficient. Rule-based classifiers are useful only if rules are reliable, accurate, stable and sufficiently simple to be understood. In real data mining problems many continuous-valued attributes may be present and large sets of rules may be needed. Most classifiers are unstable [31] and lead to rules that are significantly different if the training set is slightly changed. Such rules contain little useful information and in fact may be rather misleading. Even if stable and robust rules are found [13] the user should be warned about potential misclassifications, other probable classification possibilities and influence of each feature on the classification probability.

Interpretation of new cases requires something more than a set of rules and yes/no decisions. Through correlation of rules, investigation of class probabilities in the neighborhood of a new case, confidence intervals for interesting features and interactive visualization the new data may be understood much better in relation to the whole database of known cases. For some classifiers probabilities $p(C_k|\mathbf{X};\rho,M)$ may be estimated analytically, for other classifiers Monte Carlo estimations are necessary. Since one new case is evaluated at a time the cost of such simulations is not so important. In practical applications users are interested in relevant features and may rarely be satisfied with answers to questions “why” based on quotation of logical rules belonging to a complex set. Similarity to prototypes, or case-based interpretation, is an alternative to logical rule-based systems. Expert systems in medical and other fields should help to understand and evaluate new cases, not just classify it.

Acknowledgement: The kNN program was written by Karol Grudziński, FSM by Rafał Adamczak, SSV by Krzysztof Grąbczewski, Fig. 3 are from Norbert Jankowski, and Fig. 3 was made with program written by Antoine Naud, all of the Dept. of Computer Methods, Nicholas Copernicus University.

References

1. Michalski, R.S. (1983) A Theory and Methodology of Inductive Learning. Artificial Intelligence **20**, 111–161
2. Michalski, R.S., Bratko, I. and Kubat, M. (eds.), Machine Learning and Data Mining: Methods and Applications, London, John Wiley and Sons, 1997.
3. Piatetsky-Shapiro, G. (2000) Knowledge Discovery in Databases: 10 years after. SIGKDD Explorations **1**: 59–61
4. Nauck, D., Klawonn, F., Kruse, R. (1997) Foundations on Neuro-Fuzzy Systems. J. Wiley, New York
5. Pal, S.K. and Mitra S. (1999) *Neuro-Fuzzy Pattern Recognition*. J. Wiley, New York
6. Michie, D., Spiegelhalter, D. J. Taylor, C. C. (eds.) (1994) Machine Learning, Neural and Statistical Classification. Ellis Horwood, New York.
7. Kosko B. (1992) Neural Networks and Fuzzy Systems. Prentice Hall
8. Duch, W., Adamczak, R., Grąbczewski, K. (1999) Neural optimization of linguistic variables and membership functions. Int. Conference on Neural Information Processing (ICONIP’99), Perth, Australia, Nov. 1999, Vol. II, pp. 616–621
9. Duch, W., Jankowski, N., Adamczak, R., Grąbczewski, K. (2000) Optimization and Interpretation of Rule-Based Classifiers. Intelligent Information Systems IX, Springer Verlag (in print)
10. Haykin, S. (1994) Neural networks: a comprehensive foundations. MacMillian.

11. Duch, W., Jankowski, N. (1999) New neural transfer functions. Neural Computing Surveys **2**, 639–658
12. Duch, W., Adamczak, R., Grąbczewski, K. (1998) Extraction of logical rules from backpropagation networks.
13. Duch, W., Adamczak, R., Grąbczewski, K. (in print) Methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Transactions on Neural Networks. Neural Processing Letters **7**, 1-9
14. Bennett K.P., Mangasarian, O.L. (1992) Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software **1**, 23–34
15. Mertz, C.J., Murphy, P.M. UCI repository of machine learning databases, http://www.ics.uci.edu/pub/machine-learning-data-bases.
16. Quinlan J.R. (1993) C4.5: Programs for machine learning. San Mateo, Morgan Kaufman
17. Grąbczewski, K., Duch, W. (1999) A general purpose separability criterion for classification systems. 4th Conf. on Neural Networks and Their Applications, Zakopane, Poland, May 1999, pp. 203–208
18. Buckley, J.J., Hayashi, Y., Czogala, E. (1993) On the Equivalence of Neural Nets and Fuzzy Expert Systems. Fuzzy Sets and Systems **53**, 129–134.
19. Yen, J., Langari, R., Zadeh, L.A. (eds) Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press, New York, 1995.
20. Duch, W., Diercksen, G.H.F. (1995) *Feature Space Mapping as a universal adaptive system*, Computer Physics Communication **87**, 341–371
21. Duch, W., Adamczak, R., Jankowski, N. (1997) New developments in the Feature Space Mapping model. 3rd Conf. on Neural Networks, Kule, Poland, Oct. 1997, pp. 65-70
22. Kay, J.W., Titterington, D.M., (1999) Statistics and neural networks. Oxford University Press, U.K.
23. Hastie T., Stuetzle, W. (1989) Principal curves. J. Am. Stat. Assoc. **84**, 502–516
24. Jankowski, N., (1999) Ontogenic neural networks and their applications to classification of medical data. PhD thesis (in Polish), Department of Computer Methods, Nicholas Copernicus University, Toruń, Poland
25. Duch, W., Adamczak, R., Grąbczewski, K. (1999) Methodology of extraction, optimization and application of logical rules. Intelligent Information Systems VIII, Ustroń, Poland, 14-18.06.1999, pp. 22-31
26. Duch, W. (1998) A framework for similarity-based classification methods, Intelligent Information Systems VII, Malbork, Poland, June 1998, pp. 288–291
27. Kruskal, J. B (1964) Non metric multidimensional scaling : a numerical method. Psychometrika **29**, 115–129.
28. Naud, A., Duch, W., (2000) Interactive data exploration using MDS mapping. 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, June 2000 (in print)
29. Duch W., Adamczak R., Grąbczewski K. (1999) Neural methods for analysis of psychometric data. Proc. of Enginnering Applications of Neural Networks (Duch W, ed.), Warsaw, Poland, Sept. 1999, pp. 45-50
30. Jankowski, N., Kadirkamanathan, V. (1997) Statistical control of RBF-like networks for classification. 7th Int. Conf. on Artificial Neural Networks, Lausanne, Switzerland 1997, pp 385-390, Springer Verlag.
31. Breiman L. (1998) Bias-Variance, regularization, instability and stabilization. In: C. Bishop (ed.) Neural Networks and Machine Learning. Springer Verlag