Optimization and Interpretation of Rule-Based Classifiers

W. Duch, N. Jankowski, K. Grąbczewski and R. Adamczak

Department of Computer Methods, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
E-mails: duch,norbert,kgrabcze,raad@phys.uni.torun.pl

Abstract. Machine learning methods are frequently used to create rule-based classifiers. For continuous features linguistic variables used in conditions of the rules are defined by membership functions. These linguistic variables should be optimized at the level of single rules or sets of rules. Assuming the Gaussian uncertainty of input values allows to increase the accuracy of predictions and to estimate probabilities of different classes. Detailed interpretation of relevant rules is possible using (probabilistic) confidence intervals. A real life example of such interpretation is given for personality disorders. The approach to optimization and interpretation described here is applicable to any rule-based system.

1 Introduction.

In many applications rule-based classifiers are created starting from machine learning, fuzzy logic or neural network methods [1]–[3]. If the number of rules is relatively small and accuracy is sufficiently high such classifiers are an optimal choice, because the reasons for their decisions are easily verified. Crisp logical rules are desirable since they are most comprehensible, but they have several drawbacks. First, using crisp rules only one class is identified as the correct one, thus providing a black-and-white picture where some gradation may be more appropriate. Second, reliable crisp rules may reject some cases as unclassified. Third, using the cost function based on the number of errors made by the crisp rule classifier leads to a difficult optimization problem, since only non-gradient optimization methods may be used.

These problems are overcomed if continuous membership functions are used, leading to fuzzy rather than crisp rules. Fuzzy rules have two disadvantages. First, they are not so comprehensible as the crisp rules, and second, they usually involve a number of parameters determining positions and shapes 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, 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 [4].

Another issue is the interpretation of the results obtained using rule-based classifiers. Although interpretation of crisp rules seems to be straightforward in fact it may be quite misleading. A small change in the value of a single feature may lead to a sudden change of the predicted class. Thus interpretation of crisp rules is not stable against small perturbations of input values. Fuzzy rules are better in this respect since estimation of probabilities of different classes change smoothly. Still a problem of tradeoff between the fuzziness and the degree of precision remains. If the membership functions are too fuzzy many classes have similar probability; if they are almost crisp perturbation of the input vector may significantly change classification probabilities, even if the size of the perturbation is within the range of accuracy of the measured input values. Believing the predicted results without exploration of alternative classes may in such cases be rather dangerous. Rough rules suffer from the same interpretative problems even to a greater degree, because rough classifiers produce a large number of unstable rules (cf. [5] on the importance of stability).

Thus although the biggest advantage of rule-based classifiers is their comprehensibility in practice reliable interpretation of sets of rules may not be so simple. A solution to these problems facing crisp and fuzzy rule-based classifiers applied to data with continuous features is presented in this paper. Neural and machine-learning methods of rule extraction from data were described in our previous publications [1]–[3]. Therefore we will assume that a small number of crisp logical rules has already been found. In the next section optimization and application of sets of logical rules is described. The third section deals with detailed interpretation of rule conditions and the fourth section illustrates optimization and interpretation of rules on a real-life psychometric data problem. The paper is finished with a short discussion.

2 Application and optimization of rule-based classifiers

Previously [1]–[3] we have described a complete methodology of rule extraction from the data. It is composed from the following steps:

- Select linguistic variables. In case of a continuos feature x linguistic variable s_k is true if the input value $x \in [X_k, X'_k]$, i.e. linguistic variables are parameterized by interval values $s_k(X_k, X'_k)$.
- Extract rules from 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.

Optimization of linguistic variables is done by minimization of the number of wrong predictions $\min_M \left[\sum_{i \neq j} P(C_i, C_j)\right]$ (where $P(C_i, C_j)$ is the confusion matrix for a rule-based classifier M), simultaneously with maximization of the predictive power of the classifier $\max_M [\mathrm{Tr}\, P(C_i, C_j)]$ over all intervals X_k, X'_k contained in the model M. This is equivalent to minimization without constraints of the following cost function $E(M)$:

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

where the parameter γ determines a tradeoff between reliability and rejection rate (number of vectors in the "don't know" class). Sets of rules of lower reliability (making larger number of errors) have lower rejection rates than sets of rules of higher reliability that have larger rejection rate. If $P(C_i, C_j)$ depends in a discontinuous way on the parameters in M minimization of this formula is difficult, requiring non-gradient minimization methods.

Real input values are obtained by measurements that are carried with finite precision, therefore it is natural to assume that instead of a crisp number X_i a Gaussian distribution $G_{X_i} = G(Y_i; X_i, S_{X_i})$ centered around X_i with dispersion S_{X_i} should be used. Performing a Monte Carlo sampling from the joint Gaussian distribution for all continuous features $G_X = G(\mathbf{Y}; \mathbf{X}, \mathbf{S}_X)$ an input vector $\mathbf{X}$ is selected and the rule-based classifier M is used to assign a class $C(\mathbf{X})$ to these vectors. Averaging results allows to compute probabilities $p(C_i|\mathbf{X})$. 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.

Assuming that uncertainties $s_i = s(X_i)$ are constants independent of the feature values X_i is a useful simplification. For a single feature $x = X_i$ to a very good approximation [2] 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 $\beta = 2.4/\sqrt{2}s$ defines the slope of the logistic function $\sigma(\beta x) = 1/(1+e^{-\beta x})$. For large dispersion s 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 slopes of these membership functions, determined by the parameter β, are inversely proportional to the uncertainty of the inputs. In our neural network approach to rule extraction such membership functions are computed by the network "linguistic units".

For uncorrelated input features X_i the probability that $\mathbf{X}$ satisfies a rule $R = R_1(X_1) \wedge \ldots \wedge R_N(X_N)$ may be defined as the product of the probabilities of $X_i \in R_i$ for $i = 1, \ldots N$. Our rule extraction methods produce very simple rules that do not contain dependent features in a single rule, therefore taking the product is a good approximation. Another problem occurs when probability of $\mathbf{X}$ belonging to a class described by more than one rule is estimated. Rules usually overlap because they use only a subset of all features and their conditions do not exclude each other. Summing and normalizing probabilities obtained for different classes may give results quite different from real Monte Carlo probabilities. To avoid this problem probabilities are calculated as:

$$P(x \in C) = \sum_{R \in 2^{R_C}} (-1)^{|R|+1} P(x \in \bigcap R) \tag{3}$$

where R_C is a set of all classification rules for class C, 2^{R_C} is a set of all subsets of R_C, and $|R|$ is the number of elements in R.

The uncertainty s_i of features may for some data dependent of the values of X_i. Classification probabilities may in such cases be based on a direct calculation of optimal soft-trapezoidal membership functions [6]. Linguistic units of neural networks with LR architecture provide such window-type membership functions, $L(x;a,b) = \sigma(\beta(x-a)) - \sigma(\beta(x-b))$. Relating the slope β to the input uncertainty allows to calculate probabilities in agreement with the Monte Carlo sampling. A network rule node (R-node) computes normalized product-type bicentral function:

$$R_j(\mathbf{X};\mathbf{p}_j) = \frac{\prod_{i \in I(R_j)} \sigma((X_i - t_{ij} + b_{ij}) s_{ij}^L)(1 - \sigma((X_i - t_{ij} - b_{ij}) s_{ij}^R))}{\sigma(b_{ij} s_{ij}^L)(1 - \sigma(b_{ij} s_{ij}^R))} \tag{4}$$

where $I(R_j)$ is a set of indices of features used in a given rule R_j and $R_j(\mathbf{X};\mathbf{p}_j) = R_j(\mathbf{X};\mathbf{t}_j,\mathbf{b}_j,\mathbf{s}_j^L,\mathbf{s}_j^R)$. Combining rules for separate clasess C_j:

$$O_j(\mathbf{X}) = \sigma\Big(\sum_{i \in I(C_j)} R_i(\mathbf{X};\mathbf{p}_i) - 0.5\Big) \tag{5}$$

where $I(C_j)$ is a set of rules indices for a given class C_j, probability of a class C_j for the given vector $\mathbf{X}$ is:

$$p(C_j|\mathbf{X};M) = O_j(\mathbf{X}) / \sum_i O_i(\mathbf{X}) \tag{6}$$

and the probability of a class C_j for a given vector $\mathbf{X}$ and rule R_i is

$$p(C_j|\mathbf{X},R_i;M) = p(C_j|\mathbf{X}) R_i(\mathbf{X};\mathbf{p}_i) \tag{7}$$

Optimization of model parameters: centers $\mathbf{t}$, biases $\mathbf{b}$ and slopes $\mathbf{s}$, may be done for example by the backpropagation gradient descend algorithm in the multilayer perceptron networks or by the Kalman filter approach in the IncNet neural networks [7]. Since probabilities $p(C_i|\mathbf{X};M)$ depend now in a continuous way on the linguistic variable parameters of the rule system M the error function:

$$E(M,\mathbf{S}) = \frac{1}{2} \sum_{\mathbf{X}} \sum_i (p(C_i|\mathbf{X};M) - \delta(C(\mathbf{X}),C_i))^2 \tag{8}$$

depends also on the Gaussian uncertainties of inputs $\mathbf{S}$ or on all parameters of the bicentral functions if full optimization of the membership functions is performed. Confusion matrix computed using probabilities instead of the yes/no error count 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. Assuming that the uncertainty of s_i is a percentage of the range of X_i values optimization is reduced to a one-dimensional minimization of the error function.

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

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

Wide classification margins are desirable to optimize the placement of decision borders, improving results on the test set. 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 a detailed analysis of the influence of each X_i feature value on the classification probability should be performed.

3 Confidence intervals and probabilistic confidence intervals

Logical rules may be replaced by *confidence intervals* or *probabilistic confidence intervals* [8]. Confidence intervals are calculated individually for a given input vector while logical rules are extracted for the whole *training set*. These intervals allow for analysis of the stabilty of rules as well as the interpretation of a given case. Suppose that for a given vector $\mathbf{X} = [X_1, X_2, \ldots, X_N]$ the highest probability $p(C_k|\mathbf{X};M)$ is found for the class k. Let the function $C(\mathbf{X}) = \arg\max_i \; p(C^i|\mathbf{X};M)$, i.e. $C(\mathbf{X})$ is equal to the index k of the most probable class for the input vector $\mathbf{X}$. The confidence interval $[X^r_{min}, X^r_{max}]$ for the feature X_r is defined by

$$\begin{aligned} X^r_{min} &= \min_{\bar{X}} \left\{ C(\bar{\mathbf{X}}) = k \;\wedge\; \forall_{X_r > \hat{X} > \bar{X}} \, C(\hat{\mathbf{X}}) = k \right\} \\ X^r_{max} &= \max_{\bar{X}} \left\{ C(\bar{\mathbf{X}}) = k \;\wedge\; \forall_{X_r < \hat{X} < \bar{X}} \, C(\hat{\mathbf{X}}) = k \right\} \end{aligned} \tag{9}$$

where $\bar{\mathbf{X}} = [X_1, \ldots, X_{r-1}, \bar{X}, X_{r+1}, \ldots, X_N]$, and $\hat{\mathbf{X}} = [X_1, \ldots, X_{r-1}, \hat{X}, X_{r+1}, \ldots, X_N]$. Confidence intervals measure maximal deviation from the value X_r that do not change the most probable classification of the vector $\mathbf{X}$, assuming that all other feature values are unchanged. If the vector $\mathbf{X}$ lies near the class border the confidence intervals are narrow, while for vectors that are typical for their class confidence intervals should be wide.

Probabilistic intervals of confidence (PIC) should guarantee that *the winning* class k is considerably more probable than the most probable alternative class:

$$X^{r,\rho}_{min} = \min_{\bar{X}}\left\{C(\bar{\mathbf{X}}) = k \wedge \forall_{X_r>\hat{X}>\bar{X}}\, C(\hat{\mathbf{X}}) = k \wedge \frac{p(C^k|\bar{\mathbf{X}})}{\max_{i\neq k} p(C^i|\mathbf{X})} > \rho\right\}$$

$$X^{r,\rho}_{max} = \max_{\bar{X}}\left\{C(\bar{\mathbf{X}}) = k \wedge \forall_{X_r<\hat{X}<\bar{X}}\, C(\hat{\mathbf{X}}) = k \wedge \frac{p(C^k|\bar{\mathbf{X}})}{\max_{i\neq k} p(C^i|\mathbf{X})} > \rho\right\}$$

The ρ factor determines the confidence level. Observation of changes in confidence intervals for different levels of ρ may be quite informative. Comparison of probabilistic intervals for the winning class and alternative classes helps to estimate the likelihood of a winning class. Such method escapes the danger of relaying only on the decision borders of logical rules. Assuming that other features are held constant for a given case $\mathbf{X}$ three probabilities for each feature X_r are displayed in Fig. 3, 4. The solid curve is the probability of the winning class defined by $p(C(\mathbf{X})|\,\mathbf{X};M)$. The class may change for different values of $\bar{\mathbf{X}}$. The dotted curve is the probability $p(C^{k_2}|\bar{\mathbf{X}})$ of the most probable alternative class $k_2 = \arg\max_i \{p(C^i|\mathbf{X};M),\ C^i \neq C(\mathbf{X})\}$. The k_2 class is determined for the point $\mathbf{X}$ only. The dashed line presents the probability $p(C^{k_M}|\bar{\mathbf{X}})$ of the most probable alternative class at $\bar{\mathbf{X}}$. The class index $k_M = \arg\max_i \{p(C^i|\bar{\mathbf{X}}),\ C^i \neq C(\mathbf{X})\}$ may change, while k_2 does not change. These three probabilities carry all information about the case given for analysis, showing the stability of classification against perturbation of each feature and the importance of alternative classes in the neighborhood of the input $\mathbf{X}$.

4 Real-life example

Using the theoretical ideas described here we have developed a rule-based expert system to support psychological diagnoses. The description of psychometric data and the test used has already been given in [9] and [10]. Here we will focus on interpretation of the results only. 14 coefficients are calculated from analysis of answers to the psychometric test, giving after normalization "psychological scales", often displayed in a histogram (called "a psychogram"). The first four coefficients are used for control, measuring consistency of answers or the number of "don't know" answers, allowing to find malingerers. The next 10 coefficients form clinical scales, developed to measure tendencies towards hypochondria, depression, hysteria, psychopathy, paranoia, schizophrenia, etc. For example values between 70 and 80 in the hypochondria scale may be interpreted as "very strong worries about own health, leading to psychosomatic reactions".

We have worked with two datasets, one for women, with 1027 cases belonging to 27 classes (normal, neurotic, drug addicts, schizophrenic, psychopaths, organic problems, malingerers, persons with criminal tendencies etc.) determined by expert psychologists, and the second for men, with 1167 cases and 28 classes. Rules were generated using C4.5 classification tree [11], a very good classification system which may generate logical rules, and the Feature Space Mapping (FSM) neural network [12,13] since these two systems were the easiest to use on such complex data.

These results are for the reclassification accuracy only using generated sets of rules. Statistical estimation of generalization by 10-fold crossvalidation gave 82-85% correct answers with FSM (crisp unoptimized rules) and 79-84% correct answers with C4.5. Fuzzification improves FSM crossvalidation results to 90-92%. A summary of results is given in Table 1. Accuracy refers there to the overall reclassification accuracy. Results from IncNet, a neural network model used in our group [7], obtained 93-95% accuracy in crossvalidation tests, comparing with 99.2% for reclassification.

Table 1. Comparison of results on psychometric data. Fuzzy accuracy refers to results with optimal uncertainty (C4.5, FSM) or results with bicentral functions obtained with IncNet.

Dataset	System	Crisp Rules	Accuracy	Fuzzy accuracy
women	C4.5	55	93.0	93.7
women	FSM	69	95.4	97.6
women	IncNet	–	–	99.2
men	C4.5	61	92.5	93.1
men	FSM	98	95.9	96.9
men	IncNet	–	–	99.2

These rules are most accurate on the available data if about 1% of the uncertainty of measurement in each of the scales is assumed, corresponding to a Gaussian dispersion centered around measured values. Larger uncertainties, on the order of 5%, lead to about the same number of classification errors as the original crisp rules, but provide softer evaluation of possible diagnoses, assigning non-zero probabilities to classes that were not covered by slightly fuzzified rules. Taking the input vector (66 74 54 68 75 69 69 52 75 69 77 58 65) for one of the cases difficult to diagnose, in Fig. 1 the influence of growing uncertainties has been presented. The top two plots show the profile and gaussian curves for each of the attributes occurring in rule No. 54 classifying the "organic problems" cases. In the first of these standard deviation of all attributes is equal to 1.3 times the range of possible values, while in the second standard deviation is equal to 3 times the range. The boxes above each feature value present the probability of belonging to a single premise of the rule. The bottom two plots show analogical properties for rule No. 59 which classifies to the "schizophrenia" class. Figure 2 shows how the probabilities in this example depend on the assumptions about the data uncertainty.

If the change of the input uncertainty has strong influence on the probability of the winning class a more detailed analysis may be useful. In contrast to rule-based classifiers we will focus here on a single case, using all features and estimation of conditional probabilities from IncNet classifier [7]. Figures 3 and 4 show probabilistic intervals of confidence for two quite different patients (the first and the last scale has been omitted, therefore only 12 features are displayed). The little squares

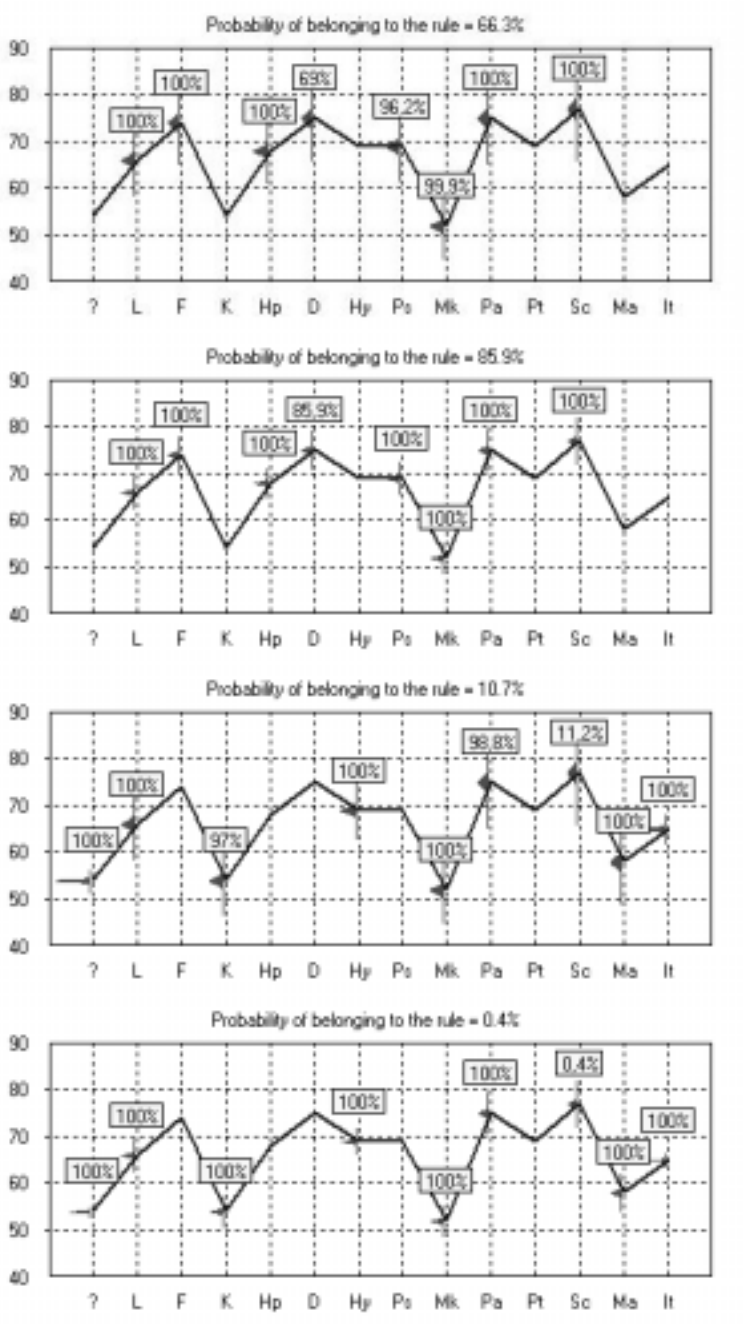

Fig. 1. Two rules applied to a case with small and large uncertainties.

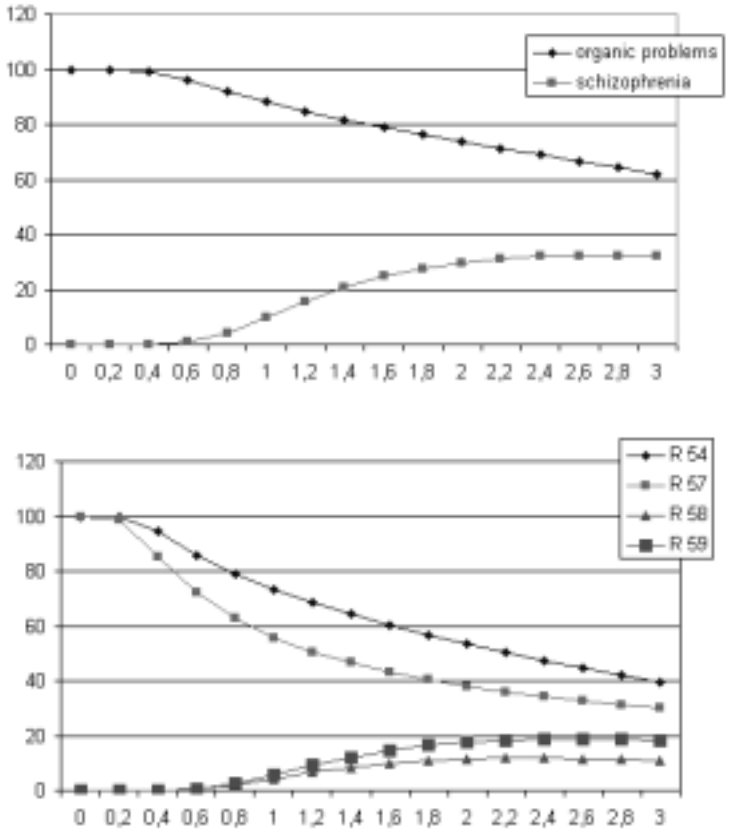

Fig. 2. Probabilities of classes and rules (in percents) for different values of assumed uncertainties s_i, in percentage of the total range of the feature values.

show the probability of the winning class corresponding to the measured input values of the psychometric scales. Figure 3 presents an easy case: the psychopathy has a large probability 0.97 and the case is quite far from any other alternative classes. The whole range of values, 0-120, is shown and an alternative class appears only for features 4, 7 and 12, but the confidence intervals are quite broad. Classification does not depend on the precise values of some features r (for example features 2, 3, 5, 6, etc) since there are no alternative classes in the whole range of values $\bar{\mathbf{X}}$ may take. The second set of plots, Fig. 4, is not so simple. The winner class, paranoia, has probability 0.68 while the alternative class, schizophrenia has probability 0.28. The analysis of plots shows that the values for scales 7 and 11 are close to the border and therefore both diagnoses are probable.

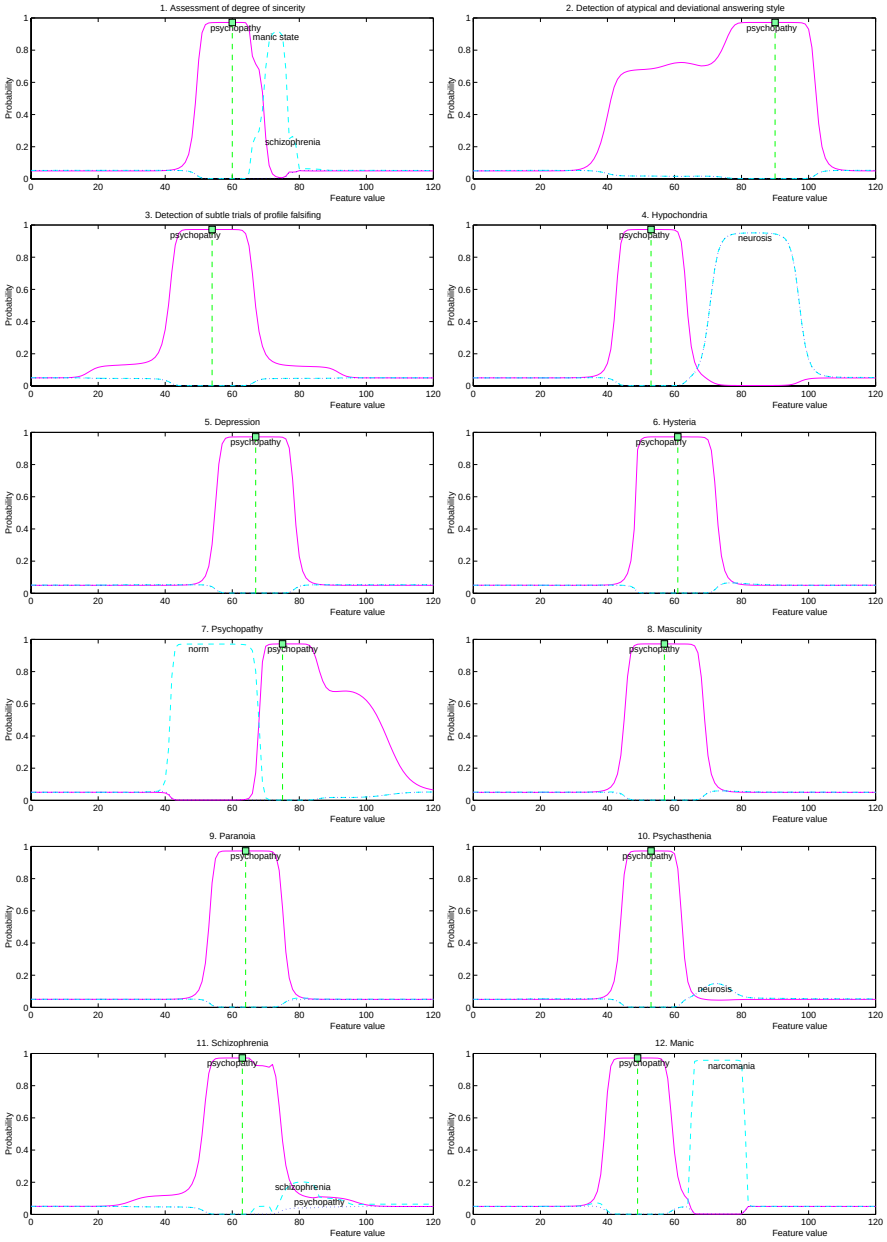

Fig. 3. Class: Psychopathy (prob. 0.97); alternative class: neurosis (prob. 0.002).

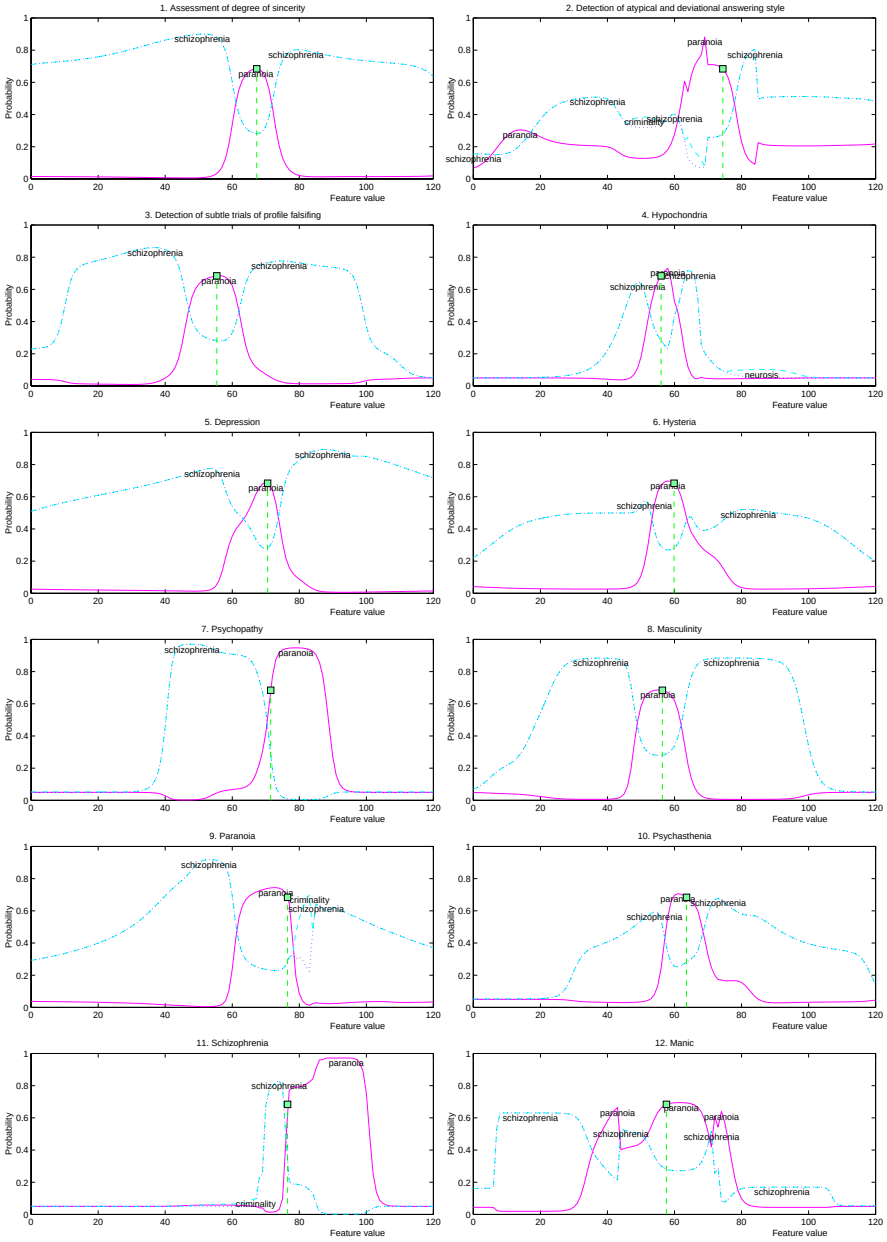

Fig. 4. Class: Paranoia (prob. 0.68); alternative class: schizophrenia (prob. 0.28).

5 Discussion

Machine Learning community has focused on artificial cases where a few symbolic attributes are defined (for example, the three Monk's problems). In real data mining problems many continuous-valued attributes may be presented and large sets of rules may be needed. Rule-based classifiers are useful only if rules are reliable, accurate, stable and sufficiently simple to be understood. Most classifiers are unstable [5] 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 [1] the user should be warned about potential misclassifications, other probable classification possibilities and influence of each feature on the classification probability.

In this paper optimization and interpretation of sets of rules have been described. The method is equivalent to a specific fuzzification of crisp membership functions, equivalent to an assumption of uncertainties in the inputs. Analysis of the change of probabilities of classification in response to the change in uncertainties allows to estimate confidence in the performance of a rule-based system. If the confidence is low a more detailed analysis of the influence of each feature on classification probability is started. Probabilistic confidence intervals may be applied to any classificator estimating $p(C_k|\mathbf{X})$, enabling detailed interpretation of cases. In practical applications users are interested in relevant features and may rarely be satisfied with answers to questions "why" based on quotation of complex sets of logical rules. Similarity to prototypes, or case-based interpretation, is an alternative to rule-based systems. Therefore one should not exaggerate the importance of logical description as the only understandable alternative to other classification methods.

Support by the KBN, grant 8 T11F 014 14, is gratefully acknowledged.

References

1. 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.
2. 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
3. Duch, W., Adamczak, R., Grąbczewski, K. (1998) Extraction of logical rules from backpropagation networks. Neural Processing Letters **7**, 1-9
4. 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
5. Breiman L. (1998) Bias-Variance, regularization, instability and stabilization. In: C. Bishop (ed.) Neural Networks and Machine Learning. Springer Verlag
6. Duch, W., Jankowski, N. (1999) *New neural transfer functions.* Neural Computing Surveys **2**, 639-658

7. 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.
8. 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
9. 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
10. Duch, W., Kucharski, T., Gomuła, J., Adamczak, R., (1999) Metody uczenia maszynowego w analizie danych psychometrycznych. Zastosowanie do wielowymiarowego kwestionariusza osobowośći MMPI-WISKAD. Toruń, March 1999; 650 pp, ISBN 83-231-0986-9
11. Quinlan J.R. (1993) C4.5: Programs for machine learning. San Mateo, Morgan Kaufman
12. Duch, W., Diercksen, G.H.F. (1995) *Feature Space Mapping as a universal adaptive system*, Computer Physics Communication **87**, 341–371
13. 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