Fuzzy Knowledge Generation Method for Data-Mining Problems

DMITRY KROPOTOV, VLADIMIR RYAZANOV, DMITRY VETROV
Situation Recognition Department
Dorodnicyn Computing Centre of the Russian Academy of Sciences
119991, Moscow, Vavilova str. 40, CCAS
RUSSIAN FEDERATION

Abstract: - Fuzzy sets have been widely used for solving data-mining problems during the last years. Another possible area of fuzzy methods application is automatic knowledge generation based on the set of precedents. This area is very important for artificial intelligence and machine learning theory. In this paper we suggest a new algorithm for fuzzy knowledge generation. It can find all significant rules with respect to wide range of reasonable criterion functions. Besides, the number of rules being generated is not high and their size is short thus simplifying decision interpretation by expert. We present the statistical criterion for knowledge quality estimation that provides high generalization ability. The theoretical results are complemented with the experimental evaluation.

Key-Words: - Data-mining, Artificial intelligence, Fuzzy sets, Knowledge generation, Rules optimization.

1 Introduction

At the present time fuzzy logic concept finds its application in many areas of human knowledge. Thus there exist a lot of successive projects of implementing fuzzy logic in control systems [2]. The ability of the theory to represent dependencies in linguistic terms facilitating understanding and managing the investigated process [3] led to development of fuzzy expert systems [1]. Such systems aimed for supervised learning or forecasting fall under the situation, in which we are given a set of fuzzy sets for each feature and knowledge base - a set of fuzzy rules. The successive system's creation depends fully on happy choice of fuzzy sets and rules appropriate for the current research field. It is a common situation that experts can't properly solve the problem with forming of fuzzy sets and rules and hence there is a need for some kind of automatic means. For the purpose methods using neural networks [4, 5], genetic algorithms [6, 7], clustering [8] and others have been proposed.

Unfortunately, these approaches have some drawbacks:

- The fuzzy system with great number of generated rules with relatively low significance level tends to sufficient overfitting;
- High rules dimensions lead to poor knowledge interpretation and inability of deep understanding for the current application field;
- Neuro-fuzzy techniques are characterized by dependence from initial approximation and sufficient calculation time needed for training;
- For clustering there is a need to determine number of clusters or number of rules beforehand;
- In genetic approaches there are a great number of parameters to be set by user and sufficient calculation time (or even infinite time in case of non-convergence).

Also the following algorithms for fuzzy rules generation as decision lists [9] and active learning procedures like boosting [10, 11] can be mentioned. In these algorithms rules are used consequently for decision making. In practice consequent scheme makes decision interpretation for expert sufficiently harder or even impossible. There is a huge amount of research in algorithms based on decision trees [12, 13, 14]. These algorithms show good performance and are frequently used in practice. However, presentation of tree structure as a set of rules leads to great number of long rules with very similar sumptions [13]. This hardens decision interpretation.

The goal of this article is to establish rules generating algorithm which avoids the mentioned drawbacks and at the same time builds a little set of short informative rules. In the next section different ways of representing fuzzy rules are considered. Section 3 provides algorithm for rules generation and investigates its properties. Then experimental results and brief conclusion are given.

Hereinafter suppose we are given d real-valued features (independent variables) and one dependent variable, which takes values in $\{1,\ldots,l\}$ for classification (pattern recognition) task with l classes or takes real values for regression task.

Training set is denoted as $\{x_i, y_i\}_{i=1}^{q}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1,\dots,l\}$ or $y_i \in \mathbb{R}$.

2 Knowledge presentation

Usually for knowledge presentation a set of rules of type "IF …, THEN…" is used [2]. At that rule sumption is some logical expression with respect to fuzzy sets of features. Denote $\mu_A(x)$ as characteristic function of fuzzy set A. Consider some real-valued feature. From expert point of view this feature can be described as ordered set of fuzzy sets, where each of them corresponds to some linguistic value. For example, feature "patient body temperature" can be presented as three fuzzy sets with labels "Low", "Medium" and "High". In general case suppose that for expert there exists some partition of feature values which determines conditional borders between different states.

Definition. *Expert interpretation* I_i *of feature* $i \in \{1,\dots,d\}$ for partition of its possible values $a_i^1, a_i^2 \dots, a_i^{n_i}$ is a set of fuzzy sets with conditional borders on neighbor points in this partition:

$$\mathrm{I}_i = \{M_i^j,\ j = \overline{1,n_i} \mid \mu_{M_i^j}(x) = \mu(x; a_i^j, a_i^{j+1}),\ x \in \mathbb{R}\}$$

and

$$\forall j = \overline{1,n_i}\ \exists x_* \in [a_i^j, a_i^{j+1}]:$$
$$\mu_{M_i^j}(x_*; a_i^j, a_i^{j+1}) = \max_x \mu_{M_i^j}(x; a_i^j, a_i^{j+1})$$

Here $\mu(x;a,b)$ means characteristic function of fuzzy set with conditional borders *a* and *b*. The particular shape of characteristic function $\mu(\cdot)$ can be chosen in different way. In the paper trapezium- and bell-shaped functions are considered. The connection between function's shape and conditional borders will be given in detail below.

2.1 Trapezium-shaped functions

Base shape of characteristic function can be given by isosceles trapezium (see fig. 1). Such trapezium includes triangle (if $b = c$) and rectangle (if $a = b, c = d$) as special case.

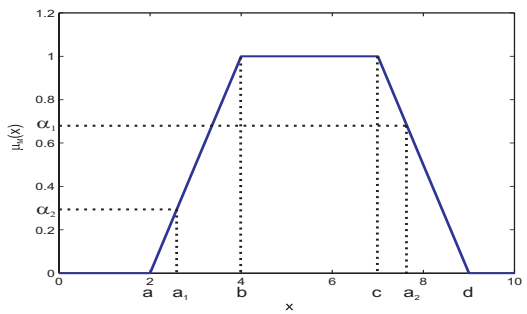

Fig. 1 Trapezium-shaped characteristic function.

Definition. $\{\vec{\alpha}, \vec{\beta}\}$*-covering* of feature for some partition of its possible values $a_1,\dots,a_n$ is a set of fuzzy sets $\{M_i\}_{i=1}^{n-1}$ with trapezium-shaped characteristic functions (fig. 1) such that:

1. $\alpha_1,\dots\alpha_n \in (0,1),\ \beta_1,\dots,\beta_{n-1} \in [0,1)$
2. $\mu_{M_i}(a_{i+1}) = \mu_{M_{i+1}}(a_{i+1}) = \alpha_{i+1},\ \forall i = \overline{1,n-2}$
3. $\mu_{M_1}(a_1) = \alpha_1, \mu_{M_{n-1}}(a_n) = \alpha_n$
4. $\dfrac{\left|\{x \in \mathbb{R} \mid \mu_{M_i}(x) = 1\}\right|}{a_{i+1} - a_i} = \beta_i,\ \forall i = \overline{1,n-1}$
5. $\exists x_* \in [a_i, a_{i+1}]: \mu_{M_i}(x_*) = \max_x \mu_{M_i}(x),$ $\forall i = \overline{1,n-1}$

It can be shown that $\{\vec{\alpha}, \vec{\beta}\}$-covering for partition $a_1,\dots,a_n$ is defined uniquely. Thus $\{\vec{\alpha}, \vec{\beta}\}$-covering can be used as expert interpretation of features. For definition of this covering it is necessary to set $2n-1$ parameters. However, in practice feature partition shows approximate borders between different states and fuzzy sets shape determines confidence level of expert with respect to these borders. Hence suppose that coefficients $\{\vec{\alpha}, \vec{\beta}\}$ characterize expert knowledge rather about whole feature values than separate fuzzy sets. That is why it seems reasonable to set $\alpha_1,\dots,\alpha_n = \alpha,\ \beta_1,\dots,\beta_{n-1} = \beta$.

2.2 Bell-shaped functions

Trapezium-shaped characteristic functions are simple and have intuitive interpretation. However, such functions are not continuously differentiable, that hardens optimization of approximate borders using precedent information. To solve this problem a set of bell-shaped functions are introduced (see fig.2):

$$\mu(x; l, r, \alpha, \beta) = \frac{1}{1 + \left(\frac{1}{\alpha} - 1\right)\left(\frac{x - (l+r)/2}{(r-l)/2}\right)^{2\beta}},$$

$$\alpha \in (0,1),\ \beta \in \mathbb{N},\ l, r, x \in \mathbb{R},\ l < r$$

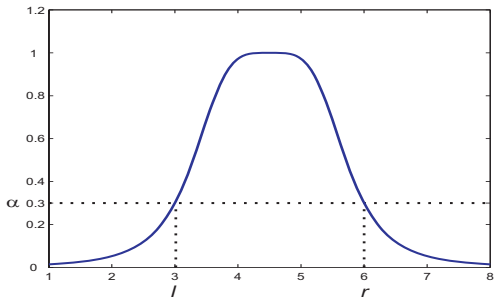

Fig. 2 Bell-shaped characteristic function.

Here parameters l and r determine approximate borders of fuzzy set, coefficient β controls fuzzy degree and $\mu(l;l,r,\alpha,\cdot)=\mu(r;l,r,\alpha,\cdot)=\alpha$. When β tends to infinity bell-shaped characteristic function determines classical set of interval $[l,r]$.

Similar to $\{\vec{\alpha},\vec{\beta}\}$-covering a covering by means of bell-shaped functions can be introduced.

Definition. $[\vec{\alpha},\vec{\beta}]$-*covering* of feature for partition of its values $a_1,\ldots,a_n$ is a set of fuzzy sets $\{M_i\}_{i=1}^{n-1}$ with bell-shaped characteristic functions $\mu_{M_i}(x)=\mu(x;a_i,a_{i+1},\alpha_i,\beta_i)$, $\alpha_1,\ldots\alpha_{n-1}\in(0,1)$, $\beta_1,\ldots,\beta_{n-1}\in\mathbb{N}$.

In $[\vec{\alpha},\vec{\beta}]$-covering parameters $\vec{\alpha}$ determine values of characteristic functions in partition points. If there are no significant reasons to set some particular values for these parameters, 0.5 can serve as appropriate choice. Similar to $\{\vec{\alpha},\vec{\beta}\}$-covering parameters $\vec{\beta}$ in $[\vec{\alpha},\vec{\beta}]$-covering can be treated as expert confidence level with respect to chosen borders between states. Hence we can assume that $\beta_1,\ldots,\beta_{n-1}=\beta$.

In the following $\mu(x;a,b)$ can be considered as both trapezium- or bell-shaped functions.

3 Rule generation

Consider fuzzy rule R in the following form:

$$IF\ x_{i_1}\in M_{i_1}^{j_1}\ \&\ldots\&\ x_{i_r}\in M_{i_r}^{j_r},THEN\ y\in N^k \quad (1)$$

Here $Sump(R)=\{M_{i_1}^{j_1},\ldots,M_{i_r}^{i_r}\}$ and $res(R)=N^k$. Denote R^* the set of all possible rules of type (1). Rule generation problem means separation of some subset of rules $\tilde{R}$ from R^*, where these rules satisfy some criterion function. Many known criterion functions can be formulated using notions of representativeness and effectiveness.

Definition. *Representativeness* of rule R is the following value:

$$rep(R)=\frac{1}{q}\sum_{k=1}^{q}\min\left(\mu_{M_{i_1}^{j_1}}(x_{i_1}^k),\ldots,\mu_{M_{i_r}^{j_r}}(x_{i_r}^k)\right)$$

Definition. *Effectiveness* of rule R is given by the following formula:

$$eff(R)=\frac{\sum_{k=1}^{q}\min\left(\mu_{M_{i_1}^{j_1}}(x_{i_1}^k),\ldots,\mu_{M_{i_r}^{j_r}}(x_{i_r}^k),\mu_{N^k}(y^k)\right)}{rep(R)q}$$

In other words representativeness is implicitly the rate of precedents, which satisfy the sumption of the given rule while effectiveness is the rate of precedents from the sumption, which satisfy the rule itself. We intend to generate rules, which have both high representativeness and effectiveness. More formally, a rule $R\in\tilde{R}$, if $C(rep(R),eff(R))=1$, where C takes one or zero if rule satisfies criterion function or not.

The simplest criterion function uses constant thresholds for representativeness and effectiveness:

$$C^h(v,w)=\begin{cases}1, if\ v\ge c_r, w\ge c_e\\ 0, otherwise\end{cases} \quad (2)$$

It is clear that for low representativeness significant rule must have high effectiveness while its effectiveness should be just a little more than prior class probability in case of high representativeness. Criterion function which takes this assumption into account can be formulated using statistical hypothesis checking. The rule R is insignificant if the information that object satisfies the rule sumption sheds no light on its affiliation to the result set of the rule. Let's check the following statistical hypothesis:

$$\mathbb{P}\{y\in N^k\mid x\in Sump(R)\}=\mathbb{P}\{y\in N^k\}$$

Without loss of generality suppose uniform prior probabilities: $\mathbb{P}\{y\in N^1\}=\ldots=\mathbb{P}\{y\in N^l\}=1/l$. Examine the value $q\cdot rep(R)\cdot eff(R)$. If the hypothesis is right, we have $n=rep(R)q$ Bernoulli trials with the probability of success equals $s=1/l$. If $ns>5$, according to Moivre-Laplace theorem, the distribution can be approximated with a normal distribution with the mean ns and variance $ns(1-s)$. This means that:

$$eff(R)\sim N(s,s(1-s)/n)$$

Fixing the level of significance α, we find the necessary criterion function:

$$C^s(v,w)=\begin{cases}1, if\ \dfrac{lw-1}{\sqrt{\dfrac{l-1}{vq}}}\ge z_\alpha\\ 0, otherwise\end{cases} \quad (3)$$

Here z_α is fractile of standard normal distribution.

In literature there are known many other possible predicates for identification of significant rules based on effectiveness and representativeness notions. Entropy-based criterion and exact Fisher test [15] can serve as examples.

Definition. $\alpha \geq 0$ is called *characteristics* of predicate $C:[0,1]^2 \to [0,1]$, if the following is true:

- $\forall (v,w) \in [0,1]^2 : C(v,w) = 1 \Rightarrow v \cdot w \geq \alpha$
- $\forall \varepsilon > 0 \; \exists (v,w) \in [0,1]^2 : C(v,w) = 1, v \cdot w < \alpha + \varepsilon$

Definition. Predicate $C^\alpha : [0,1]^2 \to [0,1]$ with characteristic α is called *maximal*, if for any other predicate $C' : [0,1]^2 \to [0,1]$ with characteristics α $\forall (v,w) \in [0,1] : C'(v,w) = 1 \Rightarrow C^\alpha(v,w) = 1$.

Definition. Rule R_b is *restriction* of rule R_a ($R_b \subset R_a$) if the next two conditions are satisfied:

- $res(R_a) = res(R_b)$
- $Sump(R_a) \subset Sump(R_b)$

During the rule restriction representativeness becomes lower while the effectiveness may become higher. In the last case we will call restriction an *effective* one.

Suppose we are given expert interpretations for all features $\mathrm{I}_1, \ldots, \mathrm{I}_d$, rules' result set N^k and some predicate C with characteristics $\alpha \geq \alpha_0 > 0$. Denote

$$c_r^* = \inf \left\{ v \in [0,1] \mid \exists w \in [0,1] : C(v,w) = 1 \right\}$$

An algorithm given below (effective restrictions method) allows finding all significant rules of minimal possible order according to learning sample. It is based on linear search over the rules order.

Step 1. Construct all possible rules of the first order

$$R' = \{ R \in R^* \mid res(R) = N^k, Sump(R) = M_i^{j_i}, j_i = \overline{1, n_i}, i = \overline{1, d} \}$$

Step 2. Reject all rules with low representativeness, i.e. $R' = \left\{ R \in R' \mid rep(R) \geq c_r^* \right\}$.

Step 3. Reject all rules that will not become significant even under the most effective restriction:

$$R' = \left\{ R \in R' \mid C^{\alpha_0}(rep(R), eff(R)) = 1 \right\}$$

Step 4. If no rules remained then go to step 6. Otherwise examine the effectiveness of residuary rules. If $C(rep(R), eff(R)) = 1$ then the rule is tolerable and should be moved to the list of final rules:

$$\tilde{R} = \tilde{R} \cup \left\{ R \in R' \mid C(rep(R), eff(R)) = 1 \right\}$$

$$R' = \left\{ R \in R' \mid C(rep(R), eff(R)) = 0 \right\}$$

Step 5. All other rules (if any) are used for restrictions in the following way. Sumption of any rule being restricted should be a subset of any other two rules, which are being restricted to the same rule of higher order:

$$R' = \{ R \in R^* \mid res(R) = N^k,$$
$$Sump(R) = Sump(R_1) \cup Sump(R_2), R_1, R_2 \in R',$$
$$Ord(R) = Ord(R_1) + 1,$$
$$\forall R_+ : R \subset R_+, Ord(R_+) = Ord(R) - 1 \Rightarrow R_+ \in R' \}$$

In other words, the union of sumptions of any two rules, which are restricted to the same rule of higher order, is exactly the sumption of this new rule (see fig. 3). If no new rules got, then go to step 6. Otherwise go to step 2.

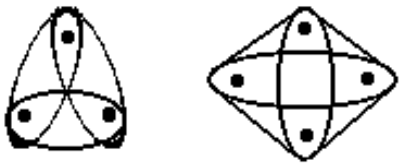

Fig. 3 Restriction of rules to third and forth order. Points represent fuzzy sets and contours encircle rules sumptions.

Step 6. If all result sets were examined then stop working, otherwise increase k by one and go to step 1.

Theorem. Effective restrictions method constructs all significant rules of minimal order for any predicate C with positive characteristics, i.e.

$$\tilde{R} = \{ R \in R^* \mid C(rep(R), eff(R)) = 1,$$
$$\forall R' \supset R \Rightarrow C(rep(R'), eff(R')) = 0 \}$$

The use of trapezium-shaped characteristic functions leads to continuous outputs with respect to continuous inputs. In case of bell-shaped functions the outputs are smooth (i.e. second derivate of output with respect to the inputs can be computed). These properties of outputs make possible the optimization of expert interpretation adjusting it to the training data using first (in case of continuous outputs) and second order (in case of smooth outputs) optimization methods.

4 Experiments and conclusion

The proposed algorithm was tested on both classification and forecasting tasks. For knowledge presentation bell-shaped characteristic functions with further borders optimization using training set were used. In classification case results of proposed

technique (ExSys) were compared with q-nearest neighbors (QNN), support vector machines (SVM), committee of linear classificators (LM), test algorithm (TA), linear Fisher discriminant (LDF) and multi-layer perceptron (MLP). The comparison was held according to three applications. The first was melanoma diagnostics (3 classes, 33 features, 48 objects in the training sample, 32 objects in the testing set), the second task was speech phoneme recognition (2 classes, 5 features, 2200 objects in the training sample, 1404 in the testing set) and the last one was drug intoxication diagnostics (2 classes, 18 features, 450 objects in the training sample and 450 in the testing set). The results of experiments (percent of correctly classified objects in the independent test sample) are shown on Figure 4.

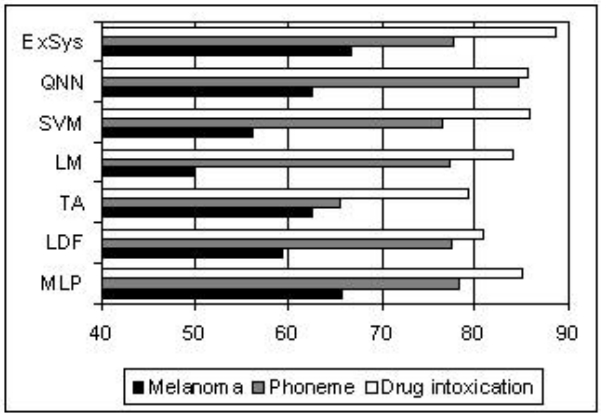

Fig. 4. The performance of different recognition algorithms on three reallife tasks.

In area of forecasting ExSys was compared with multiple linear regression and MatLab fuzzy logic toolbox. There was considered the following task: predictions of magnetic amplitude oscillations in accelerating cavity of a klystron. The necessary data was taken for linear accelerator in DESY, Hamburg. The source information was oscillations on other cavities within the same klystron. The same table was used for learning of both systems. The results of their work on the control sample are shown on Figure 5. The tests show that the methods described above can be successfully used for fuzzy expert systems development. The proposed algorithm for knowledge base generation provides not a great number of rules which are both statistically significant and easily interpreted by experts. The approach focuses on the essence of research problem, not on particular samples, thus preventing the whole system from catastrophic overtraining.

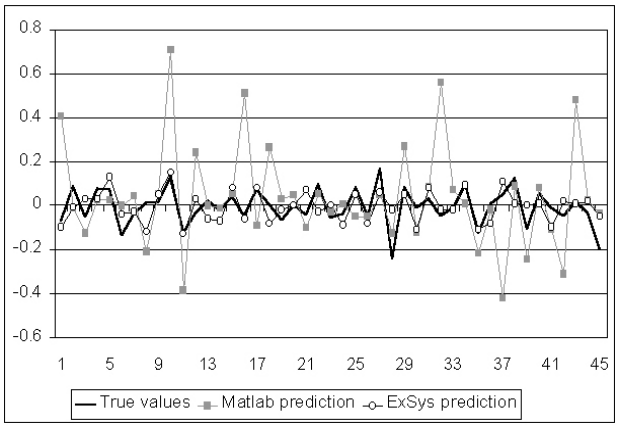

Fig. 5. Oscillations of magnetic field amplitude**.**

The work was supported by the Russian Foundation for Basic Research (grants 04-01-00161, 05-07-90333, 06-01-00492, 06-01-08045).

References:

[1] I. Perfil'eva, Applications of fuzzy sets theory, *Itogi nauki i tehniki*, Vol. 29, 1990, pp. 83-151.

[2] T. Terano, K. Asai, M. Sugeno, *Applied Fuzzy Systems*, 1993

[3] L. Zadeh, *The Concept of a linguistic variable and its application to approximate reasoning*, Elsevier Pub. Co., 1973

[4] T. Ojala, *Neuro-Fuzzy Systems in Control*, Master of Science thesis, Tampere, Finland, 1994

[5] J.-S. R. Jang, C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997

[6] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Construction of Fuzzy Classification Systems with Rectangular Fuzzy Rules Using Genetic Algorithms, *Fuzzy Sets and Systems*, Vol. 65, No. 2/3, 1994, pp. 237-253.

[7] H. Inoue, K. Kamei, K. Inoue, Rule Pairing Methods for Crossover in GA for Automatic Generation of Fuzzy Control Rules, http://citeseer.ist.psu.edu/200265.html

[8] A. F. Gomez-Scarmeta, F. Jimenez, Generating and Tuning Fuzzy Rules Using Hybrid Systems, *Proc. of the 6th IEEE International Conference on Fuzzy Systems*, Vol. 1, 1997, pp. 247-252.

[9] R. L. Rivest, Learning Decision Lists, *Machine Learning*, Vol. 2, No. 3, 1987, pp. 229-246.

[10] Y. Freund, R. E. Schapire, Experiments with a new boosting algorithm, *Proc. 13th Internat. Conf. Mach. Learn.*, 1996, pp. 148-156.

[11] W. W. Cohen, Y. Singer, A Simple, Fast, and Effective Rule Learner, *Proc. 16th Nat. Conf. Artif. Intell*, 1999.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*, 1984

[13] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann, 1993

[14] L. A. Breslow, D. W. Aha, Simplifying decision trees: a survey, *Knowledge Engineering Review*, Vol. 12, No. 1, 1997, pp. 1-40.

[15] K. V. Vorontsov. Lectures on logical classification algorithms. *http://www.ccas.ru/voron/download/LogicAlgs.pdf*, 2006.