GENETIC TUNING FUZZY DEMPSTER-SHAFER DECISION RULES

JAROSŁAW S. WALIJEWSKI AND ZENON A. SOSNOWSKI

Department of Computer Science,
Technical University of Bialystok,
Wiejska 45a, 15-351 Bialystok, Poland
{jarekw, zenon}@ii.pb.bialystok.pl

(Received 10 June 2002; revised manuscript received 19 August 2002)

Abstract: The objective of this paper is to employ the Dempster-Shafer theory (DST) as a vehicle supporting the generation of fuzzy decision rules. The concept of fuzzy granulation realized via fuzzy clustering is aimed at the discretization of continuous attributes. Next we use Genetic for tuning fuzzy decision rules. Detailed experimental studies are presented concerning well-known medical data sets available on the Web.

Keywords: genetic algorithms, fuzzy modelling, Dempster-Shafer theory

1. Introduction

Genetic Algorithms (GA) are problem solving methods, based upon an abstraction of the process of Natural Selection. If Darwinian theory is to be believed, living creatures have come about through the actions of evolution. The simplicity of GA makes them a powerfull tool. The randomly assigned initial pool is usually pretty poor. However, successive generations improve via *Selection* and *Mutation* mechanisms. In each generation the parents are selected to produce new children. The selection of parents is biased by fitness, so that well fit parents produce more children, while very unfit solutions produce no children. Thus, the genes of good solutions begin to proliferate through the population. Small changes (mutations) are made to at least some of the newly born children. Some of these mutations may be harmful. However, this is not significant, because bad mutations will be soon purged by selection. On the other hand, the good mutations will succeed, causing further increases in fitness.

In this study, we discuss the use of Dempster-Shafer theory as a well-rounded algorithmic vehicle in the construction of fuzzy decision rules. The concept of fuzzy granulation realized via fuzzy clustering is aimed at the discretization of continuous attributes. Detailed experimental studies are presented concerning well-known medical data sets available on the Web. Next we use GA to find the best points of division for discretization of continuous attributes. The rules, generated using Fuzzy Dempster-Shafer model (FDSM), were verified by the GA methods. The natural crossover improved by random changes (mutation and selection) can help us to find the best set of rules. Fuzzy modeling is regarded to be one of the possible classification architectures of machine learning and data mining. There is a significant number of studies devoted to generating fuzzy decision rules from sample cases or examples. These include attempts to extend many classical machine learning methods to learn fuzzy rules.

The objective of this paper is to employ the Dempster-Shafer theory as a vehicle supporting the generation of fuzzy decision rules. More specifically, we concentrate on the role of fuzzy operators, and on the problem of discretization of continuous attributes. We show how they can be effectively used in the quantization of attributes for the generation of fuzzy rules.

The material is arranged in the following way. First, we summarize the underlying concepts of the Dempster-Shafer theory and briefly discuss the nature of the underlying construction. By doing so, the intention is to make the paper self-contained and help to identify some outstanding design problems emerging therein. Next we explain essential features of our model. Finally, we report exhaustive experimental studies. This paper is a continuation of our earlier works [1, 2]. Here we apply the theoretical vehicle, introduced in previous research, to new input data in order to find possible area of applications. Our important objective here is to reveal a way in which this approach becomes essential to a more comprehensive treatment of continuous attributes.

2. Fuzzy Dempster-Shafer model

In FDSM [3] we consider rules R_r as:

$$\textit{If } (X_1 \textit{ is } A_{r,1,j_1}) \ldots \textit{ and } \ldots (X_n \textit{ is } A_{r,n,j_n}) \textit{ then } (D \textit{ is } m_r),$$

where X and D stand for input and output, recpectively, and m_r is a fuzzy belief structure, that is a standard belief structure with focal elements $S_{r,p}$ as fuzzy subset of frame of discernment Θ with basic probability assignment $m_r(S_{r,p})$, and $m_r(S_{r,p})$ is the belief, that the conclusion should be represented as class $S_{r,p}$.

2.1. Learning – rules construction

In antecedent construction, let us assume that we have n features (attributes) in antecedents of testing example. We consider a collection of m generic linguistic terms, characterized by membership functions defined in a universe of discourse being a domain of each attribute.

For each element of data t we build a collection:

$$\begin{matrix} A_{1,1,t} & A_{2,1,t} & \cdots & A_{n,1,t} \\ A_{1,2,t} & A_{2,2,t} & \cdots & A_{n,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,m,t} & A_{2,m,t} & \cdots & A_{n,m,t} \end{matrix}, \tag{1}$$

where $A_{i,j,t}$ are the values of j^{th} membership function for i^{th} feature and for t^{th} element of data.

Example

We demonstrate the calculations on the set of synthetic data presented in Table 1.

Table 1. Sample data set

L1	1.0	1.0	1.0	1.0	1
L2	9.0	8.0	8.0	9.0	2
L3	1.0	1.0	3.0	4.0	1
L4	2.0	1.0	2.0	2.0	1
L5	2.0	2.0	2.0	2.0	2
L6	5.0	6.0	7.0	8.0	2
T1	3.0	3.0	3.0	2.0	1
T2	1.0	2.0	2.0	1.0	1
T3	4.0	7.0	7.0	9.0	2
T4	2.0	8.0	7.0	8.0	2

The first six rows (L1–L6) will constitute learning data, while the remaining ones (T1–T4) will form testing data. All the features are numbers from the ⟨0;9⟩ interval. The last column represents the decision class equal to 1 or 2. We will consider four membership quadratic functions uniformly distributed along the space of all attributes. Other membership functions will be discussed in the next section.

According to (1) for row T1 we have:

1.0	1.0	0.0625	0.0156
0	0	0.9375	0.9844
0	0	0	0
0	0	0	0

On the base of (1), for a given data point t we can calculate vectors:

$$A_{\mu,t}: \quad A_{1,\max_1,t} \quad A_{2,\max_2,t} \quad \cdots \quad A_{n,\max_n,t}$$

and

$$I_{c,t}: \quad I_{1,\max_1,t} \quad I_{2,\max_2,t} \quad \cdots \quad I_{n,\max_n,t},$$

called index of membership functions. Here $A_{i,\max_i,t}$ is a maximum value of all the membership functions designed for the feature i, and $I_{i,\max_i,t}$ is the number of the best membership function for feature i.

Then we have the following candidate for a rule:

$$R_t: \quad I_{1,\max_1,t} \quad I_{2,\max_2,t} \quad \cdots \quad I_{n,\max_n,t}.$$

The firing level of the rule is calculated according to the following formula

$$\tau_t = \overset{n}{\underset{i=1}{\phi}} \left(A_{i,\max_i,t}\right),$$

where ϕ means the operator of fuzzy matching (see Section 5.2). The rule candidate is added to rules set if $\phi[\tau_r, m_r] \geq Th$ (where Th threshold value, and ϕ matching operator). This can help to eliminate the worst rule from the final rule set.

More than one rule can have the same antecedent part and it is also possible that the conclusion of these rules are different. Then we have to use appropriate counters $c_{t,1}, \ldots, c_{t,|S|}$, where $|S|$ denotes the power of decision class set. These counters can show us how many data, according to rule pattern, vote for each decision class.

Example

In our sample (T1) the vectors are:
$A_{\mu,1}$: 1.0000, 1.0000, 0.9375, 0.9844
$I_{c,1}$: 1, 1, 2, 2 with counters vector 1, 0

In our sample matching value equals to 0.9229, where multiplication was used as the matching operator $(1.0000\cdot 1.0000\cdot 0.9375\cdot 0.9844=0.9229)$. For the threshold set on 0.75, we obtain a new rule.

The product is a new belief structure on X:

$$\hat{m}_r=\tau_r \bigwedge m_r.$$

Focal elements are fuzzy subsets given as:

$$F_{r,p}(x)=\tau_r \bigwedge S_{r,p}(x)$$

and appropriate distributions of new focal elements are defined as:

$$\hat{m}_r(F_{r,p})=m_r(S_{r,p}).$$

So we can build an aggregate:

$$m=\bigcup_{r=1}^{R}\hat{m}_r.$$

Than for each collection:

$$\mathfrak{I}=\left\{F_{r_1,p_1},F_{r_2,p_2},\ldots,F_{r_R,p_R}\right\},$$

where F_{r_t,p_t} are focal elements of $\hat{m}_r$, we have focal element E of m described as:

$$E=\bigcup_{t=1}^{R}F_{r_t,p_t}$$

with appropriate probability distribution:

$$m(E)=\prod_{t=1}^{R}m(F_{r_t,p_t}).$$

At this point, the rule generalization process is complete.

Example

Our sample data produce the following rule set.

	$I_{1\max}$	$I_{2\max}$	$I_{3\max}$	$I_{4\max}$	C_1	C_2	m
R1:	1	1	2	2	1	0	2.5000
R2:	1	1	1	1	2	1	2.0833
R3:	4	4	4	4	0	1	1.2500
R4:	2	3	3	4	0	1	1.2500

The first four elements are numbers of the best membership function for proper features, the next two are counters for decision classes and the last one is a probability distribution. Let us observe that rule R2 covers the data L1, L 4 and L5. L1 and L4 produce decision class C_1 but L5 decision class C_2.

Now we can move to the testing of new rules.

2.2. Test

In testing we ignore the value from the last column in Table 1, that is decission class number, because our goal is to calculate it.

To compute the firing level of rule k for a given data

$$X_k:\quad X_{1,k}\quad X_{2,k}\quad \cdots\quad X_{n,k}\quad D_k,$$

where $X_{i,k}$ is the feature's value, and D_k is the conclusion decision class that we have to compare with the result of inference, we build a rule matrix $\mu_{k,t}=\underset{i=1}{\overset{n}{\phi}}\,(A_{i,l,k}(X_{i,t}))$,

$l = I_{i,\max,k}$. We are interested only in active rules *i.e.* rows with matching value $\mu_{k,t} > 0$.

Example

In the test we will demonstrate calculations on

L5	2	2	2	2
T1	3	3	3	2

For sample data L5 we have two active rules:

R1:	1	1	2	2	1	0	0.859375	0.859375	0.609375	0.609375	0.274242
R2:	1	1	1	1	2	1	0.859375	0.859375	0.859375	0.859375	0.54542

The first four elements are the rule pattern, the next two are the counters for decision classes. The next four numbers are the values of appropriate membership function. The number 0.859375 is the value of the first membership function, according to the first number in the rule, on the first feature. The next three numbers are calculated in a similar way. The last numbers in the above rows is the matching value for the rule. It has been calculated by matching operator for the values of membership function.

We focused only on the rows with matching value grater than zero. For sample data T1 we have:

R1:	1	1	2	2	1	0	0.4375	0.4375	0.9375	0.6094	0.1093
R2:	1	1	1	1	2	0	0.4375	0.4375	0.4375	0.8594	0.0720

For each collection of F_{r_t,p_t} focal elements $\hat{m}_r$ we define an aggregate:

$$E = \bigcup_{t=1}^{R} F_{r_t,p_t}$$

with basic probability assignment:

$$m(E) = \prod_{t=1}^{R} m(F_{r_t,p_t}).$$

The results of classification are D is m, with focal elements $E_k (k = 1,\dots,R^{|S|})$ and distribution $m(E_k)$. Those results are calculated using focal elements and appropriate counters $c_{t,1},\dots,c_{t,|S|}$.

Example

For sample point L5 and T1 the counters are 3, 1, and 3, 0, respectively.

Then we perform defuzzification according to COA method [4]:

$$\bar{y} = \sum_{k=1}^{R^{|S|}} \bar{y}_k m(E_k),$$

where $\bar{y}_k$ are defuzzified values for focal element E_k defined as:

$$\bar{y}_k = \frac{\sum_{1 \le t \le n} x_t \mu_{k,t}(x_t)}{\sum_{1 \le t \le n} \mu_{k,t}(x_t)}.$$

In the next step, the rules structure is simplified to:

$$\textit{If antecedent}_r \textit{ then } (D \textit{ is } H_r),$$

where $H_r = \left\{\frac{1}{\gamma_r}\right\}$ is a singleton fuzzy set for factor $\gamma_r = \sum_{p=1}^{|S|} \bar{y}_p m_r(S_{r,p})$.

Example

For both L5 and T1 we calculate decision class 1. It is correct for T1, but wrong for L5. The values of H_r are 0.4283 and 0.4800, respectively.

3. Empirical learning for FDS model

In this section we compare and analyze the performance of several membership functions and matching operators. We start from a standard solution used in the introduction to fuzzy modelling, then we consider more complicated models. We compute results for the following membership functions: *Linear*, *Quadratic*, *Gaussian*, and *Fuzzy c–Means* (*FCM*). We concentrate on *Minimum*, *Multiply* and *Implication* as matching operators. The most valuable is comparing the results of all calculations. In the end of this section we show some results of experimental research.

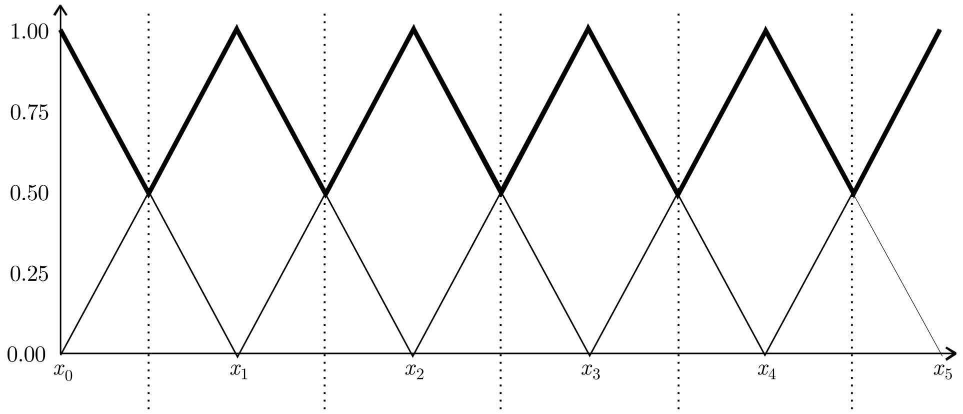

Figure 1. Linear function

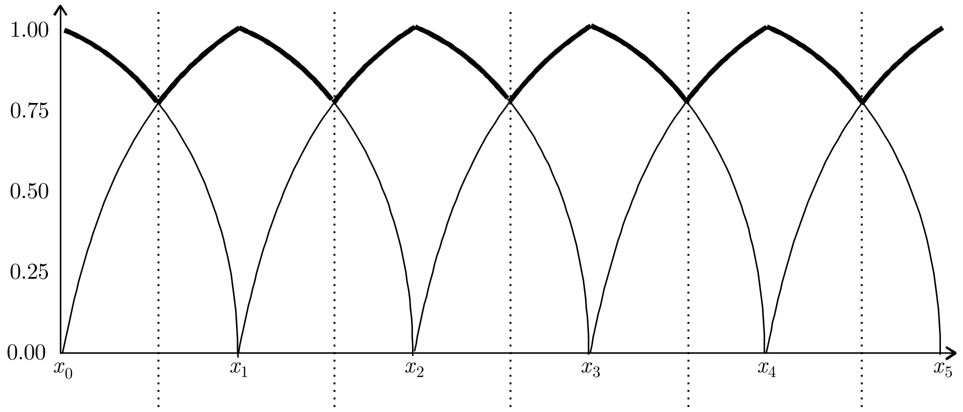

Figure 2. Quadratic function

3.1. Membership functions

The membership function makes possible the division of data into n intervals. It is a way of discretization of the input data. Hence, we get the best result for continuous data or for data with several discrete (nominal) values. If we have single discrete or binary data, then results of the proposed model are not good enough.

The choice of membership function has great influence on the quality of rules being received. Although the quantity of rules is different, the quality of classification is comparable.

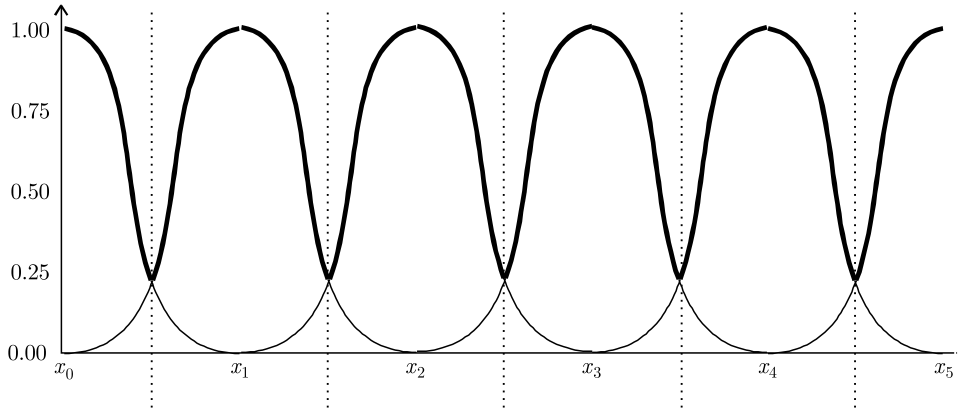

Figure 3. Gaussian function

The most interesting membership function was generated by *FCM*) algorithm. The main idea of FCM is described in [4]. Figures 1, 2 and 3 show the universe of discourse divided into six intervals (x_0–x_5) – as in our experiment. The results of experimental research with membership functions have been summarized in Table 2.

Table 2. Membership function

	Function	Formula	Features
1	*Linear* *Triangle* (see Figure 1)	F1: $Y=(X-x_{i+1})/(x_i-x_{i+1})$ F2: $Y=(X-x_i)/(x_{i+1}-x_i)$	The simplest one. The result are not so good.
2	*Quadratic* (see Figure 2)	$Y=1-(X-x_i)^2/(x_{i+1}-x_i)^2$	Also calculated in a simple way. The result are better than in the case of *Linear* function.
3	*Gaussian* (see Figure 3)	$y=e^{-\left(\frac{X-x_i}{x_{i+1}-x_i}\right)^2}$	Exponential function. The usage of it leads, in general, to proper conclusions, especially in learning sample.

3.2. Matching operators

It was shown in our experiments that a matching operator applied to data sample with existing rules plays a very important role in the accuracy of diagnoses. It occurred as early as the rules were generated. A matching operator influences the quality of the generated rules. Of course, this quality has secondary means, but in general, the more rules the better accuracy.

From the analysis of the results of experiments shown in Table 4 (see Section 5), we can infer that the most powerful operator is *Implication*. This not all true, because Table 4 shows the results only for one fixed threshold value. It is not optimal in all instances, especially for *Multiply* operator. The change of the threshold value (*e.g.* to 0.25) gives almost the same results as for *Implication* operator. Anyway, the choice of the threshold value is of minor importance here, but it can have influence on the result of the receiving of rules. Of course, we can not analyze the threshold value without keeping in mind the features of the membership function and the number of intervals. The choice of the threshold value will be subject of future works.

The results of the investigation of various matching operators are collected in Table 3.

Table 3. Matching operators

	Name	Formula	Features
1	*Minimum*	$\max(0,\min(x,y))$	The simplest case. In many experiments the results are good.
2	*Multiply*	$\max(0,x\cdot y)$	It is a special case of *Implication*, with $\alpha=-1$. The results are better than in the case of *Minimum* operator. The setting of threshold value is important.
3	*Implication*	$\max(0,(\alpha+1)\cdot(x+y-1)-\alpha\cdot x\cdot y)$	Function seems to be complex. The best results can be obtained. Setting of constant α – to be investigated in future.

4. Genetic tuning of the given rules

In the previous section we divided the space of all possible values of each feature. We assume that all intervals are equal. But, in general, it is not a good decision. To find the points of the division we should consider the decision class for the point of data. But how we can we find a decision class before running the application? Of course it is not possible to say anything about the worth of the rule set before testing. But if we have the results of the rules then we can use these results to tuning the rule set. The points of division are of particular interest. In general it is difficult to find the direction of points' changes. Almost all of them seem to be the same. We can move them randomly, but every time we have to exam the given rules. If we find the rule set that can produce more accurate conclusion, then we can use these rules in the future. In the opposite case we reject the rules.

This algorithm is a simplification of Genetic Algorithms (GA). The natural way is to use the GA theory to find the best division points.

First we start from the basic FDSM. Initially, all the division intervals of the data features are equal. Although not optimal, it is a quite good choice. Next we generate new division points at random, and then we can test them. We generate rules by FDSM algorithm using the new division points. The accuracy of the rules is a survival function. We use it to test the data set. We generate the next generation of population using the GA methods. Of course instead of only one, we can generate more random divisions, and crossover them. In the next point we use previous population to produce a new one. We observed, that after a few generations the quality of calculated rules does not change any more. Thus, the cost of rule generation is only a few times higher than using basic FDSM. When we analyze the results of the classification using the rules set. Let us illustrate it on a sample Dermatology data set [5]. Data contain 366 records, 244 of them we use to learn, 122 for test. There is no floating data, but 33 discrete and only 1 binary. The rules generated by FDSM have the best accuracy 51.64% for *Gaussian* function and *Multiply* operator. After 10 generations of GA, the conclusion was good almost in 100% of cases of testing data. For *Gaussian* function and *Implication* operator the accuracy indicator achieved 100%.

5. Experimental studies

In this section we compare and analyze the performance of several membership functions and matching operators. We start from a standard solution used in introduction to fuzzy modeling then we consider more complicated models. We compute results for the following membership function: *Linear, Quadratic* and *Gaussian.* We concentrate on *Minimum*, *Multiply* and *Implication* as matching operators.

Some results of experimental research are shown in Table 4. We fixed here count of membership functions on 6 and threshold value on 0.75. The features of all data are described in Table 5. All data sets have been divided into two parts: learning (training) data (about 2/3 of the entire data set) and testing data (remaining 1/3).

Table 4. Experimental result for GA

		Linear		Quadratic		Gaussian		Decision Trees
		Test	GA	Test	GA	Test	GA	Test
Minimum								
	Iris	93.33	93.33	96.67	96.67	93.33	93.33	91.30
	Ulcers	9.52	16.67	28.57	59.52	52.38	59.52	—
	Diabetes	0.00	2.70	37.84	62.16	56.76	82.22	—
	Derm.	0.00	5.55	66.39	66.39	50.82	94.26	87.50
	EKG	5.56	5.56	38.89	66.67	66.67	99.44	59.00
Multiply								
$\alpha=-1$	Iris	0.00	0.00	96.67	96.67	93.33	93.33	
	Ulcers	0.00	4.76	2.38	61.90	2.38	59.52	
	Diabetes	0.00	0.00	29.73	64.86	54.05	67.67	
	Derm.	0.00	5.55	66.67	77.87	51.64	88.52	
	EKG	5.56	5.56	5.56	72.22	11.11	100.00	
Implication								
$\alpha=-20$	Iris	83.33	83.33	96.67	96.67	96.67	96.67	
	Ulcers	9.52	9.52	28.57	57.14	52.38	64.28	
	Diabetes	0.00	2.70	37.84	67.57	56.76	96.30	
	Derm.	0.00	5.55	86.89	86.89	50.82	100.00	
	EKG	0.00	5.56	38.89	77.78	66.67	88.89	

Table 5. Features of data

	Iris	Ulcers	Diabetes	Breast Cancer Wisconsin	Dermatology	Echocardiogram
Data	150	122	107	683	366	62
Learn	120	80	70	466	244	44
Test	30	42	37	217	122	18
Decision class	3	5	2	2	6	2
Total Features	6	11	8	9	34	11
Continuous	1	5	1	0	0	8
Discrete	5	5	2	9	33	0
Binary	0	1	5	0	1	3

The learning data have been used to generate the rule set. Testing data have been applied to test the produced rule set. To obtain reliable results, we carried out the experiment several times.

The most valuable is comparing the results of all calculations. In the end of this section we show some results of experimental research. To apply GA methods we use standard procedures from package SUGAL [6] written by Andrew Hunter from the University of Sunderland (UK). Some results of experimental research have been shown in Table 4. We compare the results of our research with standard decision trees algorithm [7–9]. For all data sets, we get better results using *Gaussian* function, and in a few points of *Quadratic* function we obtained also better accuracy.

6. Conclusions

The study has focused on the use of Fuzzy Dempster-Shafer model for generating fuzzy decision rules. Fuzzy sets are useful in discretization of continuous attributes. The approach is discussed in the concrete applications of two real medical data sets (especially to problems of identification of diseases) and several well-known data sets available on the Web. The results are used to classify objects. The vehicle of Genetic Algorithms, as additional approach for generation of the rules give us better indicator of accuracy. It can be used in the case of features with many possible values.

Acknowledgements

Support from the Technical University of Bialystok, grant W/WI/8/02, is gratefully acknowledged.

References

[1] Sosnowski Z A and Walijewski J S 1999 *Proc. 13th European Simulation Multiconference*, Warsaw, Poland, p. 419

[2] Sosnowski Z A and Walijewski J S *Proc. 7th Workshop of the Polish Society of Computer Simulation*, Zakopane, Poland, p. 273

[3] Bianaghi E and Madella P 1997 *Proc. 7th IFSA World Congress*, Prague, Czech Republic, **1** 197

[4] Bezdek J C, Sabin M J and Tucker W T 1987 *IEEE Trans. On System, Man and Cybernetics* **17** (5) 873

[5] http://www.ics.uci.edu/m̃learn/MLRepository.html

[6] Hunter A 1995 *SUGAL Genetic Algorithm Simulator*, University of Sunderland, UK

[7] Pedrycz W and Sosnowski Z A 2000 *IEEE Trans. on Systems, Man and Cybernetics* **A 30** 151

[8] Quinlann J R 1986 *Machine Learning* **1** 81

[9] Weber R 1992 *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp. 265–268