A STUDY ON THE BEHAVIOUR OF THE ALGORITHM FOR FINDING RELEVANT ATTRIBUTES AND MEMBERSHIP FUNCTIONS

NOZĪMĪGĀKO ATRIBŪTU UN PIEDERĪBAS FUNKCIJU MEKLĒŠANAS ALGORITMA UZVEDĪBAS PĒTĪŠANA

M. Gasparoviča, L. Aleksejeva

Keywords: ***Iris data, relevant attributes, membership functions, decision table, IF-THEN fuzzy classification rules***

Abstract - One of the most recent approaches in machine learning is fuzzy rules usage for solving classification problems. This paper describes the algorithm for finding relevant attributes and searching for membership functions. Experimental results are used to clarify – which data sets can be used to automatically gain primary membership functions from primary data. This quality – gaining of membership functions – is one of the pros of the algorithm, because it eases resolution of classification task. The ability to use it with fuzzy data is one more merit. As a result, there are obtained reliable fuzzy classification rules to separate classes. By reconstructing primary membership functions also the number of IF-THEN rules gained from decision tables is reduced up to three times. Four experiments are conducted with different training and testing data set sizes. Conclusions are made about the optimal size of the training and testing data set that is necessary for achieving better results as well as about the data this algorithm is appropriate for. Finally, possible directions for further research are outlined.

Introduction

Fuzzy systems that are able to extract IF-THEN rules from numerical data have been developed relatively lately. Nevertheless they have gained popularity in the recent years because of their properties to describe the task issues with the uncertainty typical to the real world. Most of the methods of the fuzzy systems, for example the fuzzy PRISM algorithm [1], need previously defined membership functions to perform the training part of an algorithm.

In 1994 the algorithm for automatic definition of IF-THEN rules and membership functions was developed [2]. The data processing was difficult because of decision table and primary membership functions complexity, so this approach was improved by finding relevant attributes [3]. Its main advantage is the automatic extraction of the primary membership functions. Experiments with this algorithm are carried out on the Iris data set [4, 5]. The goal of the experiments is to clear up the dependency of the results on the size of training and testing sets as well as establishing other suitable data sets and advice which features of the data makes a data set suitable for this algorithm.

The idea of the algorithm and its work is step-by-step described in the first section of the paper. The next section describes the data set used and explains why it was chosen for working with the algorithm for finding relevant attributes and membership functions. The experiments carried out in the study and the results obtained are also discussed. The ending part of the paper contains the conclusions about the performance of the algorithm and the requirements for a data set to be used with this algorithm as well as directions for future research.

Algorithm Description

The algorithm for finding relevant attributes and membership function consists of three main stages – find relevant attributes, then build initial membership functions, and at the end derive decision rules [3, 6]. Next we will describe the algorithm step by step in more detail.

Stage I. Find relevant attributes:

1. Sort each attribute value of A_i, which appear in training instances in ascending order.
2. For each attribute A_{ij}, count how many instances belong to the same classes.
3. Sum how many numbers of A_{ij} with attribute values belong to only one class.
4. Calculate the fitness degree f_i of each attribute. Several approaches can be used for that purpose, for example, the first method described [3, 6] is based on average fitness (Equation 1):

$$f_i = \frac{t_i}{n}, \tag{1}$$

where

t_i – the number of instances of attribute A_i that belong to only one class,

n – the total number of training instances.

The second method uses the concept of entropy:

$$f_i = \left\{ -\frac{1}{q_i} \sum_{j=1}^{q_i} \sum_{k=1}^{p} \left[\left(\frac{D_{ijk}}{\sum_{k=1}^{p} D_{ijk}} \right) \log_p \left(\frac{D_{ijk}}{\sum_{k=1}^{p} D_{ijk}} \right) \right] \right\}, \quad (2)$$

where

D_{ijk} – number of instances that fall within attribute value A_{ij};

p – total number of output class;

q_i – number of values of A_i.

5. Sort the attributes in ascending order of fitness degrees;
6. Select relevant attributes using procedure (3), using threshold β_1. An attribute with higher fitness degree can be taken as relevant.

```
PROCEDURE Select_Relevant_Attributes:
$l = 1$, $Total\_E = 1$
DO WHILE $l \le m$
    $Total\_E = Total\_E \cdot (1 - f_l')$        (3)
    IF $Total\_E \le \beta_1$ THEN exit
    ENDIF
$l = l + 1$
ENDDO
```

Stage II. Build initial membership functions:

7. Find the initial default group number G of each relevant attribute as $G = 1 + 3.3 \log n$;
8. Find the range of each attribute $R_i = \max(A_i') - \min(A_i')$;
9. Find the group interval of each attribute $H_i = R_i / (G - 1)$;
10. Extend the possible minimum attribute value as $V_i = \min(A_i') - H_i / 2$;
11. Divide the possible range of each attribute into G groups;
12. Find the typical points of triangular membership function (see Fig.1) a_{ij}, b_{ij} and c_{ij} for each initial membership function (Equation 4):

$$b_{ij} = \sum_{s=1}^{r_{ij}} \frac{A_{ij}'(I_{ijs})}{r_{ij}}, \quad a_{ij} = b_{i(j-1)} \quad c_{ij} = b_{i(j+1)}, \quad (4)$$

where

$A_{ij}'(I_{ijs})$ represents the attribute value of instance I_{ijs} in A_{ij}';

r_{ij} – number of instances that fall into attribute value range A_{ij}'.

Stage III. Derive decision rules:

13. Construct an initial decision table using relevant attributes ranges;

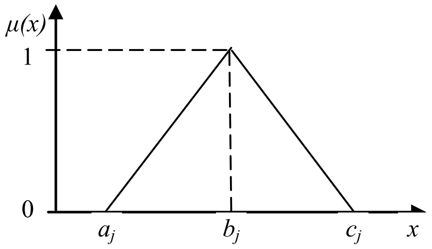

Fig. 1. Triangular membership function

14. Simplify the initial decision table by eliminating redundant and unnecessary table cells;
15. Rebuild membership functions using initial decision table cell merging operations;
16. Derive decision rules from the modified decision table.

Description of the Data Set

This research uses the most popular data set of the UCI Machine Learning Repository – Fisher's Iris flowers [4]. The part of Iris data set used in the research for classification is shown in Table 1.

Field of this data set is nature, all attributes are real numbers. The number of records in the data set is 150; they can be classified into three classes that denote classes of Iris plant – SETOSA, VERSICOLOUR, VIRGINICA, each class containing 50 records of the training data set. The Iris data set consists of four attributes: A_1 - SEPAL LENGTH (cm), A_2 - SEPAL WIDTH (cm), A_3 - PETAL LENGTH (cm) and A_4 - PETAL WIDTH (cm).

Table 1

IRIS data set (fragment)

SETOSA				VERSICOLOR				VIRGINICA			
A_1	A_2	A_3	A_4	A_1	A_2	A_3	A_4	A_1	A_2	A_3	A_4
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
…				…				…			

This data set is often chosen for classification tasks because the classes are clearly divided (see Figure 2). The class SETOSA is easily distinguished, but there can be problems with classes VERSICOLOUR and VIRGINICA, which overlap in the attributes A3 and A4 interval (4.5; 1.0) – (5.5; 2.2).

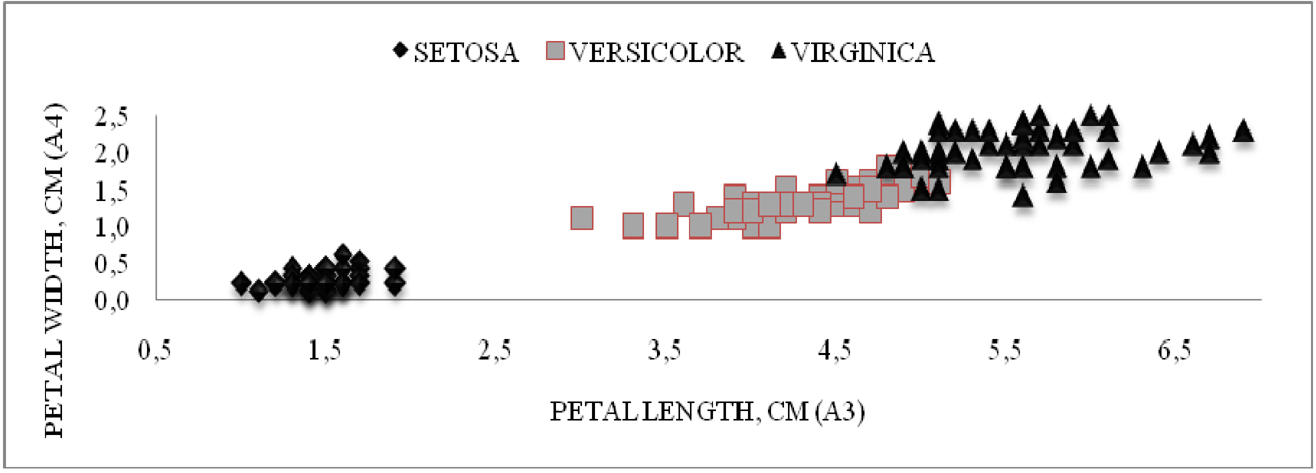

Fig. 2. Iris Flower data set representation

Experiments

Within this research four experiments were carried out using different sizes of training and testing data sets and different structures of the data sets creating them by randomly choosing the records from the primary data set.

Description of Experiments

In the first experiment the ratio between training and testing data was the usual – 70% of records in the training data and 30% in the test data. In the second experiment the training set was expanded and the test data set was reduced. Finally, in the third experiment 30 random records were discarded; the training data set consisted of 105 records and the test set consisted of 15 records.

The fourth experiment was conducted using *n*-fold cross-validation. Accordingly, the data set used in this experiment was split into three subsets of almost equal size: M_1, M_2 and M_3. The used split condition – belonging to a subset is determined randomly but all three subsets have to have equal number of records. The training set consists of 2/3 of all records and the test set consists of the remaining 1/3. The algorithm is executed as follows: I iteration – training set consists of M_1 and M_2, test set consists of M_3, II iteration – training set consists of M_1 and M_3, testing is conducted on M_2; III iteration – training set consists of M_2 and M_3, test set is M_1. Cross-validation guarantees that every record will be used (n–1) times in the training and once in the testing (where n is the number of subsets). [7]. As a result, a classifier is described, but the total classification error and the accuracy are calculated as the mean of all three iterations (classifiers).

Data Processing

Fitness degree for each attribute was calculated using the first method (see Equation 1) – using the number of attribute value representing each class and the number of records in the training data set.

Coefficient $\beta = 0.1$ was used to choose the relevant attributes and while executing the procedure it was taken into account (see Equation 3). Within the experiments conducted in this study, in one experiment (3-fold cross-validation) in two iterations out of three there was only one relevant attribute, in all other cases there were two relevant attributes and the membership functions defined by the combination of the values of these two attributes indicated class of a record. The mentioned relevant attributes are PETAL LENGTH (A_3) and PETAL WIDTH (A_4) labelled as A_1' and A_2', respectively.

When constructing the membership functions, we build a multi–dimensional decision table, where each dimension represents a relevant attribute (see Table 2). There were some conflicting cells having instances at the same position (attribute value interval) but having different output classes. In the experiments, conflicts were resolved using the so-called „majority voting" strategy [8]. The underlying idea of the strategy is as follows: if there are two records with class O_1 and six records with class O_2 in the same interval, then the class O_2 is chosen. It is also possible to use other strategies to solve conflicts like giving priority to a certain class, or using a measure of the ”interestingness” of each class [8].

When decision basis is acquired and all conflicts are solved it can be simplified by merging several intervals which consist of records with the same class into one interval. In similar manner, all intervals which hold no class can be merged. This way from initial 16 by 16 intervals, the table is scaled down to 6 by 6 intervals.

This way it is obvious (see Table 2), how to extract a simplified decision base by performing some simple activities (for example, merging intervals 1 to 3) and reconstructing the initial membership functions. In e Table 3 the first record shows the results obtained with the training data set proposed by the authors which consists of 150 records, but there is no certainty about the test data and the error in classifying the data.

The rules extracted from the decision base are tested using the values of the relevant attributes and checking if the record that is being tested belongs to any class with its values of the attributes. As a result, it is possible to compare the results of the algorithm to the real classes.

A method of calculating accuracy is used to distinguish the accuracy of the experiment – the error is calculated using the calculation of the total classification error. To do the calculus, it is first needed to create a confusion matrix that holds the numbers of correctly and incorrectly classified records [7].

Table 2

Initial decision table

		A_2'									
		Beg.	End	1	2	3	…	12	13	14	15
	Beg.			0.8	1.2	1.6	…	5.1	5.5	5.9	6.3
	End			1.2	1.6	2.0	…	5.5	5.9	6.3	6.7
	1	0.0	0.2	O_1	O_1						
	2	0.2	0.3	O_1	O_1	O_1					
	3	0.3	0.5		O_1	O_1					
	…	…	…	...	...	...	...	...	...	...	...
	12	1.8	1.9				...	O_3	O_3	O_3	
	13	1.9	2.1				...	O_3	O_3		O_3
	14	2.1	2.3				...		O_3		O_3
A_1'	15	2.3	2.4				...	O_3	O_3		

Table 3

Experimental results

Experi-ment No.	Training data set	Testing data set	Incorrectly classified examples	Accuracy	Classification error	Comments
0.	150			0.9667	0.0333	The authors proposed [4, 7]
1.	105	35	2	0.9429	0.0571	Classical – 70% in training data set, 30% in testing data set
2.	135	15	1	0.9333	0.0667	Bigger training data set than in previous experiment
3.	105	15	1	0.9333	0.0667	Eliminated 30 random examples
4.	78	42	1	0.9579	0.0421	Three fold cross validation (with data sets from previous experiment)

Analysis of Experimental Results

As shown in Table 3, the highest accuracy among all the experiments that were carried out in this research was in the case with three fold cross–validation (see Experiment No.4 in Table 3). It is logical because only in that experiment every record was used for both training and testing.

The next best result is for the second experiment – the classic ratio of the training (70%) and test (30%) data, despite the fact that it is the only experiment with 2 incorrectly classified records of the test set. However, the test set in this experiment was relatively large with 35 records (other experiments, with the exception of cross-validation, had significantly smaller test sets) and it explains the accuracy level in this case.

We can draw a conclusion that in these experiments a larger test set leads to better accuracy, but for this to be convincing there should be a greater number of experiments.

Conclusions

The algorithm is convenient and most of the work involved is to establish which attribute (or attributes) is relevant. Afterwards when attributes of less relativity have been eliminated, calculations are made only using the relevant attributes. Of course, some information is lost this way but the time attained using the algorithm to find relative attributes and membership functions is greater than the loss. Another great benefit of this algorithm is its ability to automatically construct the initial membership functions using the training data. It excludes matters of chance and factor of subjectivity while constructing membership functions because there is no way to interpret membership functions – they are calculated using mathematical formulas using the initial data.

It is not surprising that the best results of all experiments were for cross-validation because it is the only way to use all records for both training and testing.

We can draw a conclusion that the classical division with 70% training data and 30% test data also leads to a good result. The difference between this experiment and the three fold cross-validation is so small that the work put in for cross-validation does not pay off.

In general this algorithm is convenient and fast to calculate, 70% of the calculations are needed in the first phase of the algorithm – while finding the relevant attributes, because all attributes and their values have to be taken into account to discover if the exact attribute is relevant.

Another great benefit is the ability to decrease the decision base because from the initial 16 intervals only five remain after carrying out the simplification operations. This way also the number of decision rules becomes smaller and it makes the testing faster – there are less rules to check. This could be one of the uppermost benefits why this algorithm should be used.

These experiments showed that data should be continuous to use this algorithm. Categorical data do not work. The data cannot have small predefined values like three values in the interval [-1, 1].

Also if the initial data hold a large number of attributes it will take a lot of time and work to distinguish which are the most relevant. Of course, computer programs like WEKA can be used to select the most relevant attributes but it can happen that some relevant attributes are eliminated and the algorithm is not able to carry out its work correctly.

Future Research

Future research plans include studying the algorithm for finding relevant attributes and membership functions and continuing the research using other data sets. It is planned to use data accessible at UC Irvine Machine Learning Repository [4].

The comparison of the gained results with those of other algorithms using the same data set is intended. The algorithm PRISM [9] and its modified version Fuzzy PRISM [1] are within interest range.

Also it is planned to search for other algorithms that find membership functions and are able to determine belonging of a record to predefined classes using membership [10].

Acknowledgments

Thanks to Dr.habil.sc.comp. Professor Arkady Borisov for help and support.

References

1. Wang C. H., Liu J. F., Hong T. P., Tseng S.S. A fuzzy inductive learning strategy for modular rules. Fuzzy sets and Systems. – Vol. 103 (1999), pp. 91–105.
2. Hong T. P., Lee C. Y. Learning fuzzy rules and membership functions from training examples. The 1994 International Computer Symposium, 1994, pp. 351–356
3. Hong T. P., and Chen J. B. Finding relevant attributes and membership functions. Fuzzy Sets and Systems. – Vol. 103, No. 3 (1999), pp. 389-404.
4. Iris data set: UCI Machine Learning Repository USA [Electronic resource] - http://archive.ics.uci.edu/ml/datasets/Iris. – Resource viewed in May 2009
5. Бэстенс Д.-Э., Ван Ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки: Принятие решений в торговых операциях. – Москва: Научное издательство ТВП, 1997, 236 с.
6. Finding relevant attributes and membership functions: NTUST CSIE Taiwan [Electronic resource]. – http://neuron.et.ntust.edu.tw/homework/88/fl/88homework/Relevant Attributes.htm. – Resource viewed in October 2005.
7. Datu ieguve: Pamati / A. Sukovs, L. Aleksejeva, K. Makejeva [u.c.]. – Rīga: SIA Drukātava, 2007, 120 lpp.
8. Bramer M. Principles of Data Mining. – London: Springer, 2007, 343 p.

9. Cendrowska J. PRISM: an Algorithm for Inducing Modular Rules. Internat. J. Man Machine Stud. – Vol. 27 (1987), pp. 349–370.
10. Chen S.-M., Fang Y.-D. A New Method to Deal with Fuzzy Classification Problems by Tuning Membership Functions for Fuzzy Classification Systems. Journal of the Chinese Institute of Engineers. – Vol. 28, No.1 (2005), pp. 169–173.

Madara Gasparoviča received her diploma of Bc. sc. ing. in Information Technology from Riga Technical University in 2008. Now she is the second course MSc student of Information Technology program at Riga Technical University. Her interests include decision support systems, data mining tasks and modular rules.

Ludmila Aleksejeva received her Dr. sc. ing. degree from Riga Technical University in 1998. She is associate professor in the Department of Modelling and Simulation of Riga Technical University. Her research interests include decision making techniques and decision support systems design principles as well as data mining methods and tasks.

Madara Gasparoviča, Ludmila Aleksejeva. Nozīmīgāko atribūtu un piederības funkciju meklēšanas algoritma uzvedības pētīšana

Viena no relatīvi jaunajām pieejām mašīnapmācībā ir izplūdušo likumu pielietošana klasifikācijas problēmu risināšanā. Darbā aprakstīta nozīmīgo atribūtu un piederības funkciju meklēšanas algoritma darbība, ar eksperimentiem pētīta šī algoritma uzvedība, lai noskaidrotu kādām datu kopām šis algoritms, kurš automātiski no sākuma datiem var iegūt sākotnējās piederības funkcijas, ir derīgs. Tieši šī īpašība – automātiska piederības funkciju iegūšana ir viens no lielākajiem šī algoritma plusiem, jo krietni atvieglo klasifikācijas uzdevuma atrisināšanu. Vēl viens šī algoritma pluss ir iespēja strādāt ar izplūdušiem datiem, un rezultātā iegūt precīzus likumus, kas atdala klases vienu no otras. Pārkonstruējot sākotnējās piederības funkcijas arī JA – TAD likumu skaits, kas iegūts no lēmumu tabulām, tiek samazināts pat trīs reizes. Darba izstrādes gaitā tika veikti četri eksperimenti ar dažādiem apmācības un testa kopas lielumiem. Rezultātā izdarīti secinājumi par to, kāds testa kopas un apmācības kopas lielums ir optimāls, lai iegūtu labākus rezultātus. Kā arī eksperimentu gaitā iegūti secinājumi par rekomendācijām kādiem datiem šis algoritms ir piemērots, lai sasniegtu augstas kvalitātes rezultātus. Doti arī nākotnes pētījumu iespējamie attīstības virzieni.

Мадара Гаспаровича, Людмила Алексеева. Анализ поведения алгоритма поиска значимых атрибутов и функций принадлежности

Использование нечетких правил для решений проблем классификации является одним из относительно новых подходов в задачах машинного обучения, достоинство которого заключается в возможности анализа и разделения примеров, принадлежащих различным классам. Статья посвящена описанию процедуры нахождения значимых аттрибутов и алгоритма определения функций принадлежности. Результаты экспериментов служат основанием для выявления особенностей множеств данных с тем, чтобы они могли быть использованы для автоматического определения начальных функций принадлежности из исходных данных. Это качество – получение функций принадлежности – является одним из достоинств данного алгоритма, поскольку упрощает решение задач классификации. В результате работы алгоритма определяются нечеткие классификационные правила, достаточно четко разделяющие объекты на классы. Преобразование исходных функций принадлежности путем их слияния приводит более чем к трехкратному уменьшению числа полученных из таблиц решений правил *IF-THEN*. В ходе исследований были проведены четыре эксперимента с различными размерами обучающей и тестовой выборки, дано заключение об оптимальных для получения лучшего результата размерах выборок. Наряду с этим, сформулированы рекомендации к множествам данных, на которых этот алгоритм может быть применен, а также определены направления дальнейших исследований.