INTERNATIONAL JOURNAL OF FUZZY SYSTEMS and ADVANCED APPLICATIONS
DOI: 10.46300/91017.2022.9.10 Volume 9, 2022

Automatic Generation of Fuzzy Classification Rules from Data

Mohammed Al-Shammaa, Maysam F. Abbod
School of Engineering and Design, Brunel University, London, UB8 3PH, UK

Abstract—In this paper, we propose a method for automatic generation of fuzzy rules for data classification. The proposed method is based on subtractive clustering optimized using genetic algorithm. It searches for the FIS structure and number of rules that have the highest fitness value. Multiple performance measures are incorporated into the fitness function to address the problem of imbalanced data. Fitness function includes both training and validation to avoid data over-fitting. Classification performance of the proposed method is evaluated using different data sets and results are compared to those of a number of models generated by fuzzy cmeans clustering with various cluster numbers. Results show that the proposed method has better accuracy and a well compromised sensitivity and specificity.

Keywords—Fuzzy rules generation, Genetic algorithm, Neurofuzzy system, Subtractive clustering.

I. Introduction

Fuzzy inference system (FIS) is a computational technique that relies on fuzzy logic [1] for performing input-output mapping. FIS is featured by its interpretability, ability to deal with vagueness and incompleteness of knowledge and the ability of describing non-linear relationships between input and output. However, it lacks learning capability and systematic design [2].

One of the most important issues of designing FIS is determining the structure of its rule base that best suits the problem. In most classification problems, it is difficult for human experts to provide sufficient information required to generate fuzzy rules. Therefore, computational methods are used to generate fuzzy rules from data automatically.

Several methods for extracting fuzzy rules from data have been proposed. Data clustering is one of the most widely used approaches for rule base generation. Data clustering methods aim to group data into clusters based on a similarity measure. Fuzzy sets of an input variable can be obtained from data clusters by projecting each cluster to the dimension that corresponds to the input. For example, subtractive clustering method presented in [3] is used to extract fuzzy rules from data. This method selects the data point with many neighboring data points as a cluster center and associates the neighboring data points to this cluster. Another clustering method used to generate fuzzy rules is fuzzy clustering [4]. In this method, data elements can be associated to more than one cluster. Each data point is assigned a set of membership levels which indicate the strength of its association to one or more clusters.

The use of optimization algorithms for fuzzy rules generation is suggested by many studies. For example, genetic algorithm is used to simultaneously estimate rule base structure and parameters of the fuzzy model from data [5]. Also, particle swarm optimization is used to obtain the antecedents, consequences, and connectives of the fuzzy rules base [6]. The use of fuzzy genetic algorithm for automatic fuzzy rule generation is investigated in [7].

In addition to data clustering and optimization algorithms, several methods are suggested by the literature. A method presented in [8] aims to generate fuzzy rules automatically by optimizing fuzzy neural network. The optimization is performed by a hybrid algorithm based on tabu search algorithm and least squares algorithm. Algorithms based on a tolerance rough sets model are used in [9] to obtain fuzzy rules. Resulted rules are then fuzzified using a genetic algorithm.

A method that uses unsupervised learning and reinforcement learning for structure identification and parameter estimations of fuzzy inference systems is suggested in [2]. In this method, unsupervised learning clustering methods to cluster data and generate fuzzy inference systems while adjustment and deletion of fuzzy rules are achieved using reinforcement learning. In [10], a data mining approach based on regularization theory is used to refine and generate fuzzy classification rules.

In this paper, we present a method for automatic generation of fuzzy classification rules from data. The proposed method is based on subtractive clustering optimized using genetic algorithm.

This paper is arranged as follows. A brief description of fuzzy inference system is given in section II. Section III introduces genetic algorithms. The proposed method is presented in section IV. Section V shows experimental results obtained by testing the proposed method. The conclusion is given in section VI.

II. Fuzzy Inference System

Fuzzy inference systems (FIS) are widely used in many applications, from system modelling, simulation and control to classification and decision support. FIS is a computational technique that relies on fuzzy logic [1] for performing input-output mapping. It inherits, from fuzzy logic, the ability to deal with vagueness and incompleteness of knowledge and the ability of describing non-linear relationships between input and output.

FIS consists of three components: a set of fuzzy if-then rules, a database that defines the fuzzy membership functions used in these rules and a reasoning mechanism that performs the inference process. Inputs are transformed to fuzzy sets using the membership functions defined in the FIS's database and the output is computed from the fuzzy rules using the reasoning mechanism. The consequent part of FIS's rules can be a fuzzy value (Mamdani model [11]), a crisp value (type II), or a function of linear combination of input variables (TSK model [12]). In the case of Mamdani model, the process of defuzzification is required to produce the final crisp value of the output.

Information required to design FIS and create its rules can be obtained from the knowledge of human experts. However, with increasing the complexity of the system, the number of inputs and the number of fuzzy variables for each input, it becomes more difficult for human experts to generate these rules. Instead, some computational methods (e.g. data clustering) can be used to generate fuzzy rules from data automatically.

Data Clustering

Data clustering methods aim to group data into clusters of data points that feature a certain level of similarity more than those in other clusters. Fuzzy sets of an input variable can be obtained from data clusters by projecting each cluster to the dimension that corresponds to the input. Thus, each cluster represents a fuzzy rule whose antecedent part is the fuzzy sets resulted from this projection. Many clustering methods require the number of clusters to be specified prior to the clustering process. Different number of clusters results in different clusters properties, hence different fuzzy rules. In addition to the number of clusters, each clustering method has its own parameters that affect the clustering process. Finding the right number of clusters and other clustering parameters for a given application is difficult in many cases. On the other hand, there are some clustering methods that do not require the number of clusters to be specified, instead, they rely on a number of parameters to perform clustering. One of these methods is subtractive clustering [3]. Initially, this method considers that each point is a potential cluster center. Then, for each data point, it computes the potential, defined as the density of the neighboring data points which is a function of the Euclidean distances to all other data points:

$$P_i = \sum_{j=1}^{n} e^{-(4/r_a^2)\left\| x_i - x_j \right\|^2} \tag{1}$$

where P_i is the potential of data point i and r_a is the radius in which other points have significant effect on the potential of data point i.

The point with greatest potential is chosen as a first cluster center, and the potentials of all other points are revised by subtracting an amount that is a function of their distances to the first center:

$$P_i \Leftarrow P_i - P_1^* e^{-(4/r_b^2)\left\| x_i - x_1^* \right\|} \tag{2}$$

where r_b is the radius in which points will have significant reduction in potential, i.e. are less likely to be selected as the next cluster center. This radius can be expressed as multiplication of r_a by a factor S called squash factor:

$$r_a = S\, r_b \tag{3}$$

After potential revision, the data point with the next higher potential is selected as a second cluster center candidate. Whether this point is accepted as a new cluster center or not depends on its potential according to the following conditions:

If $P_k^* > \bar{\varepsilon} P_1^*$, x_k^* is accepted as a new cluster center and the above procedure is repeated.

If $P_k^* < \underline{\varepsilon} P_1^*$, x_k^* is rejected as a new cluster center and the clustering is ended.

If $\bar{\varepsilon} P_1^* > P_k^* > \underline{\varepsilon} P_1^*$, then if $\frac{d_{\min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$, x_k^* is accepted as a new cluster center and the above procedure is repeated. Otherwise x_k^* is rejected, its potential set to 0 and the procedure is repeated with the data point with next higher potential. Here, $d_{\min}$ is the shortest distance between x_k^* and all previously found cluster centers and $\bar{\varepsilon}$ and $\underline{\varepsilon}$ are accept and reject ratios respectively. So, the properties of the final clusters resulted from this method depends on the following parameters: squash factor (S), accept ratio ($\bar{\varepsilon}$), reject ratio ($\underline{\varepsilon}$), and radius (r_a) (or a number of different radii equal to the dimension of the clustered data).

III. Genetic Algorithm

Genetic algorithm (GA) is a stochastic optimization algorithm that imitates natural evolution. Like in natural evolution, genetic algorithms generate populations of different possible solutions successively. Each generation consists of a number of individual solutions, called chromosomes, which compete with each other. The fittest chromosomes with respect to some criteria get higher probabilities of being selected to reproduce the next generation. A chromosome comprises a set of coded properties, called genes, which describe the solution. Genes are either binary coded or real coded [13] [14]. The next generation of chromosomes is produced from selected chromosomes, called parents, by a reproduction method. New parents are selected for each new child to be generated. There are various procedures to select the parents like: roulette

wheel, linear ranking and geometric ranking. The reproduction of a new generation from parents is usually done using two main operators: crossover and mutation. In crossover, chromosomes of one child or more are produced from genes of two or more parent chromosomes using a crossover technique. On the other hand, mutation produces one new chromosome by altering one or more genes in a single chromosome [15].

IV. Proposed Method

In this paper, we present a method that aims to generate fuzzy rules from data automatically to build a FIS used in data classification. The proposed method relies on subtractive clustering to create data clusters from which fuzzy rules can be obtained. The clustering process is optimized using genetic algorithm to produce a FIS that classify data as accurate as possible. Input membership functions of the resulted FIS are then fine-tuned using GA. The final FIS can be used as an initial FIS for neuro-fuzzy system. The block diagram of the proposed method is shown in Fig. 1. It comprises of two main parts: subtractive clustering and genetic algorithm.

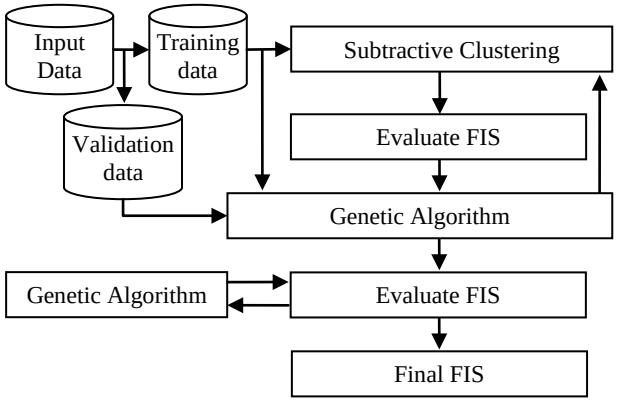

Fig.1. Block diagram of proposed method.

Modeling data are divided into two sets: training data (70%) and validation data (30%).

The parameters of subtractive clustering algorithm (squash factor, accept ratio, reject ratio, and a number of radii equal to the dimension of the clustered data) are adjusted by the GA. Each chromosome contains a set of possible values for these parameters. Table I depicts the structure of a chromosome. For each chromosome in a generation, training data are clustered by subtractive clustering algorithm with clustering parameters contained in the chromosome and a FIS is built. Both training and validation data are classified by the resulted FIS and the classification performance for each data set is evaluated. GA selects parent chromosomes to produce the next generation using a fitness function which incorporates the evaluated performances. Including classification performance of validation data in fitness function is necessary to prevent over-fitting training data by generating too many rules.

Three different metrics are used to evaluate classification performance: accuracy, sensitivity and specificity. Accuracy is defined as the ratio of the number of all correctly classified data points to the number of overall data points:

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where TP (true positive) is the number of correctly classified positives, TN (true negative) is the number of correctly classified negatives, FP (false positive) is the number of negatives misclassified as positives, and FN (false negative) is the number of positives misclassified as negatives.

Accuracy gives an overall indication of classification performance; however, it does not provide an insight into classification performance for each class separately. To measure the classification performance for positive and negative classes separately, sensitivity and specificity are used respectively. Sensitivity is defined as the ratio of the number of correctly classified positives to the number of all actual positives:

$$Sn = \frac{TP}{TP + FN} \tag{5}$$

Similarly, specificity is defined as the ratio of the number of correctly classified negatives to the number of all actual negatives:

$$Sp = \frac{TN}{TN + FP} \tag{6}$$

Using these three metrics for validation and training data, the fitness function can be written as:

$$F = W_1 * Ac_{tr} + W_2 * Sn_{tr} + W_3 * Sp_{tr} + W_4 * Ac_{va} + W_5 * Sn_{va} + W_6 * Sp_{va} \tag{7}$$

where Ac_{tr}, Sn_{tr} and Sp_{tr} are the training accuracy, sensitivity, and specificity respectively, Ac_{av}, Sn_{av} and Sp_{av} are the validation accuracy, sensitivity, and specificity respectively, and W_1 to W_6 are their corresponding weights.

Table I. Chromosome structure for subtractive clustering optimization.

Radius1	Radius 2	……	Radius n
Squash factor	Accept ratio	Reject ratio	

The final step of proposed method is fine-tuning of input membership functions. This is done by a GA whose chromosomes contain a tuning variable for each membership function parameter. The input membership function used is Gaussian function which has two parameters. Gaussian function has a symmetric bell shape and is defined as follows:

$$G(x, \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{8}$$

where c is the center of the of the bell shape and σ (standard deviation) controls the width of the bell. The chromosome structure is shown in Table II, where c_{ij} and σ_{ij} are the center and standard deviation of the *j*th membership function of the *i*th input.

Table II. Chromosome structure for input membership function parameters optimization.

c_{11}	σ_{11}	c_{12}	σ_{12}	…	…	c_{1j}	σ_{1j}
c_{21}	σ_{21}	c_{22}	σ_{22}	…	…	c_{2j}	σ_{2j}
:	:	:	:	:	:	:	:
c_{i1}	σ_{i1}	c_{i2}	σ_{i2}	…	…	c_{ij}	σ_{ij}

V. Experimental Study

To evaluate performance of the proposed method, three different data sets are used for testing. Each data set is divided into two sets: modeling data (70%) and testing data (30%). Modeling data are used to build a FIS using the proposed method and to train the resulted FIS using a neuro-fuzzy system. The neuro-fuzzy system used is "Adaptive Network-based Fuzzy Inference System" (ANFIS). ANFIS, introduced by Jang (1993), is a method for training FIS parameters through a hybrid learning rule which combines the back-propagation gradient descent method and a least-squares method [16]. Accuracy, sensitivity and specificity of the resulted FIS are computed for each data set using testing data partition. These results are compared to results of a number of ANFIS models whose fuzzy rules are generated by fuzzy c-means (FCM) clustering method with various cluster numbers (number of rules). Testing results for each model are the average results of multiple evaluations each with different random data sampling. Number of GA population used is 40 and maximum number of generations is 100.

The three data sets used in testing are: Pima data set, BUPA data set and bladder cancer data set. All data are normalized before being used in training or testing. Pima and BUPA data sets are taken from University of California in Irvine (UCI) Machine Learning Repository [17].

A. *Pima data set*

Pima data set contains 768 records of Pima Indian patients from USA tested for diabetes. All patients are females at least 21 years old. Table III shows details of Pima data set.

Table III. Details of Pima data set.

Number of instances	768
Number of +ve/-ve	268/500
Number of attributes	9

Classification results of BUPA data set are shown in Table IV. The first eight rows are results of classification using 8 FIS's trained by ANFIS and generated by fuzzy c-means clustering with different number of clusters. Bold results indicate best results among these rows. The last row shows the results of classification using a FIS generated by the proposed method (Subtractive Clustering and Genetic Algorithm, SCGA) and trained by ANFIS.

As shown in the Table IV, the accuracy and sensitivity of the proposed method are higher than the highest accuracy and sensitivity of the eight ANFIS models. However, there are two ANFIS models whose specificities are higher than that of the proposed method. In general, the proposed method tries to find the optimal solution which has the best classification performance with respect to all the three metrics. Fig. 2 and Fig. 3 show plots of sensitivity and specificity respectively, for the eight ANFIS models. The dotted line in each figure represents the result of the proposed methods. These figures clearly show the tradeoff between sensitivity and specificity.

Table IV. Classification results for Pima data set.

Rule generation method	Number of rules	Accuracy (%)	Sensitivity (%)	Specificity (%)
FCM	2	76.30	56.60	86.60
FCM	3	**77.10**	52.20	89.30
FCM	5	76.60	57.00	86.50
FCM	10	75.80	**57.20**	85.90
FCM	15	72.40	55.00	81.50
FCM	25	70.87	38.69	86.64
FCM	40	67.39	18.54	**93.18**
FCM	50	65.83	27.17	86.83
SCGA	Auto	77.13	58.30	87.12

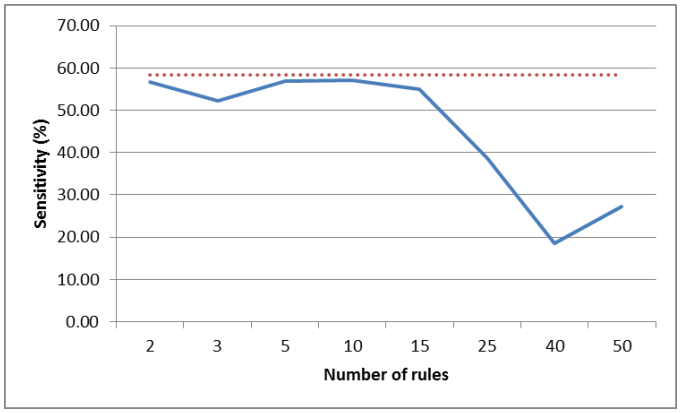

Fig.2. Sensitivity versus number of rules for the eight ANFIS models for Pima data set.

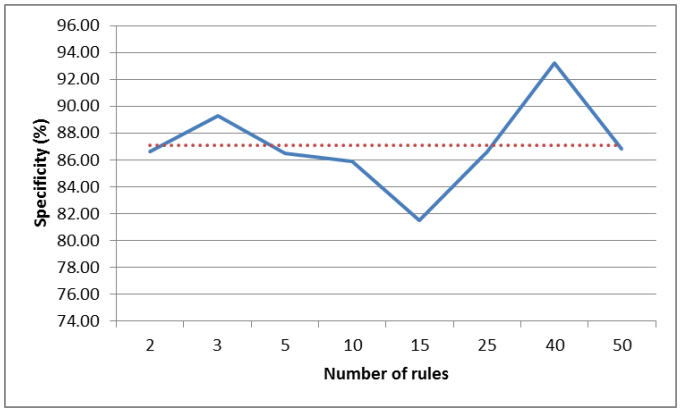

Fig.3. Specificity versus number of rules for the eight ANFIS models for Pima data set.

Fig. 4 and Fig 5 depict plots of testing and training accuracies respectively. The dotted line in each figure represents the result of the proposed methods. It is noticed that

increasing the number of rules increases training accuracy; however, it reduces testing accuracy (over-fitting). The proposed method tries to avoid the problem of over-fitting by finding the optimal solution for both training and validation.

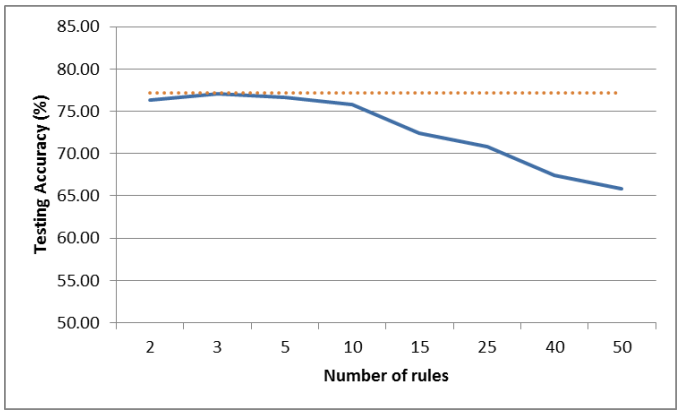

Fig.4. Testing accuracy versus number of rules for the eight ANFIS models for Pima data set.

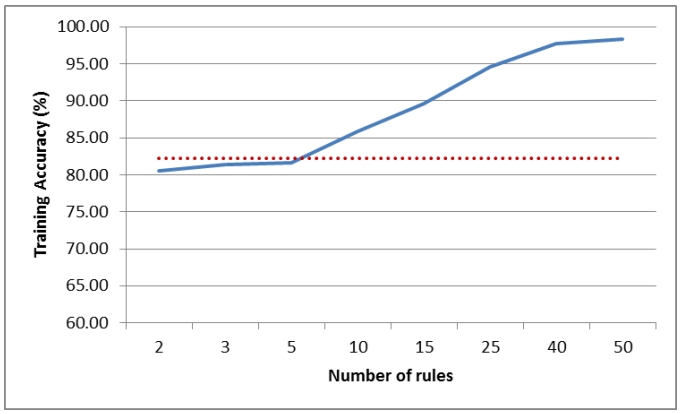

Fig.5. Training accuracy versus number of rules for the eight ANFIS models for Pima data set.

B. BUPA data set

BUPA data set contains 345 records of liver patients from USA. Details of BUPA data set are shown in Table V.

As with Pima data set, BUPA data set is used to compare classification performance of the proposed method to the performance of eight ANFIS models. The results are shown in Table VI. The proposed method has higher accuracy than all the eight models and sensitivity that is very close to the highest one. The plots of testing and training accuracies are shown in Fig. 6 and Fig. 7 respectively.

Table V. Details of BUPA data set.

Number of instances	345
Number of +ve/-ve	200/145
Number of attributes	7

Table VI. Classification results for BUPA data set.

Rule generation method	Number of rules	Accuracy (%)	Sensitivity (%)	Specificity (%)
FCM	2	70.20	79.10	58.30
FCM	3	**71.35**	79.05	59.56
FCM	5	70.40	87.30	45.80
FCM	10	64.80	69.80	58.10
FCM	15	64.20	**88.60**	24.70
FCM	25	64.81	73.66	53.56
FCM	40	63.85	64.01	**61.06**
FCM	50	65.77	80.22	44.26
SCGA	Auto	71.92	78.29	60.96

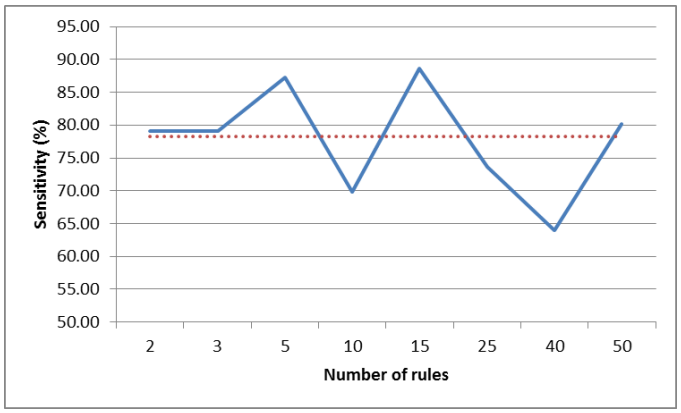

Fig.6. Sensitivity versus number of rules for the eight ANFIS models for BUPA data set.

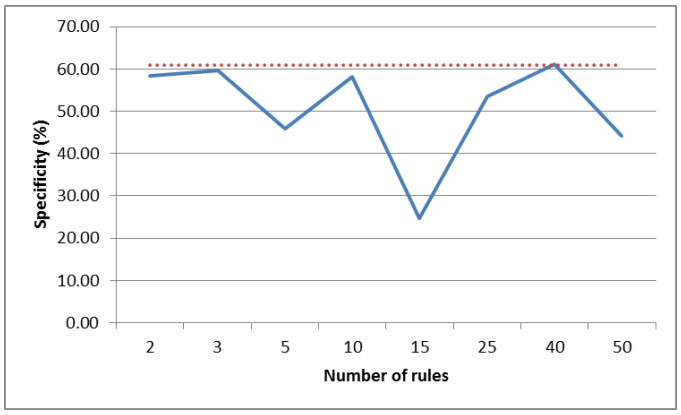

Fig.7. Specificity versus number of rules for the eight ANFIS models for BUPA data set.

C. Bladder cancer data set

Bladder cancer data set comprises of progression information of 234 patients who had undergone surgical tumor removal for bladder cancer. Table VII shows details of this data set. Results of classification testing and comparison are shown in Table VIII. The plots of testing and training accuracies are shown in Fig. 6 and Fig. 7 respectively.

Table VII. Details of bladder cancer data set.

Number of instances	234
Number of +ve/-ve	163/71
Number of attributes	8

Table VIII. Classification results for bladder cancer data set.

Rule generation method	Number of rules	Accuracy (%)	Sensitivity (%)	Specificity (%)
FCM	2	71.70	20.00	94.40
FCM	3	**75.90**	16.20	98.90
FCM	5	70.30	**26.20**	90.10
FCM	10	74.50	26.00	94.80
FCM	15	64.10	24.50	89.70
FCM	25	62.80	8.25	97.60
FCM	40	66.20	8.00	**100.00**
FCM	50	73.10	10.60	97.60
SCGA	Auto	77.93	14.44	**99.00**

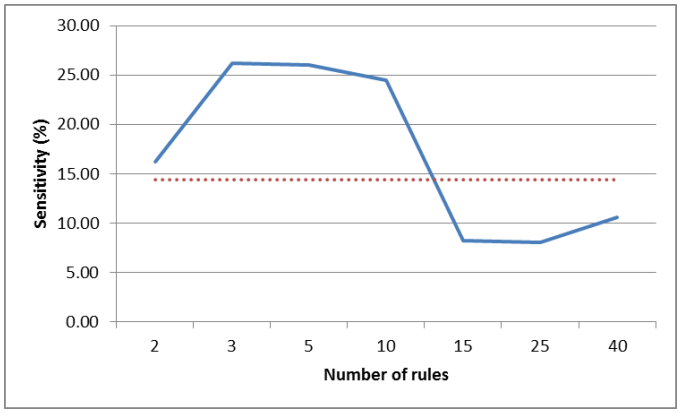

Fig.8. Sensitivity versus number of rules for the eight ANFIS models for bladder cancer data set.

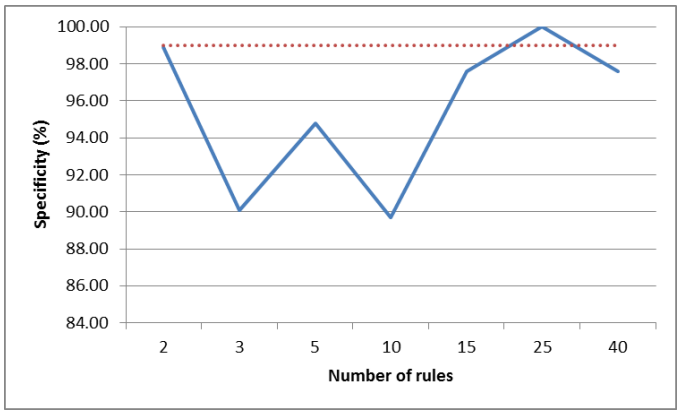

Fig.9. Specificity versus number of rules for the eight ANFIS models for bladder cancer data set.

VI. Conclusion

In this paper, we proposed a method based on subtractive clustering and genetic algorithm to automatically generate fuzzy rules for data classification. The proposed method searches for the FIS structure and the number of rules that best compromise between classification performance of training and validation. Experimental results show that the proposed method has better accuracy than one of the widely used clustering method (fuzzy c-means) and a well compromised sensitivity and specificity. Results also show that FIS generated by the proposed method avoids data over-fitting.

References

[1] L. Zadeh, "Fuzzy sets," Inf. Control, vol. 353, pp. 338–353, 1965.
[2] M. J. Er and Y. Zhou, "Automatic generation of fuzzy inference systems via unsupervised learning," Neural Netw., vol. 21, no. 10, pp. 1556–66, Dec. 2008.
[3] S. L. Chiu, "Extracting fuzzy rules from data for function approximation and pattern classification," Fuzzy Information Engineering: A Guided Tour of Applications, pp. 1–10, 1997.
[4] A. F. Gómez-Skarmeta, M. Delgado, and M. A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification," Fuzzy Sets Syst., vol. 106, no. 2, pp. 179–188, Sep. 1999.
[5] P. P. Angelov and R. A. Buswell, "Automatic generation of fuzzy rule-based models from data by genetic algorithms," Inf. Sci. (Ny)., vol. 150, pp. 17–31, 2003.
[6] R. P. Prado, S. García-Galán, J. E. Exposito, and A. J. Yuste "Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization," Fuzzy Syst. IEEE Trans., vol. 18, no. 6, pp. 1083–1097, 2010.
[7] H. Zhang, B. Zhang, and F. Wang, "Automatic Fuzzy Rules Generation Using Fuzzy Genetic Algorithm," 2009 Sixth Int. Conf. Fuzzy Syst. Knowl. Discov., pp. 107–112, 2009.
[8] L. Xiao and G. Liu, "Automatic fuzzy rule extraction based on fuzzy neural network," Adv. Neural Networks–ISNN 2005, no. 104262, Springer Berlin Heidelberg, pp. 688–693, 2005.
[9] M. Sikora, "Fuzzy Rules Generation Method for Classification Problems Using Rough Sets and Induction of Decision Rules Based Upon Rough Set," in Lecture Notes in Computer Science Volume, 2005, pp. 383–391.
[10] D. Wang, T. S. Dillon and E. J. Chang, "A data mining approach for fuzzy classification rule generation," Proc. Jt. 9th IFSA World Congr. 20th NAFIPS Int. Conf. (Cat. No. 01TH8569), vol. 5, no. C, pp. 2960–2964, 2001.
[11] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," Int. J. Man. Mach. Stud., 1975.
[12] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst. Man. Cybern., vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.
[13] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating point representations in genetic algorithms.," ICGA, 1991.
[14] D. E. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking," Urbana, pp. 1–21, 1990.
[15] C. R. Houck, J. A. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a Matlab implementation," NCSU-IE TR, no. 919, pp. 1–14, 1995.
[16] J. Jang, "ANFIS: adaptive-network-based fuzzy inference system," Syst. Man Cybern. IEEE Trans. …, vol. 23, no. 3, 1993.
[17] A. Frank and A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.