International Journal of Engineering Research
Volume No.6, Issue No.2, pp :55-60

ISSN:2319-6890 (online),2347-5013(print)
1 Feb. 2017

Inductive Learning of Fuzzy Rule-Based Classifier with Self-Constructing Clustering

Chie-Hong Lee[a], Cheng-Ru Wang[b], Yann-Yean Su[a], Shie-Jue Lee[b]
[b]Department of Electrical Engineering, NationalSun Yat-Sen University,Kaohsiung 804,Taiwan
[a]Department of Digital Content Application and Management,Wenzao Ursuline University of Languages, Kaohsiung 807,Taiwan
chichhong@gmail.com, yysu@mail.wzu.edu.tw,crwang@water.ee.nsysu.edu.tw,leesj@mail.ee.nsysu.edu.tw

Abstract: ***The inductive learning of fuzzy rule-based classification systems usually encounters an issue of exponential growth of the fuzzy rulesearch space when the number of patternsand/or variables becomes large.This issue makes the learning process more difficultand, in most cases, may lead to scalability problems. Alcalá-Fdezet al.proposed a fuzzy association rule-based classification method for high-dimensional problems, whichis based on three stages to obtain an accurate and compact fuzzyrule-based classifier with a low computational cost.But there is a serious drawback with this method: the initial linguistic termsmust be predefined by the user. Weapply a self-constructing clustering technique for determining the linguistic terms automatically according to thecharacteristics of the training data. Therefore, the resulting classification system can be more friendly and time-saving for use to the user. Furthermore, more accurate classification results can usually be obtained for the user.***

Keywords:self-constructing clustering, associative classification,fuzzy association rules, membership functions, genetic algorithms.

1. Background/ Objectives and Goals

FUZZY rule-based classification (FRBC) is one of the techniques used in the area of pattern recognition and classification. One of the big advantages it possesses is that it can provide anhumanly interpretable model by using linguistic terms in the antecedents of the rules [1]. FRBC has been applied in various applications including anomaly intrusion detection [2], image processing [3], domotics [4], and medical diagnosis [5],among others.For low-dimensional cases, FRBC works fine. However, in those cases where the available or useful data consist of a high number of patterns (instances or examples) and/or variables, the inductive learning of FRBC suffers from an exponential growth of search space. Either the number of rules or the number of variables involved in the rules becomes huge. This growth makes the learning process more difficult and may lead to problem of scalability in terms of the time and memory consumed [6].Alcalá-Fdez*et al.* [7] proposed a fuzzy association rule-based classification method for high-dimensional problems, which is based on three stages to obtain an accurate and compact fuzzy rule-based classifier with a low computational cost. But there is a serious drawback with this method: the initial linguistic terms must be predefined by the user. We apply a self-constructing clustering technique [8] for determining the linguistic terms automatically according to the characteristics of the training data. Therefore, the resulting classification system can be more friendly and time-saving for use to the user.

2. Methods

We present an improved fuzzy association rule-based classification method to develop an accurate and compact fuzzy rule-based classifier with a low computational costfor high-dimensional problems. The method is based on the following four stages:

1. Use SCC to define linguistic terms:

Self-constructing clustering (SCC) is a progressive clustering algorithm [8]. For each data input, the most similar cluster is computed. If the similarity between the input and an existing cluster is bigger or equal to a predefined threshold, the input is included in the cluster and the mean and standard deviation of the cluster are updated accordingly. Otherwise, a new cluster is created. After all the data points are considered, we get a set of clusters from which a set of linguistic terms can be obtained.

Suppose $x_p = [x_{p,1}, x_{p,2}, \cdots, x_{p,n}] \in \Re^n$ is a training pattern,$1 \leq p \leq l$, where n is the featuredimensionalityof x_p and l is the number of training data. $C_j$denotes the set of existing clusters, where $C_j = \{m_j, \sigma_j\}$ woith $m_j = [m_{j,1}, m_{j,2}, \cdots, m_{j,n}]$ being the mean and $\sigma_j = [\sigma_{j,1}, \sigma_{j,2}, \cdots, \sigma_{j,n}]$ being the deviation. Let J be the number of all existing clusters and$S_j$represent thesize of clusterC_j, i.e., the number of data patterns included inC_j, for $1 \leq j \leq J$. In principle, our algorithm works like a five-layer network. The five layers are referred to as the input layer, layer 1, layer 2, layer 3, and the output layer, respectively. The operation of each layer is described in detail below.

- Input layer. This layer transfers the n components of x_p forward to the nodes in layer 1. Node 1 of this layer passes $x_{p,1}$ to the first node of all the groups in layer 1, node 2 of this layer passes $x_{p,2}$ to the second node of all the groups in layer 1, etc.
- Layer 1.This layeris a MIMO (multi-input multi-output) layer. Its main task is to calculatethe one-dimensional Gaussian membership degreeof each cluster for the inputs. Each node of this layer corresponds one-dimensional of Gaussian function. The Gaussian membership degreein each dimension is calculated as follows:

$$O_{j,i}^{(1)} = g(x_{p,i}; m_{j,i}; \sigma_{j,i}) = \exp\left[-(\frac{x_{p,i} - m_{j,i}}{\sigma_{j,i}})^2\right] \quad (1)$$

for$1 \leq i \leq n$ and $1 \leq j \leq J$, where $x_{p,i}$ is the i-th dimension of the p-th training pattern and$m_{j,i}$ and $\sigma_{j,i}$ are the mean and deviation, respectively,of the i-th dimension of cluster C_j.

- Layer 2.This layeris a MISO(multi-input single-output) layer. The relationship between the inputs and output of each node is represented by

$$O_j^{(2)} = G_j(x_p) = \prod_{i=1}^{n} O_{j,i}^{(1)} \quad (2)$$

for $1 \leq j \leq J$. The outputs of this layer represent the similarities between x_p and the existing clusters.

- Layer 3.This layer contains only one node,icon-labeled by**C**in the figure. It performs competitive learning. The inputs to this node are the similarity degrees provided from all the existing clusters. The output of this node, $O^{(3)}$, takes the largest of its inputs, i.e.,

$$O^{(3)} = \max_{1 \leq j \leq J} O_j^{(2)} \quad (3)$$

Let node **a** be the winner node, i.e.,

$$a = \arg \max_{1 \leq j \leq J} O_j^{(2)} \quad (4)$$

- Output layer .The layer contains a single node. It applies its input through a hard limit function as follows:

$$O^{(4)} = \text{hardlim}(O^{(3)}) = \begin{cases} 1, if\ O^{(3)} \geq \rho \\ 0, otherwise \end{cases} \quad (5)$$

whereρ is a predefined threshold, $0 \leq \rho \leq 1$.

Two cases may occur for x_p:

[Case 1]If $O^{(4)} = 0$, we regard x_p to be most similar to node a. Then we assign $x_p$to C_a, and m_a and σ_a of cluster C_a are modified to include x_p as its member:

$$m_{a,i} = \frac{S_a m_{a,i} + x_{p,i}}{S_a + 1} \quad (6)$$

$$\sigma_{a,i} = \sqrt{\frac{(S_a - 1)(\sigma_{a,i} - \sigma_{0,i})^2 + S_a m_{a,i}{}^2 + x_{p,i}{}^2}{S_a} - \frac{(S_a m_{a,i} + x_{p,i})^2}{S_a(S_a + 1)}} + \sigma_0 \quad (7)$$

for $1 \leq i \leq n$, where σ_0 is the initial standard deviationdetermined by the user.

Finally, we update the size of clusterC_a:

$$S_a = S_a + 1 \qquad (8)$$

[Case 2]If $O^{(4)} = 1$, there are no existing clusters to which x_p is similar enough. In this case, we add a new cluster by

$$C_{J+1} = \{m_{J+1}, \sigma_{J+1}\} \qquad (9)$$

$$m_{j+1} = x_p \qquad (10)$$

$$\sigma_{j+1} = \sigma_0 \qquad (11)$$

$$\mathrm{J} = \mathrm{J} + 1 \qquad (12)$$

The above process is iterated for all the patterns until the end of $p = l$. At the end, we have J clusters and $C_j = \{m_j, \sigma_j\}$ for $1 \leq \mathrm{j} \leq \mathrm{J}$.

2. **Fuzzy association rule extraction for classification (For each class $\boldsymbol{C_i}$):**

Fuzzy association rule extraction basically follows the steps proposed in [7], as stated below:

Step 1: Calculate the minimum support of class C_i:

$\text{Mini\_Support}_{C_i} = \alpha * f_{C_i}$(13)

where α is the minimum support determined by an expert and f_{C_i} is the pattern ratio of the class C_i.

Step 2: Create levels 0 and 1 of the search tree.

Thelevel 0 (root) of a search tree is an empty set. All features are assumed to have an order, and the one-item sets that correspond to the attributes are listed in the first level, i.e., level 1, of the search tree according to their order.

Step 3: Createone new level in the search tree.

The following conditions have to be satisfied in order to expand frequent item-sets:

1. The support of an n-item set in a node J is higher than the minimum support.
2. A candidate item set generates a classification rule with confidence less than the maximum confidence ($conf_{max}$).

$$\text{Support}(A \rightarrow C_i) = \frac{\sum_{x_p \in Class\ C_i} \mu_A(x_p)}{|N|} \qquad (14)$$

$$\text{Confidence}(A \rightarrow C_i) = \frac{\sum_{x_p \in Class\ C_i} \mu_A(x_p)}{\sum_{x_p \in T} \mu_A(x_p)} \qquad (15)$$

where$|N|$is the number of transactions in T and $\mu_A(x_p)$ is the matching degree of the pattern x_p with the item set. The matching degree $\mu_A(x_p)$ of x_p to the different fuzzy regions is computed by the use of a conjunction operator. We adopt the product T-norm as the conjunction operator.

Step 4: If there are more than two nodes in a new level, and the depth of the tree is less than Depth_max (Depth_max is determined by an expert), go to Step 3.

Step 5: The rules with class C_i are then obtained.

3. **Candidate rule prescreening (For each class $\boldsymbol{C_i}$):**

Candidate rule prescreening basically follows the steps proposed in [7], as stated below:

Step 1: Initialize the weight of the pattern to 1.

Step 2: Calculate the wWRAcc'' for each rule by

$$\text{wWRAcc}''(A \rightarrow C_i) = \frac{n''(A \cdot C_i)}{n'(C_i)} * \left(\frac{n''(A \cdot C_i)}{n''(A)} - \frac{n(C_i)}{N}\right) \qquad (16)$$

where $n''(A)$ is the sum of the products of the weights of all covered patterns by their matching degrees with the antecedent part of the rule, $n''(A \cdot C_i)$ is the sum of the products of the weights of all correctly covered patterns by their matching degrees with the antecedent part of the rules, and $n'(C_i)$ is the sum of the weights of all the patterns of class C_i.

Step 3: Select the best rule as a part of the initial RB and remove it from the candidate rule set. Each time a rule is selected, a count r for each pattern of how many times the pattern has been covered is stored.

Step 4: Decrease the weight of a pattern covered by the selected rules by

$$\mathrm{w}(e_i, \mathrm{r}) = \frac{1}{r+1} \qquad (17)$$

where$r$is the number of times that the pattern has been covered.

Step 5: If any pattern has been covered less than r_t times and there are more rules in the candidate rule set, go to**Step 2.**

4. **Apply genetic algorithms (GAs) for rule selection and lateral tuning**

Rule selection and lateral tuning basically follow the steps proposed in [7], as stated below:

Step 1: Generate the initial population with Pchromosomes.

Chromosomes are composed by the rule selection C_S and lateral tuning C_T.

A. For the C_S part, each chromosome is a binary vector that determines when a rule is selected or not. If a rule is selected, the corresponding element is 1. Otherwise it is 0. Considering the set $C_S$with M rules that are contained in the candidate rule set, $C_S = \{c_1, \cdots, c_M\}$, which represents a subset of rules composing the final RB so that IF $c_i = 1\, THEN\ (R_i \in RB) else\ (R_i \notin RB)$,with R_i being the corresponding ith rule in the candidate rule set and RB being the final RB.

B. For the C_T part, a real coding is considered. This part is the joint of the symbolic translation parameters of each fuzzy partition. Assume that there are m^i terms associated with the ithvariable in the system. For n variables, this part has a form $C_T = (c_{11}, \cdots, c_{1m^1}, c_{21}, \cdots, c_{2m^2}, c_{n1}, \cdots, c_{nm^n})$where each gene is associated with the tuning value of the corresponding term.

The initial population is obtained with the first individual having all genes with value "1" in the C_S part and all genes with value "0.0" in the C_T part. The remaining individuals are generated at random.

Step 2: Evaluate the population.

$$\text{Fitness(C)} = \frac{\#Hits}{N} - \delta * \frac{NR_{ini\ tial}}{NR_{initial} - NR + 1.0} \qquad (18)$$

where $\#Hits$ is the number of patterns that are correctly classified, $NR_{initial}$ is the number of candidate rules, NR is the number of selected rules, and δ is a weighting percentage given by the system expert that determines the tradeoff between accuracy and complexity. If there is at least one class without covering by selected rules or if there are no covered patterns, the fitness value of a chromosome will be penalized with the number of classes without covering by selected rules and the number of uncovered patterns.

Step 3: Initialize the threshold value by taking into account Gray codings, i.e., $L = L_{initial}$.

The threshold value is initialized as the maximum possible distance between two individuals divided by 4.

Step 4: Generate the next population as follows:

I. Shuffle the population.

II. Select twopairs of parents at a time.

III. Each pair is crossed ifthe hamming distance between the parent Gray codingsdivided by 2 is more than L.

The crossover operator depends on the chromosome part where it is applied.

A. For the C_T part, the Parent Centric BLX (PCBLX) operator [9] (an operator that is based on BLX- α) is used. This operator is based on the concept of neighborhood, which allows the offspring genes to be around the genes of one parent or around a wide zone that is determined by both parent genes. Assume that $X = (x_1, \cdots, x_n)$, and $Y = (y_1, \cdots, y_n)$, where $x_i, y_i \in [a_i, b_i] \subset \mathfrak{R}, i = 1, \cdots, n$, are two real-coded chromosomes that are going to be crossed. The following two offspring chromosomes are generated.

i. $O_1 = (o_{11} \cdots o_{1n})$, where O_{1i} is a randomly (uniformly) chosen number from the interval $[l_i^1, u_i^1]$, with $l_i^1 = \max\{a_i, x_i - I_i \cdot \alpha\}$, $u_i^1 = \max\{b_i, x_i + I_i \cdot \alpha\}$, and $I_i = |x_i - y_i|$.

ii. $O_2 = (o_{21} \cdots o_{2n})$,where O_{2i} is a randomly (uniformly) chosen number from the interval $[l_i^2, u_i^2]$, with $l_i^2 = \max\{a_i, y_i - I_i \cdot \alpha\}$, and $u_i^2 = \max\{b_i, y_i + I_i \cdot \alpha\}$.

B. In the C_S part, the half-uniform crossover scheme (HUX) is employed [10]. The HUX crossover exactly interchanges the mid of the alleles that are different in the parents (the genes to be crossed are randomly selected from among those that are different in the parents). By this, maximum distance of the offspring to their parents is ensured (exploration).

In this case, four offspring chromosomes are generated by the combination of the two from the part C_T with the two from the part C_S. The two best offspring chromosomes obtained in this way are considered as the two corresponding descendents.

IV. Evaluate the new individuals.

V. Join the parents with their offspring chromosomes, and select the best Pindividuals to take part in the next

ISSN:2319-6890 (online),2347-5013(print)
1 Feb. 2017

population.

Step 5: If the population does not change or there areno new individuals in the population, then $\mathrm{L} = \mathrm{L} - (L_{initial} * 0.1)$.

Step 6: If$\mathrm{L} < 0$, restart the population and initialize L.

Step 7: If the maximum number of evaluations is not reached,go to **Step 4**.

3. Results

To analyze the performance of our proposed approach, we have conducted experiments on 12 real-world datasets and made a comparison with the FARC-HD [7].For convenience, our proposed approach is called improved FARC-HD. The characteristics of these 12 datasets, taken from the Evolutionary Learning (KEEL)-dataset repository[11], are listed in Table 1. In this table, the column "**Attributes(R/I/N)**" is the number of (Real/Integer/Nominal) attributes, "**Patterns"** is the number of patterns, and "**Classes"** is the number of classes in each dataset. For example, for the Iris dataset, there are 4 attributes which are real and none are integer or nominal attributes. Therefore, 4(4/0/0) is shown in the column "**Attributes(R/I/N)**". Furthermore, Iris has 150 patterns in total, each belonging to one of 3 classes. These numbers are shown in the columns of "**Patterns"**and "**Classes"**, respectively, in Table 1.

We use a tenfold cross-validation for experiments. Theexperimental results are shown in Table 2. In this table, we can see that our method can get higher classification accuracy than FARC-HD.For instance, for the Phoneme dataset, our method has 84.06% in classification accuracy, while FARC-HD has 82.14% instead. The percentage of improvement is about (84.06-82.14)/82.14 =2.34% as shown in the last column of Table 2.

4. Conclusion

We have proposed an improved fuzzy association rule-based classification method for high-dimensional problems. Our approach generates the linguistic terms automatically by applying a self-constructing clustering technique, rather than requiring experts to define them instead. The resulting classification system can be more friendly and time-saving for use to the user. Furthermore, more accurate classification results can usually be obtained for the user. Experimental results have shown that our improved method can average provide 2% better in classification accuracy over the original method. However, when the number of attributes or patterns increases, the time GAtakes also increases significantly.Our future work is to further reduce the dimensionality involved in the fuzzy association rules and to apply parallel computing to speed up the constructing process of the classification system.
In Candidate rule prescreening,weights of positive patterns covered by the selected rule decrease according to the formula $\mathrm{w}(e_i, \mathrm{r}) = \frac{1}{r+1}$, but we can try to determine in accordance with the degree of coverage to reduce the weightin the future. This way, except forthe uncovered target class patterns whose weights have not been decreased, the low level of covered patternwill have a greater chance of being covered in the following iterations.

5. References

i. *H. Ishibuchi, T. Nakashima, and M. Nii, "Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining," 2004, Springer-Verlag.*

ii. *C. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," Pattern Recognition, vol. 40, no. 9, pp. 2373-2391, 2007.*

iii. *T. Nakashima, G. Schaefer, and Y. Yokota, "A weighted fuzzy classifier and its application to image processing tasks," Fuzzy Sets and Systems, vol. 158, no. 3, pp. 284-294, 2007.*

iv. *F. Ch´avez, F. Fern´andez, R. Alcal´a, J. Alcal´a-Fdez, G. Olague, and F. Herrera, "Hybrid laser pointer detection algorithm based on template matching and fuzzy rule-based systems for domotic control in real home environments," Applied Intelligence, vol. 36, no. 2, pp. 407–423, 2012.*

v. *J. Sanz, M. Galar, A. Jurio, A. Brugos, M. Pagola, and H. Bustince, "Medical diagnosis of cardiovascular diseases using an* *interval-valued fuzzy rule-based classification system," Applied Soft Computing, vol. 20, pp. 103–111, 2014.*

vi. *Y. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement," IEEE Transactions on Fuzzy Systems, vol. 8, no. 2, pp. 212-221, 2000.*

vii. *J. Alcalá-Fdez, R. Alcalá,and F. Herrera,"A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning,"IEEE Transactions on Fuzzy Systems, vol. 19, no. 5, pp. 857-872, 2011.*

viii. *S.-J. Lee and C.-S. Ouyang, "A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning," IEEE Transactions on Fuzzy Systems, vol. 11, no. 3, pp. 341-353, 2003.*

ix. *M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Real-coded memeticalgorithms with crossover hill-climbing," Evolutionary Computation, vol. 12, no. 3, pp. 273-302, 2004.*

x. *L. Eshelman and J. Schaffer, "Real-coded genetic algorithms and intervalschemata," in Foundations of Genetic Algorithms, vol. 2, D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 187–202.*

xi. *J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," Journal of Multiple-Valued Logic &Soft Computing, vol. 17, nos. 2–3, pp. 255–287, 2011.*

Table 1. Characteristics of the 12 datasets

Name	Attributes(R/I/N)	Patterns	Classes
Iris	4(4/0/0)	150	3
Phoneme	5(5/0/0)	5404	2
Monks	6(0/6/0)	432	2
Ecoli	7(7/0/0)	336	8
Yeast	8(8/0/0)	1484	10
Glass	9(9/0/0)	214	7
Page-blocks	10(4/6/0)	5472	5
Wine	13(13/0/0)	178	3
Heart	13(1/12/0)	270	2
Cleveland	13(13/0/0)	297	5
Vowel	13(10/3/0)	990	11
Crx	15(3/3/9)	653	2

[Available at http://sci2s.ugr.es/keel/datasets.php]

Table 2. Comparison on classification accuracy

Dataset	FARC-HD (%)	**Improved** FARC-HD (%)	Improvement (%)
Iris	96.00	96.22	+0.23
Phoneme	82.14	84.06	+2.34
Monks	99.77	99.81	+0.04
Ecoli	82.19	84.73	+3.09
Yeast	58.50	62.10	+6.15
Glass	70.24	72.15	+2.72
Page-blocks	95.01	96.47	+1.54
Wine	94.35	96.53	+2.31
Heart	84.44	87.57	+3.71
Cleveland	55.24	58.40	+5.72
Vowel	71.82	75.65	+5.33
Crx	86.03	88.15	+2.46
mean	**81.31**	**83.49**	**+2.68**