FCM-Fuzzy Rule Base: A new Rule Extraction Mechanism

Hossein Khosravi R.
Dept. of Engineering
Ferdowsi University of Mashhad, Mashhad, Iran

M.H.Yaghmaee Moghaddam
Dept. of Engineering
Ferdowsi University of Mashhad, Mashhad, Iran

Amirhossein Baradaran Shahroudi
Dept. of IT, Islamic Azad University , Mashhad Branch, Mashhad, Iran

Hadi Sadoghi Yazdi
Dept. of Engineering
Ferdowsi University of Mashhad, Mashhad, Iran

Abstract— **Regardless of creation method, Fuzzy rules are of great importance in the implementation and optimization systems. Although using human knowledge in creating Fuzzy rules, has the advantage of readability and is near the experimental expertise, but it cannot be implemented in all systems. Since Output of a system is based on its correct function over the time, output data is reliable with higher percentage. In this paper, Fuzzy rules are extracted from a decision tree, constructed from the output of the system. In fact, traversing the decision tree leads to producing fuzzy rules. Decision tree which presented, is innovative, in comparison with previous implementations, and could also be regarded as new solution in classification. First advantage of the new decision tree to C4.5 (which is the most widely used as a common decision-making structure), is its capability of deciding on more than one feature simultaneously which is not provided in C4.5. Not producing a definite answer and result improvement in iterative processes are also other benefits of the new presented method.**

Keywords-component; Fuzzy systems ; decision tree ; clustering ; FCM ; C4.5;

I. INTRODUCTION

Using decision tree as a classifier is variously applied now fully functional different types of implementations including ID3, C4.5, C5 are used in numerous applications. Decision tree is created based on analysis of input data aimed to find a feature as the basis for taking decision in each node. Different features of data will be analyzed in each node and the one which yields the minimum entropy would be selected. [1] And [2] are using the similar method. Creating rules by decision tree traversing had been presented in previous studies [3] and the resulted rules had also been optimized in [4].

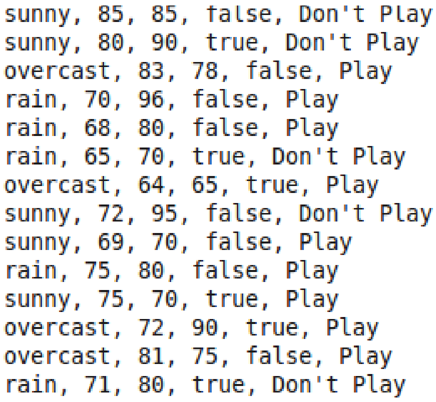
In [4] rules are extracted from a C4.5 decision tree and the efficiency of the method has been emphasized by the authors. In this paper, a new decision tree is presented based on a technique known as Clustering -rather than the concept of entropy - i.e., FCM. The proposed method chooses the best feature for each node based on how it minimizes the FCM function. With the selected feature, clustered data would be isolated and then in each stage (for each node), remaining data develops the decision tree in a recursive process. FCM Method, [5], [6] as it will be presented in section 1, is a Clustering method aimed to minimize the total distance from cluster centers to data points – which dynamically changes by an algorithm. Since cluster centers are randomly selected in the beginning, same output cannot always be expected. This paper is organized as follows: In the second section of the paper, the background of decision trees, FCM clustering and fuzzy rules will be reviewed. In 3rd Section, the innovated method will be presented. Finally in Section IV, simulation results are presented.

II. BACKGROUND

A. Decision Trees

Decision tree, is a unique style of presentation of a system that facilitates future decisions and defines the system properly. Since the majority of engineering, computational and executive systems can be defined in the form of a series of data (features and the corresponding outputs), an algorithm (tree construction) can be applied in order to analyze the output and presents it based on the given data in the form of a decision tree. For example, the following data (which includes a series of features and corresponding outputs) resulted in the tree below “Fig. 1”, using the algorithm presented in [7]. In table 1 data are classified in two classes: (Play and Don `t Play), the first four fields are feature fields and last one is classification data.

TABLE I. DATA INPUT



```
sunny, 85, 85, false, Don't Play
sunny, 80, 90, true, Don't Play
overcast, 83, 78, false, Play
rain, 70, 96, false, Play
rain, 68, 80, false, Play
rain, 65, 70, true, Don't Play
overcast, 64, 65, true, Play
sunny, 72, 95, false, Don't Play
sunny, 69, 70, false, Play
rain, 75, 80, false, Play
sunny, 75, 70, true, Play
overcast, 72, 90, true, Play
overcast, 81, 75, false, Play
rain, 71, 80, true, Don't Play
```

978-1-4577-0314-0/11/$26.00 ©2011 IEEE

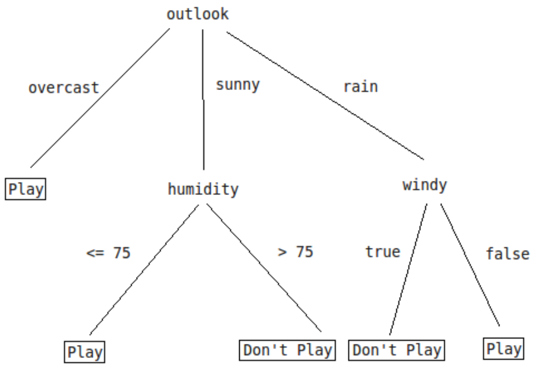
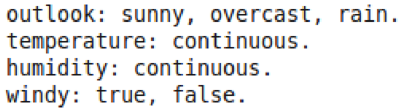


Figure 1. The corresponding tree to the data in Table 1.

Methods applied in creating the tree use the variables introduced in Table 2. It can be seen that the method, correctly recognized the temperature variable as an ineffective feature and removed it.

TABLE II. CORRESPONDING VARIABLE TYPES TO "FIG 1"



```
outlook: sunny, overcast, rain.
temperature: continuous.
humidity: continuous.
windy: true, false.
```

After creating a tree, structure can be used for the classification of the test data. By this rapid formation, test data will be separated in different classes. Generally, the training data used to create the tree are different from the test data that was created to evaluate the tree. Number of errors in detecting test data is suitable criterion for algorithm evaluation.

ID3, C4.5 and C5

ID3 is an algorithm for creating decision trees [1]. In this algorithm the concept of entropy is used for data classification. And the algorithm tries to keep the amount of entropy in the upper nodes of the tree to the minimum which leads to a tree with the least possible height. First, for all the possible features of primary data, entropy is calculated with (1), and then the most useful features will be selected as the root.

$$I = -\sum_c p(c) \log_2 p(c) \qquad (1)$$

To determine the effectiveness of each feature, (2) is used.

$$\mathrm{Gain(A) = I -\ I_{res}(A)} \qquad (2)$$

In which I_{res} is the amount of remaining entropy in the categories due to the use of each feature. That can be achieved by summing the total probability of each division. (3)

$$I_{res} = -\sum_v \mathrm{p(v)} \sum_c \mathrm{p(c|v)} \log_2 \mathrm{p(c|v)} \qquad (3)$$

This algorithm (ID3) can only be used for classification of discrete and limited range of data features and it is not applicable for noisy or distorted data.

C4.5 algorithm is an improvement to ID3.it can classify both continues and noisy data. For this purpose, it first sorts the data then calculates the benefit value for all the split modes for the sorted data and chooses the splitter corresponding to the biggest benefit as the separator. Latest version of the C4.5 decision tree marketed commercially by [8] and is called C5. C5 version is capable of analyzing several records and can utilize processing capabilities of multi-core CPU`s to improve the algorithm performance. Presented results have been achieved using the trial version provided by the website [8].

B. FCM(Fuzzy c-means) clustering Method

In pattern recognition concepts, clusters are sets of data that grouped due to their similarities. Clustering attempts data to be divided into clusters which has the maximum similarity possible between data in each cluster and minimum similarity between data in different clusters without any supervision. [2]. FCM is one of the most common methods of clustering. Similar to k-means algorithm, the FCM method is one of the clustering algorithms which aims to minimize an objective function [15] [16] [17] [18]. In this algorithm samples divide into C number of clusters which has a predefined value. Objective function for Fuzzy C-Means (FCM) Clustering algorithm is calculated by (4):

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m}\, d_{ik}^{2} = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m}\ ||x_k - v_i||^2 \qquad (4)$$

In (4), m is a real number greater than 1, which is 2 in most cases. If m is set equal to 1 in (4) objective function of classical c- means clustering will be obtained.

In the above formula x_k is the k_{th} sample and v_i is the cluster center and n is the number of feature samples. u_{ik} is the membership function of i_{th} sample in k_{th} cluster. ||*|| shows the degree of membership (distance between sample and the center of cluster) which could be measured by any function that defines the distance measure. U_{ik} can provide a matrix named U with n rows and c columns and it`s entries can be set on any value between 0 to 1. If all entries of the U matrix were set 0 or 1, the algorithm would be similar to the classic C-mean algorithm.

Although the entries of U matrix can be any value between 0 and 1, but the sum of each column should be equal to 1 and we have:

$$\sum_{i=1}^{\mathrm{c}} \mathrm{u_{ik}} = 1, \forall \mathrm{k} = 1, \ldots, n \qquad (5)$$

This condition means that total membership of each sample to c-clusters must be equal to 1. To get the formulas related to v_i and u_{ik} the defined objective function should be minimized. By using the above conditions and setting the derivation of the objective function equal to zero we get:

$$u_{\mathrm{ik}} = \frac{1}{\sum_{\mathrm{j=1}}^{\mathrm{c}} (\frac{\mathrm{d_{ik}}}{\mathrm{d_{jk}}})^{2}/(\mathrm{m-1})} \qquad (6)$$

$$v_{\mathrm{i}} = \frac{\sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{u_{ik}^{m} x_k}}{\sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{u_{ik}^{m}}} \qquad (7)$$

Using formulas 6 and 7, FCM clustering algorithm is as follows:

Algorithm steps:

1. Setting primary values for c, m, and U_0. Initial clusters are conjectured.
2. Cluster centers are calculated (v_i)
3. Calculation of the membership matrix from the calculated clusters in 2.
4. If $||U^{i+1}-U^i|| \le \varepsilon$ algorithm ends, else go to 2.

Although FCM algorithm always converges to a local minimum and is an unsupervised clustering algorithm it has some disadvantages. This method consumes lots of computing time and can be very sensitive to the initial guesses (Speed local minima) and noise. It may also get stuck at local minimums. Algorithm sensitivity to noise has been solved by considering a cluster for the data noise. In addition, if there were some labeled data, the initial guesses for the cluster centers would be optimized. [19]

C. *Fuzzy rules base*

Fuzzy logic was presented for the first time by Professor Zadeh [9] it has been applied in many fields of science and also for industrial purposes. The first and most successful application of fuzzy logic is in control systems and the most important practical aspect of this theory is implementing complex systems in the form of variables and fuzzy rules. Variables that are defined in fuzzy form (which are continuous and usually ranged between zero and one) and rules which are more familiar with human knowledge and are closer to real-world practical situations.

Fuzzy rules base includes rules in forms of IF-THEN which are imposed on the fuzzy variables. Definition method for “AND” and “OR” are decided by the expert, implication and aggregation must also be appropriately selected.

This paper aims to provide a method for producing rules which will be discussed in detail in Section 3. Considering the extensive discussion of fuzzy theory, reviewing [10] [11] [12] [13] is advised.

III. THE PROPOSED ALGORITHM

In a nutshell, the proposed algorithm can be divided into four parts:

1. Creating a decision tree based on input data using FCM.

2. Mining the rules of the tree (decision tree traversing).

3. Creating a fuzzy system using the rules created in the previous section

4. Fuzzy rules system optimization.

Section A; which can be mentioned as the strength of the proposed algorithm, analyzes the input data in each node and after revising individual features, constructs the decision tree. Basically, the utilization of decision tree for producing rules has been presented before in [3] and [4], but providing a different tree structure and the use of FCM in data separation are the innovative concepts which are effective in result optimization. To clarify the proposed method, sample data having four features and classified in three classes will be experimented. Selecting a feature per node, means that this feature, if elected, could split the data more. Consequently the final tree would lead the user to the final decision much more faster .For example, C4.5 tree separates the data using the entropy concept in such a way that the best features to minimize entropy will be selected and also a threshold for dividing data and the decision making will be set. In our example, since we have 3 classes of data, the threshold value is not one number but three.

Y1 separates the class 1 data from class 2, Y2 separates the data of class 2 from 3 and Y3 is the separator for classes 1 and 3 “Fig. 2”.

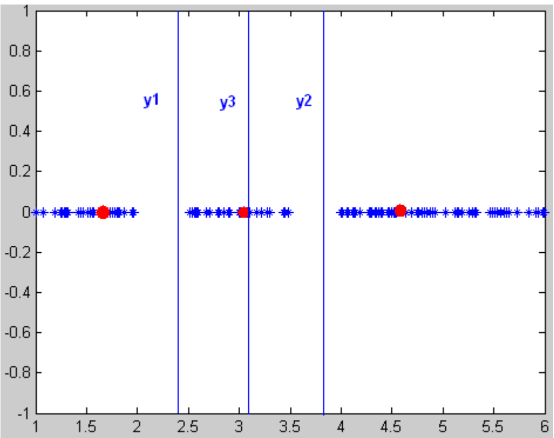


Figure 2. Data clustering, the cluster centers and separator lines.

In addition to detection method for detecting y1, y2, y3, that is based on FCM clustering. Providing a definition for preferred feature to choose for the current node, is also very important. Two proposed methods, i.e. the threshold detection method and method of selecting the best feature will be discussed in the A and B. it is also necessary to mention that although there is just one feature based for deciding each node in the sample data, decision making based on two or more features simultaneously is also possible and there are no limits on the number of features selected, which is one of the advantages of the presented algorithm itself.

A. *Separator threshold selection*

As mentioned in the background section, FCM is simply a clustering method. FCM cluster centers are randomly selected and optimized after some iterative processes. Suppose, the first feature (of 4 features) is selected as the one that should be analyzed, (As mentioned before in this process will be implemented for the other features as well). FCM will run on one-dimensional numbers related to the first feature. The number of classes (3) would be considered as an input to the FCM beforehand. FCM will define three cluster centers after the run. Splitter Line for Class 1 and 2 (y1) will be the mean center point of the relevant clusters. Y2 and y3 are also similarly selected.

B. *Deciding the best characteristics in each node for separation*

After selecting the separator points for the first features, the same process would be continued for the others. Being the top Feature in each node would depend on how it minimizes the objective function of FCM. In other words, based on the objective function (4), the best selected feature would be the one which minimizes the objective function most. After selecting the best features, in the next stage, algorithm re-examines the data class in the decision area. If data in decision area were all in the same class, a definite decision has been reached (final node) for the desired decision area. In this case, the data (which have led to the final decision) will be extracted from the primary data and the process returns again to the other nodes for running on the remaining data. Otherwise (failure to obtain the final decision), data of each cluster will be refined in the relevant returning process again. Recursive process will continue until all nodes reach the final end. Presented algorithm can be summarized as follows:

```
1. Start.
2. For each feature (x):
3. run FCM on the input data and create clusters
4. For the selected features ($x_i$) with the lowest objective function:
    a. In case clusters contained data from the same class, make a final node and delete data from the recursive cycle.
    b. Else, for each cluster`s data, rerun the process from 2 to 4.
```

Tree created by the above algorithm is displayed in “Fig. 3”.

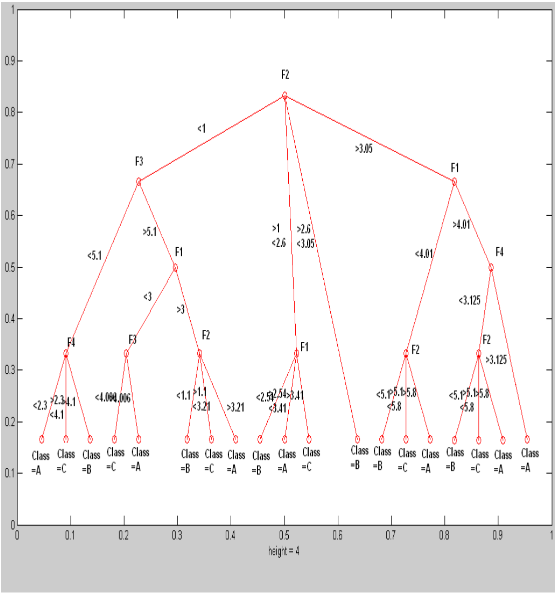


Figure 3. The implemented tree after running the algorithm

C. *extracting fuzzy rules and fuzzy making*

Traversing the created tree, converts it to a series of rules with conditions based on features, which can be utilized for fuzzy implementation. For example, part of the rules extracted from the tree of “Fig. 3”, with Depth-first Traversal is:

If f2<1 and f3<5.1 and f4<2.3 then Class=A

If f2<1 and f3<5.1 and 2.3<f4<4.1 then Class=C

If f2<1 and f3<5.1 and f4>4.1 then Class=B

Fuzzy implementation process can be summarized as follows:

- For each feature, a fuzzy variable will be defined.
- For each variable, all the nodes will be examined, to see if the condition applied to any specific point, so that this point can be inserted into the membership function. (Triangular or trapezoidal or any other similar types could be utilized as membership functions).
- Output variables are created. for as many as the number of classes,
- The extracted rules from the tree would be inserted into the fuzzy rules base.

For example, the membership function for the first input variable (feature 1) can be displayed as “Fig. 4”:

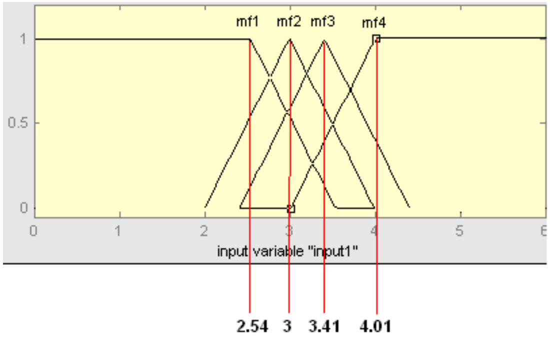


Figure 4. Variable membership function for F1 feature

(Center of functions of triangular membership has been created after traversing of all branches of the tree and surveying all the conditions which have been imposed on it).

D. *Optimization of the created fuzzy rules.*

In order to optimize the rules, as presented in [4] the use of genetic algorithm is suggested, however evaluation of that part is not the aim of the authors of this paper. Which could be performed in future attempts.

IV. Simulation Results

In order to simulate proposed algorithm, the MATLAB software is used. Training and testing data extracted from a valid database [14] and simulation have helped the claims to be proved. 64 samples of data (as described in Table 3) have been trained on 3 decision tress, C4.5, C5 and the presented decision tree (which is named as FCMDT). And 150 data samples have been used for testing. Evaluation results are displayed in Table 4.

TABLE III. DATASET USED IN THE EXPERIMENT

Number of test points	Number of training points	Number of classes	Number of features
150	64	3	4

TABLE IV. COMPARISON OF C4.5, C5 AND FCMDT TREES

Tree Type	Tree depth	Training Errors	Test Errors
C5	5	2	4
C4.5	4	2	5
FCMDT	8	0	4

According to the results, system response improvement compared to the previous trees (especially C4.5) is significant. Furthermore, not providing the same response in repetitive execution of the FCMDT algorithm is one of the advantages of the proposed solution. FCM does not always provide the same center for the clusters, so the resulted tree would not be the same in successive repetitions of the algorithm. This is also beneficial since creating a suitable tree is the only expected aim of the algorithm and the algorithm can be implemented as many times until a good tree with minimum error occurs on test data results. For example, Table 5 displays the Results of algorithm execution after 15 replicates:

TABLE V. RESULTS OF FCMDT ALGORITHM EXECUTION AFTER 15 REPLICATES

Repeat count	Tree Depth	Training Data error	Test Error
1	7	3	6
2	8	1	6
3	7	2	7
4	9	1	7
5	9	1	6
6	6	3	6
7	8	0	4
8	7	2	4
9	6	4	10
10	8	3	10
11	7	2	7
12	7	2	4
13	8	1	6
14	8	0	6
15	8	2	7

V. CONCLUSION

In this paper a new method for generating fuzzy rules is presented. Rules are extracted of traversing a decision tree. A tree which utilizes the FCM algorithm to separate data and produces rules using a new and efficient algorithm. Extracted rules, become fuzzy along with related variables (which are the features of the data) and the output of the system are evaluated under different conditions. Regarding the Good performance of the new decision tree (FCMDT), fuzzy rules created in the system are evaluated as efficient and predicted to become a new strategy for further researches in this field.

REFERENCES

[1] Quinlan, J. R. Induction of Decision Trees. Mach. Learn. 1, 1, 81-106, 1986.

[2] J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.

[3] Guillaume, S., Designing fuzzy inference systems from data: an interpretability-oriented review. IEEE Trans. Fuzzy Systems. v9 i3. 426-443.

[4] Pietari Pulkkinen , Hannu Koivisto, Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms, International Journal of Approximate Reasoning, v.48 n.2, p.526-543, June, 2008

[5] J. C. Bezdek: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.

[6] F. H¨oppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, (Methods for Classification, Data Analysis and Image Recognition), John Wiley & Sons, Ltd., 1999.

[7] Ross Qinlan Personal Academic Page. http://www.rulequest.com/Personal.

[8] RuleQuest Research Co. http://www.rulequest.com

[9] Zadeh, L. Fuzzy sets. Information and Control, 8(3):338-353. 1965.

[10] M. Sugeno, “An introductory survey of fuzzy control,” Inf. Sci., vol. 36, no. 1/2, pp. 59–83, 1985.

[11] C. C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller,” IEEE Trans. Syst., Man, Cybern., vol. 20, pp. 404–435, 1990.

[12] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” IEEE Trans. Syst., Man, Cybern., vol. 15, pp. 116–132, Jan./Feb. 1985.

[13] K. M. Passino and S. Yurkovich, Fuzzy Control Menlo Park, CA: Addison Wesley Longman, 1998.

[14] UCI machine Learning Repository web site: http://archive.ics.uci.edu/ml/datasets.html

[15] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.

[16] R. O. Duda, P. E. Hart, G. D. Stork, Pattern Classification. wiley-Interscience, ISBN-0-47b-0sbb9-3.

[17] R. Krishnapuram, J. M. Keller, A Possibilistic approach to clustering, IEEE Transactions on fuzzy systems, Vol. 1, No. 2, May 1993.

[18] Haj-Ahmadi, Homayoun poor, Ahadi -improved fuzzy c-mean clustering, the ability to cluster large and noisy data sets - the first conference on Fuzzy and Intelligent Systems, Mashhad, 2007.

[19] Dr.Saeed Shiry Ghidary Academic website, http://ceit.aut.ac.ir/~shiry