TAXONOMY OF CLUSTERING METHODS USED IN FUZZY LOGIC SYSTEMS

Essam F. Natsheh

Department of Management Information Systems,
College of Applied Studies & Community Services,
King Faisal University, Saudi Arabia

Email: dr_natsheh@hotmail.com

Abstract

Fuzzy logic systems have many applications in every field of moderate science. Most of the fuzzy logic systems are rule based reasoning, which are not easy to generate since the conflict between rules always arise in acquiring new knowledge. In recent years, there has been increasing interest in clustering-based fuzzy systems, which are easier to generate rules since they built from input-output training data. Clustering training data make the fuzzy system easier to maintain and more flexible in acquire real world knowledge. In this paper, we present taxonomy of clustering methods used in fuzzy logic systems. In particular, the exposition includes a discussion of strength and weakness of these methods and how they can be improved.

Keywords: *Fuzzy logic systems, clustering methods, fuzzy inference system, Sugeno-type fuzzy system.*

I. INTRODUCTION

AS a general rule, a good engineering approach should be able to make effective use of all the available information. If the mathematical model of a system is too hard to obtain (this is true for many practical systems), then the most important information comes from human experts who provide linguistic descriptions about the system and control instructions. Conventional mathematical-based systems cannot incorporate the linguistic fuzzy information into their designs. If in some situations the most important information comes from human experts, then the fuzzy logic systems (FLS) is the best choice.

Many design methods for the FLS have been developed during the last decade. For most of these design methods it was implicitly assumed that expert information is available. This information included a number of the IF ... THEN rules in the knowledge base, and rough estimates of the parameters defining the antecedent and consequent templates. In this paper, we discuss an alternative and more user-friendly approach to rule generation based upon a clustering of the input-output data [1].

Data Clustering is an effective knowledge acquisition methodology. It has been employed to build a clustering-based fuzzy inference system that best models the data behavior using a minimum number of rules. The success of this model has been proved through the routine of using it in literature with high accuracy. Its knowledge base is organized as a binary tree; each node of a tree contains a rule and a cluster center that invoke the creation of the rule in this node. It can provide only one conclusion for a data case. It has been conceived as a methodology to use all these experts' behavioral features when they are maintaining a knowledge base. In particular, it focuses on adding a refinement to capture the identified difference instead of attempting to modify the existing knowledge base. The resulting structure of the knowledge base can be viewed as a binary tree where every node is an "if ..then" rule containing only conjunctions that remember the data which was mis-classified and resulted in the rule being added. These data are termed "cluster center".

In this paper we present a survey on various clustering algorithms used in fuzzy logic systems. We outline the main steps that have been used by every clustering method and discuss some drawbacks of these methods and possible solutions.

The remainder of this paper is structured as follows: we present a review of current research efforts classified as supervised clustering algorithms and unsupervised clustering algorithms. Later sections discuss the related works and the paper's conclusion with further researches possibilities.

II. TAXONOMY OF CLUSTERING METHODS USED IN FUZZY LOGIC SYSTEMS

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to distill natural groupings of data from a large data set, producing a concise representation of a system's behavior.

The traditional principle of grouping objects into clusters utilizes some measure of object similarity, usually the reciprocal of a distance measure. The objective of clustering is to group a set of objects into clusters such that the objects within a cluster have a high degree of similarity, while elements belonging to different clusters have a high degree of dissimilarity. In contrast to methods of clustering that exactly assign each object to one cluster (this is called hard clustering), fuzzy-clustering allows of vagueness of the data [2]. In spite of the simplicity of hard clustering algorithms, they have two major problems:

a) Each of the neighbors is considered equally important in determining the classification of the input data. A far neighbor is given the same weight as a close neighbor of the input.
b) The algorithm only assigns a class to the input data; it does not determine the "strength" of membership in the class.

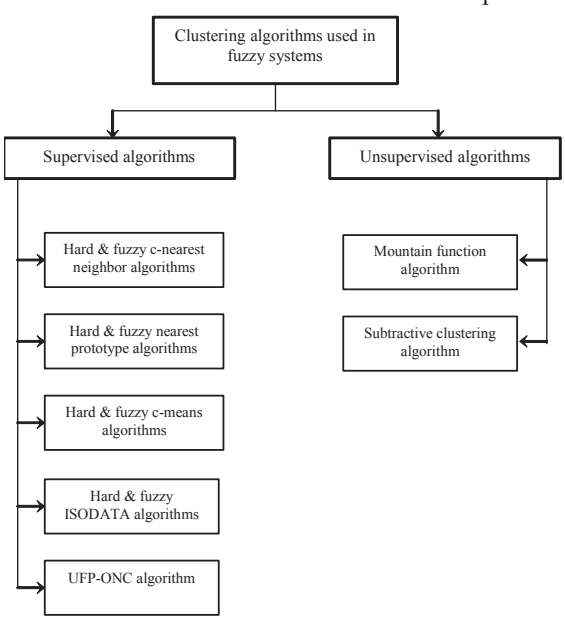

Figure 1: Taxonomy of clustering algorithms used in fuzzy logic systems.

Figure 1 show various clustering algorithms used in fuzzy logic systems. We classify them as supervised and unsupervised algorithms that depend on previous determination of the number of clusters that the algorithms will generate.

III. SUPERVISED CLUSTERING ALGORITHMS

In supervised algorithms the number of clusters C should be known before applying the algorithm. The classification of this type will be discussed in the following.

A. Hard and fuzzy c-nearest neighbor algorithms

The hard c-nearest neighbor (c-NN) algorithm rule assigns an input sample vector, which is of unknown classification, to the cluster of its neighbor. This algorithm is widely used with fuzzy logic systems for its simplicity [3-5].

While the fuzzy c-nearest neighbor (FCNN) algorithm [6] is also a clustering algorithm the form of its results differs from the crisp c-nearest neighbor. The FCNN algorithm assigns cluster membership to a sample vector rather than assigning the vector to a particular cluster. The advantage is that the algorithm makes no arbitrary assignments. In addition, the vector's membership values should provide a level of assurance to accompany the resultant classification. For example, if a vector is assigned 0.9 membership in one cluster and 0.05 membership in other cluster we can be reasonably sure the cluster of 0.9 membership is the cluster to which the vector belongs [5-6].

B. Hard and fuzzy nearest prototype algorithms

These clustering algorithms bear a marked resemblance to the one nearest neighbor algorithm. Actually, the only difference is that for the nearest prototype algorithm the labeled samples are a set of cluster prototypes, whereas in the nearest neighbor algorithm we use a set of labeled samples that are not necessarily prototypical. Of course, the nearest prototype algorithm could be extended to multiple prototypes representing each cluster, similar to the c-nearest neighbor routine. The prototypes may be used as the cluster means of the labeled sample set [6]. The major problem of these two algorithms is they produce a large error rate.

C. Hard and fuzzy c-means algorithms

The hard c-means (HCM) algorithm tries to locate clusters in the multi-dimensional feature space. The goal is to assign each point in the feature space to a particular cluster. The fuzzified c-means algorithm allows each data point to belong to a cluster to a degree specified by a membership grade, and thus each point may belong to several clusters.

The fuzzy c-means (FCM) algorithm partitions a collection of K data points specified by m-dimensional vectors u_k (k=1, 2,…,K), into C fuzzy clusters, and finds a cluster center in each, minimizing an objective function. Fuzzy c-means is different from hard c-means, mainly because it employs *fuzzy partitioning* where a point can belong to several clusters with degrees of membership. To accommodate the fuzzy partitioning, the membership matrix M is allowed to have elements in the range [0, 1]. The membership matrix M must satisfy the following property:

The sum of each column is one (1)

Formally, the c-means algorithm finds a center in each cluster, minimizing an objective function of a distance measure. The objective function depends on the distances between vectors u_k and cluster centers c_i, the expression for the objective function is:

$$J(\mathrm{M}, \mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_C) = \sum_{i=1}^{C} J_i = \sum_{i=1}^{C} \sum_{k=1}^{K} m_{ik}^{q} \, d_{ik}^{2} \quad (2)$$

where J_i is the objective function within cluster i, K is the number of data points, C is the number of clusters, m_{ik} is a membership between 0 and 1, c_i is the center of fuzzy cluster i, $d_{ik} = || u_k - c_i ||$ is the Euclidean distance between the ith cluster center and kth data point, and $q \in [1, \infty)$ is a weighting exponent (typically q = 2 [7]). There are two necessary conditions for J to reach a minimum,

$$c_i = \frac{\sum_{k=1}^{K} m_{ik}^{q} u_k}{\sum_{k=1}^{K} m_{ik}^{q}}, \tag{3}$$

and

$$m_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}} \tag{4}$$

Algorithm: The fuzzy c-means algorithm determines the cluster centers ci and the membership matrix M using the following steps [8]:

1. Initialize the membership matrix M with random values between 0 and 1 within the constraints of (1).
2. Calculate C cluster centers, using (3).
3. Compute the objective function according to (2). Stop if either it is below a certain threshold level or its improvement over the previous iteration is below a certain tolerance.
4. Compute a new M using (4)
5. Go to step 2.

The cluster centers can alternatively be initialized first, before carrying out the iterative procedure. The algorithm may not converge to an optimum solution and the performance depends on the initial cluster centers. Although of this problem a lot of authors use this algorithm in their fuzzy systems [2], [5], [7-10].

D. Hard and fuzzy ISODATA algorithms

The ISODATA algorithm can be considered to be an enhancement of the approach taken by c-NN algorithm and c-means algorithm. Like those algorithms, it tries to minimize the squared error by assigning objects (samples) to the nearest center. Unlike those algorithms, it does not deal with fixed number of clusters but rather it deals with C clusters where C is allowed to vary over an interval that includes the number of clusters requested by the user. It discards clusters with few elements. Clusters are merged if the number of clusters grows too large or if clusters are too close together. A cluster is split if the number of clusters is too few or if the cluster contains very dissimilar objects [11]. Some of the work that used hard and fuzzy ISODATA algorithms in their fuzzy systems are [12-14].

E. Unsupervised fuzzy partition-optimal number of classes (UFP-ONC) algorithm

Gath and Geva [15] derive new algorithm from a combination of the fuzzy c-means algorithm and the fuzzy maximum likelihood estimation, called unsupervised fuzzy partition-optimal number of classes (UFP-ONC). This algorithm performs well in situations of large variability of clusters shapes, densities, and number of data points in each cluster. The algorithm use unsupervised tracking of classification prototypes but maximum number of clusters that the algorithm generates must be determined.

IV. UNSUPERVISED CLUSTERING ALGORITHMS

The major problem of supervised algorithms that it is necessary to tell the algorithm how many clusters C to look for. If C is not known beforehand, it is necessary to apply an unsupervised algorithm.

A. Mountain clustering method

Yager and Filev [16] proposed a simple and effective algorithm called the Mountain Method, for estimating the number and initial location of cluster centers. Their method is based on griding the data space and computing a potential value for each grid point based on its distances to the actual data points; a grid point with many data points nearby will have a high potential value. The grid point with the highest potential value is chosen as the first cluster center. The key idea in their method is that once the first cluster center is chosen, the potential of all grid points are reduced according to their distance from the cluster center. Grid points near the first cluster center will have greatly reduced potential. The next cluster center is then placed at the grid point with the highest remaining potential value. This procedure of acquiring new cluster center and reducing the potential of surrounding grid points repeats until the potential of all grid points fall below a threshold [5], [7].

Although this method is simple and effective, the computation grows exponentially with the dimension of the problem. For example, a clustering problem with 4 variables and each dimension having a resolution of 10 grid lines would result in 104 grid points that must be evaluated [7].

B. Subtractive clustering method

Subtractive clustering, developed by Stephen Chiu [7] is based on a measure of the density of data points in the feature space (subtractive clustering is an extension of Yager and Filev's Mountain Method). The idea is to find regions in the feature space with high densities of data points. The point with the highest number of neighbors is selected as center for a cluster. The data points within a prespecified, radius are then removed (subtracted), and the algorithm looks for a new point with the highest number of neighbors. This continues until all data points are examined [8], [17].

Consider a collection of K data points specified by m-dimensional vectors u_k (k=1, 2,…, K). Without loss of generality, the data points are assumed normalized. Since each data point is a candidate for a cluster center, a density measure at data point uk is defined as:

$$D_k = \sum_{j=1}^{K} \exp\left(- \frac{\| u_k - u_j \|}{(r_a / 2)^2} \right), \tag{5}$$

where r_a is a positive constant. Thus, the measure of density for a data point is a function of its distances to all other data points. A data point with many neighboring data points will have a high-density value. The constant r_a is effectively the radius defining a neighborhood; data points outside this radius have little influence on the density measure.

After calculating the density measure for each data point, the point with the highest density is selected as the first cluster center. Let u_{C1} be the point selected and D_{C1} its density measure. Next, the density measure for each data point u_k is revised by the formula:

$$D'_k = D_k - D_{C1} \exp\left(- \frac{\| u_k - u_{C1} \|}{(r_b / 2)^2} \right), \tag{6}$$

where rb is a positive constant. Thus, we subtract an amount of density from each data point as a function of its distance from the first cluster center. The data points near the first cluster center uC1 will have greatly reduced density, and therefore will unlikely be selected as the next cluster center. The constant rb is effectively the radius defining the neighborhood which will have measurable reductions in density. To avoid obtaining closely spaced cluster centers, we set rb to be somewhat greater than ra ; a good choice is rb = 1.5 ra .

Chiu [7] proposes the following procedure for accepting/rejecting cluster centers:

If $D_{k'} > \bar{\varepsilon} D_{C1}$

Accept uk' as a cluster center and continue.

Else if $D_{k'} < \underline{\varepsilon} D_{C1}$

Reject u_k' and end the clustering process.

Else

Let $d_{\min}$ = shortest of the distances between u_k' and all previously found cluster centers.

$$\frac{d_{\min}}{r_a} + \frac{u_k'}{u_{C1}} \geq 1$$

If

Accept u_k' as a cluster center and continue.

Else

Reject u_k' and set the density at uk' to 0.
Select the data point with the next highest
density as the new uk' and re-test.
End if

End if

Here $\bar{\varepsilon}$ specifies a threshold for the density above which the procedure will definitely accept the data point as a cluster center; $\underline{\varepsilon}$ specifies a threshold below which the procedure will definitely reject the data point. Chiu uses $\bar{\varepsilon}$ = 0.5 and $\underline{\varepsilon}$ = 0.15. If the density falls in the gray region, the procedure check if the data point provides a good trade-off between having a sufficient density and being sufficiently far from existing cluster centers.

Although the number of clusters is automatically determined by this method, we should note that the user specified parameter ra (i.e., the radius of influence of a cluster center) strongly affects the number of clusters that will be generated. A large ra generally results in fewer clusters and hence a coarser model, while a small ra can produce excessive number of clusters and a model that does not generalize well (i.e., by over-fitting the training data) [8]. To illustrate this principle, Chiu [7] applied this algorithm with different values of ra ranging from 0.15 to 0.5 for the Mackey-Glass time series problem as benchmark. This produced models of varying size, ranging from 69 clusters to 9 clusters.

V. RELATED WORKS

Baldwin [13] proposes a new approach for clustering namely the semantic unification method. This approach has many applications for fuzzy logic systems in the guise of case base reasoning.

All methods for designing neural networks used for clustering and used to design fuzzy logic systems, related works are [13], [18]. Anan [4] proposes simple taxonomy of these methods.

VI. SUMMARY

One of the powerful techniques to generate the rules of the fuzzy systems is using the clustering algorithms. Taxonomy of clustering algorithms used with fuzzy logic systems was reviewed in this paper. The main problem of supervised clustering algorithms is determination of the number of clusters that satisfactorily represent the system. Unsupervised clustering algorithms solve this problem. In spite of that, unsupervised algorithms have another problem. They require determination of some parameters that will affect on (increase/decrease) the number of generated clusters. In future work, we'll propose two non-parametric clustering algorithms to overcome this problem. These algorithms will be used to extract the rule-base of the fuzzy logic systems.

ACKNOWLEDGMENTS

The author would like to express his appreciation to Deanship of Scientific Research at King Faisal University for supporting this research.

REFERENCES

[1] G. M. Abdel-Nour, C. Chang, F. Hung and J. Y. Cheung, "Design of a Fuzzy Controller Using Input and Output Mapping Factors", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-21, No. 5, pp. 952-960, 1991.

[2] K. Hirota and W. Pedrycz, "Subjective Entropy of Probabilistic Sets and Fuzzy Cluster Analysis", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-16, No. 1, pp. 173-178, 1986.

[3] L. Wang, "Adaptive Fuzzy Systems and Control: Design and Stability Analysis", PTR Prentice Hall, 1994.

[4] H. Anan, "Two Novel Fuzzy Knowledge Acquisition Techniques", M. Sc. thesis, Alexandria University, 1997.

[5] J. Yen and R. Langari, "Fuzzy Logic: Intelligence, Control and Information", Prentice-Hall, 1999.

[6] J. Keller, M. R. Gray and J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-15, No. 4, pp. 580-585, 1985.

[7] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation", J. of Intelligent & Fuzzy Systems, Vol. 2, No. 3, pp. 267-278, 1994.

[8] J. Jantzen, "Neurofuzzy Modeling", Technical University of Denmark: Department of Automation, Technical report number 98-H-874, 30 Oct 1998, URL: http://www.iau.dtu.dk/

[9] J. C. Bezdek, R. J. Hathaway, M. J. Sabin and W. T. Tuker, "Convergence Theory for c-Means: Counterexamples and Repairs", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 5, pp. 873-877, 1987.

[10] R. L. Cannon, J. V. Dave and J. C. Bezdek, "Efficient Implementation of the Fuzzy c-Means Clustering Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 248-255, 1986.

[11] E. Gose, R. Johnsonbaugh and S. Jost, "Pattern Recognition and Image Analysis", Prentice Hall PTR, 1996.

[12] H. Bandemer and S. Gottwald, "Fuzzy Sets, Fuzzy Logic and Fuzzy Methods with Applications", John Wiley & Sons, 1995.

[13] J. F. Baldwin, "Fuzzy Sets, Fuzzy Clustering and Fuzzy Rules in AI", Proceedings of IJCAI'93 Workshop, pp.10-23, Chamberry, France, August 28, 1993.

[14] S. Z. Selim and M. A. Ismail, "On the Local Optimality of the Fuzzy ISODATA Clustering Algorithm", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 2, pp. 284-288, 1986.

[15] Gath and A. B. Geva, "Unsupervised Optimal Fuzzy Clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 773-781, 1989.

[16] R. R. Yager and D. P. Filev, "Approximate Clustering Via the Mountain Method", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-24, No. 8, pp. 1279-1284, 1994.

[17] S. Chiu, "Selecting Input Variables for Fuzzy Models", J. of Intelligent & Fuzzy Systems, Vol. 4, No. 4, pp. 243-256, 1996.

[18] S. Mitra and S. K. Pal, "Self-Organizing Neural Network as a Fuzzy Classifier", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-24, No. 3, pp. 385-399, 1994.