

A New Hierarchical Clustering Algorithm on Fuzzy Data (FHCA)

Mohammad GhasemiGol, Hadi Sadoghi Yazdi, Reza Monsefi

Abstract—This paper presents a novel approach to cluster Fuzzy numbers using hierarchical method to be called (FHCA). In this approach a Dendrogram is drawn over fuzzy numbers until we could cluster fuzzy numbers using hierarchical cluster tree with inconsistency coefficient or other useful measures. All the similar previous methods extended FCM (Fuzzy Clustering Method) to support fuzzy data. On contrary, the present work is based on hierarchical method, i.e., we extended the hierarchical clustering algorithm to cluster fuzzy data for the first time. Finally this approach has been compared with some of the newly presented methods in the literature. The major advantage of the algorithm is its fault tolerance against noisy samples.

Index Terms—Fuzzy data, Hierarchical clustering algorithm, Fuzzy dendrogram, Dissimilarity measure.

I. INTRODUCTION

For a long time, clustering methods has attracted researcher in various applications in diverse areas. In many clustering methods crisp data can only be used. There are two main groups in clustering algorithms:-

- Clustering algorithm that works on only crisp data
- Clustering algorithm that can work on fuzzy data as well as crisp data

Here we are concerned with the second group.

Fuzzy data is a kind of data that is imprecise or with some source of uncertainty. This data type has been readily used in natural language, social science, knowledge representation, etc. Recently some methods have been presented in this field; Amir B. Geva presented a new recursive algorithm for hierarchical fuzzy partitioning [8]. Miin-Shen Yang, et al. presented Fuzzy clustering algorithms for mixed feature variables [4]; Hathaway et al. proposed FCM for fuzzy data [12]; Vicenc Torra presented another type of Fuzzy c-means [7].

Some of these papers are concentrated on crisp data [5,7,8,13], and some others extended FCM (that is the one of the widely used fuzzy clustering models [3,9]) to cluster

fuzzy data [4,12,14,17]. In the present work, we focused on the hierarchical clustering method for clustering fuzzy data (a non-fuzzy algorithm for clustering fuzzy data). In other words, an extension for hierarchical clustering algorithm that can work on fuzzy data too. Due to the usage of hierarchical methods, the proposed algorithm has inherited all the benefits of these methods, since it is very illustrative and clear that we can guesstimate dispersion of clusters in a glance at the dendrogram as well as its fault tolerance against noisy data.

We use the Miin-Shen Yang dissimilarity measures and build a dissimilarity matrix based on it. Then we can use different linkage algorithms (single-link, complete-link, average, median, centroid and weighted average) to create hierarchical cluster tree. Due to the usage of fuzzy data, we named this algorithm “FHCA”, and the created tree “Fuzzy dendrogram”. Finally we cluster data by proper measures like inconsistency coefficient [16] or maximum number of clusters.

As an important deficiency, all of similar works cannot tolerate noise on samples. We demonstrate our proposed method can resolve this defect by discovering noisy samples and clustering them into the separate clusters.

This paper is organized as follows. In the next section we review hierarchical clustering method. The proposed method is explained in Section 3. Finally, in the last section we evaluate the performance of FHCA and compare it to similar proposed algorithms.

II. REVIEW OF HIERARCHICAL CLUSTERING METHOD

Hierarchical clustering procedures are the most commonly used method of summarizing data structure. A hierarchical tree is a nested set of partitions represented by a tree diagram or dendrogram (see Fig 1). Sectioning a tree at a particular level produces a partition into g disjoint groups. If two groups are chosen from different partitions (the results of partitioning at different levels) then either the groups are disjoint or one group wholly contains the other [1].

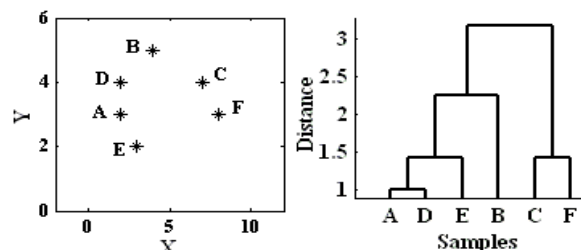


Figure 1. Dendrogram is drawn over six samples. The hierarchical algorithm contained the following

Manuscript received June 24, 2009.

Mohammad GhasemiGol is with the Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran (phone: +98-915-9620831; fax: +98-561-4434070; e-mail: ghasemigol@wali.um.ac.ir).

Hadi Sadoghi Yazdi is with the Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran (phone: +98-915-1738312; e-mail: h-sadoghi@um.ac.ir).

Reza Monsefi is with the Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran (phone: +98-915-1135763; e-mail: rmonsefi@um.ac.ir).

procedure, where c is the desired number of final clusters. If $c=1$ then the dendrogram could be created [2].

Algorithm (Agglomerative hierarchical clustering)

- *Begin*
- Initialize $\hat{c}, \hat{c} \leftarrow n, D_i \leftarrow \{x_i\}, i = 1, \dots, n$
- Do $\hat{c} \leftarrow \hat{c} - 1$
- Find nearest clusters, say, D_i and D_j
- Merge D_i and D_j
- Until $\hat{c} = c$
- Return c clusters
- *End*

Here $c=1$. By this technique we will draw cluster's dendrogram and use it to specify clusters. So at first, dissimilarity matrix is created. This matrix shows the distance between each pair of samples. Suppose that at the beginning, every sample is a cluster with one sample. Then in each step two clusters that are closer to each other get selected and joined as a new cluster. At the end, we have a nested set of clusters that can be analyzed.

In the hierarchical method we use several mechanisms to obtain the distance of two clusters. One of which is single-link method. In this method the distance between two clusters is defined as the distance between their closest members of two clusters. In other words the distance between two groups, A and B, is defined as:

$$d_{AB} = \min_{i \in A, j \in B} (d_{ij})$$

Another mechanism is complete-link. In this method the distance of two clusters is defined as the distance between their furthest members of two clusters, i.e., the distance between two groups, A and B, is

$$d_{AB} = \max_{i \in A, j \in B} (d_{ij})$$

In this method, we make sure that other samples of two clusters are closer than the distance between of them.

III. THE PROPOSED HIERARCHICAL CLUSTERING ALGORITHM (FHCA)

In this section we explain how we can generalize hierarchical method for fuzzy data. For this reason first we should define fuzzy data and measure to obtain the distance between them. We can extend symmetric trapezoidal fuzzy numbers (TFNs) to all TFNs by defining its parameterization as shown in Fig 2 Parameterization of a trapezoidal fuzzy number A is denoted by $A = m(a_1, a_2, a_3, a_4)$ where a_1, a_2, a_3, a_4 are called center, inner diameter, left outer radius and right outer radius respectively [4,10,15].

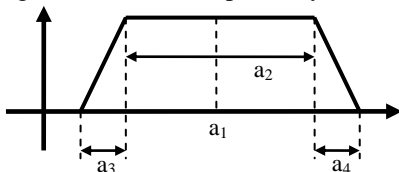


Figure 2. Parameterization of trapezoidal fuzzy data.

The benefit of this representation is that we can easily show four kinds of fuzzy data (see Fig 3). According to the representation above, $A = [a_1, 0, 0, 0]$, $B = [b_1, b_2, b_3, b_4]$, $C = [c_1, c_2, 0, 0]$, $D = [d_1, 0, d_3, d_4]$.

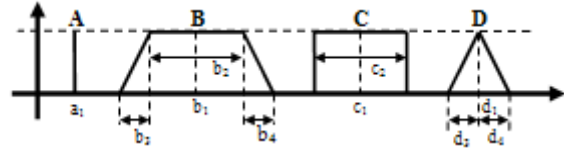


Figure 3. Four kinds of trapezoidal fuzzy data.

Let $A = m(a_1, a_2, a_3, a_4)$ and $B = m(b_1, b_2, b_3, b_4)$ be any two fuzzy data. Hathaway et al. [12] defined dissimilarity for two TFNs A and B as follows:

$$d_h^2(A, B) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + (a_4 - b_4)^2.$$

However, they did not consider the left or right shapes of numbers (i.e. LR- type TFN). Yang et al. [4] gave a distance for two symmetric TFNs A and B based on yang and Ko's distance definition [15] as follows:

$$d_f^2(A, B) = \frac{1}{4} (g_-^2 + g_+^2 + (g_- - (a_3 - b_3))^2 + (g_+ - (a_4 - b_4))^2)$$

$$\text{Where } g_- = 2(a_1 - b_1) - (a_2 - b_2)$$

$$\text{And } g_+ = 2(a_1 - b_1) + (a_2 - b_2)$$

For example if $A = [0,0,1,1]$ and $B = [0,0,3,3]$ then:

$$g_- = 2(0 - 0) - (0 - 0) = 0$$

$$g_+ = 2(0 - 0) + (0 - 0) = 0$$

$$d_f^2(A, B) = \frac{1}{4} (0^2 + 0^2 + (0 - (1 - 3))^2 + (0 + (1 - 3))^2) = 2$$

Now, we explain our proposed algorithm for clustering fuzzy data. This algorithm has five steps that will be explained in this section.

The Proposed Hierarchical Clustering Algorithm (FHCA)

- *Begin*

$$\text{Initialize } x = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{bmatrix},$$

(n =number of samples, each row of x identifies a fuzzy data.)

- Compute the distance between each pair of fuzzy data
- Create dissimilarity matrix.
- Create Fuzzy dendrogram.
- Extract clusters from fuzzy dendrogram.(with inconsistency coefficient or max number of clusters)
- *End*

Step1: For each fuzzy data, we need four crisp numbers to

show it. So with x (is an n -by-4 matrix) we can present n fuzzy data samples.

Here we explain the proposed algorithm to cluster fuzzy data by a simple example. Suppose that we have six fuzzy data (see Fig 4).

$A = [2.5; 2; 0.5; 1]$, $B = [2.5; 0; 1; 1]$,
 $C = [7; 1; 0.5; 1]$, $D = [8; 0; 1; 1]$,
 $E = [8.5; 0; 0.5; 0.5]$, $F = [4.5; 0; 1; 1]$.

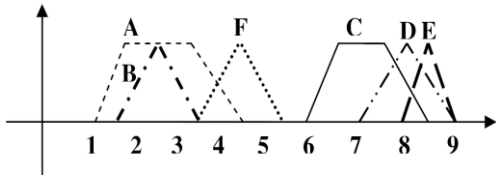


Figure 4. Six fuzzy data A, B, C, D, E and F.

So, x matrix is shown as below.

$$x = \begin{bmatrix} 2.5 & 2 & 0.5 & 1 \\ 2.5 & 0 & 1 & 1 \\ 7 & 1 & 0.5 & 1 \\ 8 & 0 & 1 & 1 \\ 8.5 & 0 & 0.5 & 0.5 \\ 4.5 & 0 & 1 & 1 \end{bmatrix}$$

Step2: In this stage, we use Yang distance formula to reach the distance between each two fuzzy data samples. (For example $d_f(A, B) = 1.89$, $d_f(A, C) = 9.05$, ...)

Step3: Now, we can easily create dissimilarity matrix. In this matrix each item shows the distance between two fuzzy data. This matrix is used for creating fuzzy dendrogram.

TABLE.1 Dissimilarity matrix for the samples in 错误! 未找到引用源。 .

	A	B	C	D	E	F
A	0	1.89	9.05	11.04	12.06	4.31
B		0	9.17	11.00	12.00	4.00
C			0	2.08	3.09	5.20
D				0	1.06	7.00
E					0	8.01
F						0

Step4: Here, we should choose a method such as single-link, complete-link, median, etc. to create fuzzy dendrogram and form our clusters. We discussed two methods of them (single-link and complete-link) in section 2. Each method has its priority and can be useful in some cases.

- 'single-link' --- nearest distance
- 'complete-link' --- furthest distance
- 'average' --- unweighted average distance (UPGMA) (also known as group average)
- 'weighted' --- weighted average distance (WPGMA)
- 'centroid' --- unweighted center of mass distance (UPGMC)
- 'median' --- weighted center of mass distance (WPGMC)
- 'ward' --- inner squared distance (minimum variance algorithm)

If we run single-link method on the dissimilarity matrix, we reach these results (see Table 2).

Table 1: The stages (a, b, c, d) of executing single-link method on 错误! 未找到引用源。 .

TABLE 2.a

	A	B	C	{D,E}	F
A	0	1.89	9.05	11.04	4.31
B		0	9.17	11.00	4.00
C			0	2.08	5.20
{D,E}				0	7.00
F					0

TABLE 2.b

	{A, B}	C	{D,E}	F
{A,B}	0	9.05	11.00	4.00
C		0	2.08	5.20
{D,E}			0	7.00
F				0

TABLE 2.c

	{A, B}	{C,D,E}	F
{A, B}	0	9.05	4.00
{C,D,E}		0	5.20
F			0

TABLE.2.d

	{A,B,F}	{C,D,E}
{A,B,F}	0	5.20
{C,D,E}		0

You can see the achieved dendrogram in Fig 5.

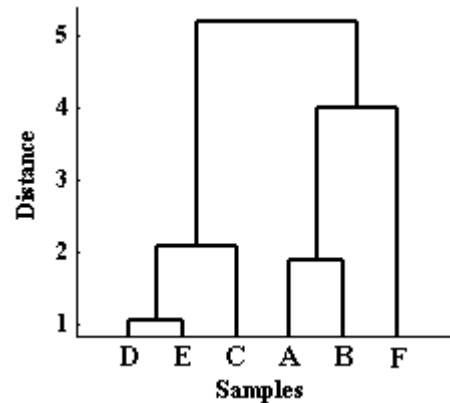


Figure 5. Fuzzy dendrogram for six fuzzy samples.

Step5: This is the most important stage in our algorithm. Here we should extract the clusters from the fuzzy dendrogram. For this purpose there are two ways. One way is based on the max number of clusters. For example if we want to have two clusters, it is enough to aggregate samples that are in each sub tree of dendrogram root, in a new cluster. In other words, we can divide dendrogram from the highest level into two clusters. This result is deducible from Fig 4.

The other way is based on the inconsistency coefficient [16]. In this way we define a label for each link in our dendrogram. This label shows how much two clusters are similar. With this measure, we can join clusters if the inconsistency value is less than specific threshold. The inconsistency coefficient characterizes each link in a cluster tree by comparing its length with the average length of other links at the same level of the dendrogram. The higher the value of this coefficient, the less similar the clusters connected by the link. To calculate inconsistency coefficient we should define two matrixes.

Linkage matrix is an $(n-1)$ -by-3 matrix containing cluster tree information. Table 3 shows the value of Linkage matrix for the samples in Fig 4.

TABLE.3 Linkage matrix for the samples in Fig 4.

D	E	1.06
A	B	1.89
C	{D,E}	2.08
{A,B}	F	4.00
{A,B,F}	{C,D,E}	5.20

The other matrix is Inconsistency matrix (Table 4) that is an (n-1)-by-4 matrix, formatted as follows.

TABLE.4 Inconsistency matrix properties

Column	Description
1	Mean of the lengths of all the links included in the calculation.
2	Standard deviation of all the links included in the calculation.
3	Number of links included in the calculation.
4	Inconsistency coefficient.

We used the distance between two clusters, as length of the link that connects them to each other. If we name Linkage matrix, Z and Inconsistency matrix, W then the inconsistency coefficient for each link, is calculated by following formula:

$$W(k, 4) = (Z(k, 3) - W(k, 1)) / W(k, 2) \quad (1)$$

For leaf nodes, nodes that have no further nodes under them, the inconsistency coefficient is set to zero.

TABLE.5 Inconsistency matrix for the samples in Fig 4.

1.06	0	1	0
1.89	0	1	0
1.57	0.72	2	0.71
2.94	1.49	2	0.71
2.85	1.70	5	1.38

To understand this matrix, pay attention to the last row of inconsistency matrix which is shown in Table 5. The calculation consists of five links, so the value of third column is five. Also the mean of used links is:

$$\mu = \frac{1.06 + 1.89 + 2.08 + 4 + 5.20}{5} = 2.85$$

And calculated Standard deviation is 1.70. For computing inconsistency coefficient we used formula (1).

$$W(5, 4) = (Z(5, 3) - W(5, 1)) / W(5, 2) \\ = (5.20 - 2.85) / 1.70 = 1.38$$

In Fig 6, you can see the inconsistency coefficient related to each link.

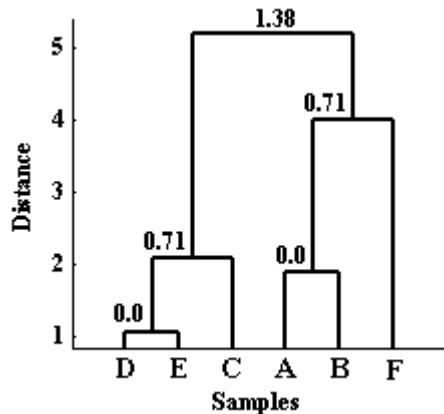


Figure 6. Inconsistency coefficient related to each link.

Now, we can use inconsistency coefficient to form clusters. If we define threshold=1, then we have two clusters C1= {A, B, F} and C2= {C, D, E}.

According to Fig 4 this result is perfectly acceptable. Therefore, in this method the number of cluster are defined via inconsistency coefficient and we do not define it.

An important benefit of this algorithm is its fault tolerance against noisy samples. We have tree structure of clusters and by analyzing that exactly, we can discover the noisy samples in two cases, low SNR¹ and medium SNR.

A. Low SNR

Indisputably, noisy samples are far from other samples, so the inconsistency coefficient between noisy samples and normal samples is larger. Thus, we can find noisy samples when the inconsistency coefficient grows up suddenly. In other words, this algorithm, noisy samples are classified in separated clusters. Usually, noisy samples appear in highest level of dendrogram.

B. Medium Level of SNR

FHCA can tolerate the medium domain of noise on some samples. For example if we have a little noise on a sample, it shouldn't have any effect on our clustering. In the other words, medium level of noise must not effect over clusters or clustering procedure must be robust against medium level of noise.

We consider a data set with six triangular fuzzy data (Fig 7). We cluster them and then put a bit noise on a sample. Afterwards we cluster them again and compare the obtained results in two stages (Low level of noise on some samples). After clustering these samples, we obtained a dendrogram that is shown in Fig 8.

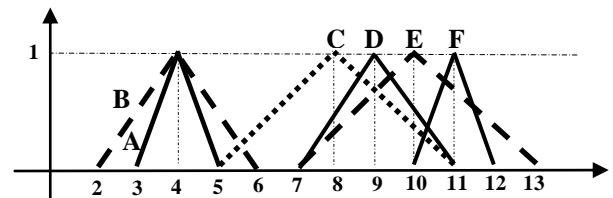


Figure 7. Representation of six fuzzy data.

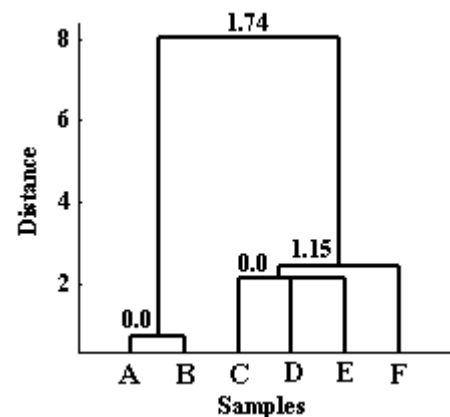


Figure 8. Fuzzy dendrogram for the samples in Fig 7.

¹ Signal to Noise Ratio

As it can be seen, we have two clusters by applying the inconsistency coefficient and supposing threshold between 1.15 and 1.74 for example mean of them (threshold=1.4450)

First, suppose that we only have noise on center of a sample. For example we received $D = [9 \pm \alpha; 0; 2; 2]$ instead of $D = [9; 0; 2; 2]$. We want to find the maximum value of α that doesn't alter our clustering. After executing this experiment, $\alpha = 2.04$.

For example if $D = [6.96; 0; 2; 2]$, we have the following dendrogram (see Fig 9). This dendrogram shows the maximum noise on the center of sample D that can be supported. If the noise is bigger than 2.04 then we have different clusters.

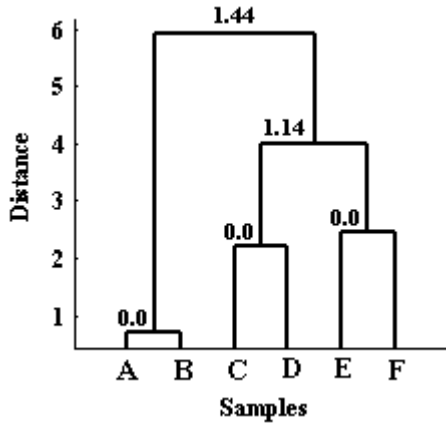


Figure 9. Fuzzy dendrogram for Fig 7 samples. (After adding noise to center of D)

So, if we suppose threshold=1.4450 and run FHCA on these samples, it can tolerate %23 noise on center of D. 错误! 未找到引用源。 shows the rest of the results.

TABLE.6 Percent of noise tolerance for Fig 7 when we have noise on center of samples.

Samples	A	B	C	D	E	F
The value of α	1.81	1.78	1.01	2.04	3.06	1.66
The percent of noise tolerance for center of fuzzy samples	45	44	13	23	31	15

We can do this experiment on right, left and length of a fuzzy sample instead of its center.

As it can be seen, we presented a fuzzy data by four crisp data, and that is possible to have noise on any of them.

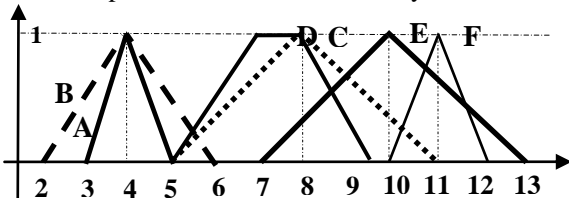


Figure 10. Representation of Fig 7 samples with noise on center, length, left and right of D

Now, suppose that we have noise on center, length, left and right of a sample simultaneously (see Fig 10). For example we received $D = [9 \pm \alpha; 0 \pm \beta; 2 \pm \lambda; 2 \pm \gamma]$ instead of $D = [9; 0; 2; 2]$. We can't find the maximum value of α, β, λ and γ that do not alter our clustering, because they are mutually depend on each other. But we can consider results in a special

state when we suppose that $\alpha = \beta = \lambda = \gamma$. In this way we can find out the minimum noise that can be tolerated by this algorithm. For example, if according to Fig 10, we received $D = [7.5; 1; 2.5; 2]$ instead of $D = [9; 0; 2; 2]$ we get the following dendrogram (Fig 11).

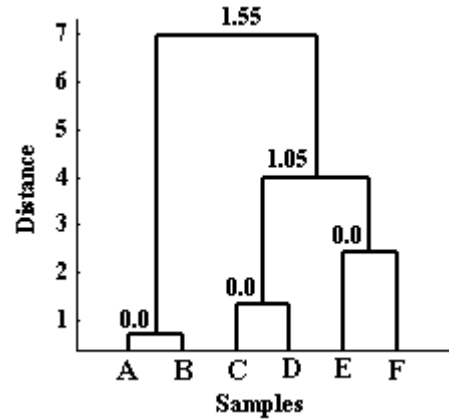


Figure 11. Fuzzy dendrogram after adding noise on center, length, left and right of D.

So, we have the same two previous clusters ($C1 = \{A, B\}$, $C2 = \{C, D, E, F\}$) and our clustering remained without any change. In Table 7, you can see the minimum noise on fuzzy data (used in this Example) that can be tolerated. These noises can be added to each part of a fuzzy sample (center, length, left and right) without change of clustering.

TABLE.7 The minimum noise that can be tolerated for each samples in Fig 7.

Samples	A	B	C	D	E	F
The value of (α)	1.66	1.06	0.65	1.34	1.40	0.91
The percent of noise tolerance for center of fuzzy samples	41	26.5	8.1	14.9	14	8.3

In Table 7 we suppose that a fuzzy data is shown as $[a_1 \pm \alpha; a_2 \pm \alpha; a_3 \pm \alpha; a_4 \pm \alpha]$, that α is the minimum noise which can be tolerated by FHCA. In this table, we just compute the percent of noise tolerance for center of fuzzy samples, because center of fuzzy samples are very important to cluster data. We can compute the percent of noise tolerance for other part of fuzzy samples (length, left and right) similarly.

According to Table 7, we can tolerate noise on inner samples of clusters more than boundary samples. Finally consider that the value of noise tolerated is different in any example and based on the distance of clusters from each other.

IV. EXPERIMENTAL RESULTS

We consider a data set G with 20 triangular fuzzy data [11]. Intuitively, the number of clusters that is suitable for data set G is two (Fig 12).

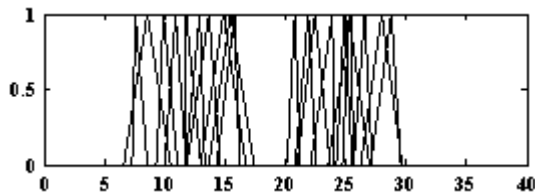


Figure 12. Data set G with 20 triangular fuzzy data.

If we run our algorithm on these fuzzy numbers, a dendrogram is formed and we can easily extract the clusters from that, by removing the highest link of dendrogram or via inconsistency coefficient measure by selecting threshold between 1.57 and 2.94 (Here we use the mean of them so $\text{threshold} = (1.57 + 2.94) / 2 = 2.2550$). In this state FHCA, FCM and AFCN give the same results.

Now, we repeat these algorithms on this data set after adding a noisy sample to it. For this reason we add a point (100; 0.71; 1.79) to the data set G. The added point is far away from the other TFNs so that it can be regarded as an outlier. In this case we reach the following dendrogram.

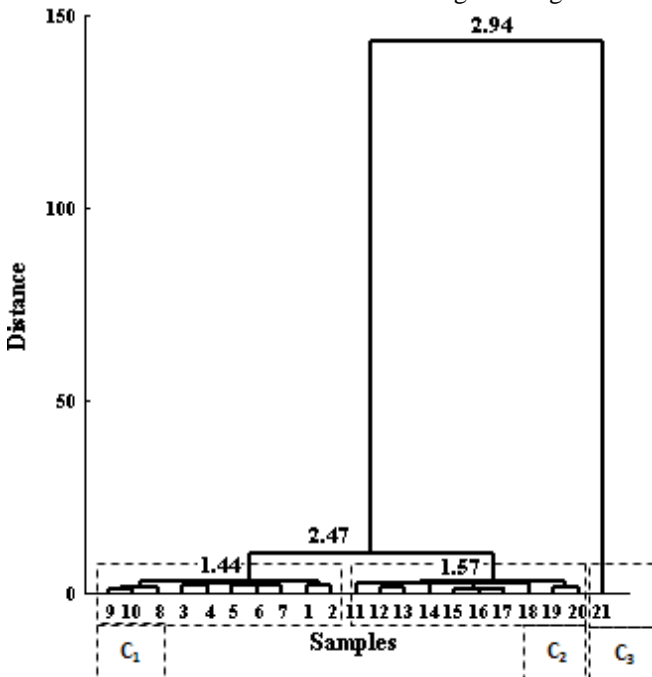


Figure 13. Fuzzy dendrogram for data set G with a noisy sample

Table 8 shows the clustering results of FHCA, FCM and AFCN after adding a noisy sample. So we have three clusters C_1 , C_2 and C_3 that are distinguished in Fig 13, because the inconsistency coefficient of two levels of dendrogram are upper than threshold (Note that threshold is 2.2550). Table 8 shows that FCM cannot tolerate noisy samples, AFCN is better than FCM but FHCA is the best because the noisy samples are classified in the separate clusters and the noisy samples cannot change the other clusters.

TABLE.8 Clustering results in data set g with a Noisy Sample Using FCM, AFCN, FHCA.

No	TFNs	FCM	AFCN	FHCA
1	(7.56, 0.27, 1.00)	C_1	C_1	C_1
2	(8.56, 1.95, 1.93)	C_1	C_1	C_1
3	(9.89, 0.56, 1.17)	C_1	C_1	C_1
4	(10.89, 0.89, 0.88)	C_1	C_1	C_1
5	(11.78, 0.12, 1.21)	C_1	C_1	C_1

6	(12.90, 1.19, 0.41)	C_1	C_1	C_1
7	(13.67, 1.82, 0.90)	C_1	C_1	C_1
8	(14.87, 1.90, 1.85)	C_1	C_1	C_1
9	(15.45, 1.79, 1.95)	C_1	C_1	C_1
10	(15.78, 1.47, 0.42)	C_1	C_1	C_1
11	(20.77, 0.63, 0.47)	C_1	C_2	C_2
12	(21.88, 1.08, 0.66)	C_1	C_2	C_2
13	(22.45, 1.48, 1.26)	C_1	C_2	C_2
14	(23.88, 1.79, 0.16)	C_1	C_2	C_2
15	(24.88, 0.66, 0.64)	C_1	C_2	C_2
16	(25.25, 0.52, 1.71)	C_1	C_2	C_2
17	(25.47, 1.95, 0.15)	C_1	C_2	C_2
18	(26.56, 0.92, 0.63)	C_1	C_2	C_2
19	(27.98, 1.74, 1.69)	C_1	C_2	C_2
20	(28.77, 1.71, 0.79)	C_1	C_2	C_2
21	(100.00, 0.71, 1.79)	C_2	?	C_3

V. CONCLUSION

In this paper we have described a new approach for clustering fuzzy data. So far several papers have been presented methods to cluster fuzzy data that are based on fuzzy c-means algorithm. Here we open a new point of view to cluster fuzzy data based on hierarchical clustering methods. In this method, computing the distance between fuzzy data and drawing fuzzy dendrogram, lead to forming clusters. The experimental results demonstrated the major advantage of the FHCA in comparison with similar methods that is its fault tolerance against noisy samples. Furthermore the fuzzy dendrogram can present us a general view from the relations between fuzzy data which help us to cluster them more accurately. Finally, FHCA is a very suitable clustering algorithm for fuzzy data because the nature of this algorithm is illustrative and clear so that we can guess estimate dispersion of clusters, with a glance at the dendrogram.

REFERENCES

- [1] Andrew R. Webb. "Statistical Pattern Recognition". Wiley, New York, 2002.
- [2] Richard O. Duda, Peter E. Hart and David G. Stork. "Pattern classification", 1997.
- [3] J. C. Bezdek. "Pattern recognition with fuzzy objective function algorithms". Plenum press, New York, 1981.
- [4] Miin-Shen Yang, Pei-Yuan Hwang, De-Hua Chen. "Fuzzy clustering algorithms for mixed feature variables". Fuzzy Sets and Systems 141, pp. 301-317, 2004.
- [5] Shyi-Ming Chen, Liang-Yu Chen. "A Fuzzy hierarchical clustering method for clustering documents based on dynamic cluster centers". Journal of the Chinese Institute of Engineers, Vol. 30, No. 1, pp. 169-172, 2007.
- [6] P. Grzegorzewski, "Metrics and orders in space of fuzzy numbers" Fuzzy sets and systems. 97, pp. 83-94, 1998.
- [7] Vicenc Torra. "Fuzzy c-means for fuzzy hierarchical clustering". IEEE transactions on Fuzzy Systems, Vol. 14, pp. 646-651, 2005.
- [8] Amir B. Geva. "Hierarchical Unsupervised Fuzzy Clustering". IEEE transactions on fuzzy systems, Vol. 7, No. 6, pp. 723-733, 1999.
- [9] R. O. Duda, P. E. Hart, "Pattern classification and scene Analysis". New York. Wiley, 1973.
- [10] L. Stefaninia, L. Sorinia, M. L. Guerra, "Parametric representation of fuzzy numbers and application to fuzzy calculus" Fuzzy Sets and Systems. 157, pp. 2423-2455, 2006.
- [11] Wen-Liang Hung, Miin-Shen Yang. "Fuzzy clustering on LR-type fuzzy numbers with an application in Taiwanese tea evaluation", Fuzzy Sets and Systems 150, pp. 561-577, 2005.
- [12] R.J. Hathaway, J.C. Bezdek, W. Pedrycz. "A parametric model for fusing heterogeneous fuzzy data". IEEE transactions on Fuzzy Systems, Vol. 4, No. 3, 1996.

- [13] Perng-Cherng Wang, Jin-Jang Leou. "New Fuzzy Hierarchical Clustering Algorithms". Journal of Information Science and Engineering, Vol. 9, No. 3, pp. 461-489, 1993.
- [14] M. Sato, Y. Sato. "Fuzzy clustering model for fuzzy data". IEEE transactions on Fuzzy Systems, Vol. 4, pp. 2123 – 2128, 1995.
- [15] Miin-Shen Yang, Chen-Hsiu Ko. "On a class of fuzzy c-numbers clustering procedures for fuzzy data". Fuzzy Sets and Systems 84, pp. 49–60, 1996.
- [16] <http://www.mathworks.com/access/helpdesk/help/toolbox/stats/index.html?access/helpdesk/help/toolbox/stats/inconsistent.html>
- [17] Bohdan S. Butkiewicz. "Robust Fuzzy Clustering with Fuzzy Data". Springer Berlin / Heidelberg , pp. 76-82, 2005.