Comparison Of K- Means And Fuzzy C- Means Algorithms

Ankita Singh
MCA Scholar

Dr Prerna Mahajan
Head of department
Institute of information technology and management

Abstract

Clustering is the process of grouping feature vectors into classes in the self-organizing mode. Choosing cluster centers is crucial to the clustering. In this paper we compared two fuzzy algorithms: fuzzy c-means algorithm and fuzzy k means algorithm. Fuzzy c-means algorithm uses the reciprocal of distances to decide the cluster centers. The representation reflects the distance of a feature vector from the cluster center but does not differentiate the distribution of the clusters [1, 10, and 11]. The fuzzy k means algorithm in data mining, is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean[10,11].

Keywords: *fuzzy c-means, fuzzy k means, classification, pattern recognition*

1. Introduction

Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each cluster share some common attributes [3].Cluster analysis attempts to isolate regions of similarity within a dataset and find the relationships between multiple clusters. The differences among members of a cluster, in terms of their absolute difference from the cluster's calculated centre or centroid, define a metric of compactness and homogeneity [3,10,11].

Fuzzy clustering plays an important role in solving problems in the areas of pattern recognition and fuzzy model identification. A variety of fuzzy clustering methods have been proposed and most of them are based upon distance criteria. One widely used algorithm is the fuzzy c-means (FCM) algorithm. It uses reciprocal distance to compute fuzzy weights and K-means algorithm that is used to solve well known clustering problems. In the following sections we discuss and compare K-means and fuzzy c-means algorithm [11]

2. K-Means Algorithm

K-means algorithm given by MacQueen,[9] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more [2]

//Assumptions:
//$A_k$is the centroid number
//Axand **Ay**are the x and y value of the centroid.
//$D_k$is the distance between centroid and point
//$X^k$and**Y^k**is the x and y value of the point
//$Bx^k$and**By^k**are the x and y value of new centroid

Input: $A_{k,}$Ax, Ay, $D_{k,}X^k$,Y^k ,Bx^k , By^k
Step1: Choose random centroids.
A_k(ax,ay) where k < 4

Step2:calculate distance between centroids and points
$D_k = | x^k - ax^k | + | y^k - ay^k |$
Step3: According minimal D_K assign the point to that cluster.
Step4: Calculate new centroids
$Bx^k = \frac{\sum_{k=1} x^k}{\sum k}$
$By^k = \frac{\sum_{k=1} y^k}{\sum k}$
Step5: Check if new centroids are equal to old centroids
$Ax_k == Bx_k$
$Ay_k == By_k$
Step6: If new centroids are equal to old centroids then program ends else goto step2
Output:K clusters

Fig 1.k-means algorithm

2.1 Results and Experiments:

We have considered a student data of 8 students and cluster the following eight points(with(x,y) representing age and marks or the student) S1(5,10) S2(6,8) S3(4,5) S4(7,10) S5(8,12) S6(10,9) S7(12,11) S8(4,6) . Initial cluster centers are S1 (5, 10), S4 (7, 10), S7 (12, 11). The distance function between two points a=(x1, y1) and b=(x2, y2) is defined as: p (a,b)= |x2-x1|+|y2-y1|.

The k means algorithm find three cluster centers after the second iteration in the considered example

Table 1:K-means computation on student data [step 1]

Iteration 1

			(5,10)	(7,10)	(12,11)	
Roll no	**Age**	**Marks**	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
S1	5	10				
S2	6	8				
S3	4	5				
S4	7	10				
S5	8	12				
S6	10	9				
S7	12	11				
S8	4	6				

The initial clusters centers-means, are (5, 10), (7, 10) and (12, 11) chosen randomly. Next we will calculate the distances from the first point (5, 10) to each of the three centroids, by using the distance function:

Point	**mean 1**
x1,y1	x2,y2
(5,10)	**(5,10)**

P(a,b)=|x2-x1|+|y2-y1|

P(point mean)= |x2-x1|+|y2-y1| (eq 1)

=|5-5|+|10-10|

= 0+0=0

Point	**mean 2**
x1,y1	x2,y2
(5,10)	**(7,10)**

P(a,b)=|x2-x1|+|y2-y1|

P(point mean)= |x2-x1|+|y2-y1|

= |7-5|+|10-10|

= 2+0=2

Point	mean 3
x1,y1	x2,y2
(5,10)	**(12,11)**

P(a,b)=|x2-x1|+|y2-y1|

P(point mean)= |x2-x1|+|y2-y1|

= |12-5|+|11-10|

= 7+1=8

So, we fill in these values in the table:

Table 2: K-means computation on student data[step 2]

			(5,10)	(7,10)	(12,11)	
Roll no	**Age**	**Marks**	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
S1	5	10	0	2	8	1
S2	6	8				
S3	4	5				
S4	7	10				
S5	8	12				
S6	10	9				
S7	12	11				
S8	4	6				

Now, we go to the second point (6, 8) and we will calculate the distance to each of the three means, by using distance function given in (eq 1) and analogically we fill all the values in the table:

Table 3:K-means computation on student data[step3]

			(5,10)	(7,10)	(12,11)	
Roll no	**Age**	**Marks**	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
S1	5	10	0	2	8	1
S2	6	8	3	3	9	1
S3	4	5				
S4	7	10				
S5	8	12				
S6	10	9				
S7	12	11				
S8	4	6				

Now, we go to the third point(4,5) and we will calculate the distance to each of the three means, by using distance function given in (eq 1) Analogically, we fill in the rest of the table, and place each point in one of the clusters:

Table 4: K-means computation on student data[step 4]

			(5,10)	(7,10)	(12,11)	
Roll no	**Age**	**Marks**	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
S1	5	10	0	2	8	1
S2	6	8	3	3	9	1
S3	4	5	6	8	14	1
S4	7	10	2	0	6	2
S5	8	12	5	3	5	2

S6	10	9	6	4	4	3
S7	12	11	8	6	0	3
S8	4	6	5	7	13	1

Next we need to re-compute the new clusters centers (means). We do so, by taking the mean of all points in each cluster.

For Cluster 1, we have (5+6+4+4)/4, (10+8+5+6)/4 = (4.75, 7.25)
For Cluster 2, we have (7+8)/2, (10+12)/2 = (7.5, 11)
For Cluster 3, we have (10+12)/2, (9+11)/2 = (11, 10)
New Clusters: 1:{S1,S2,S3,S8}, 2:{S4,S5} , 3:{S6,S7}

Centers of the new clusters:

C1= (5+6+4+4)/4, (10+8+5+6)/4 = (4.75, 7.25)
C2= (7+8)/2, (10+12)/2 = (7.5, 11)
C3= (10+12)/2, (9+11)/2 = (11, 10)

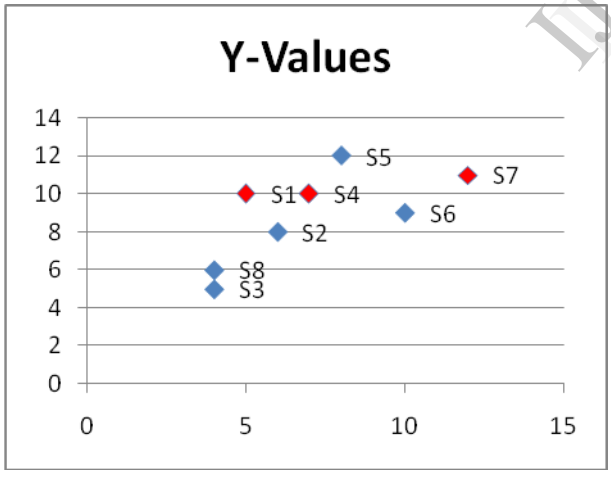

Fig 2: initial clusters

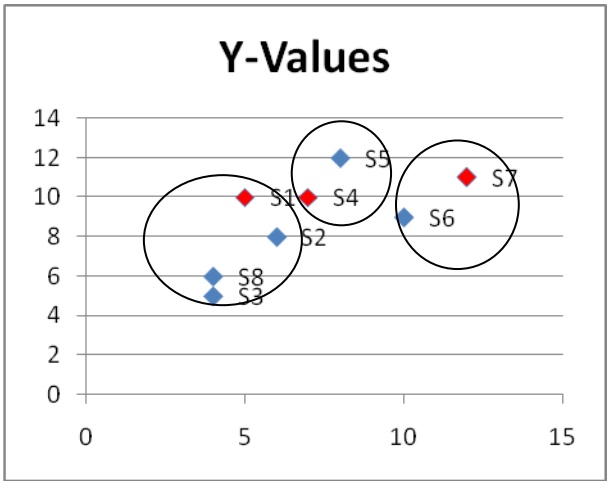

Fig 3: Merging clusters

Red dots are the centroids.

Blue dots are the points.

3.Fuzzy C-Means Algorithm

The most well-known fuzzy clustering algorithm is fuzzy c-means, a modification by Bezdek of an original crisp clustering methodology. Bezdek introduced the idea of a fuzzification parameter **(m)** in the range **[1, n]**, which determines the degree of fuzziness in the clusters. When m = 1 the effect is a crisp clustering of points. when m > 1 is the degree of fuzziness among points in the decision space increases[3] where:

//Assumptions
//$\mathbf{x_i}$: is the ith data point
//**Cj**:is the centroid of a fuzzy cluster (j = 1, 2, . . ., p). This value is repeatedly calculated by the algorithm
//$\mathbf{d_{ij}}$:isthe distance of the ith data point from the jth cluster center with using the Euclidean distance.
//**P:** is the number of fuzzy clusters specified as part of the algorithm.
//**M**:isafuzzification parameter
//$\mu_j(x_i)$: is a fuzzy membership qualification indicating the membership of sample xi to the jthcluster[3]

Input: x_i,**Cj**,d_{ij},p,M,$\mu_j(x_i)$
Step 1: Randomly initializing the cluster center
Step2: Creating distance matrix from a point xi to each of the cluster centers to with taking the Euclidean distance between the point and the cluster center.

$$d_{ji} = \sqrt{\sum (x_j - c_j)^2}$$

Step3: Creating membership matrix takes the fractional distance from the point to the cluster center and makes this a fuzzy measurement by raising the fraction to the inverse fuzzification parameter.
This is divided by the sum of all fractional distances, thereby ensuring that the sum of all memberships is 1.

$$\mu_j(x_i) = \frac{(\frac{1}{d_{ji}})^{\frac{1}{m-1}}}{\sum_{k=1}^{p}(\frac{1}{d_{ji}})^{\frac{1}{m-1}}}$$

Step4: Creating membership matrix
Fuzzy c-means imposes a direct constraint on the fuzzy membership function associated with each point, as follows. The total membership for a point in sample or decision space must add to 1

$$\sum_{j=1}^{p} \mu_j(x_i) = 1$$

Step5: Generating new centroid for each cluster

$$C_j = \frac{\sum_i [\mu_j(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m}$$

Step6: Generating new centroid for each cluster with iteration all this step optimize cluster centers will generate.
Step7: Weight Acceleration Cluster Assignments
Output: weighted cluster assignments

Fig 4: fuzzy c means algorithm

3.1 Results and experiments:

We have assumed a automobile property information database and applied fuzzy c means algorithm on it, whereclustering is done in two attributes ACCEL (acceleration) and WGT (weight) where m= 1.25and P= 2[3].

Table 5: Fuzzy c-means computation on automobile data[step 1]

	ACCEL	WGT
Object1	12.0	3504.0
Object2	11.5	3693.0
Object3	11.0	3436.0
Object4	12.0	3433.0
Object5	10.5	3449.0
Object6	10.0	4341.0
Object7	9.0	4354.0
Object8	8.5	4312.0
Object9	10.0	4425.0
Object10	8.5	3850.0
Object11	10.0	3563.0
Object12	8.0	3609.0
Object13	9.5	3761.0
Object14	10.0	3086.0
Object15	15.0	2372.0
Object16	15.5	2833.0
Object17	15.5	2774.0
Object18	16.0	2587.0
Number of cluster:	2	
Fuzzification parameter	1.25	

Step 1: Randomly initializing the cluster center

Cluster Center Initialization		
	ACCEL	WGT
Centroid 1	6.00	1379.00
Centroid 2	5.00	817.00

Step2: Creating distance matrix from a point xi to each of the cluster centers to with taking the Euclidean distance between the point and the cluster center.

$$d_{ji} = \sqrt{\Sigma (x_j - c_j)^2} \qquad d_{11} = \sqrt{(12-6)^2 + (3504-1379)^2}$$

Table 6: Fuzzy c-means computation on automobile data[step 2]

	Cluster1	Cluster2
Object1	2125.0	2687.0
Object2	2314.0	2876.0
Object3	2057.0	2619.0
Object4	2054.0	2616.0
Object5	2070.0	2632.0
Object6	2962.0	3524.0
Object7	2975.0	3537.0
Object8	2933.0	3495.0
Object9	3046.0	3608.0
Object10	2471.0	3033.0
Object11	2184.0	2746.0
Object12	2230.0	2792.0
Object13	2382.0	2269.0

Object14	1707.0	1555.0
Object15	993.0	815.0
Object16	1454.0	2016.0
Object17	1395.0	1957.0
Object18	1208.0	1770.0

Step3: Creating membership matrix takes the fractional distance from the point to the cluster center and makes this a fuzzy measurement by raising the fraction to the inverse fuzzification parameter.

This is divided by the sum of all fractional distances, thereby ensuring that the sum of all memberships is 1.

$$\mu_j(x_i) = \frac{(\frac{1}{d_{ji}})^{\frac{1}{m-1}}}{\sum_{k=1}^{p}(\frac{1}{d_{ji}})^{\frac{1}{m-1}}}$$

Step4: Creating membership matrix
Fuzzy c-means imposes a direct constraint on the fuzzy membership function associated with each point, as follows. The total membership for a point in sample or decision space must add to 1.

$$\sum_{j=1}^{p} \mu_j(x_i) = 1$$

Table 7: Fuzzy c-means computation on automobile data[step 3]

	Cluster1	Cluster2	Sum of DFM
Object1	0.72	0.28	1.00
Object2	0.70	0.3	1.00
Object3	0.72	0.28	1.00
Object4	0.72	0.28	1.00
Object5	0.72	0.28	1.00
Object6	0.67	0.33	1.00
Object7	0.67	0.33	1.00
Object8	0.67	0.33	1.00
Object9	0.66	0.34	1.00
Object10	0.69	0.31	1.00
Object11	0.71	0.29	1.00
Object12	0.71	0.29	1.00
Object13	0.45	0.55	1.00
Object14	0.41	0.59	1.00
Object15	0.31	0.69	1.00
Object16	0.79	0.21	1.00
Object17	0.79	0.21	1.00
Object18	0.82	0.18	1.00

Step5: Generating new centroid for each cluster

$$C_j = \frac{\sum_i [\mu_j(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m}$$

Cluster Center after cycle 1		
	ACCEL	WGT
Centroid 1	10.18	3767.12
Centroid 2	11.96	3690.81

Step6: Generating new centroid for each cluster with iteration all this step optimize cluster centers will generate.

cycle 1		
	AC	WG
Cent	10.1	376
Cent	11.9	369

cycle 2		
	AC	WG
Cent	15.6	2906
Cent	14.6	3693

cycle 3		
	AC	WG
Cent	16.3	458.
Cent	14.1	3980

cycle 4		
	AC	WG
Cent	16.2	2426
Cent	14.3	3944

Final Cluster		
	AC	WG
Cent	16.2	2426
Cent	14.3	3944

Step7: Weight Acceleration Cluster Assignments

Table 8: Fuzzy c-means computation on automobile data[step 4]

Object	Cluster1	Cluster2
Object1	0.002	0.998
Object2	0.002	0.998
Object3	0.009	0.991
Object4	0.009	0.991
Object5	0.007	0.993.
Object6	0.007	0.993
Object7	0.000	1.000
Object8	0.000	1.000
Object9	0.000	1.000
Object10	0.000	1.000
Object11	0.000	1.000
Object12	0.000	1.000
Object13	0.000	1.000
Object14	0.000	1.000
Object15	1.000	0.000
Object16	1.000	0.000
Object17	1.000	0.000
Object18	1.000	0.000

4. Comparison of K means and Fuzzy C-means Algorithms.

Table 9: Comparative analysis of K-means and Fuzzy c-means algorithm

Basis	K means	Fuzzy C means	Reason
Efficiency	Fairer	Slower	K-Means just needs to do a distance calculation, whereas fuzzy c means needs to do a full inverse-distance weighting[4]
Objective function	$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)}\right\|$ [2]	[2] $J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\left\|$	**The objective functions are virtually identical,** the only difference being the introduction of a vector which expresses the percentage of belonging of a given point to each of the clusters

			[4]
Performance	Traditional and Limited use	Can be used in variety of clusters and can handle uncertainty.	FCM may converge faster than hard K-Means, somewhat offsetting the bigger computational requirement of FCM[4]
Applications	In image retrieval algorithms[5]	-Segmentation of magnetic resonance imaging (MRI)[6] - Analysis of network traffic[7] - Fourier-transform infrared spectroscopy (FTIR)[8]	

5. Conclusion

In this paper we have evaluated k-means & fuzzy c – means algorithms on various datasets. *k*- means clustering is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into k clusters. Whereas .In fuzzy c-means, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. In this paper we have concluded that fuzzy c-means algorithm is slower than k means algorithm in efficiency but gives better results in cases where data is incomplete or uncertain and has a wider applicability.

6.References

[1] Liyan Zhang(2001), Comparison of Fuzzy c-means Algorithm and New Fuzzy Clustering and Fuzzy Merging Algorithm Computer Science Department University of Nevada Reno Reno, May, 2001,NV 89557 , lzhang@cs.unr.edu

[2]http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[3]Maedeh Zirak Javanmard, 2010 ,Fuzzy c-means clustering And Its application in case of forest fires intelligent system

[4]http://stackoverflow.com/questions/2345903/whats-is-the-difference-between-k-means-and-fuzzy-c-means-objective-function

[5] HangZhou, , 28 November 2007, College of Computer science and Information Engineering Zhejiang Gongshang University, China,http://asp.eurasipjournals.com/content/2008/1/468390

[6] Pham DL, Xu CY, Prince JL, 2000, A survey of current methods in medical image segmentation.Ann. Rev. Biomed.Eng.,Johns Hopkins University.

[7] Lampinen, Timo; Koivisto, Hannu ,Honkanen, Tapan Institute of Automation and Control Tampere University of Technology, FINLAND

[8] Xiao Ying Wang, Jon Garibaldi, Turhan Ozen Department of Computer Science and Information Technology The University of Nottingham, United Kingdom

[9] J. B. Macqueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297

[10]A.K. JAIN, Michigan State University, September 1999,Data Clustering: A Review**,** ACM Computing Surveys, Vol. 31, No. 3, http://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf.

[11]M.-S. YANG, Department of Mathematics, Chung Yuan Christian University, Chungli, Taiwan 32023, October 1993,A Survey of Fuzzy Clustering, Vol. 18, No. 11, http://www2.math.cycu.edu.tw/TEACHER/MSYANG/yang-pdf/yang2-survey.pdf.