FUZZY C - MEANS CLUSTERING IN MATLAB

Makhalova Elena

Abstract

Paper is a survey of fuzzy logic theory applied in cluster analysis. Fuzzy logic becomes more and more important in modern science. It is widely used: from data analysis and forecasting to complex control systems. In this article we consider clustering based on fuzzy logic, named Fuzzy Clustering. Clustering involves the task of dividing data points into homogeneous classes or clusters so that items in the same class are as similar as possible and items in different classes are as dissimilar as possible. (Řezanková, 2013). In hard clustering, data is divided into crisp clusters, where each data point belongs to exactly one cluster. In fuzzy clustering, the data points can belong to more than one cluster, and associated with each of the points are membership grades which indicate the degree to which the data points belong to the different clusters. There are many methods of Fuzzy Clustering nowadays. In our work we review the Fuzzy c - means clustering method in MATLAB.

Key words: Fuzzy clustering; Fuzzy c – means; Hard c-means; MATLAB

JEL code: C18, C38, C69

Introduction

Data clustering is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed. In contradistinction to hard clustering, in fuzzy clustering each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster (Löster, 2012). Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. This method was developed by Dunn in 1973 and improved by Bezdek in 1981.

Actually, there are many programmes using Fuzzy C-Means Clustering, for instance: C++, MATLAB, R-ko. In this papir we will review Fuzzy C-Means Clustering in MATLAB.

1 Comparison Fuzzy c – means clustering algorithm with hard C – means clustering algorithm

Let‘s start by considering, what is it fuzzy c-means clustering. We can see some differences in comparison with c-means clustering (hard clustering). Moreover, one should note here: c – means clustering algorithm is a starting point for the fuzzy extensions (Löster, 2012).

These algorythms are based on objective functions J, which are mathematical criterial that quantify the goodness of cluster models that comprise prototypes and data partition. Objective functions serve as cost functions that have to be minimized to obtain optimal cluster solutions (Řezanková, 2011). Thus, for each of the following cluster models the respective objective function expresses desired properties of what should be regarded as ‘‘best’’ results of the cluster algorithm. Steps of the algorithm follow from the optimization scheme that they apply to approach the optimum of J. Thus, in our paper of the hard and fuzzy c-means we discuss their respective objective functions first. In their basic forms the hard and fuzzy algorithms look for a predefined number of c clusters in a given data-set, where each of the clusters is represented by its center vector (Zheru, 1996). However, hard and fuzzy differ in the way they assign data to clusters, i.e., what type of data partitions they form.

The main advantage of fuzzy c – means clustering, it allows gradual memberships of data points to clusters measured as degrees in [0,1]. This gives the flexibility to express that data points can belong to more than one cluster (Höppper, 2000).

Fig. 1a: Hard c-means clustering **Fig. 1b: Fuzzy c – means clustering**

$$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

The matrix U (Fig.1a, Fig.1b) whose factors are the ones taken from the membership functions presents the data for hard and fuzzy clustering. In our case we have two clusters (the number of rows) and we can see the following: in case a. (hard clustering) each data point x_j in the given data-set X=$\{x_1....x_n\}$assigned to exactly one cluster. In case b. a data point can

belongs to more than one cluster (fuzzy clustering) with degree of membership between data and centers of clusters. Figure 2a and 2b illustrate this idea. A hard clustering can be obtained from a fuzzy partition by thresholding the membership value (Gan, 2007).

Fig. 2a: Hard clustering.

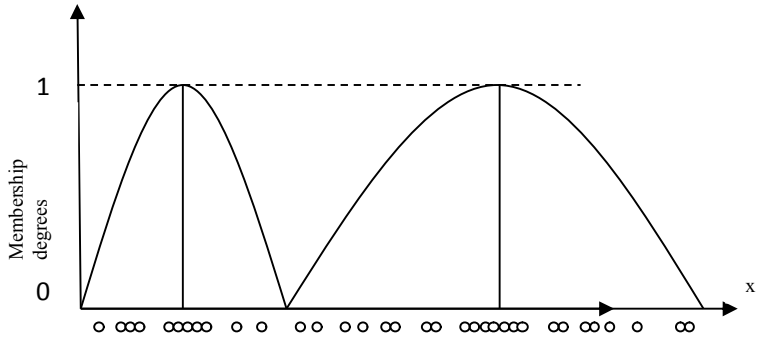
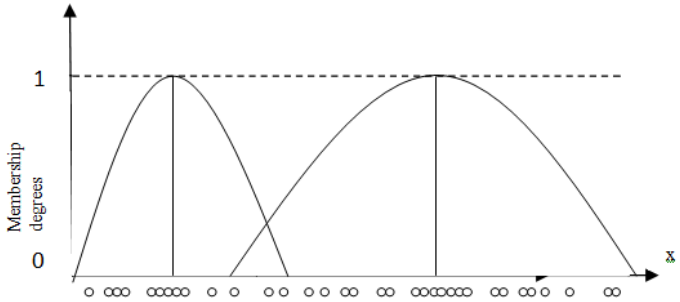


Fig. 2b: Fuzzy clustering.



Membership degrees can also express how ambiguously or definitely a data point should belong to a cluster (Löster, 2012). So, this partitioning is carried out through an iterative optimization of the objective function, with the update of membership (u_{ij}) and the cluster centers (c_j). This function show below:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , 1 \leq m \leq \infty \qquad (1)$$

Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j (De Oliveria, 2007), by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (2)$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m * x_i}{\sum_{i=1}^{N} u_{ij}^m} \quad (3)$$

This iteration will stop when $max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas *k* are the iteration steps. This procedure converges to a local minimum of J_{m} (Meloun, 2006).

The algorithm is composed of the following steps:

1. Initialization

Selected the following parameters:
• the required number of clusters *N*,
2 <N <k;
• measure distances as Euclidean distance;
• a fixed parameter q (usually 1.5);
• initial (at zero iteration) matrix $U^{(0)} = (c_i)^{(0)}$ object ownership x_{i} with the given initial cluster centers c_j (Everitt, 2011).

2. Calculate the centers vectors *C*(*k*)=[c_j] with *U*(*k*)

In the t-th iteration step in the known matrix is computed in accordance with the above solution of differential equations (Everitt, 2011).

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m * x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

3. Modified membership measure u_{ij}

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

4. If $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ **then STOP; otherwise return to step 2.**

| | | - matrix norm (for example, Euclidean norm);

ε - predetermined level of accuracy.

2 Fuzzy c – means clustering in MATLAB

In this abstrakt we consider the simple case of a Fuzzy c – means clustering in MATLAB. For clarity, we restrict ourselves to the simplest form of cluster prototypes. The data set has n=45 points in an s=3 dimensional space. There are two ways solve this in MATLAB: using the command line or graphical user interface. Consider the first of these methods.

To find the cluster centers in MATLAB we can with the help fcm function (built-in function), which is given below**.**

Description of function:

[center, U, obj_fcm] = fcm(data,cluster_ n)

The arguments of this function are:

1) *data* - lots of data to be clustering, each line describes a point in a multidimensional feature space;

2) *cluster_n* - number of clusters (more than one).

The function returns the following parameters:

1) *center* - the matrix of cluster centers, where each row contains the coordinates of the center of an individual cluster;

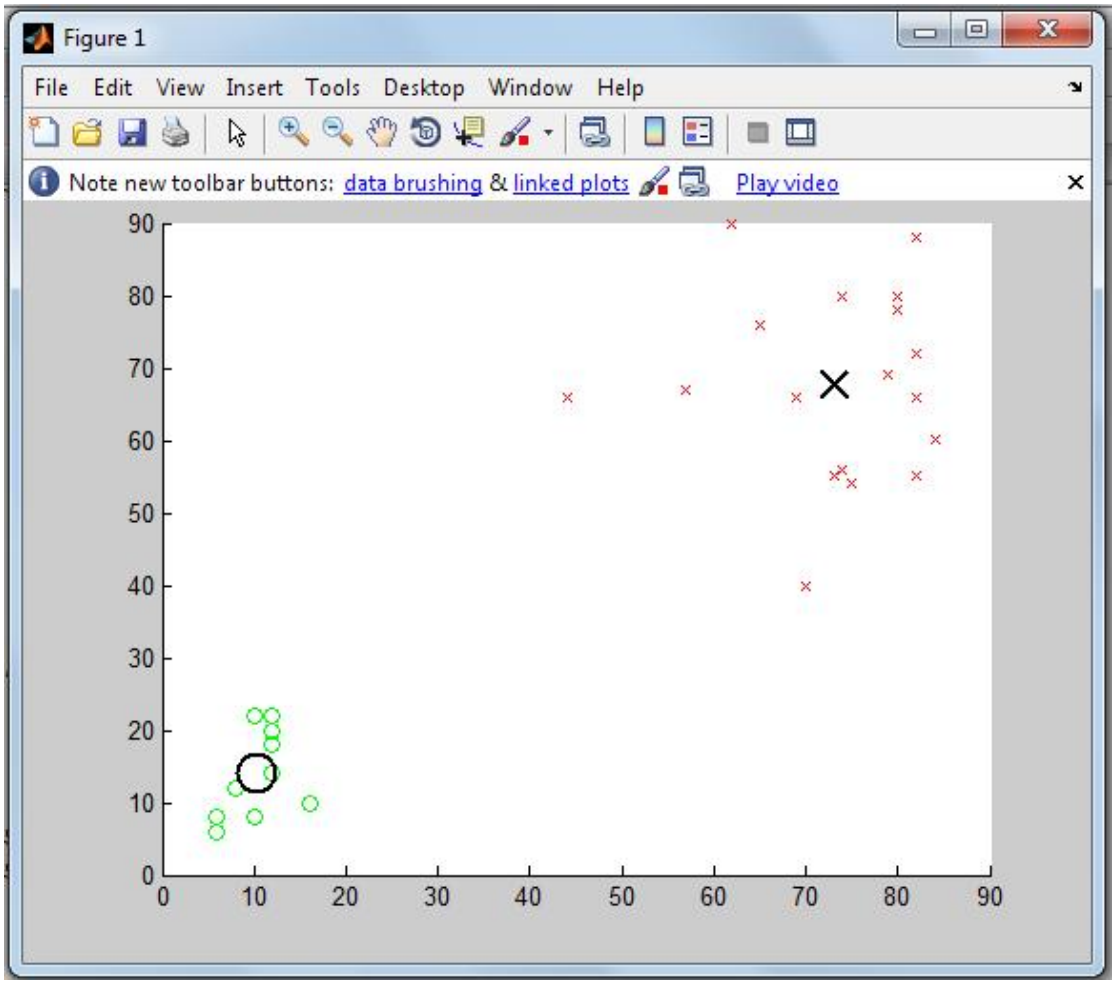
2) *U* - resulting matrix;

3) *obj_fcn* - the objective function value at each iteration.

So, for this example we should write (results are shown in Figure 3):

```
load ccc.dat
[center, U, obj_fcn] = fcm(ccc, 2);
maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :) == maxU);
line(ccc(index1, 1), ccc(index1, 2), 'linestyle',...
'none','marker', 'o','color','g');
line(ccc(index2,1),ccc(index2,2),'linestyle',...
'none','marker', 'x','color','r');
hold on
plot(center(1,1),center(1,2),'ko','markersize',15,'LineWidth',2)
plot(center(2,1),center(2,2),'kx','markersize',15,'LineWidth',2)
```

Fig. 3: Results of the C - means fuzzy clustering in MATLAB



Source: Generated data

Tab. 1: Results of iterations

Number of iteration count	Value of the objective function
Iteration count = 1	obj. fcn = 169197.014483
Iteration count = 2	obj. fcn = 117766.228513
Iteration count = 3	obj. fcn = 21602.841537
Iteration count = 4	obj. fcn = 7893.034365
Iteration count = 5	obj. fcn = 7887.892094
Iteration count = 6	obj. fcn = 7887.890663
Iteration count = 7	obj. fcn = 7887.890662

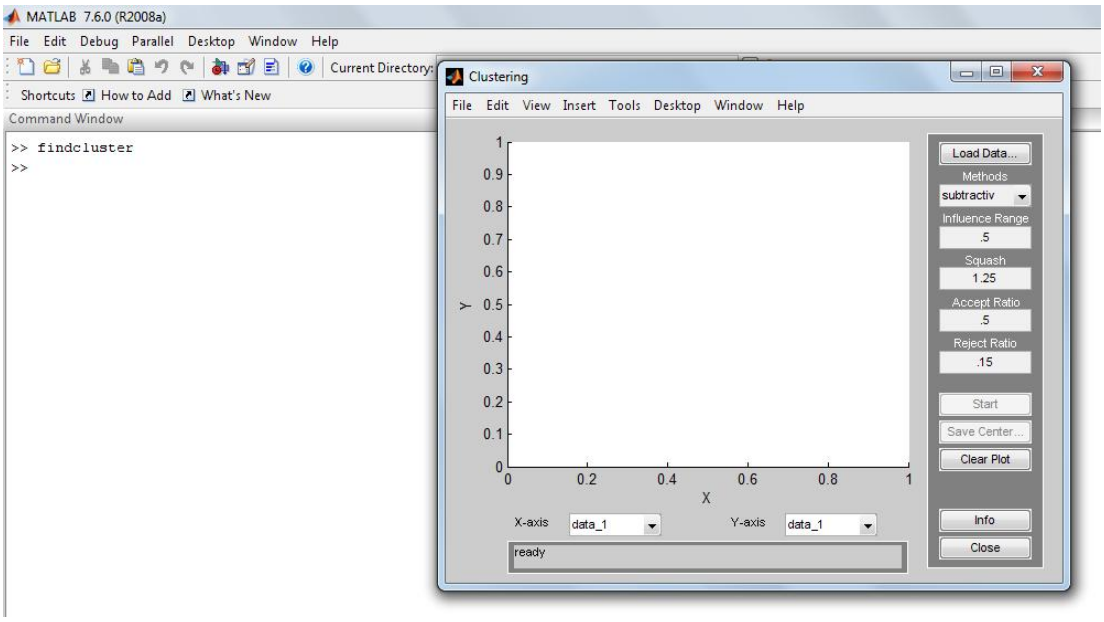
Source: Generated data

The second way to solve the problems of clustering in MATLAB based on the using the Clustering GUI Tool.

The Clustering GUI Tool shown in the next figure (Figure 4) lets you perform the following tasks:

1. Load and plot the data.
2. Start the clustering.
3. Save the cluster center.

Fig. 4: The Clustering GUI Tool



Source: Generated data

Let's consider each of tasks.

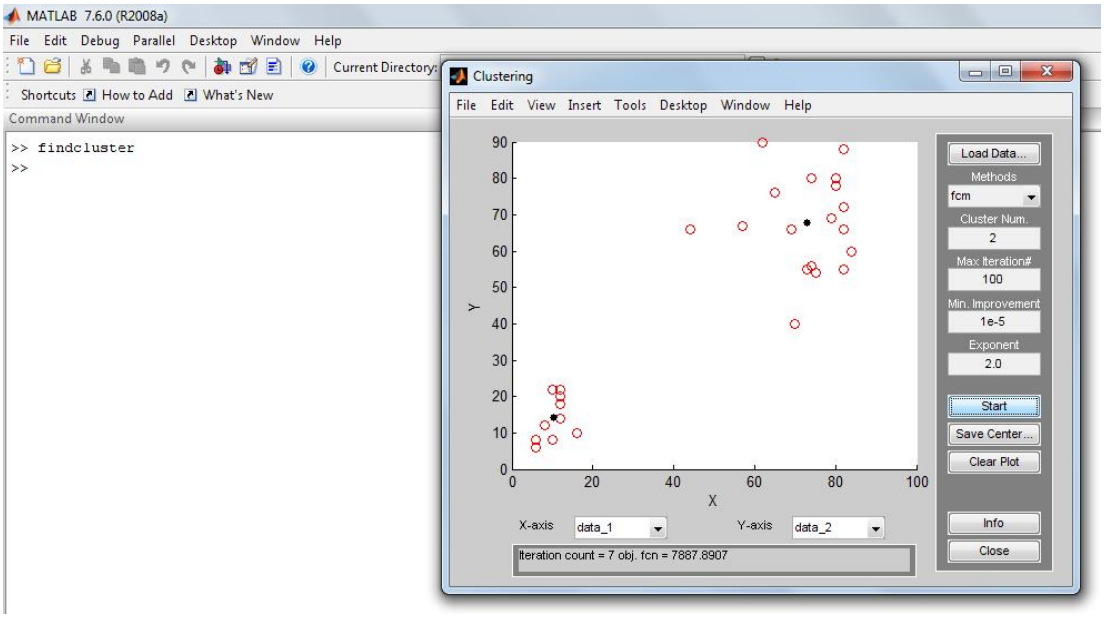
1. Loading and Plotting the Data (shown in Fig.5)

To load a data set in the GUI, perform either of the following actions:

a. Click Load Data, and select the file containing the data.

b. Open the GUI with a data set directly by invoking find cluster with the data set as the argument, in the MATLAB Command Window. The data set must have the extension *.dat* ('Name.dat').

Fig. 5: Loading and Plotting the Data



Source: Generated data

2. The Clustering GUI Tool works on multidimensional data sets, but displays only two of those dimensions on the plot. To select other dimensions in the data set for plotting, you can use the drop-down lists under X-axis and Y-axis.
3. Starting the Clustering

To start clustering the data: choose the clustering function fcm (fuzzy C-Means clustering) from the drop-down menu under Methods.

Set options for the selected method using the Influence Range, Squash, Aspect Ratio, and Reject Ratio fields. Begin clustering by clicking Start.

4. Save the cluster center.

After clustering gets completed, the cluster centers appear in black as shown in the next figure.

Conclusion

In the first phase of this research it was revealed: the results of the Fuzzy c – means clustering are more accurate in comparison with the results of the Hard c – means clustering, because of Fuzzy algorithm allows gradual memberships of data points to clusters measured as degrees in [0,1]. This gives the flexibility to express that data points can belong to more than one cluster. Indisputable advantage of this program is the ability to select the way of data processing: writing code or using the built-in functions.

In the second phase of our research we plan to process real data from Spain, German and Russian banks in the same way.

Acknowledgment

This article was created with the help of the Internal Grant Agency of University of Economics in Prague No. 6/2013 under the title „Evaluation of results of cluster analysis in Economic problems."

References

[1] De Oliveira, J. V., & Pedrycz, W. (2007). Advances in fuzzy clustering and its applications. (1 ed., pp. 4-69). London: Wiley.

[2] Everitt, S., Landau, S., Leese, M. (2011). Cluster Analysis. (5 ed., pp. 76-80). London: Wiley.

[3] Gan, G., Ma, C., Wu, J. (2007). Data Clustering: Theory, Algorithms, and Applications. (1 ed., pp. 18-25). London: Wiley.

[4] Höppper, F., Klawonn, F., Krause, R., & Runkler, T. (2000). Fuzzy cluster analisys. (2 ed., pp. 5-17). New York: Wiley.

[5] Meloun, M., Militký, J. (2006). Kompendium statistického zpracování dat. (2 ed., pp. 279-295). Prague: Academy.

[6] Zheru, C., Hong, Y., & Tuan, P. (1996). Fuzzy algorithms : with applications to image processing and pattern recognition. (Vol. 10, pp. 57-58, 86-89). Singapore: World Scientific Publishing.

[7] Loster, T., & Langhamrova, J. (2012). *Disparities between regions of the czech republic for non-business aspects of labour market*. In Loster Tomas, Pavelka Tomas (Eds.), *6th International Days of Statistics and Economics* (pp. 689-702). ISBN 978-80-86175-86-7

[8] Löster, T. (2012). Hodnocení výsledků fuzzy shlukování. International collection of scientific work on the occasion of 60th anniversary of university education at faculty of Business Economy with seat in Košice of University of Economics in Bratislava, 1–14. ISBN 978-80-86175-80-5.

[9] Löster, T. (2012). Kritéria pro hodnocení výsledků shlukování se známým zařazením do skupin založená na konfuzní matici. Retrieved from http://ssds.sk/casopis/archiv/2012/fss0712.pdf

[10] Řezanková, H., Löster, T., & Húsek, D. (2011). Evaluation of categorical data clustering. Advances in Intelligent Web Mastering, 3, 173–182. ISBN 978-3-642-18028-6. ISSN 1867-5662.

[11] Řezanková, H., Löster, T. (2013). Shluková analýza domácnosti charakterizovaných kategoriálními ukazateli. E+M. Ekonomie a Management, 16(3), 139-147. ISSN: 1212-3609

Contact

Makhalova Elena
University of Economics
Czech Republic, Prague, W. Churchill Sq. 4, 130 67
makhalova007@gmail.com