Course: “Introduction to Soft Computing Theory”

Laboratory Work №5

Clustering using the fuzzy center algorithm

Objective: master the method of finding cluster centers.
Task: find the centers of clusters using the fuzzy center algorithm with the use of the Clustering program included in the Fuzzy Logic Toolbox of the MATLAB mathematical environment.

Basic theoretical information:

Clustering (clusterization) - a combination of objects into groups (clusters) based on the similarity of features for objects of the same group and differences between groups.

Cluster - a bunch, group of elements characterized by some common property.

Clustering helps to present heterogeneous data in a visual form for further, more convenient, use of this data.

Clustering can be used for solving the following tasks:

- image processing;
- classification;
- thematic analysis of document collections;
- construction of a representative sample.

Cluster analysis is intended for dividing a set of objects into a given or unknown number of classes based on some mathematical criterion of classification quality. The clustering quality criterion to some extent reflects the following informal requirements:

- objects must be closely related to each other within groups;
- objects from different groups should be far from each other.

These requirements express the standard concept of density for partition classes. The focal point in cluster analysis is the choice of the measure of proximity of objects (metrics), which decisively determines the final version of dividing objects into groups for a given partitioning algorithm. In each specific task, this choice is made in its own way, taking into account the main objectives of the study, the physical and statistical nature of the information used, etc.

In cluster analysis, the distance between entire groups of objects is also important. Here are some examples of the most common measures of distance and proximity, characterizing the relative position of individual groups of objects.

Let it be:

ω_i - *i*-th group (class, cluster) of objects;

Ni - the number of objects forming a group ω_i, each object is a point in *n*-dimensional space;

μ*i* - the arithmetic mean of objects included in ω_i (μ*i* is the "center of gravity" of the i-th group);

p - the number of groups (clusters);

$\mathbf{x}_i$ - *i*-th cluster object;

q (ω_l, ω_m) - distance between groups ω_l and ω_m;

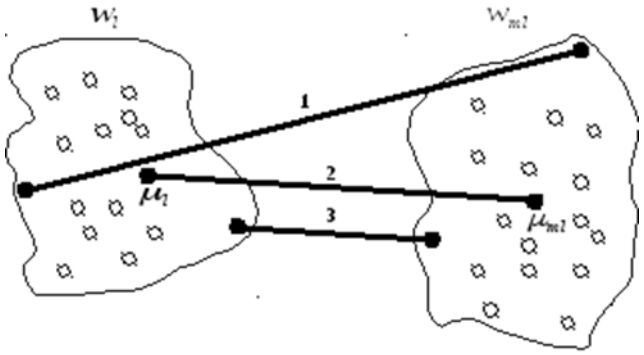

Fig. 1 - Different methods for determining the distance between the clusters ω_l and ω_m: 1 - by the most distant objects, 2 - by the centers of gravity, 3 - by the nearest objects

The nearest neighbor distance - the distance between the nearest cluster objects (fig. 1):

$$q_{\min}(\omega_l, \omega_m) = \min d(\mathbf{x}_i, \mathbf{x}_j), i = \overline{1, p}, j = \overline{1, p}$$

The farthest neighbor distance - the distance between the most distant cluster objects (fig. 1):

$$q_{\min}(\omega_l, \omega_m) = \max d(\mathbf{x}_i, \mathbf{x}_j), i = \overline{1, p}, j = \overline{1, p}$$

The distance of the centers of gravity is equal to the distance between the center points of the clusters (fig. 1):

$$q(\omega_l, \omega_m) = d(\mu_l, \mu_m).$$

The choice of one or another measure of the distance between clusters influences the type of geometric groupings of objects allocated in the feature space by the cluster analysis algorithms. Thus, algorithms based on the distance of the nearest neighbor work well in the case of groupings that have a complex, in particular, chain structure. The farthest neighbor distance is used when the desired groupings form globular clouds in the feature space. And the intermediate place is taken by algorithms that use the distances of the centers of gravity and average connection, which work best in the case of ellipsoidal groupings.

There is a variety of cluster analysis algorithms. For example, there are algorithms that implement a full enumeration of combinations of objects or perform random partitions of a set of objects. At the same time, most such algorithms consist of two stages:

- at the first stage, an initial (possibly artificial or even arbitrary) partition of the set of objects into classes is set and a certain mathematical criterion for the quality of automatic classification is determined;
- at the second stage, objects are transferred from class to class until the criterion value stops improving.

In such a way, clustering algorithms are based on the similarity of objects and place close objects into one cluster.

The identification of centers is a significant stage in the preliminary data processing, as it allows one to compare the membership functions of variables with these centers when designing a fuzzy inference system.

The Clustering program of the Fuzzy Logic Toolbox of the MATLAB mathematical environment identifies the centers of the clusters, i.e. points in a multidimensional data space, around which experimental data are grouped (accumulated).

The Clustering program uses two algorithms for identifying cluster centers: Subtractive clustering and Fuzzy c-means (fuzzy center algorithm).

The first algorithm is based on the proposition that each experimental point can be the center of a cluster. In this case, for each point, the likelihood measure of this assumption ("point potential") is calculated based on the density of points in a

given neighborhood of the considered point. Further calculations are done iteratively:

1. The point with the highest potential is declared as the center of the first cluster.

2. All other points are removed from the marked neighborhood of this point.

3. The center of the next cluster is declared from the remaining points and so on until all points have been considered (excluded or declared as centers).

Fuzzy's c-means algorithm is more accurate. It requires to specify such options as the number of clusters and the number of iterations. Let's consider it in detail.

Clustering based on the fuzzy centers algorithm. Data clustering is performed based on fuzzy c-means algorithm. This clustering algorithm was proposed by James Bezdek in 1981.

There are many clustering methods that can be classified into non-fuzzy and fuzzy. Non-fuzzy clustering methods divide the original set of objects *X* into several disjoint subsets. Furthermore, any object from *X* belongs to only one cluster. Fuzzy clustering methods allow one object to belong simultaneously to several (or even all) clusters, but with different truth degree. Fuzzy clustering in many situations is more "natural" than non-fuzzy, for example, for objects located on the border of clusters.

The fuzzy clustering problem is set in the following way:

It is given that:

- $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, ... \mathbf{x}_n^T)^T$ objects to be clustered, where *n* - number of objects, T - transposition symbol. Each object $\mathbf{x}_k = (x_{k_1}, x_{k_2}, ..., x_{k_p})^T$, $k = \overline{1.n}$ is a point in the *p*-dimensional feature space;
- *c* - number of clusters ($2 \le c < n$).

It is necessary to put each element of the set $\mathbf{X}$ in accordance with the degree of belonging to the classes.

The elements of one cluster should be as close to each other as possible, and, at the same time, the clusters should be at the greatest distance from each other. To ensure manageability of the clustering process, it is necessary to use a measure of proximity, which is usually defined as the distance between two objects $\mathbf{x}_k$ and

$\mathbf{x}_l$ (points in p-dimensional space) and in the form of a real function $d : x \times x \to R^+$ such that:

$d(\mathbf{x}_k, \mathbf{x}_l) = d_{kl} \geq 0;$
$d_{kl} = 0 \Leftrightarrow \mathbf{x}_k = \mathbf{x}_l;$
$d_{kl} = d_{lk}.$

Additionally, if the function d satisfies the triangle rule, i.e. $d_{kl} \leq d_{kj} + d_{jl}$, then this function is a metric, although this property is not always necessary for clustering tasks.

Any partition of the set $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, ...\mathbf{x}_n^T)^T$ into fuzzy subsets *Si* ($i = \overline{1.c}$) can be fully described by the membership function $\eta_{S_i} : \mathbf{X} \to [0,1].$

Let us denote by η_{ik} the degree of belonging of the object $\mathbf{x}_k = (x_{k_1}, x_{k_2}, ..., x_{k_p})^T$, to the subset *Si*, i.e. $\eta_{ik} \equiv \eta_{S_i}(\mathbf{x}_k)$, and through $\mathbf{V}_{cn}$ the set of all real matrices of size $c \times n$. Then a fuzzy c-partition (or a membership degree matrix) is a matrix $\mathbf{M} = [\eta_{ik}] \in V_{cn}$ when following conditions are met:

$$\eta_{ik} \in [0,1],\ i = \overline{1.c},\ k = \overline{1.n}; \quad (4.1)$$

$$\sum_{i=1}^{c} \eta_{ik} = 1,\ k = \overline{1.n}; \quad (4.2)$$

$$\sum_{k=1}^{n} \eta_{ik} \in (0,n),\ i = \overline{1.c}. \quad (4.3)$$

Unlike a non-fuzzy one, with a fuzzy c-partition, any object simultaneously belongs to different clusters, but with different degrees. Conditions (4.2) and (4.3) only require that the sum of the degrees of belonging of the object to all clusters should be normalized to 1, and also that the number of clusters to which the object belongs does not exceed c.

Let's designate the centers of clusters, i.e. points in p-dimensional space around which the corresponding objects are concentrated, through $\mathbf{v}_i = (v_{i1}, v_{i2}, ..., v_{ip})$, $i = \overline{1.c}$.

When using the Euclidean distance, the problem of fuzzy clustering is to find such a matrix of degrees of membership **M** and such coordinates of cluster centers $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_c)$ that provide the minimum of the following criterion:

$$\sum_{i=1}^{c} \sum_{k=1}^{n} (\eta_{ik})^m \cdot \|\mathbf{x}_k - \mathbf{v}_i\|^2 \to \min,$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} (\eta_{ik})^m \cdot \mathbf{x}_k}{\sum_{k=1}^{n} \eta_{ik}}$$

where - the center of the *i-th* cluster, $i = \overline{1.c}$, *m* is the so-called exponential weight ($m \geq 1$). In mathematics, a notation $\|\bullet\|$ is usually understood as a norm, i.e. a function defined on a vector space and generalizing the concept of the length of a vector.

The exponential weight value is set before clustering begins. The exponential weight *m* affects the membership degree matrix M. The larger *m*, the more "smeared" the final *c*-partition matrix, and for $m \to \infty$ all objects belong to all clusters with the same membership degree, which is a very bad decision. Exponential weight also allows, when forming the coordinates of cluster centers, to enhance the influence of objects with large values of degrees of membership and to reduce the influence of objects with small values of degrees of membership. There is currently no theoretically valid rule for choosing the value of m. Usually it is set m = 2.

There is no analytical solution to the problem of finding the optimal coordinates of cluster centers and the membership degree matrix, so it is solved numerically. In MATLAB, the fuzzy center algorithm is implemented in the *fcm* function.

The *fcm* function can take three input arguments:

1. **X** is a matrix representing the data to be clustered. Each row of the matrix corresponds to one object;
2. **c** - the number of clusters that should be obtained as a result of executing the *fcm* function. The number of clusters must be greater than 1 and less than the number of objects specified by the matrix X.
3. **options** is an optional argument that sets the parameters of the clustering algorithm:

- options (1) - exponential weight value (default - 2.0);
- options (2) - maximum number of clustering algorithm iterations (default value is 100);
- options (3) - the minimum admissible value of the objective function improvement in one iteration of the algorithm (the default value is 0.000001);
- options (4) - output of intermediate results while the fcm function is running (default value is 1).

To use the default values, you can enter NaN as the value of the corresponding coordinate of the options vector.

The clustering algorithm stops when the maximum number of iterations has been completed or when the improvement in the objective function for one iteration is less than the specified minimum allowable value.

The *fcm* function has three output arguments:

1. **V** - matrix of coordinates of cluster centers obtained as a result of clustering. Each row of the matrix corresponds to the center of one
2. **M** - matrix of degrees of belonging of oblects to clusters. Each row of the matrix corresponds to the membership function of one
3. **obj_fcn** is the vector of values of the objective function at each iteration of the clustering algorithm.

Findcluster GUI Module:

The **Findcluster** GUI module automatically finds multidimensional data cluster centers using the fuzzy c-means algorithm and the subtractive clustering algorithm. The **Findcluster** module is loaded by **findcluster** command. The main graphic window of the **Findcluster** module with an indication of the purpose of functional areas is shown in figure 1.

The **Findcluster** module contains the 7 top sample menus of the graphics window (**File**, **Edit**, **View**, **Insert**, **Tools**, **Windows** and **Help**), visualization area, area of data load, clustering area, current information display area, as well as the **Info** and **Close** buttons that allow you to call the help window and close the module, respectively.

Visualization area

In this area, in two-dimensional space, experimental data (objects) and found cluster centers are displayed. A red circle marker (o) is used for images, and a black dot marker (.) for cluster centers.

The area also contains menus for selecting the **X-axis** and **Y axis**, which allow associating the features of images with the abscissa and ordinate axes.

Data loading area

This area, located in the upper right corner of the window, contains the **Load Data** button. Clicking this button loads the clustering data stored on disk. After clicking the **Load Data…** button, a typical file opening window opens. In file, the data must be written line by line, that is, one line of the data file must correspond to each object.

Current information display area

In this area, which is located at the bottom of the graphics window, the most important current information is displayed, for example, the state of the module, the iteration number of the clustering algorithm, the value of the objective function, etc.

Clustering area

In this area, the user can select the clustering algorithm, set the parameters of the clustering algorithm, perform clustering and save the coordinates of the cluster centers as a file. The area contains the following menus and buttons.

The **Method** menu allows you to select one of two clustering algorithms: **subtractiv** - subtractive clustering algorithm; **fcm** is a fuzzy c-means algorithm. When the subtractive clustering algorithm is selected, the **Findcluster** module graphical window looks like the one shown in Figure 1. In this case, the user can set the values of the following algorithm parameters **Influence Range**, **Squash**, **Accept Ratio** and **Reject Ratio**, the meaning of which is explained in the description of the **subclust** function. When choosing a fuzzy c-means algorithm, the clustering region takes the form shown in Figure 2. In this case, the user can set the values of the following parameters: Cluster Num. - the number of clusters; **Max Iteration #** - maximum number of iterations of the algorithm; **Min** is the minimum admissible value of the objective function improvement in one iteration of the algorithm; **Exponent** - exponential weight values.

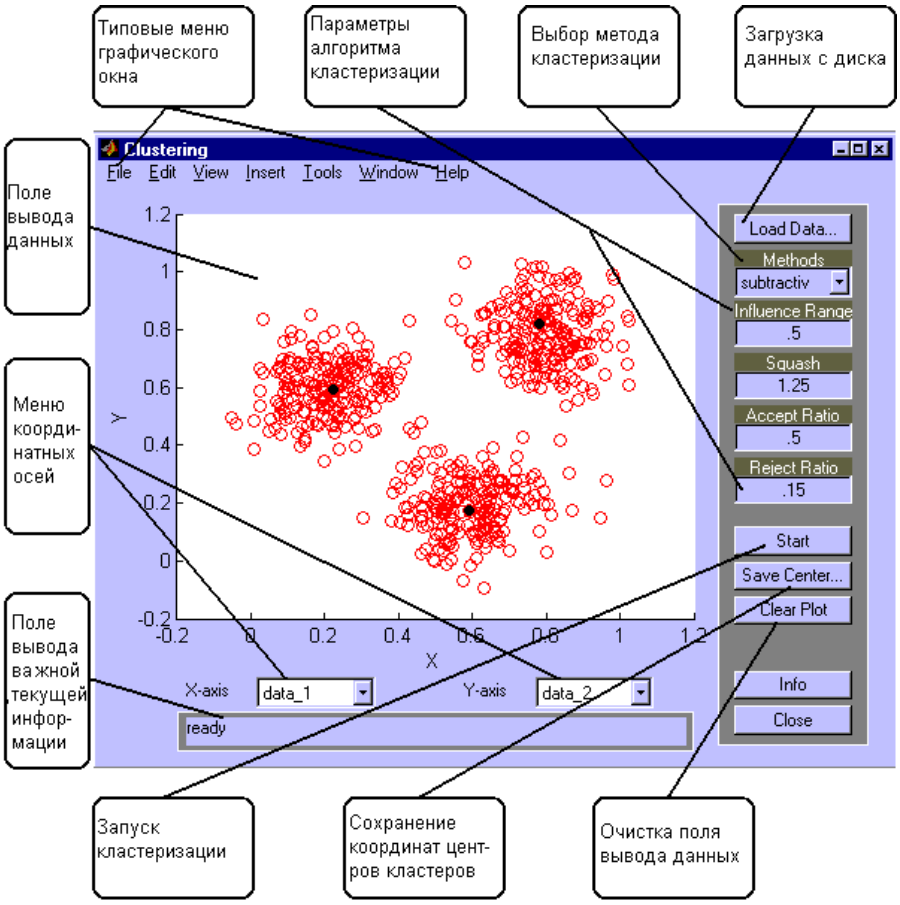

Figure 1 - Findcluster main window

Figure 2 - Clustering area

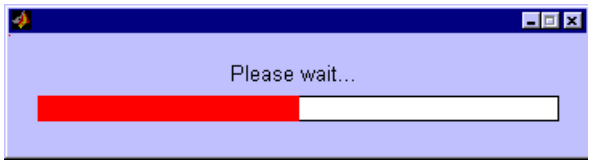

Figure 3 - Subtractive clustering algorithm execution window

Start button starts clustering. When using the *fcm* algorithm, the coordinates of the cluster centers are displayed in the visualization window after each iteration. When using the subtractive algorithm, an additional window opens (fig. 3) showing the dynamics of the clustering process. The coordinates of cluster centers are displayed at the end of the algorithm execution.

The **Clear Plot** button allows you to clear the data output field.

To identify cluster centers (points in a multidimensional data space), open the clusterdemo.dat file to view its structure (fig. 4). The file is an array of numbers (experimental data) grouped into three columns (multidimensional array).

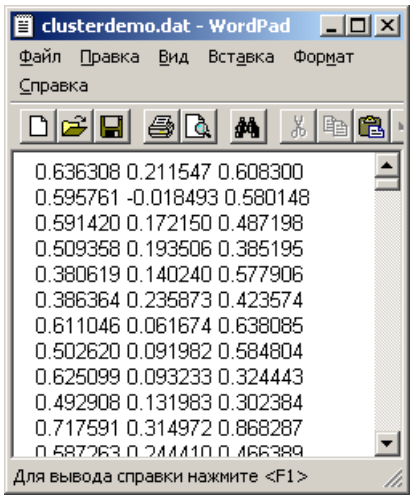

Figure 4 - Clusterdemo.dat file

***Example*:**

1. Load the matlab data file\toolbox\fuzzy\fuzdemos\clusterdemo.dat using the "Load data" Button;

2. Select the fuzzy c-means (fcm) clustering algorithm using the "Method" Button;

3. Set the number of clusters to 3 using the "Cluster num" Option button;

4. Set the number of iterations equal to 100 using the "Max iteration #" Option button;

5. Press the start button and get the result (fig. 5)

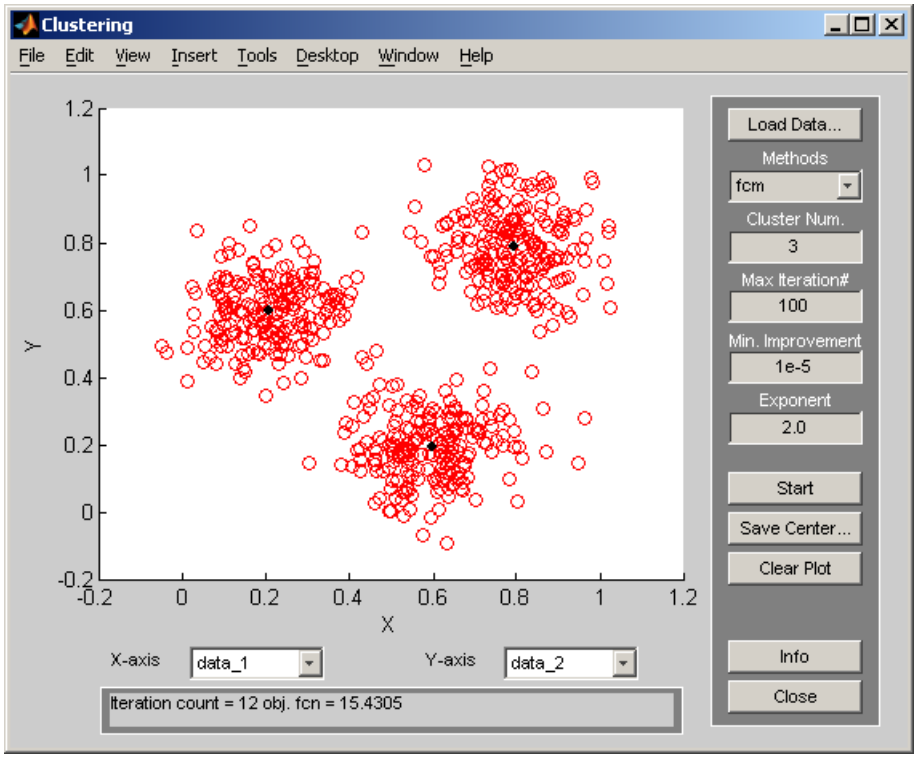

Figure 5 - The output of the Clustering program (the centers of the clusters are colored black)

With an increase in the number of clusters to 30, we get the result shown in fig. 6.

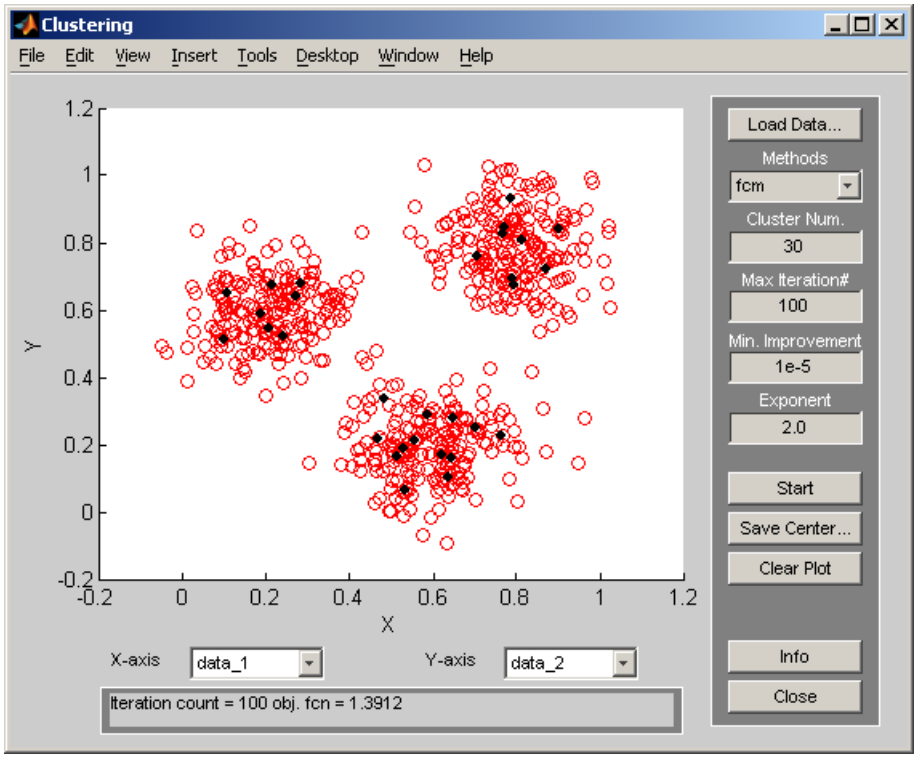

Figure 6 - The result of the Clustering program (cluster centers are colored black)

Conclusions (example):

During the laboratory work, we got acquainted with the graphical environment of the clustering program - Clustering, the Fuzzy Logic Toolbox package, which identifies the centers of clusters. We made sure that using this program, you can quickly find the centers of the clusters.

Using a demo example, we compared the operation of two algorithms for finding cluster centers (subtractive clustering algorithm and fuzzy centers algorithm). As a result of the comparison, it was revealed that …….(!make your own research!)

Test questions:

1. What program was used to identify cluster centers?
2. Give a definition of clustering.
3. For what tasks can clustering be used?
4. How can the distance between clusters be determined?
5. Name the stages of the cluster analysis algorithm.
6. How are cluster centers determined?