IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

Data Stream Online Clustering Based on Fuzzy Expectation-Maximization Approach

Anastasiia O. Deineko
Artificial Intelligence Department
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
anastasiya.deineko@gmail.com

Polina Ye. Zhernova
System Engineering Department
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
polina.zhernova@gmail.com

Boris Gordon
Computer Systems Department
Tallinn University of Technology
Tallinn, Estonia
boris.gordon@ttu.ee

Oleksandr O. Zayika
Artificial Intelligence Department
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
zaiqa.alexander@gmail.com

Iryna Pliss
Control Systems Research Laboratory
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
iryna.pliss@nure.ua

Nelya Pabyrivska
Department of Mathematics
Lviv Polytechnic National University
Lviv, Ukraine
nelyapab@gmail.com

***Abstract*—In the paper the online fuzzy clustering recurrent procedure has been introduced that allows the forming of hyperellipsoidal clusters with an arbitrary orientation of the axes is proposed. Such clustering system is the generalization of a number of known algorithms, it is intended to solve tasks within the general problems of Data Stream Mining (DSM) and Dynamic Data Mining (DDM), when information is sequentially fed to processing in online mode.**

Keywords— big data; dynamic data mining; data stream mining; computational intelligence; EM-algorithm; fuzzy clustering; Kohonen's self-learning; soft clustering.

I. Introduction

Clustering task is an integral part and an important direction in the global problem of Data Science and Data Mining [1, 2]. Many approaches and methods were proposed for solution of this problem [3-6]. They differ from each other in apriori assumptions, problem formulation and in used mathematical apparatus. Currently, the most intensively growing direction of Data Mining is *DSM* [7], in which data are fed to processing in the online mode, observation by observation. This direction is closely related with tasks of processing large amounts of data, so-called Big Data [8], when it is simply impossible to process increasing volume of data in the batch mode.

Computational Intelligence *(CI)* can be successfully used for many tasks of *DSM*. And first of all the methods based on soft computing and neural networks among which the Fuzzy Clustering *(FC)* methods [4, 9, 10] are the most wide spread. At the same time, the overwhelming number of known methods are oriented to the batch mode processing. And known intelligent systems of sequential data processing and first of all Kohonen's clustering neural networks, which also are known as Self-Organizing Maps [11], can solve crisp clustering tasks with the assumption of linearly separable classes.

Capabilities of crisp clustering algorithms are restricted by the fact that real data usually form overlapping classes, thus each vector-observation could belong to several classes at once, with different probabilities (or belonging) levels. In this case soft calculations come to the fore. In the class of probabilistic methods most widely used is so-called Expectation-Maximization *(EM)* algorithm [10, 12-15]. And in the class of fuzzy methods the most popular is J.C. Bezdek's Fuzzy *C*-means *(FCM)* algorithm [9, 10]. It can be noted that in [15] the hybrid clustering algorithm has been discussed. It unites both of these approaches.

The mentioned clustering procedures which are based on soft computing are oriented to information processing only in batch mode. Naturally this fact makes usage of these methods in *DSM* possible. Note that in [16, 17] group of recursive *FC* algorithms were introduced. But clusters, what they are formed have a spherical shape. This fact limits their capabilities in situations where data form classes of an arbitrary form.

In this connection, it seems appropriate to develop recurrent procedures for probabilistic and fuzzy clusterization, which allow to process data in online mode and to form clusters of the hyperellipsoidal form with the axes of arbitrary orientation in the features space.

II. Batch Procedures for Probabilistic and Fuzzy Clustering (FC) in the Case of Hyperellipsoidal Classes

The batch clustering problem can be described in general case: it is assumed that the initial data array contains N multidimensional observations, which are described by vectors-features of order n $\left(x_1(k),..., x_i(k),..., x_n(k)\right)^T \in R^n$, $k = 1, 2, 3, ..., N$ (k – number of observation in initial data array), which has to be partitioned for $m(1<m<N)$ overlapping clusters.

In a standard *EM* approach, it is also assumed that the density of the distribution of observations in each cluster is Gaussian:

$$p_j(x) = \left((2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j} \right)^{-1} \exp\left(-\frac{1}{2} (x - c_j)^T \Sigma_j^{-1} (x - c_j) \right),$$

$$j = 1, 2, \ldots, m \qquad (1)$$

where c_j – vector-centroid with order *n* of the *j*-th cluster, Σ_j – the correlation matrix of the *j*-th cluster of size $(n \times n)$:

$$\Sigma_j = \frac{1}{N} \sum_{k=1}^{N} (x(k) - c_j)(x(k) - c_j)^T \qquad (2)$$

It is obvious that the joint density of the distribution of all data is described by the expression

$$\begin{aligned} p(x) &= \sum_{k=1}^{m} p_j p_j(x) = \\ &= \sum_{k=1}^{m} p_j \left((2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j} \right)^{-1} \exp\left(-\frac{1}{2} (x - c_j)^T \Sigma_j^{-1} (x - c_j) \right) = \\ &= \sum_{k=1}^{m} p_j \left((2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j} \right)^{-1} \exp\left(-\frac{1}{2} d_M^2 (x, c_j) \right) \end{aligned} \qquad (3)$$

where p_j – a priori probabilities-weights that satisfy the obvious condition

$$\sum_{j=1}^{m} p_j = 1. \qquad (4)$$

It can be noted that condition (4) completely coincides with the constraint on the levels of belonging of the *k*-th observation to the *j*-th cluster $u_j(k)$, which is the basis of the fuzzy c-means method

$$\sum_{j=1}^{m} u_j = 1. \qquad (5)$$

In connection with that fact, *FCM* is sometimes called the method of fuzzy probabilistic clustering [10].

The main feature of the *EM* approach is that the exponent in (1), (3) contains the Mahalanobis distance between centroids c_j and observations $x(k)$

$$d_m^2(x(k), c_j) = (x(k) - c_j)^T \Sigma_j^{-1} (x(k) - c_j), \qquad (6)$$

which allows, in contrast to *FCM* that restores spherical clusters to form classes in the form of hyperellipsoids with an arbitrary orientation of axes in the space of features.

The solution of the clustering problem in the context of the *EM* approach is related to the maximization of the log likelihood function

$$E(x(k), c_j, \Sigma_j, p_j) = \sum_{k=1}^{m} \log\left(\sum_{k=1}^{m} p_j p_j(x(k)) \right), \qquad (7)$$

that leads to estimates [12]

$$\begin{cases} p_j(x(k)) = \exp\left(-\frac{1}{2} d_M^2(x(k), c_j) \right) \Big/ \sum_{l=1}^{m} \exp\left(-\frac{1}{2} d_M^2(x(k), c_j) \right), \\ c_j = \sum_{k=1}^{N} p_j(x(k)) x(k) \Big/ \sum_{k=1}^{N} p_j(x(k)). \end{cases} \qquad (8)$$

A particular crisp version of the *EM* algorithm is the widely used k-means method coinciding with *EM* at $p_j = m^{-1}$ and identity matrix Σ_j.

K-means is based on minimizing the objective function

$$\begin{aligned} E(x(k), c_j) &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j(k) \left\| x(k) - c_j \right\|^2 = \\ &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j(k) d_E^2(x(k), c_j) \end{aligned} \qquad (9)$$

where

$$u_j(k) = \begin{cases} 1, \text{ if } x(k) \in j - th \text{ cluster}, \\ 0, \text{ otherwise}. \end{cases}$$

It should be noted that k-means is based on the Euclidean distance, although the method of Mahalanobis k-means is also known. It is based on the minimization of the goal function in the form

$$\begin{aligned} E(x(k), c_j) &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j(k) (x(k) - c_j)^T \Sigma_j^{-1} (x(k) - c_j) = \\ &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j(k) d_M^2(x(k), c_j). \end{aligned} \qquad (11)$$

As a result of optimization (9), (11), it is not difficult to obtain estimates of the centroids coordinates in the form

$$c_j = \sum_{k=1}^{N} u_j(k) x(k) \Big/ \sum_{k=1}^{N} u_j(k) = \frac{1}{N_j} \sum_{x(k) \in u_j} x(k) \qquad (12)$$

where N_j – the number of observations assigned to *j*-th cluster.

A generalization of crisp objective functions (9), (11) in the case of overlapping classes is fuzzy objective functions [18]

$$E(x(k), c_j, u_j) = \sum_{k=1}^{N} \sum_{j=1}^{m} u_j^{\beta}(k) d^2(x(k), c_j) \qquad (13)$$

where β – a non-negative fuzzifier parameter. Such parameter determines the "bluring" of the boundaries between classes (usually $\beta = 2$), $d^2\left(x(k), c_j\right)$ – estimate of Euclidean distance between $x(k)$ and c_j.

Minimization (13), taking into account the constraint (5), leads to the result [3]:

$$\begin{cases} u_j(k) = d^{\frac{2}{1-\beta}}\left(x(k), c_j\right) \Big/ \sum_{l=1}^{m} d^{\frac{2}{1-\beta}}\left(x(k), c_l\right), \\ c_j = \sum_{k=1}^{N} u_j^{\beta}\left(x(k), c_j\right) \Big/ \sum_{k=1}^{N} u_j^{\beta}(k), \end{cases} \quad (14)$$

that for $\beta = 2$ *FCM* takes the form:

$$\begin{cases} u_j(k) = \dfrac{d_E^{-2}\left(x(k), c_j\right)}{\sum_{l=1}^{m} d_E^{-2}\left(x(k), c_l\right)} = \dfrac{\left\|x(k) - c_j\right\|^{-2}}{\sum_{l=1}^{m}\left\|x(k) - c_l\right\|^{-2}}, \\ c_j = \sum_{k=1}^{N} u_j^2(k)x(k) \Big/ \sum_{k=1}^{N} u_j^2(k). \end{cases} \quad (15)$$

The first relation (14) can be easily transformed to the form

$$u_j(k) = \frac{1}{1 + \dfrac{d^{\frac{2}{\beta-1}}\left(x(k), c_j\right)}{\gamma_j}},$$
$$\gamma_j = \left(\sum_{\substack{l=1, \\ l \neq j}}^{m} d^{\frac{2}{\beta-1}}\left(x(k), c_l\right)\right)^{-1}, \quad (16)$$

corresponding to the generalized Gaussian function [19], with $\beta = 2$ we get

$$u_j(k) = \frac{1}{1 + \dfrac{d^2\left(x(k), c_j\right)}{\gamma_j}},$$
$$\gamma_j = \left(\sum_{\substack{l=1, \\ l \neq j}}^{m} d^2\left(x(k), c_l\right)\right)^{-1}, \quad (17)$$

corresponding to the Cauchy probabilities density function.

Thus, it can be noted that if the *EM* approach is based on the Gaussian distribution, then for fuzzy procedures, the Cauchy distribution is implicit. Among the fuzzy clustering procedures based on the objective function (13), the closest to the *EM* approach is the algorithm introduced in [20]. It uses as an estimate of the distance expression

$$d_{GG}^2\left(x(k), c_j\right) = q_j\left(\sqrt{\det \Sigma_j}\right)^{-1} \exp\left(-\frac{1}{2}\left(x - c_j\right)^T \Sigma_j^{-1}\left(x - c_j\right)\right) = \\ = q_j\left(\sqrt{\det \Sigma_j}\right)^{-1} \exp\left(-\frac{1}{2} d_M^2\left(x, c_j\right)\right) \quad (18)$$

where

$$q_j = \sum_{k=1}^{N} u_j^{\beta}(k) \Big/ \sum_{k=1}^{N}\sum_{l=1}^{m} u_l^{\beta}(k) \quad (19)$$

.

Minimization of the (18), taking into account (5) and (19), leads to the result

$$\begin{cases} u_j(k) = d_{GG}^{\frac{2}{1-\beta}}\left(x(k), c_j\right) \Big/ \sum_{l=1}^{m} d^{\frac{2}{1-\beta}}\left(x(k), c_j\right), \\ c_j = \sum_{k=1}^{N} u_j^{\beta}(k)x(k) \Big/ \sum_{k=1}^{N} u_j^{\beta}(k), \\ \Sigma_j = \sum_{k=1}^{N} u_j^{\beta}(k)\left(x(k) - c_j\right)\left(x(k) - c_j\right)^T \Big/ \sum_{k=1}^{N} u_j^{\beta}(k). \end{cases} \quad (20)$$

for $\beta = 2$ (20) becomes

$$\begin{cases} u_j(k) = d_{GG}^{-2}\left(x(k), c_j\right) \Big/ \sum_{l=1}^{m} d^{-2}\left(x(k), c_j\right), \\ c_j = \sum_{k=1}^{N} u_j^2(k)x(k) \Big/ \sum_{k=1}^{N} u_j^2(k), \\ \Sigma_j = \sum_{k=1}^{N} u_j^2(k)\left(x(k) - c_j\right)\left(x(k) - c_j\right)^T \Big/ \sum_{k=1}^{N} u_j^2(k) \end{cases} \quad (21)$$

close to (15) and is a generalization of *FCM* for the case of hyperellipsoidal clusters.

III. ADAPTIVE ONLINE PROCEDURES FOR PROBABILISTIC AND FUZZY CLUSTERING IN THE CASE OF HYPERELLIPSOIDAL CLUSTERS

Let's consider further a case when the data are fed to processing sequentially one after another in the form of a stream $x(1), x(2), ..., x(k), x(k+1), ...$, where k has the sense of the current discrete time. It is clear that the fuzzy clustering procedures discussed above in this case are ineffective. It is known that the optimization problem solution of the objective function (9) corresponding to the k-means method can be obtained with the help of the self-learning *WTA*-rule of the clustering neural network of T. Kohonen [21] in the form

$$c_j(k+1) = \begin{cases} c_j(k) + \eta(k+1)\left(x(k+1) - c_j(k)\right), \\ \text{if } c_j(k) - \text{"winner"}, \\ c_j(k) - \text{otherwise} \end{cases} \quad (22)$$

where $0 < \eta(k+1) < 1 -$ the learning rate parameter chosen in the accordance with the stochastic approximation conditions.

Here it should be noted that it is possible to draw a clear analogy between self-learning according to T. Kohonen and the *EM* algorithm: the step of competition corresponds to the E-step (expectation), and the step of synaptic adaptation is the M-step (maximization). At the step of synaptic adaptation, a step of gradient minimization of the distance

$$d_E^2\left(x(k+1), c_j(k)\right) = \left\|x(k+1) - c_j(k)\right\|^2 \quad (23)$$

is realized, i.e., procedure (22) can be represented in the form

$$c_j(k+1) = \begin{cases} c_j(k) - \eta(k+1)\nabla_{c_j} d_E^2\left(x(k+1), c_j(k)\right), \\ if\ c_j(k) - "winner", \\ c_j(k) - otherwise. \end{cases} \quad (24)$$

Similarly, the Mahalanobis metric (6), used in the EM algorithm, can be minimized [22]:

$$c_j(k+1) = \begin{cases} c_j(k) - \eta(k+1)\nabla_{c_j} d_M^2\left(x(k+1), c_j(k)\right), \\ if\ c_j(k) - "winner", \\ c_j(k) - otherwise, \end{cases} \quad (25)$$

or

$$c_j(k+1) = \begin{cases} c_j(k) + \eta(k+1)\Sigma_j^{-1}(k)\left(x(k+1), c_j(k)\right), \\ if\ c_j(k) - "winner", \\ \Sigma_j(k) = \frac{1}{k_j}\sum_{\tau=1}^{k}\left(x(\tau) - c_j(k)\right)\left(x(\tau) - c_j(k)\right)^T = \\ = \Sigma_j(k-1) + \frac{1}{k_j}\Big(\left(x(k) - c_j(k)\right)\left(x(k) - c_j(k)\right)^T - \\ -\Sigma_j(k-1)\Big), c_j(k) - otherwise \end{cases} \quad (26)$$

where the index k_j shows how many times the *j*-th neuron of Kohonen's *SOM* was the winner in the process of self-learning.

In the case of overlapping classes, procedure (26) can be supplemented by an estimate of the membership level of the first relation type (15):

$$u_j(k) = d_M^{-2}\left(x(k), c_j(k)\right) \Big/ \sum_{l=1}^{m} d_M^{-2}\left(x(k), c_l(k)\right) = \\ = \frac{\left(\left(x(k) - c_j(k)\right)^T \Sigma_j^{-1}(k)\left(x(k) - c_j(k)\right)\right)^{-1}}{\sum_{l=1}^{m}\left(\left(x(k) - c_l(k)\right)^T \Sigma_l^{-1}(k)\left(x(k) - c_l(k)\right)\right)^{-1}}. \quad (27)$$

The task of fuzzy objective function recurrent minimization of type (13) with constraint (5) reduces to the solution of the non-linear programming problem by the Arrow-Hurwitz-Uzawa procedure by optimizing the Lagrange function. In this case [16]:

$$\begin{cases} c_j(k+1) = c_j(k) + \eta(k+1)u_j^\beta(k+1)\left(x(k+1) - c_j(k)\right), \\ u_j(k+1) = d_E^{\frac{2}{1-\beta}}\left(x(k+1), c_j(k)\right) \Big/ \sum_{l=1}^{m} d_E^{\frac{2}{1-\beta}}\left(x(k+1), c_l(k)\right), \end{cases} \quad (28)$$

and for $\beta = 2$ *FCM*:

$$\begin{cases} c_j(k+1) = c_j(k) + \eta(k+1)u_j^2(k+1)\left(x(k+1) - c_j(k)\right), \\ u_j(k+1) = \left\|x(k+1) - c_j(k)\right\|^{-2} \Big/ \sum_{l=1}^{m}\left\|x(k+1) - c_l(k)\right\|^{-2}, \end{cases} \quad (29)$$

where the factors $u_j^\beta(k+1)$ and $u_j^2(k+1)$ play the role of the neighborhood function in the *WTM*-rule of self-learning, instead of the traditionally used Gaussians, generalized Gaussian is used in (28), and in (29) - Cauchian, while the width parameter of these functions is given automatically.

As for the Gath-Geva algorithm [20], described by the expression (18-21), recurrent modifications were introduced in [23]. However, they are not related to optimization procedures. Thus, in [23] a simplified procedure of the form

$$\begin{cases} c_j(k+1) = c_j(k) + \eta(k+1)\left(x(k+1) - c_j(k)\right), \\ if\ c_j(k) - "winner", \\ \Sigma_j(k+1) = \left(1 - \eta(k+1)\right)\Sigma_j(k) + \eta(k) * \\ *\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T \end{cases} \quad (30)$$

that is essentially a *WTA*-rule of self-learning, supplemented with the procedure for the correlation matrix correcting. In this case, this matrix does not influence the centroids correction process.

More flexible is the algorithm proposed in [23], where an additional variable of accumulated memberships were introduced into consideration:

$$U_j(k+1) = \sum_{\tau=1}^{k+1} u_j^\beta(\tau) = U_j(k) + u_j^\beta(k+1). \quad (31)$$

In this case, the *FC* procedure becomes the form

$$\begin{cases} c_j(k+1) = c_j(k) + \frac{u_j^\beta(k+1)}{U_j(k+1)}\left(x(k+1) - c_j(k)\right), \\ \Sigma_j(k+1) = U_j(k)/U_j(k+1) * \\ *\left(\Sigma_j(k) + \frac{u_j^\beta(k+1)}{U_j(k+1)}\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T\right), \\ u_j(k+1) = d_{GG}^{\frac{2}{1-\beta}}\left(x(k+1), c_j(k)\right) \Big/ \sum_{l=1}^{m} d_{GG}^{\frac{2}{1-\beta}}\left(x(k+1), c_j(k)\right). \end{cases} \quad (32)$$

The algorithm (32) coincides with (28) for $\eta(k+1) = U_j^{-1}(k+1)$ and differs only in the used distance estimate $d_{GG}^2\left(x(k+1), c_j(k)\right)$ instead of $d_E^2\left(x(k+1), c_j(k)\right)$. The correlation matrix $\Sigma_j(k)$ does not affect the process of centroids correction.

Returning to the algorithm (25) based on the gradient of the Mahalanobis distance, it is easy to introduce a recurrent version of the Gath-Geva algorithm, which is a generalization of the procedures (25-26, 28):

$$\begin{cases} c_j(k+1) = c_j(k) + \frac{u_j^\beta(k+1)}{U_j(k+1)} \Sigma_j^{-1}\left(x(k+1) - c_j(k)\right), \\ \Sigma_j(k+1) = \frac{U_j(k)}{U_j(k+1)} * \\ * \left(\Sigma_j(k) + \frac{u_j^\beta(k+1)}{U_j(k+1)}\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T\right), \\ u_j(k+1) = d_{GG}^{\frac{2}{1-\beta}}\left(x(k+1), c_j(k)\right) \Big/ \sum_{l=1}^{m} d_{GG}^{\frac{2}{1-\beta}}\left(x(k+1), c_j(k)\right). \end{cases} \quad (33)$$

For $\beta = 2$ we obtain

$$\begin{cases} U_j(k+1) = U_j(k) + u_j^2(k+1), \\ c_j(k+1) = c_j(k) + \frac{u_j^2(k+1)}{U_j(k+1)} \Sigma_j^{-1}\left(x(k+1) - c_j(k)\right), \\ \Sigma_j(k+1) = \frac{U_j(k)}{U_j(k+1)} * \\ * \left(\Sigma_j(k) + \frac{u_j^2(k+1)}{U_j(k+1)}\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T\right), \\ u_j(k+1) = d_{GG}^{-2}\left(x(k+1), c_j(k)\right) \Big/ \sum_{l=1}^{m} d_{GG}^{-2}\left(x(k+1), c_j(k)\right). \end{cases} \quad (34)$$

IV. Results of Experiments

Tree data set from *UCI Machine Learning Repository* [24] are used in the experimental analysis. The information about used data sets is shown in Table I.

TABLE I. THE DESCRIPTION OF THE DATA SETS

Data sets	**Properties**		
	Attributes	Classes	Number of samples
Iris	4	3	150
WDBC	30	2	569

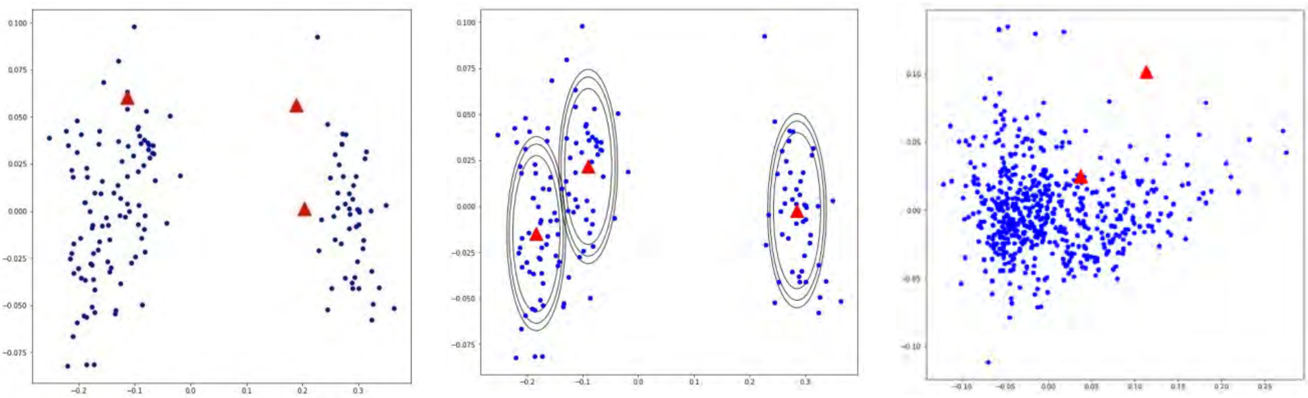
The performance of described in these paper systems were compared in series of experiments. For comparison of proposed soft clustering system, the standard *FCM* algorithm and standard *EM* algorithm were taken. The clustering accuracy of proposed soft clustering system was measured and compered with *FCM* and *EM* algorithms. The clusterization results were shown in Table II. These clustering results of the proposed soft clustering system, *FCM* algorithm and standard *EM* algorithm were estimated using the well-known *Xie-Benie criterion* for fuzzy clustering. From Table II easy to see that proposed soft clustering system demonstrated a better performance of clustering quality. The changes centroids coordinates from initial initialization to the final iteration are shown at the Fig. 1.

TABLE II. THE MEAN CLUSTERING ACCURACIES OF THE COMPARED ALGORITHMS

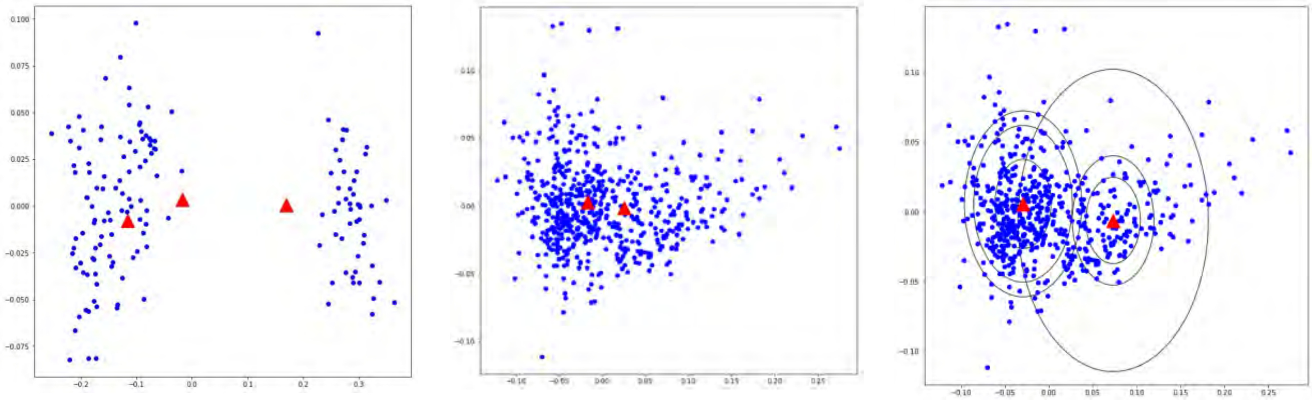
Algorithms for comparison	**Clustering accuracies**	
	Iris	**WDBC**
FCM	0,82	0,86
EM	0,84	0,85
Soft clustering system	0,89	0,90

V. Conclusion

The online fuzzy clustering problem was considered. The recurrent procedure has been introduced that allows the forming of hyperellipsoidal clusters with an arbitrary orientation of the axes. The proposed procedure is the generalization of a number of known algorithms, it is quite simple in computational implementation and is intended to solve tasks within the general problem of Data Stream Mining, when information is sequentially fed to processing.



a) Initial centroids coordinates "Iris" c) Final centroids coordinates "Iris d) Initial centroids coordinates "WDBC"



b) Middle iteration "Iris" e) Middle iteration "WDBC" f) Final centroids coordinates "WDBC"

Fig. 1. The changes centroids coordinates

REFERENCES

[1] C. C. Aggarwal, Data Mining. Cham: Springer, Int. Publ., Switzerland, 2015.

[2] M. Bramer, Principles of Data Mining. Springer-Verlag London, 2016.

[3] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, Machine Learning for Data Streams with Practical Examples in MOA. The MIT Press, 2018.

[4] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. John Wiley & Sons. Chichester, 1999.

[5] C. C. Aggarwal and C. K. Reddy, Data Clustering. Algorithms and Application. Boca Raton: CRC Press, 2014.

[6] R. Xu and D. C. Wunsch, Clustering. IEEE Press Series on Computational Intelligence. Hoboken, NJ: John Wiley & Sons, Inc., 2009.

[7] A. Bifet, Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams, IOS Press, 2010.

[8] J. Kacprzyk, and W. Pedrycz, Springer Handbook of Computational Intelligence, Berlin Heidelberg: Springer, Verlag, 2015.

[9] K.-L. Du and M. N. S. Swamy, Neural Networks and Statistical Learning. London: Springer-Verlag, 2014.

[10] J.-C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, N.Y.: Plenum Press, 1981.

[11] Ye. V. Bodyanskiy, A. O. Deineko, and Y. V. Kutsenko, "On-line kernel clustering based on the general regression neural network and T. Kohonen's self-organizing map," Automatic Control and Computer Sciences, 51(1), pp. 55-62, 2017.

[12] J. Keller, J. C. Bezdek, R. Krishnapuram and N. Pal, Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. The Handbook of Fuzzy Sets. Kluwer, Dordrecht, Netherlands: Springer, vol. 4, 1999.

[13] B. Quost, and T. Denœux "Clustering and classification of fuzzy data using the fuzzy EM algorithm," Fuzzy Sets and Systems. vol. 286, pp. 134-156, March 2016.

[14] J. Yu, Ch. Chaomu, and M. S. Yang, "On convergence and parameter selection of the EM and DA-EM algorithms for Gaussian mixtures," Pattern Recognition, vol. 77, pp. 188-203, May 2018.

[15] X. L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm:a general framework," Biometrica, vol. 80, pp. 267-278, 1993.

[16] Ye. Bodyanskiy, "Computational intelligence techniques for data analysis," Lecture Notes in Informatics, Bonn: GI, pp. 15 – 36, 2005.

[17] Ye. Gorshkov, V. Kolodyaznhiy and Ye., Bodyanskiy, "New recursive learning algorithms for fuzzy Kohonen clustering network," 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems, Rapperswil, Switzerland, pp. 58-61, 2009.

[18] L. Jain and C. Mumford, Computational Intelligence, Collaboration, Fuzzy and Emergence, Berlin: Springer, Vergal, 2009.

[19] S. Osowski, Sieci neuronowe do przetwarzania informacji, Warszawa: Oficijna Wydawnicza Politechniki Warszawskiej, 2006.

[20] A. B. Geva and I. Gath "Unsupervised optimal fuzzy clustering," Pattern Analysis and Machine Intelligence, vol. 2, n.7, pp. 773-787, 1989.

[21] T. Kohonen, Self-Organizing Maps. Berlin: Springer-Verlag, 1995.

[22] Ye. Bodyanskiy, A. Deineko, Y. Kutsenko and O. Zayika, "Data streams fast EM-fuzzy clustering based on Kohonen`s self-learning," 1th IEEE International Conference on Data Stream Mining & Processing (DSMP 2016), Lviv, Ukrane, pp. 309-313, 2016.

[23] A. B. Geva, "Clustering as a basis for evolving neuro-fuzzy modeling," Evolving Systems, pp. 59-71, 2010.

[24] UCI Repository of machine learning databases. CA: University of California, Department of Information and Computer Science. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html