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

Adaptive Kernel Data Streams Clustering Based on Neural Networks Ensembles in Conditions of Uncertainty About Amount and Shapes of Clusters

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

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

Yevgeniy V. Bodyanskiy
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yevgeniy.bodyanskiy@nure.ua

Vladyslav O. Riepin
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
revanmax626@gmail.com

***Abstract*—The neural network's approach for data stream clustering task, that in online mode are fed to processing in assumption of uncertainty about amount and shapes of clusters, is proposed in the paper. The main idea of this approach is based on the kernel clustering and idea of neural networks ensembles, that consist of the T. Kohonen's self-organizing maps. Each of the clustering neural networks consists of different number of neurons, where number of clusters is connected with the quality of these neurons. All ensemble members process information that sequentially is fed to the system in the parallel mode. Experimental results have proven the fact that the system under consideration could be used to solve a wide range of Data Mining tasks when data sets are processed in an online mode.**

***Keywords*—clustering, X-means method, ensemble of neural networks, self-learning, T. Kohonen's neural network.**

I. Introduction

Data stream clustering is an important part of Data Mining. Many approaches to its solution have been developed [1, 2]. Processing of large information volumes requires, first of all, a high speed and simple numerical implementation of clustering algorithms. One of the most popular procedures is the K- means method due to its simplicity, clarity of results and possibilities for their explicit interpretation [3]. This method refers to the algorithms based on calculation of centroids-prototypes. In the frame of this approach, an initial data set (possibly growing) $X=\{x(1),...,x(k),...,x(N)\}$, $x(k)=(x_1(k),...x_i(k),...,x_n(k))^T\in R^n$, $k=1,2,...N$ is partitioned into m clusters where their number m is defined a priori or chosen empirically.

The X-means method is alternative approach for empirical methods of clustering, but it is more bulky from computational point of view and connected with strict apriori statistical assumptions about character of initial data distribution [4, 5].

Besides, both these methods require multi epoch procedure for initial data set *X*, that has limited opportunities for processing big data sets (Big Data) and data streams, when information is fed to the inputs of the clustering system sequentially observation by observation in the online mode (Data Stream Mining). In this situation number of observation *k* has the sense of current discrete time, but data volume *N* practically doesn't limited.

In similar situations clustering self-learning artificial neural networks [6-9] show themselves rather effective. First of all, it concerns self-organizing T. Kohonen's maps (*SOM*) [10] that can process data in sequential mode. *SOM* processing results coincide with the K-means results, wherein the number of *m* clusters is known apriori.

Saving capabilities of online processing using *SOM* and establishing the number of m clusters with K-means is possible, using the idea of clustering ensembles [11-14]. As elements of the ensemble it is needed to use Kohonen's clustering neural networks SOM^m [15], where every network is tuned for a different number of possible classes *m=2,3,…,M*. Under this approach, first member of the ensemble SOM^2 in Kohonen's layer contains only two neurons with vectors of synaptic weights w_1^2, w_2^2 . The last member SOM^M contains *M* neurons with centroids-weights $w_1^M, w_2^M, ..., w_M^M$.

In ensemble self-learning process, all SOM^m are operate in parallel. As the final result is chosen clustering network-winner, which shows the most appropriate results in terms of the applied quality criterion for clustering [2,16]. Note that in every SOM^m at each cycle k of information processing neuron-winner is chosen exactly the same as neural network-winner is chosen in ensemble at each tact. It is the best result of clustering.

An essential restriction that reduces approach capabilities is the requirement of formed clusters linear separability and convexity. Whereas the real data have the ability to form classes of completely arbitrary form. In such a situation it can be useful to use T. Cover's theorems of linear separation in spaces of higher dimension and J. Mercer's kernels, which provide this increasing [17, 18]. Based on this approach so-called, kernel self-organizing maps (KSOM) were developed

978-1-5386-2874-4/18/$31.00 ©2018 IEEE

[19-21]. They show quite good results under conditions of clusters arbitrary forms with a known number m of them, using fixed volume N of the processed selection. Therefore, this seems expedient to develop an ensemble of kernel clustering neural networks. It is intended for data streams online processing under conditions of an unknown or changing number of classes.

The architecture of kernel clustering neural networks ensemble is shown on Fig. 1. It contains five layers of information processing.

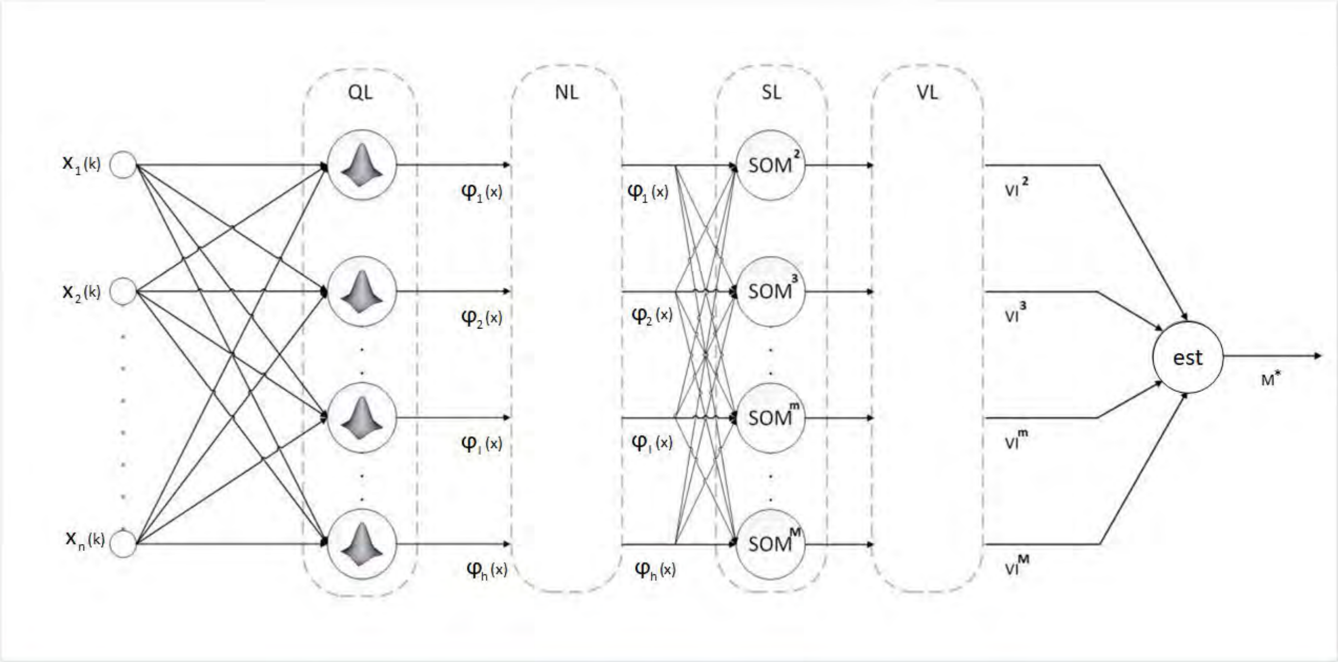


Fig. 1. The architecture of kernel clustering neural networks ensemble

The initial information to be clustered is fed to the zero (input) layer of the system as a sequence $x(1), x(2),..., x(k),..., x(N),...$ Then, it enters to the first hidden layer (RL) of radial-basis functions, formed by R-neurons. Right in this layer increasing in the dimensionality of the input space with the help of kernel functions system $\varphi_1(x), \varphi_2(x),..., \varphi_l(x),..., \varphi_h(x)$, $h>n$, occurs. As a functions, either Gaussians or other bell-shaped functions are used, for example,

$$\varphi_l(x)=\left(1+\frac{\|x-c_l\|^2}{\gamma_\varphi}\right)^{-1}=\frac{\gamma_\varphi}{\gamma_\varphi+\|x-c_l\|^2}$$

where $c_l-(\mathrm{n}\times1)$ - vector that sets the "center" of the radial-basis function $\varphi_l(x)$, γ_φ - a scalar parameter that determines the receptive field area, which is the same as "width" of this function.

Thus, when a vector signal enters to the system input $x(k)=\left(x_1(k),...x_i(k),...,x_n(k)\right)^T \in R^n$, at the output of the first RL hidden layer, a vector signal is formed: $\varphi(x(k))=(\varphi_1(x(k)),...,\varphi_l(x(k)),...,\varphi_h(x(k)))^T \in R^h$, $h>n$.

The second NL hidden layer realizes an elementary signal $\varphi(x(k))$ normalization operation in the form

$$\tilde{\varphi}(x(k))=\frac{\varphi(x(k))}{\|\varphi(x(k))\|}$$

which is needed for effective work of the third SL hidden layer. It was formed at the expense of (*M-1*) Kohonen's self-organizing maps *SOM^m*, each of which works under the assumption that in the data sample being processed, there are m classes. Clustering quality is provided by each *SOM^m* and is estimated using one or another validation index [2] in the fourth VL hidden layer. It calculates corresponding indices $VI^2, VI^3,..., VI^m,..., VI^M$ for every possible m=2,3,…,M.

Finally, in the output layer containing a single node, an optimum detector, the particular *SOM^m* is determined. It provides best clustering quality, wherein assumed that in the analyzing data array there are m clusters.

II. THE KERNEL CLUSTERING SYSTEM AND ITS SELF-LEARNING BASED ON NEURAL NETWORKS ENSEMBLE

Self-learning process of considered system is realized in the first hidden layer RL, where the centers $c_l, l=1,2,...,h$ of kernel functions $\varphi_l(x)$ are tuned. Also, it realized in a third hidden layer SL, where the synaptic weights w_j^m , $m=2,3,...,M$, $j=1,2...,m$ are estimated for each neural network of *SOM^m* ensemble.

Let's consider the tuning process for the centers of kernel functions, consisting the following steps [22]:

Step 0: set threshold value Δ that determines the indiscernibility level of two neighboring kernel functions.

After that, the maximum possible number h of these functions and receptive fields parameter γ_φ.

Step 1: when the first vector-observation $x(1)$ is fed to the system input, the first center c_1 and radial-basis function are being formed.

$$\varphi_1(x) = \frac{\gamma_\varphi}{\gamma_\varphi + \|x - c_1\|^2}$$

where $c_1 = x(1)$.

Step 2: when the second vector-observation $x(2)$ is fed to the system input, inequality has been checked

$$\|x(2) - c_1\| \le \Delta$$

and if it is satisfied, then $x(2)$ does not form a new center. And if the following condition is satisfied

$$\Delta < \|x(2) - c_1\| \le 2\Delta,$$

then c_1 is being corrected in accordance with the T. Kohonen's self-learning rule “Winner takes all” [10]:

$$c_1(2) = \mathrm{c}_1(1) + \eta(2)(x(2) - c_1(1))$$

where $c_1(1) = x(1)$, $0 < \eta(2) < 1$ is learning rate parameter. If the condition

$$2\Delta < \|x(2) - c_1\|$$

is satisfied then a new kernel function is formed

$$\varphi_2(x) = \frac{\gamma_\varphi}{\gamma_\varphi + \|x - c_2\|^2} = \frac{\gamma_\varphi}{\gamma_\varphi + \|x - x(2)\|^2}.$$

After every new observation *x(k)* this process is realized.

If on the step N h radial-basic functions are generated then in the future amount of them doesn't grow. Refinement centers c_l , $l = 1, 2, ..., h$ that were already generated can be provide only according to the condition (1) and the self-learning rule (2).

The adaptation process also consists of three steps [10]: competition, cooperation and synaptic adaptation for every SOM^m in ensemble, wherein synaptic weights vectors w_j^m describe h-dimensional centroids of formed clusters.

On the competition step input signal of second hidden layer NL $\tilde{\varphi}(x(k)) \in R^h$ is fed to every input of all SOM^m where they are compared with each of synaptic weights vectors $w_j^m(k-1)$ in the sense of distance

$$D(\tilde{\varphi}(x(k)), w_j^m(k-1)) = \|\tilde{\varphi}(x(k)) - w_j^m(k-1)\|, \qquad (3)$$

$$j = 1, 2, \dots \mathrm{m}\ ; \ m = 2, 3, \dots M \ .$$

Because $\|\tilde{\varphi}(x(k))\| = 1$, instead of the Euclidean metric (3) more easier is to use cosine similarity measure

$$sim(\tilde{\varphi}(x(k)), w_j^m(k-1)) = \tilde{\varphi}^T(x(k)) w_j^m(k-1)$$

by the help of which for every SOM^m its neuron-winners are determined, for that

$$\tilde{\varphi}^T(x(k)) w_j^{m*}(k-1) = \max_j \tilde{\varphi}^T(x(k)) w_j^m(k-1) \ .$$

On the cooperation step all neurons-winners of the ensemble generate topological neighborhoods areas, in which not only winners tuned, but and their nearest neighbors.

This area is described by the membership function $\varphi(j,l)$, that are similar to the radial-basis functions of the first hidden layer:

$$\varphi(j,l) = \frac{\gamma}{\gamma + \|w_l^m(k-1) - w_j^{m*}(k-1)\|^2} \ .$$

The synaptic centroids-weights specification of every SOM^m is occurs on the synaptic adaptation step by the T. Kohonen's self-learning rule “Winners takes more”:

$$w_l^m(k) = w_l^m(k-1) + \eta(k)\varphi(j,l)(\tilde{\varphi}(x(k)) - w_l^m(k-1)) \ . \qquad (4)$$

It's easy to see, that for winner w_j^{m*} (4) coincides with the learning rule (2). It has to be noted, that in the self-learning rule (4) learning rate parameters $\eta(k)$ and γ usually are selected according to the empirical considerations and must be decrease monotonically in the tuning process.

This process is easy to organize by the system of the recurrent relations

$$\begin{cases} \eta(k) = r^{-1}(k);\ r(k) = \alpha r(k-1) + \|\tilde{\varphi}(x(k))\|^2 = \alpha r(k-1) + 1, \\ \gamma(k) = \eta(k)\gamma(k-1),\ 0 < \alpha \le 1, \end{cases}$$

that at the $\alpha = 1$ automatically is transformed to the stochastic approximation procedure. It's easy to see too, that first and third layers of the system in fact are tuned according to the same type procedures like WTA and WTM [10].

III. Tuning of the fourth hidden layer

The estimation of the clustering quality is produced in the fourth hidden layer by the validation index VI^m [1], wherein this index is calculated for every of the T. Kohonen's maps SOM^m , $m = 2, 3, \dots M$.

As the such index it's useful to implement Davies-Bouldin criterion [23], with the help of which clustering quality can be estimated even in the case of non-convex classes.

In the case of the *m* clusters this index can be written in the form

$$DB(m)=$$

$$=\sum_{j=1}^{m}\max_{\substack{1\le q\le m\\ q\ne j}}\frac{\mathrm{s}(w_j^m(k),u_j(k),\tilde{\varphi}(x(k))-s(w_q^m(k),u_q(k),\tilde{\varphi}(x(k)))}{D(w_j^m(k),w_q^m(k))}$$

where $D(w_j^m(k),w_q^m(k))$ - distance between centroids

$$D(w_j^m(k),\mathrm{w}_q^m(k))=\left\|w_j^m(k)-w_q^m(k)\right\|,$$

$\mathrm{s}(w_j^m(k),u_j(k),\tilde{\varphi}(x(k)))-$ the intracluster scattering characteristics for *j*-th cluster:

$$\mathrm{s}(w_j^m(k),u_j(k),\tilde{\varphi}(x(\mathrm{k})))=\left(\frac{\sum_{k=1}^{N}u_j(k)\left\|\tilde{\varphi}(x(k))-\mathrm{w}_j^m(k)\right\|^2}{\sum_{k=1}^{N}u_j(k)}\right)^{\frac{1}{2}},$$

$u_j(k)$ -crisp membership function of the vector $\tilde{\varphi}(x(k))$ to the *j*-th cluster type:

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

As the optimal number of clusters m^* value, providing minimum of the *DB(m)*, is selected:

$$DB(m^*)=\min_{m}\left\{DB(2),DB(3),...,DB(M)\right\},$$

that is calculated in the output layer.

In the situation then non-stationary data are processed in online mode, is necessary to modify *DB(m)* index for processing data on the “sliding-window” mode of dimension $1<s<N$. Wherein only intercluster distance characteristics, that are calculated on the “sliding-window”, are modified by expression

$$\mathrm{s}(w_j^m(k),\mathrm{u}_j(k),\tilde{\varphi}(x(k)),\mathrm{s})=\left(\frac{\sum_{\tau=k-s+1}^{k}u_j(\tau)\left\|\tilde{\varphi}(x(\tau))-w_j^m(k)\right\|^2}{\sum_{\tau=k-s+1}^{k}u_j(\tau)}\right)^{\frac{1}{2}}$$

when the data volume *N* isn’t limited and grows with time $k=1,2,...,\mathrm{N},\mathrm{N}+1,...$

IV. Experimental Results

We have tested proposed method with two different training data sets. The first data set is artificial generated so that it contains 3 clusters, 300 observations were every observation has 3 features. The second data set “Iris” is taken from UCI-Repository [24]. This data set consists of 150 observations that are divided into 3 classes where every observation has 3 random features. The clusters are clearly visible in the artificial generated data set and shown in Figure 2.

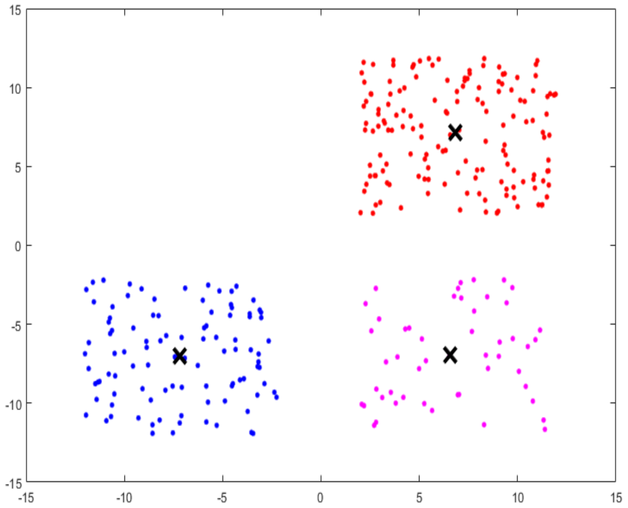


Fig. 2. The artificial generated data set

The computational accuracy of proposed method was compared with known K-means algorithm. These clustering results were estimated by the Davies-Bouldin criterion. The clustering accuracies for a series of 50 experiments are shown in Table I and Table II.

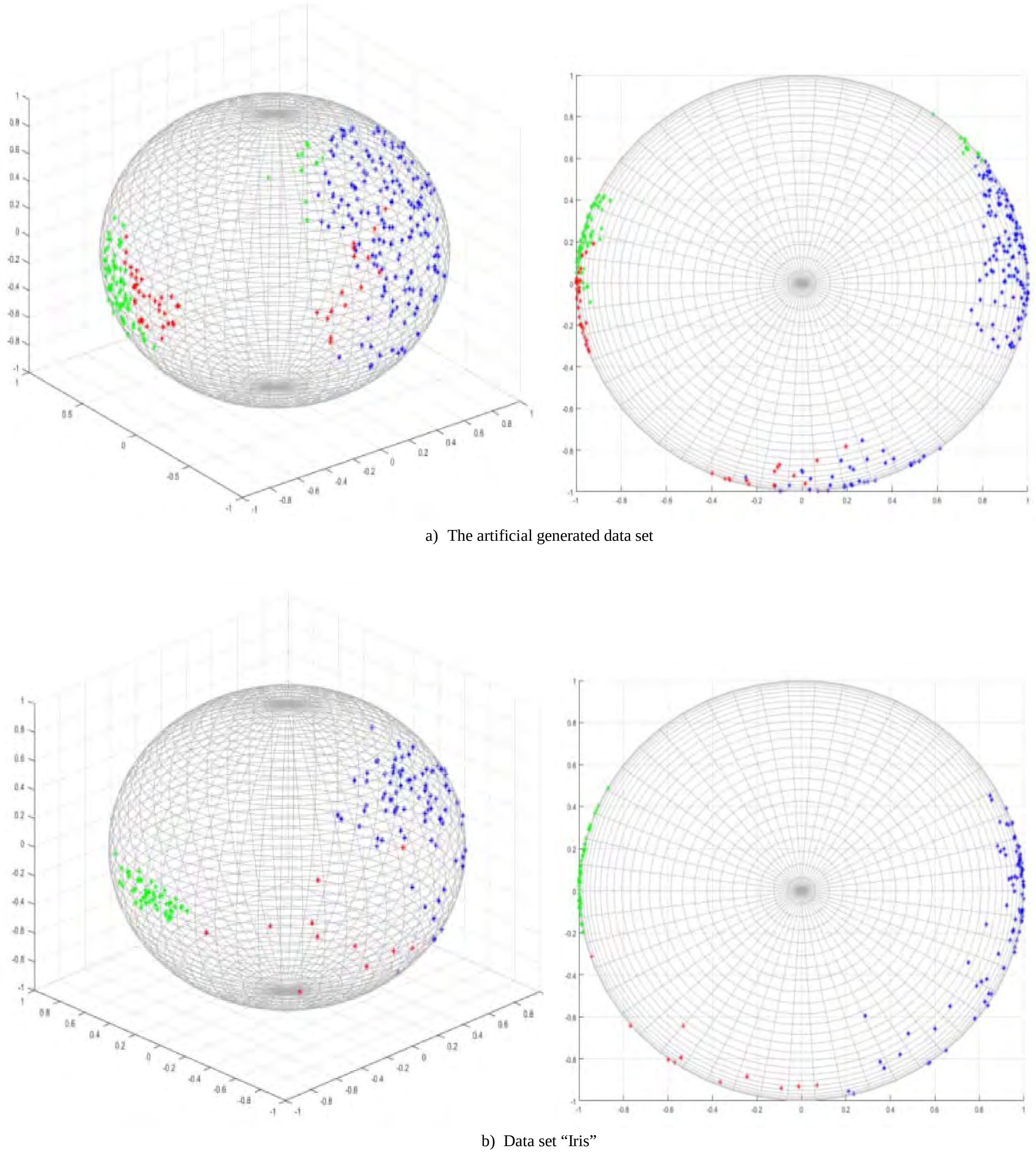
TABLE I. The mean clustering accuracies for the different numbers of clusters (The artificial generated data set)

Method	SOM^m	k-means
clustering accuracies for 2 clusters	0,71	0,70
clustering accuracies for 3 clusters	**0,89**	0,76
clustering accuracies for 4 clusters	0,68	0,67

TABLE II. The mean clustering accuracies for the different numbers of clusters (Iris)

Method	SOM^m	k-means
clustering accuracies for 2 clusters	0,84	0,83
clustering accuracies for 3 clusters	**0,91**	0,87
clustering accuracies for 4 clusters	0,72	0,73

For visualization, taken data sets were projected by the PCA (principal component analysis) method to three principal components. Visualization results of the proposed ensemble are shown in (Fig. 3).



a) The artificial generated data set

b) Data set “Iris”

Fig. 3. Visualization results of the proposed ensemble

V. Conclusion

The neural network approach for data stream clustering task, that in online mode are fed to processing in assumption that, neither the number of clusters nor their shape are known, is proposed in the paper. The main idea of this approach is based on the kernel clustering and neural networks ensembles, that consist of the T. Kohonen’s self-organizing maps.

The proposed system is characterized by the simplicity of numerical implementation, high speed, and can be used for solving different tasks of processing data streams in conditions of apriori uncertainty of their properties.

References

[1] G. Gan, Ch. Ma and J. Wu, Data Clustering: Theory, Algorithms and Applications. Philadelphia: SIAM, 2007.

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

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

[4] D. Pelleg, and A. Moor, “X-means: extending K-means with efficient estimation of the number of clusters,” 17th Int. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, pp.727-730, 2000.

[5] T. Ishioka, “An expansion of X-means for automatically determining the optimal number of clusters,” 4th IASTED Int. Conf. Computational Intelligence, Calgary, Alberta, pp. 91-96, 2005.

[6] L. Rutkowski, Computational Intelligence. Methods and Techniques. Berlin-Heidelberg: Springer-Verlag, 2008.

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

[8] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher and P. Held, Computational Intelligence. A Methodological Introduction. Berlin: Springer, 2013.

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

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

[11] A. Strehl, J. Ghosh, “Cluster ensembles – A knowledge reuse framework for combining multiple partitions,” Journal of Machine Learning Research, pp. 583-617, 2002.

[12] A. Topchy, A. K. Jain, and W. Punch, “Clustering ensembles: models of consensus and weak partitions,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1866-1881, 2005.

[13] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, “To improve the quality of cluster ensembles by selecting a subset of base clusters”, Journal of Experimental & Theoretical Artificial Intelligence, pp. 127-150, 2013.

[14] M. Charkhabi, T. Dhot, and S. A. Mojarad, “Cluster ensembles, majority vote, voter eligibility and privileged voters”, Int. Journal of Machine Learning and Computing, vol. 4, no. 3, pp. 275-278, 2014.

[15] Ye. V. Bodyanskiy, A. A. Deineko, P. Ye. Zhernova, and V. O. Riepin, “Adaptive modification of X-means method based on the ensemble of the T. Kohonen’s clustering neural networks,” VI Int. Sci. Conf. “Information Managements Systems and Technologies”, Odessa, Ukrane, pp. 202-204, 2017.

[16] J. C. Bezdek, J. Keller, 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.

[17] T. M. Cover, “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,” IEEE Trans. on Electronic Computers, no. 14, pp. 326-334, 1965.

[18] M. Girolami, “Mercer kernel-based clustering in feature space”, IEEE Trans. on Neural Networks, vol. 13, no. 3, pp. 780-784, 2002.

[19] D. MacDonald and C. Fyfe, “Clustering in data space and feature space,” ESANN'2002 Proc. European Symp. on Artificial Neural Networks, Bruges (Belgium), pp. 137-142, 2002.

[20] M. Girolami, “Mercer kernel-based clustering in feature space,” IEEE Trans. on Neural Networks, vol. 13, no. 3, pp. 780-784, 2002.

[21] F. Camastra, and A. Verri, “A novel kernel method for clustering,” IEEE Trans. on Pattern Analysis and Machine Intelligence, no. 5, pp. 801-805, 2005.

[22] Ye. V. Bodyanskiy, A. A. 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.

[23] D. L. Davies, and D. W. Bouldin, “A Cluster Separation Measure”, IEEE Transactions on Pattern Analysis and Machine Intelligence. No. 4, pp. 224-227, 1979.

[24] P. M. Murphy, and D. Aha, UCI Repository of machine learning databases. URL: http://www.ics.uci.edu/mlearn/MLRepository.html. Department of Information and Computer Science