Hybrid Intelligent Systems

ADAPTIVE GUSTAFSON-KESSEL FUZZY CLUSTERING ALGORITHM BASED ON SELF-LEARNING SPIKING NEURAL NETWORK

Yevgeniy Bodyanskiy, Artem Dolotov, Iryna Pliss

Abstract: The Gustafson-Kessel fuzzy clustering algorithm is capable of detecting hyperellipsoidal clusters of different sizes and orientations by adjusting the covariance matrix of data, thus overcoming the drawbacks of conventional fuzzy c-means algorithm. In this paper, an adaptive version of the Gustafson-Kessel algorithm is proposed. The way to adjust the covariance matrix iteratively is introduced by applying the Sherman-Morrison matrix inversion procedure. The adaptive fuzzy clustering algorithm is implemented on the base of self-learning spiking neural network known as a realistic analog of biological neural systems that can perform fast data processing. Therefore, the proposed fuzzy spiking neural network that belongs to a new type of hybrid intelligent systems makes it possible both to perform fuzzy clustering tasks efficiently and to reduce data processing time considerably.

Keywords: computational intelligence, hybrid intelligent system, fuzzy clustering, adaptive Gustafson-Kessel algorithm, self-learning spiking neural network, spiking neuron center, the temporal Hebbian learning.

ACM Classification Keywords: I.2.6 [Artificial Intelligence]: Learning – Connectionism and neural nets; I.5.1 [Pattern Recognition]: Models – Fuzzy set, Neural nets; I.5.3 [Pattern Recognition]: Clustering – Algorithms.

Introduction

Computational intelligence aims to achieve biologically and naturally plausible computing. Data clustering methods as a part of computational intelligent means has been successfully evolved in these directions over the several last decades.

On the one hand, data clustering methods progressed considerably toward more realistic approaches of clusters detection and separation. Unnatural ‘all or none’ membership restriction of hard clustering was overcome by fuzzy probabilistic clustering [Bezdek, 1981]. High degree of membership that fuzzy probabilistic clustering unnaturally produced for outliers that were equidistant from clusters centers was avoided by applying fuzzy possibilistic approach [Bezdek, 2005]. Adaptive versions of the mentioned clustering methods were proposed allowing of data processing in on-line mode [Park, 1994; Bodyanskiy, 2005]. Finally, the Gustafson-Kessel fuzzy clustering algorithm made it possible to detect clusters of different sizes and orientations [Gustafson, 1979; Krishnapuram, 1999].

On the other hand, data clustering means enriched by biologically plausible methods of data processing, namely, self-learning spiking neural networks [Natschlaeger, 1998; Bohte, 2002; Gerstner, 2002]. Being highly realistic models of biological neural systems, self-learning spiking neural networks appeared to be a considerably powerful and fast clustering tool of computational intelligence. Moreover, new type of hybrid intelligent systems that combined capabilities of self-learning spiking neural networks and fuzzy probabilistic and possibilistic clustering approaches was shown to be successfully applied in various data clustering problems [Bodyanskiy, 2008a-f; Avdiyenko, 2008].

In this paper, adaptive version of the Gustafson-Kessel fuzzy clustering algorithms is introduced. It is shown how the proposed algorithm can be implemented on the base of self-learning spiking neural network.

Adaptive Version of the Gustafson-Kessel Fuzzy Clustering Algorithm

In the general case, the clustering quality criterion can be stated as following:

$$E(\mu_j(x(k)), v_j) = \sum_{k=1}^{N}\sum_{j=1}^{m} \mu_j^{\zeta}(x(k)) \left\| x(k) - v_j \right\|_A^2, \; k = 1, 2, ..., N, ...; \; j = 1, 2, ..., m \,, \tag{1}$$

where $\mu_j(x(k)) \in [0,1]$ is membership level of the input pattern $x(k)$ to the j-th cluster, v_j is the center of the j-th cluster, N is the number of input patterns, m is the number of clusters, $\zeta \geq 0$ is the fuzzifier that determines boundary between clusters and controls the amount of fuzziness in the final partition, A is a norm matrix that defines distance measure. Under restriction

$$\sum_{j=1}^{m} \mu_j(x(k)) = 1 \,, \tag{2}$$

minimization of (1) by applying the method of indefinite Lagrange multipliers leads us to the following solution:

$$\mu_j(x(k)) = \frac{\left(\left\| x(k) - v_j \right\|_A^2 \right)^{\frac{1}{1-\zeta}}}{\sum_{\iota=1}^{m} \left(\left\| x(k) - v_\iota \right\|_A^2 \right)^{\frac{1}{1-\zeta}}} \,, \tag{3}$$

$$v_j = \frac{\sum_{k=1}^{N} \mu_j^{\zeta}(x(k)) x(k)}{\sum_{k=1}^{N} \mu_j^{\zeta}(x(k))} \,. \tag{4}$$

In the case when norm matrix A is the identity matrix and $\zeta = 2$, equations (3)-(4) present conventional fuzzy c-means algorithm. The Gustafson-Kessel algorithm uses the Mahalanobis distance measure, i.e. the inverse covariance matrix Σ_j^{-1} for the j-th cluster is used as a norm matrix in (3).

Minimization of (1) under restriction (2) based of the Arrow-Hurwicz-Uzawa gradient method [Arrow, 1958] produces the following recursive solution for the Mahalanobis distance measure:

$$\mu_j(x(k+1)) = \frac{\left((x(k+1) - v_j(k+1))^T \Sigma_j^{-1}(k+1)(x(k+1) - v_j(k+1)) \right)^{\frac{1}{1-\zeta}}}{\sum_{\iota=1}^{m} \left((x(k+1) - v_\iota(k+1))^T \Sigma_\iota^{-1}(k+1)(x(k+1) - v_\iota(k+1)) \right)^{\frac{1}{1-\zeta}}} \,, \tag{5}$$

$$v_j(k+1) = v_j(k) + \eta_v(k) \mu_j^{\zeta}(x(k)) \Sigma_j^{-1}(k+1)(x(k+1) - v_j(k)) \,, \tag{6}$$

where $\eta_v(k) > 0$ is a learning rate that defines to what extent information retrieved from new patterns will override old information .

Let us obtain recursive expression for the inverse covariance matrix. The covariance matrix for k input patterns is

$$\Sigma(k)=\frac{1}{k}\sum_{r=1}^{k}\left(x(r)-\bar{x}(k)\right)\left(x(r)-\bar{x}(k)\right)^{T}, \tag{7}$$

where $\bar{x}(k)$ is the average over k patterns. For a new incoming pattern $x(k+1)$, the covariance matrix can be written in form of

$$\Sigma(k+1)=\frac{1}{k+1}\left(k\Sigma(k)+\left(x(k+1)-\bar{x}(k+1)\right)\left(x(k+1)-\bar{x}(k+1)\right)^{T}\right). \tag{8}$$

By applying the Sherman-Morrison matrix inverse procedure

$$\left(B+ab^{T}\right)^{-1}=B^{-1}-\frac{B^{-1}ab^{T}B^{-1}}{1+b^{T}B^{-1}a} \tag{9}$$

to (8), the inverse covariance matrix can be expressed in recursive form:

$$\Sigma^{-1}(k+1)=\frac{k+1}{k}\left(\Sigma^{-1}(k)-\frac{\Sigma^{-1}(k)\left(x(k+1)-\bar{x}(k+1)\right)\left(x(k+1)-\bar{x}(k+1)\right)^{T}\Sigma^{-1}(k)}{k+\left(x(k+1)-\bar{x}(k+1)\right)^{T}\Sigma^{-1}(k)\left(x(k+1)-\bar{x}(k+1)\right)}\right). \tag{10}$$

Finally, the recursive average upon incoming pattern $x(k+1)$ is

$$\bar{x}(k+1)=\bar{x}(k)+\frac{1}{k+1}\left(x(k+1)-\bar{x}(k)\right). \tag{11}$$

In this paper, it is proposed to use centers of spiking neurons that are adjusted on each step of self-learning spiking neural network learning. This way, clusters centers adjustment (6) becomes unnecessary.

Self-Learning Spiking Neural Network

Architecture of self-learning spiking neural network is depicted on Figure 1. As illustrated, it is a heterogeneous two-layered feed-forward neural network with lateral connections in the second hidden layer [Bohte, 2002].

The first hidden layer performs pulse-position transformation of $n\times 1$-dimensional input patterns $x(k)$ (here, $k=0,1,\ldots,N$ is a pattern number) to the input vector of spikes $t^{[0]}(x(k))$ where each spike is defined by its firing time. The transformation is performed by population coding that implies that an input $x_i(k)$, $i=1,2,\ldots,n$, is processed at the same time by a pool of h receptive neurons RN_{li}, $l=1,2,\ldots,h$.

As a rule, a receptive neuron activation function is bell-shaped (Gaussian usually), and activation functions of the neurons within a pool are shifted, overlapped, and of different widths. Generally firing time of a spike emitted by receptive neuron lies in a certain interval $[0,t^{[0]}_{max}]$ referred to as a coding interval and is defined by the expression

$$t^{[0]}_{li}(x_i(k))=\left\lfloor t^{[0]}_{max}\left(1-\psi\left(\left|x_i(k)-c^{[0]}_{li}\right|,\sigma_i\right)\right)\right\rfloor, \tag{12}$$

where $\lfloor\bullet\rfloor$ is the floor function, $\psi(\bullet,\bullet)$, $c^{[0]}_{li}$, and σ_i are the receptive neuron's activation function, center, and width respectively.

It is worth to note that a pool of receptive neurons RN_{li}, $l=1,2,\ldots,h$ can be treated as a certain linguistic variable of input data and each receptive neuron (more precisely, fuzzy receptive neuron) within the pool – as its linguistic term with membership function $\psi_{li}\left(x_i(k)\right)$ [Bodyanskiy, 2008d]. Having any a priori knowledge of data structure, it is possible to adjust the first layer neurons activation functions as membership functions to fit the knowledge and thus, to get better clustering results.

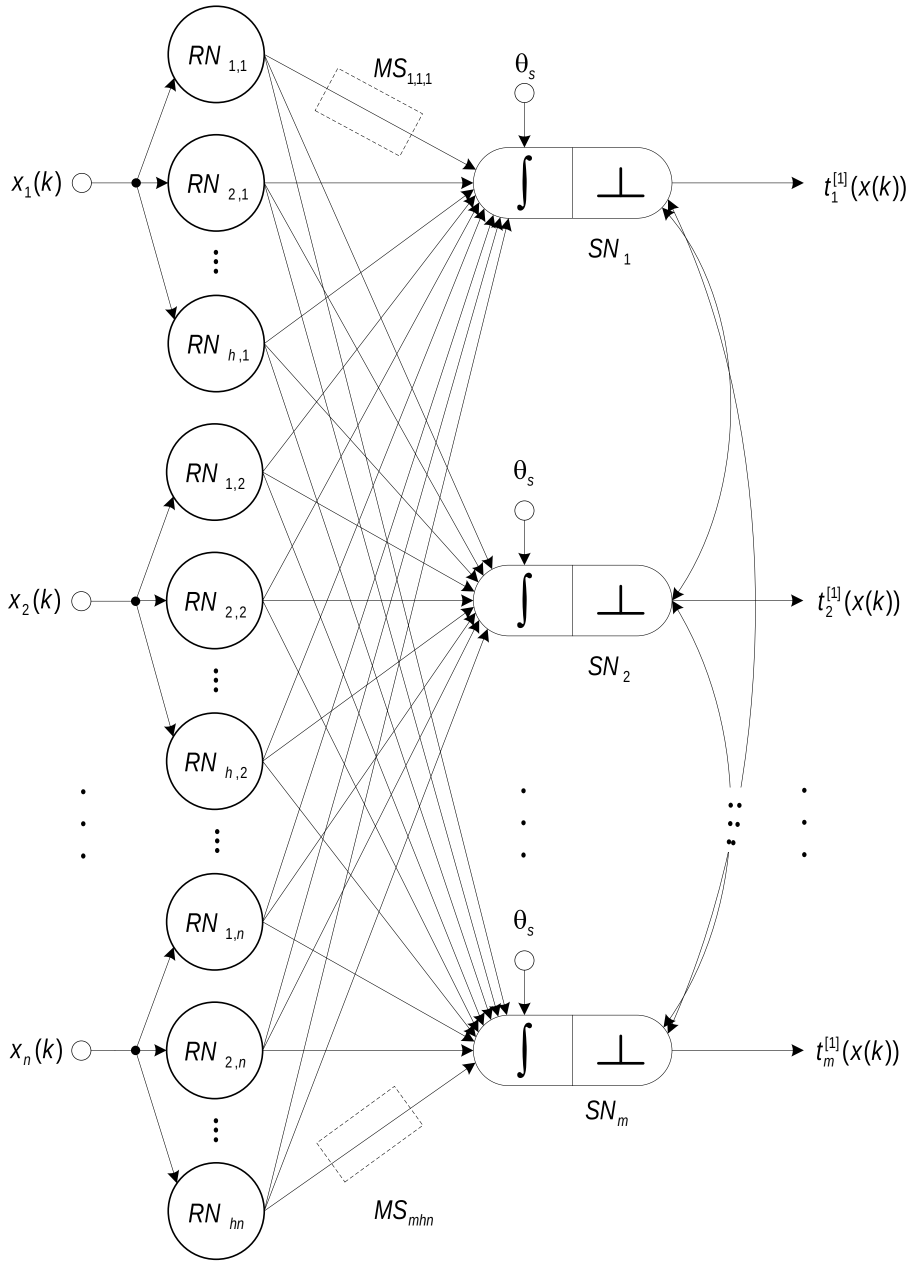

Figure 1. Self-learning spiking neural network architecture

The second hidden layer consists of m spiking neurons SN_j , $j = 1, 2, \dots, m$ (m – number of clusters). They are connected with neurons of the previous layer by multiple synapses. As shown on Figure 2, a multiple synapse MS_{jli} consists of a set of q subsynapses with different time delays d^p , $d^p - d^{p-1} > 0$, $d^q - d^1 > t^{[0]}_{\max}$, and varying weights w^p_{jli} (here $p = 1, 2, \dots, q$). It should be noted that number of subsynapses within a multiple synapse are fixed for the whole network. Having a spike $t^{[0]}_{li}(x_i(k))$ from the li -th receptive neuron, the p -th subsynapse of the j -th spiking neuron produces a delayed postsynaptic potential

$$u^p_{jli}(t) = w^p_{jli}\varepsilon^p_{jli}(t) = w^p_{jli}\varepsilon(t - (t^{[0]}_{li}(x_i(k)) + d^p)) , \qquad (13)$$

where $\varepsilon(\bullet)$ is a spike-response function usually described by the expression

$$\varepsilon(t)=\frac{t}{\tau}e^{1-\frac{t}{\tau}}H(t)\,, \quad (14)$$

τ is the membrane potential decay time constant whose value can be obtained empirically, $H(\bullet)$ is the Heaviside step function. Output of the multiple synapse MS_{jli} forms total postsynaptic potential

$$u_{jli}(t)=\sum_{p=1}^{q}u_{jli}^{p}(t)\,. \quad (15)$$

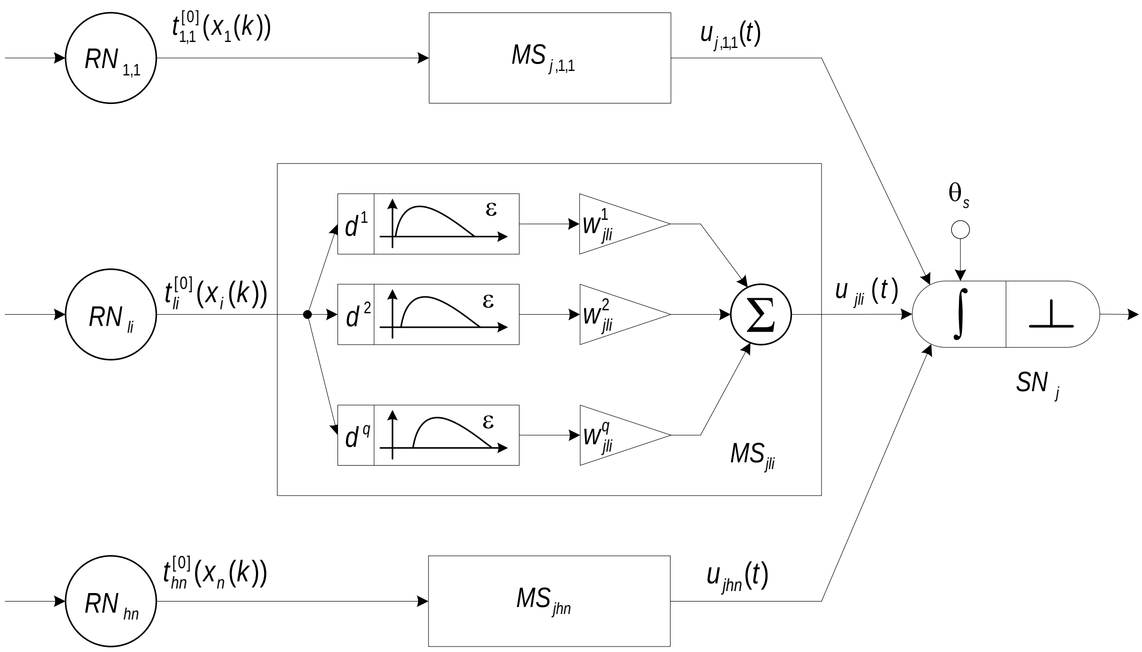

Figure 2. Multiple synapse

Each incoming total postsynaptic potential contributes to membrane potential of spiking neuron SN_j as follows:

$$u_j(t)=\sum_{i=1}^{n}\sum_{l=1}^{h}u_{jli}(t)\,. \quad (16)$$

Spiking neuron SN_j generates at most one outgoing spike $t_j^{[1]}(x(k))$ during a simulation interval (the presentation of an input pattern $x(k)$), and fires at the instant the membrane potential reaches firing threshold θ_s. After neuron firing, the membrane potential is reset to the rest potential (0 usually) until the next input pattern is presented.

The spiking neuron described above corresponds to the integrate-and-fire model that is the one of the well-known models of a biological neuron [Gerstner, 2002].

Spiking neurons are linked with lateral inhibitory connections that disable all other neurons to fire after the first one has fired. In other words, lateral connections leads spiking neurons to fire according to the ‘winner-takes-all’ rule. After learning stage, each spiking neuron firing time reflects the distance of the input pattern to the neuron center. Thus, pattern supplied to the learned spiking neural network fires the spiking neuron whose center is the closest to it. This way self-learning spiking neural network performs clusters separation.

The purpose of the learning algorithm of self-learning spiking neural network is to move the center of the neuron-winner closer to input pattern by adjusting the neuron synaptic weights. The weights are adjusted according to the temporal Hebbian rule in the following way: weights of those subsynapses which contributed to the neuron’s firing are strengthened, whereas weights of subsynapses which did not contribute are weakened. As a rule, the temporal Hebbian rule used in spiking neural networks learning has the following form [Berredo, 2005]:

$$w_{jli}^{p}(K+1)=w_{jli}^{p}(K)+\eta_w(K)\left((1+\beta)\exp\left(\frac{\left(t_{li}^{[0]}(x_i(k))+d^p-t_j^{[1]}(x(k))-\alpha\right)^2}{2(1-\kappa)}\right)-\beta\right), \quad (17)$$

$$\kappa=1-\frac{\nu^2}{2\ln\left(\frac{\beta}{1+\beta}\right)}, \quad (18)$$

where K is the epoch number, $\eta_w(\bullet)$ is the learning rate, α, β, ν are shape parameters of the learning function.

One can readily see that the unsupervised learning of the spiking neurons layer is identical to the one of self-organizing maps (SOM).

Spiking Neuron Center

A spiking neuron center is considered to represent a spiking neuron response to the input pattern in a convenient way. In the general case, it is considered to possess the following property: the closer input pattern is to the neuron's center, the earlier output spike fires. Apparently, the earliest time of spiking neuron response can appears when all postsynaptic potentials reach the neuron soma simultaneously, i. e. the firing time of outgoing spike depends on synchronization degree of incoming spikes. It is worth to note here that synchronization phenomena is of primary importance in nature [Pikovsky, 2001].

In this work, spiking neuron center calculation is required to obtain the difference of incoming pattern and cluster center (represented by the neuron center) $x(k+1)-v_j(k+1)$ that is used in (5). Since the neuron center represents temporal aspect of the network functioning, we pursue the temporal difference of the input vector of spikes and the center $t^{[0]}(x(k))-c_j^{[1]}$ ($c_j^{[1]}$ is the center of spiking neuron SN_j). Apparently, original form of pattern $x(k)$ should be replaced with time-pulsed form $t^{[0]}(x(k))$ in (5), (10), and (11) in this case.

There exist several ways to define spiking neuron center formally. The ones that were successfully used in fuzzy clustering are the Natschlaeger-Ruf center [Natschlaeger, 1998, Bodyanskiy, 2008c] and the Goren center [Goren, 2001, Avdiyenko, 2008]. Both of them requires calculation of the mean weighted synaptic delay d_{jli}. In this paper, it is proposed to calculate it in a general way, by using the quasi-arithmetic mean [Kolmogorov, 1985]:

$$d_{jli}=f^{-1}\left(\frac{\sum_{p=1}^{q}w_{jli}^{p}f(d^p)}{\sum_{p=1}^{q}w_{jli}^{p}}\right), \quad (19)$$

where $f(\bullet)$ is a strictly monotonic function.

T. Natschlaeger and B. Ruf proposed to calculate the center of spiking neuron SN_j as follows:

$$c_{jli}^{[1]}=d_{jli}-\min\{d_{jab}\,|\,1\le a\le h, 1\le b\le n\}, \quad (20)$$

where $c_{jli}^{[1]}$ is a component of vector $c_j^{[1]}=(c_{j,1,1}^{[1]},c_{j,1,2}^{[1]},\ldots,c_{jh,1}^{[1]},c_{jh,s}^{[1]},\ldots,c_{jhn}^{[1]})^T$. Transforming the input vector of spikes as follows:

$$\rho_{li}(x_i(k)) = \max\{t_{ab}^{[0]}(x_b(k)) \mid 1 \le a \le h, 1 \le b \le n\} - t_{li}^{[0]}(x_i(k)), \quad (21)$$

the difference in (5) will be

$$x(k+1) - v_j(k+1) = \rho(x(k)) - c_j^{[1]}. \quad (22)$$

Goren center does not require the input vector of spikes transformation:

$$c_{jli}^{[1]} = \max\{d_{jab} \mid 1 \le a \le h, 1 \le b \le n\} - d_{jli}, \quad (23)$$

$$x(k+1) - v_j(k+1) = t^{[0]}(x(k)) - c_j^{[1]}. \quad (24)$$

It is notable that the Natschlaeger-Ruf center takes into account only the synchronization degree, whereas the Goren centers incorporates also temporal distance of spikes and the origin of coding interval. Anyway, even the Goren center does not reflect all peculiarities of spiking neuron firing. For example, it does not consider how value of firing threshold θ_s influences on the neuron firing time. Pursuing a center definition that would describe spiking neuron firing more precisely is a purpose of the further work.

In summary, the whole learning process of the proposed hybrid system can be formulated. Upon supplying a new pattern, self-learning spiking neural network adjusts the neurons centers using learning algorithm (17), (18) (as a rule, a few epochs are sufficient). Then a new fuzzy partitioning is performed according to (5), (10), and (11) using either Natschlaeger-Ruf (20)-(22) or Goren centers (23),(24). The simulation experiments show good results of the proposed hybrid system using in various data clustering problems solving.

Conclusion

The adaptive version of the Gustafson-Kessel fuzzy clustering algorithm is proposed. The new hybrid system is designed on the based of self-learning spiking neural network. It utilizes temporal difference of incoming vector of spikes and spiking neurons centers to perform fuzzy partitioning. The obtained hybrid system allows of combining tractability of the Gustafson-Kessel algorithm and velocity capabilities of spiking neural network and processes data in on-line mode.

Bibliography

[Arrow, 1958] K.J.Arrow, L.Hurwitz, H.Uzava. Studies in Linear and Nonlinear Programming. Stanford University Press, Stanford, CA, 1958.

[Avdiyenko, 2008] L.Avdiyenko, A.Dolotov, I.Pliss. Self-learning spiking neural network with synapse fuzzy inference based on Goren centers. Proc. Int. Sci. Conf. “Intellectual Systems for Decision Making and the Problems of Computational Intelligence – 2008” (May 19-23, 2008, Yevpatoria, Ukraine), Kherson National Technical University, Vol. 3, Part 1, Kherson, 2008, P. 11-14 (in Russian).

[Berredo, 2005] R.C.De Berredo. A review of spiking neuron models and applications. M. Sc. Dissertation, University of Minas Gerais, 2005.

[Bezdek, 1981] J.C.Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[Bezdek, 2005] J.C.Bezdek, J.Keller, R.Krishnapuram, N.R.Pal. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer, New York, 2005.

[Bodyanskiy, 2005] Ye.Bodyanskiy. Computational intelligence techniques for data analysis. Lecture Notes in Informatics, Vol. 72, 2005, P. 15-36.

[Bodyanskiy, 2008a] Ye.Bodyanskiy, A.Dolotov. A self-learning spiking neural network for fuzzy clustering task. Sci. Proc. of Riga Technical University: Information Technology and Management Science, 36, 2008, P. 27-33.

[Bodyanskiy, 2008b] Ye.Bodyanskiy, A.Dolotov. Image processing using self-learning fuzzy spiking neural network in the presence of overlapping classes. Proc. 11th Int. Biennial Baltic Electronics Conf. "BEC 2008" (October 6-8, 2008, Tallinn, Estonia),Tallinn University of Technology, Tallinn, 2008, P. 213-216.

[Bodyanskiy, 2008c] Ye.Bodyanskiy, A.Dolotov. Self-learning spiking neural network with synapse fuzzy inference based on Natschlaeger-Ruf centers. Proc. 10th Int. Sci. and Tech. Conf. “System Analysis and Information Technologies” (May 20-24, 2008, Kyiv, Ukraine), National Technical University of Ukraine “KPI”, Kiyv, 2008, P. 173 (in Ukrainian).

[Bodyanskiy, 2008d] Ye.Bodyanskiy, A.Dolotov, I.Pliss. Fuzzy receptive neurons using in self-learning spiking neural network. Proc. Int. Sci. and Tech. Conf. "Automation: Problems, Ideas, Solutions" (September 8-12, 2008, Sevastopol, Ukraine), Sevastopol National Technical University, Sevastopol, 2008, P. 12-14 (in Russian).

[Bodyanskiy, 2008f] Ye.Bodyanskiy, A.Dolotov, I.Pliss, Ye.Viktorov. Fuzzy possibilistic clustering using self-learning spiking neural network. Wissenschaftliche Berichte der Hochschule Zittau/Goerlitz, Heft 100, Nr. 2360-2395, 2008, S. 53-60.

[Bohte, 2002] S.M.Bohte, J.S.Kok, H.La Poutre. Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks. IEEE Trans. on Neural Networks, 13, 2002, P. 426-435.

[Gerstner, 2002] W.Gerstner, W.M.Kistler. Spiking Neuron Models. The Cambridge University Press, Cambridge, 2002.

[Goren, 2001] Ya.Goren. Clustering with spiking neurons. 2001 http://cis.cs.technion.ac.il/Done_Projects/Projects_done/SpikingNeurons/RBF.doc.

[Gustafson, 1979] D.E.Gustafson, W.C.Kessel. Fuzzy clustering with a fuzzy covariance matrix. Proc. IEEE Conf. Decision Contr. San Diego, CA, 1979, P. 761-766.

[Kolmogorov, 1985] A.N.Kolmogorov. Mathematics and Mechanics. Nauka, Moscow, 1985 (in Russian).

[Krishnapuram, 1999] R. Krishnapuram, J.Kim. A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms. IEEE Trans. on Fuzzy Systems, Vol. 7, 4, 1999, P. 453-461.

[Natschlaeger, 1998] T.Natschlaeger, B.Ruf. Spatial and temporal pattern analysis via spiking neurons. Network: Computations in Neural Systems, 9, 1998, P. 319-332.

[Park, 1994] D.C.Park, I.Dagher. Gradient based fuzzy c-means (GBFCM) algorithm. Proc. IEEE Int. Conf. on Neural Networks. IEEE Press, Orlando, FL, 1994, P. 1626-1631.

[Pikovsky, 2001] A.Pikovsky, M.Rosenblum, J.Kurths. Synchronization: A Universal Concept in Nonlinear Sciences. The Cambridge University Press, New York, 2001.

Authors' Information

Yevgeniy Bodyanskiy *– Professor, Dr.-Ing. habil., Scientific Head of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics; 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: bodya@kture.kharkov.ua.*

Artem Dolotov *– Ph. D. student, Junior Researcher of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics; 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: dolotov.artem@gmail.com.*

Iryna Pliss *– Ph. D., Leading Researcher of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics; 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: pliss@kture.kharkov.ua.*