Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 9, September 2015, pg.270 – 275

RESEARCH ARTICLE

An Evolving Cascade Neuro-Fuzzy System for Data Stream Fuzzy Clustering

Yevgeniy Bodyanskiy[1], Oleksii Tyshchenko[2], Daria Kopaliani[3]

[1]Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Ukraine
[2]Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Ukraine
[3]Kharkiv National University of Radio Electronics, Ukraine
[1] yevgeniy.bodyanskiy@nure.ua; [2] lehatish@gmail.com; [3] daria.kopaliani@gmail.com

Abstract—An evolving cascade neuro-fuzzy system and its online learning procedure are proposed in this paper. The system is based on nodes of a special type. A quality estimation process is defined by finding an optimal value of the used cluster validity index.

Keywords— Evolving cascade system, neuro-fuzzy network, data stream, fuzzy clustering.

I. INTRODUCTION

There is an acute problem of online methods' development for data processing under conditions of significant uncertainty about data streams' properties. Another important point is the data are usually nonstationary, nonlinear, random and fuzzy; furthermore, there's no information about clusters' number and type which are formed with the data. Hybrid systems of computational intelligence may be an effective solution for this kind of problems, but most of the well-known systems which are widely used in these tasks are focused mostly on batch processing with a certain predefined number of classes. It seems appropriate to develop adaptive systems of computational intelligence that can adjust both their parameters and their structure.

New high-speed adaptive methods of computational intelligence based on the neuro-fuzzy technology should be proposed to deal with online tasks under conditions of the current uncertainty. They should be able to estimate a previously unknown number of classes/clusters which may change during data processing. These adaptive methods should be effective when a class/cluster form has an arbitrary shape and when there's a high level of class/cluster intersection.

There's a wide class of Dynamic Data Mining and Data Streams Mining tasks, when data come in the form of a sequential stream in an online mode. It's clear that a sample volume N in this case is not limited, and it acquires the meaning of the current discrete time.

The necessity of solving the clustering tasks in real time (as data streams come to a system) has led to self-learning neuro-fuzzy systems, which are hybrid systems consisting of the Kohonen self-organizing map (SOM) and the fuzzy c-means algorithm by Bezdek (FCM) [1-11].

Evolving systems of computational intelligence can help determine an amount of clusters in a data stream [12-26]. The most popular evolving systems are DENFIS [13], EFuNN [12, 14, 18], eTS [19-21], FLEXFIS [21-23], SAFIS [24], SOFNN [7], SONFIN [25], PANFIS [17] and others.

II. AN EVOLVING NEURO-FUZZY SYSTEM'S ARCHITECTURE

We consider a rather simple case of an evolving cascade neuro-fuzzy system in this paper: there's only one node in each cascade of a pool although it's possible to have a set of nodes [27, 28]. There's an architecture of the proposed system in Fig.1.

Data are fed sequentially to the system's input layer (a zero layer) in the form of a vector signal $x(k)=\left(x_1(k),x_2(k),...,x_n(k)\right)^T$, $k=1,2,\ldots,N,N+1,\ldots$ is an index of the current discrete time. These signals are fed to all system's nodes $N^{[m]}$ ($m=1,2,\ldots$ is a cascade's number). The node of each cascade is designated for online data stream clustering and it differs from neighbour nodes either by a used self-learning algorithm or (if the same clustering method is used) by the algorithm's parameters. It is assumed that an amount of clusters equals to $(m+1)$ for each cascade, which means that the first cascade splits data into two classes, the second cascade – into three classes and so on till the required clustering quality is achieved like hierarchical divisive clustering.

Then a system's element $XB^{[m]}$ estimates the general clustering quality in the pool taking into consideration the accepted amount of classes which equals to $(m+1)$. Thus, the system solves a clustering task of non-stationary data stream under uncertainty conditions (an amount of clusters as well as their shape and level of mutual overlapping).

When the system achieves some predefined number of clusters, a process of cascades' increasing stops. The system's result is an output of the last cascade. If a sequence changed its properties, the process of cascades' increasing could be continued.

Numerical implementation of the proposed system is not really difficult due to the fact that an incoming data stream is processed in a parallel and independent manner [10, 26] with the system's nodes $N^{[m]}$.

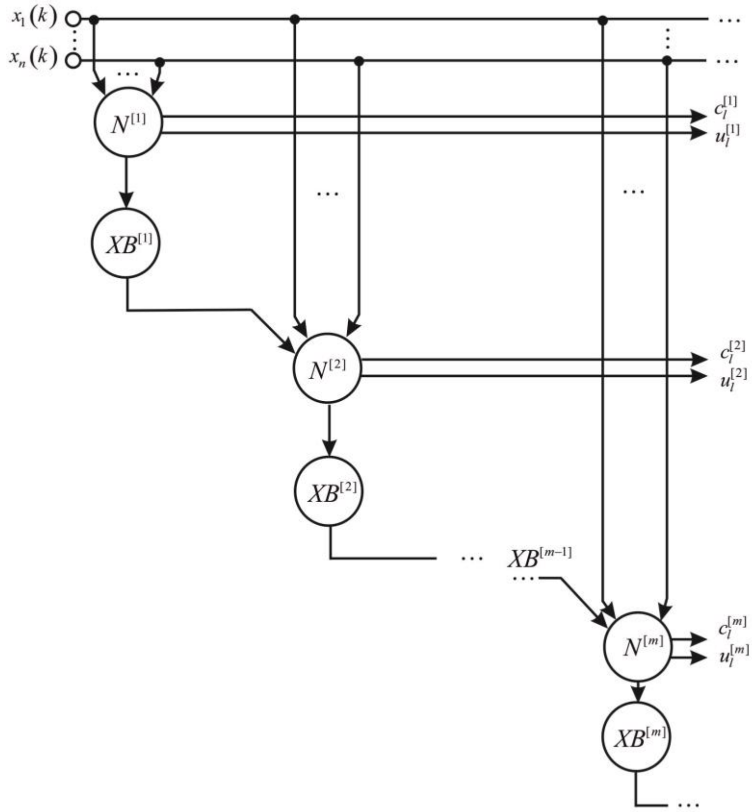

Fig.1 An evolving cascade clustering neuro-fuzzy system

III. AN ADAPTIVE LEARNING PROCEDURE FOR THE NODES

Learning procedures of the system's nodes are based on fuzzy clustering algorithms that deal with goal functions [29] and solve a task of their optimization under a priori assumptions. The most popular algorithm is the probabilistic approach based on minimization of a goal function under the constraints

$$E\left(u_l^{[m]}(k), c_l^{[m]}\right) = \sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_l^{[m]}(k)\right)^{\beta}\left\|x(k) - c_l^{[m]}\right\|^2$$

$$\sum_{l=1}^{m+1} u_l^{[m]}(k) = 1,\ 0 \le \sum_{k=1}^{N} u_l^{[m]}(k) \le N$$

where $0 \le u_l^{[m]}(k) \le 1$ is a membership level of an observation $x(k)$ to the $l-$th cluster in the $m-$th cascade, $c_l^{[m]}$ is a $(n \times 1)-$ vector of prototypes of the $l-$ th cluster in the $m-$ th cascade, $\beta > 1$ is a fuzzification parameter (a fuzzifier), which determines blurriness between cluster boundaries, $k = 1, 2, \ldots, N, \ldots$ is a sample size to be clustered (it's considered to be fixed and determined a priori within the Bezdek traditional framework).

Recurrent algorithms were proposed in [30, 31] to process data streams sequentially that come in an online mode. Here $\eta(k+1)$ is a learning rate parameter. This method is based on the Arrow-Hurwicz-Uzava nonlinear programming procedure and underlies an online neuro-fuzzy system proposed in [32]:

$$\begin{cases} u_{jl}^{[m]}(k+1) = \dfrac{\left\|x(k+1) - c_{jl}^{[m]}(k)\right\|^{\frac{2}{1-\beta_j}}}{\sum_{l=1}^{m+1}\left\|x(k+1) - c_{jl}^{[m]}(k)\right\|^{\frac{2}{1-\beta_j}}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \\ \quad + \eta(k+1)\left(u_{jl}^{[m]}(k+1)\right)^{\beta_j}\left(x(k+1) - c_{jl}^{[m]}(k)\right) \end{cases} \tag{1}$$

The Algorithm (1) is a generalization of the Chung-Lee learning procedure [5] and when $\beta_j = 2$ it's rather close to the Park-Dagher gradient procedure [1]

$$\begin{cases} u_{jl}^{[m]}(k+1) = \dfrac{\left\|x(k+1) - c_{jl}^{[m]}(k)\right\|^{-2}}{\sum_{l=1}^{m+1}\left\|x(k+1) - c_{jl}^{[m]}(k)\right\|^{-2}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \\ + \eta(k+1)\left(u_{jl}^{[m]}(k+1)\right)^{2}\left(x(k+1) - c_{jl}^{[m]}(k)\right). \end{cases} \tag{2}$$

Considering the ratio (1) from the perspective of the Kohonen self-organizing map's (SOM) learning, it should be noticed that the multiplier $\left(u_{jl}^{[m]}(k+1)\right)^{\beta_j}$ corresponds to a neighbourhood function (a bell-shaped function) in a learning rule based on the "winner takes more" principle.

In a general case, the learning algorithm (1) of the node $N^{[m]}$ may be considered as a self-learning rule of the Kohonen SOM fuzzy modification. There's an architecture of a two-layer fuzzy modification in Fig.2.

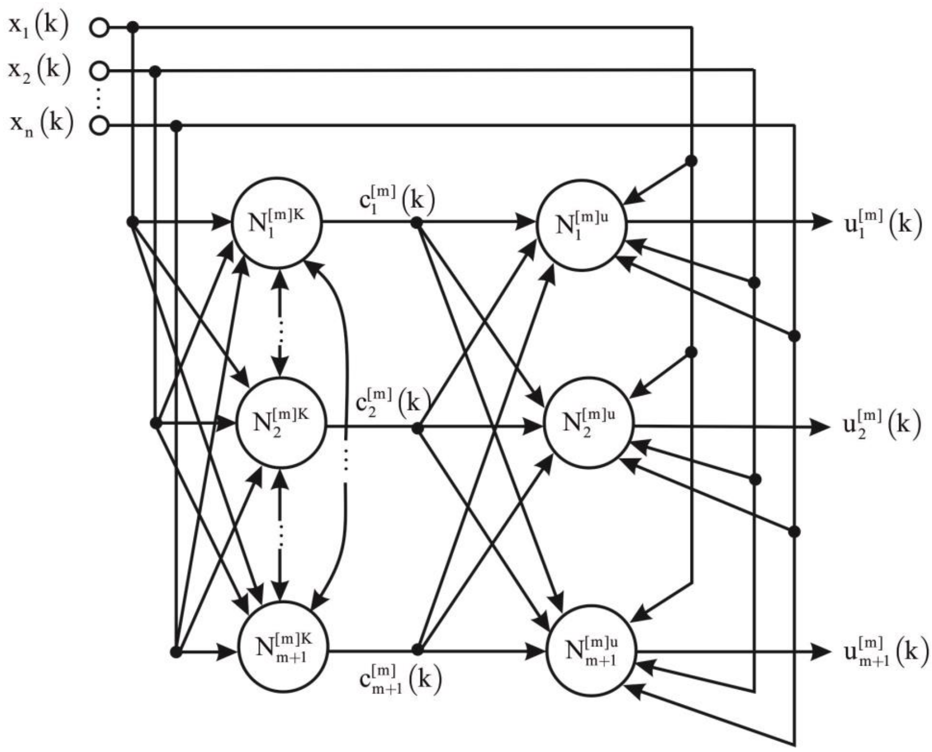

Fig. 2 An architecture of a two-layer Kohonen SOM fuzzy modification

Here $N_l^{[m]K}$ are conventional Kohonen neurons, interconnected with lateral connections, whose tuning is fulfilled according to the WTM-learning rule based on the second ratio at (1). The nodes $N_l^{[m]u}$ calculate membership levels according to the first ratio at (1). The node of every subsequent cascade contains additionally one Kohonen neuron $N_{m+2}^{[m+1]K}$ and one element $N_{m+2}^{[m]u}$ for membership levels' calculation.

We can improve a clustering quality under uncertainty conditions about a number of clusters if we move on to fuzzy clustering algorithms (which are based on the possibilistic approach [33]) from the conventional probabilistic approach (which underlies the algorithms (1), (2)). In this case, an expression is used as a clustering object function

$$E\left(u_l^{[m]}, c_l^{[m]}\right) = \sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_l^{[m]}(k)\right)^{\beta}\left\|x(k)-c_l^{[m]}\right\|^2 + \sum_{l=1}^{m+1}\mu_l^{[m]}\sum_{k=1}^{N}\left(1-u_l^{[m]}(k)\right)^{\beta} \quad (3)$$

where a scalar parameter $\mu_l^{[m]} > 0$ defines a distance where a membership level takes on a value of 0.5 which means that if

$$\left\|x(k)-c_l^{[m]}\right\|^2 = \mu_l^{[m]}$$

then

$$u_j(k) = 0.5.$$

Recurrent optimization of the object function (3) leads to a self-learning algorithm [30]

$$\begin{cases} u_l^{[m]}(k+1) = \dfrac{1}{1+\left(\dfrac{\left\|x(k+1)-c_l^{[m]}(k)\right\|^2}{\mu_l^{[m]}(k)}\right)^{\frac{-1}{1-\beta}}}, \\ c_l^{[m]}(k+1) = c_l^{[m]}(k) + \dfrac{\left(u_l^{[m]}(k+1)\right)^{\beta}}{k+1}\left(x(k+1)-c_l^{[m]}(k)\right), \\ \mu_l^{[m]}(k+1) = \left(\sum_{p=1}^{k+1}\left(u_l^{[m]}(p)\right)^{\beta}\right)^{-1}\left(\sum_{p=1}^{k+1}\left(u_l^{[m]}(p)\right)^{\beta}\left\|x(p)-c_l^{[m]}(k+1)\right\|^2\right). \end{cases} \quad (4)$$

Although the possibilistic procedure (4) is a little more complicated from a computational point of view than the probabilistic algorithm (1), its advantage is the fact that new clusters may be detected with the help of the possibilistic approach during online data processing. If a membership level of a new incoming observation $x(k+1)$ to all classes turns out to be lower than some predefined threshold then we can assume that there's a new $(m+2)-\text{th}$ cluster in the $m-$th cascade and its initial prototype coordinates are $c_{m+2}^{[m]}(0)=x(k+1)$.

IV. NODES' CONTROL

Clustering quality provided by a node $N^{[m]}$ in a cascade may be estimated with the help of any fuzzy clustering indexes [34, 35]. Wherein one of the simplest and most effective indexes is the so-called "partition coefficient" which is a mean value of squared membership levels of all observations to each cascade.

This coefficient has a clear physical sense: the better clusters are expressed, the higher the value is and its minimum is reached if data belong to all clusters evenly. Calculating the partition coefficient is fulfilled for every node of the system simultaneously with its parameters' tuning.

Each cascade of the proposed system differs from others with a number of clusters which are the result of the partition procedure of a processed data stream. The system's nodes $XB^{[m]}$ estimate the results taking into account a number of clusters in each cascade $(m+1)$. One of these indexes is the Xie-Beni index [34] which can be written down for a fixed sample of N observations in this way

$$XB^{[m]}=\frac{\left(\sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_l^{[m]}(k)\right)^2\left\|x(k)-c_l^{[m]}\right\|^2\right)/N}{\min_{l\neq q}\left\|c_l^{[m]}-c_q^{[m]}\right\|^2}=\frac{NXB^{[m]}}{DXB^{[m]}}. \tag{5}$$

The Xie-Beni index is in fact a ratio between variation within clusters ($NXB^{[m]}$) and a value of clusters' separation($DXB^{[m]}$). A minimum value at the ratios (5) corresponds to an optimal number of clusters in the cascade $(m+1)$. That's why a process of cascades' increasing in the system goes on till the index value starts increasing. initial prototype coordinates are $c_{m+2}^{[m]}(0)=x(k+1)$.

V. CONCLUSION

An evolving cascade neuro-fuzzy system for online fuzzy clustering was proposed in this paper. Every node of each system's cascade solves the clustering task independently from others which makes it possible to increase a speed of the whole data processing. A quality estimation process is defined by finding an optimal value of the used cluster validity index which current estimation is also fulfilled in an online mode.

REFERENCES

[1] D.C. Park and I. Dagher, "Gradient based fuzzy c-means algorithm", in *Proc. IEEE Int. Conf. on Neural Networks*, 1984, pp. 1626-1631.

[2] P. Vuorimaa, "Fuzzy self-organizing map", *Fuzzy Sets and Systems*, vol. 66, 1994, pp. 223-231.

[3] P. Vuorimaa, "Use of the fuzzy self-organizing map in pattern self-recognition", in *Proc. 3rd IEEE Int. Conf. Fuzzy Systems "FUZZ-IEEE'94"*, Orlando, USA, 1994, pp. 798-801.

[4] E.C.-K. Tsao, J.C. Bezdek, and N.R. Pal, "Fuzzy Kohonen clustering networks", *Pattern Recognition*, vol. 27, 1994, pp.757–764.

[5] F.L. Chung and T. Lee, "Fuzzy competitive learning", *Neural Networks*, vol. 7(3), 1994, pp. 539-552.

[6] R.D. Pascual-Marqui, A.D. Pascual Moutano, K. Kochi, and J.M. Carazo, "Smoothly distributed fuzzy c-means: a new self-organizing map", *Pattern Recognition*, vol. 34, 2001, pp. 2395-2402.

[7] G. Leng, G. Prasad, and T.M. McGinnity, "An on-line algorithm for creating self-organizing fuzzy neural networks", *Neural Networks*, vol. 17(10), 2004, pp. 1477-1493.

[8] Ye. Bodyanskiy, "Computational intelligence techniques for data analysis", in *Lecture Notes in Informatics*, vol. 72, 2005, pp.15-36.

[9] F. Crespo and R. Weber, "A methodology for dynamic data mining based on fuzzy clustering", *Fuzzy Sets and Systems*, vol. 150(2), 2005, pp. 267-284.

[10] J. Beringer and E. Huellermeier, "Online clustering of parallel data streams", *Data and Knowledge Engineering*, vol. 58(2), 2006, pp. 180-204.

[11] W. Pedrycz and P. Rai, "Collaborative clustering with the use of fuzzy c-means and its quantification", *Fuzzy Sets and Systems*, vol. 159(18), 2008, pp. 2399-2427.

[12] N.K. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 31(6), 2001, pp. 902–918.

[13] N.K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction", *IEEE Transactions on Fuzzy Systems*, vol. 10(2), 2002, pp. 144–154.

[14] N. Kasabov, "Evolving Connectionist Systems", London: Springer-Verlag, 2003, 307 p.

[15] P. Angelov and X.W. Zhou, "Evolving fuzzy systems from data streams in real-time", in *Proc. 2006 Int. Symp. on Evolving Fuzzy Systems (EFS 2006)*, Ambleside, Lake District, UK, 2006, pp. 29–35.

[16] E. Lughofer, "Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications", Berlin-Heidelberg: Springer-Verlag, 2011, 454 p.

[17] M. Pratama, S.G. Anavatti, P. Angelov, and E. Lughofer, "PANFIS: A novel incremental learning machine", *IEEE Trans. Neural Networks and Learning Systems*, vol. 25(1), 2014, pp. 55-68.

[18] N. Kasabov, "Ensembles of efunns: An architecture for a multimodule classifier", in *Proc. International Conference on Fuzzy Systems*, vol. 3, 2001, pp. 1573–1576.
[19] P. Angelov and D. Filev, "Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models", in *Proc. FUZZ-IEEE 2005*, Reno, Nevada, U.S.A., 2005, pp. 1068–1073.
[20] P. Angelov, 'Evolving Takagi-Sugeno fuzzy systems from streaming data, eTS+", *Evolving Intelligent Systems: Methodology and Applications*, 2010, pp. 21–50.
[21] P. Angelov and E. Lughofer, "Data-driven evolving fuzzy systems using eTS and FLEXFIS: Comparative analysis", *International Journal of General Systems*, vol. 37(1), 2008, pp. 45–67.
[22] P. Angelov, E. Lughofer and E. Klement, "Two approaches to data-driven design of evolving fuzzy systems: eTS and FLEXFIS", in *Proc. NAFIPS 2005*, Ann Arbor, Michigan, U.S.A., 2005, pp. 31–35.
[23] E. Lughofer, "FLEXFIS: A robust incremental learning approach for evolving TS fuzzy models", *IEEE Trans. on Fuzzy Systems*, vol. 16(6), 2008, pp. 1393–1410.
[24] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction", *Fuzzy Sets and Systems*, vol. 157(9), 2006, pp. 1260-1275.
[25] C.-F. Juang and C.-T. Lin, "An online self-constructing neural fuzzy inference network and its applications", *IEEE Trans. on Fuzzy Systems*, vol. 6(1), 1998, pp. 12-32.
[26] B.-R. Dai, J.-W. Huang, M.-Y. Yeh, and M.-S. Chen, "Adaptive clustering for multiple evolving streams", *IEEE Trans. on Knowledge and Data Engineering*, vol. 18(9), 2006, pp. 1166-1180.
[27] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture", Adv. Neural. Inf. Process. Syst., vol. 2, 1990, pp. 524–532.
[28] Ye. Bodyanskiy, O. Tyshchenko, and D. Kopaliani, "A hybrid cascade neural network with an optimized pool in each cascade", *Soft Computing. A Fusion of Foundations, Methodologies and Applications*, DOI: 10.1007/s00500-014-1344-3, pp.1-10.
[29] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", N.Y.: Plenum Press, 1981, 272 p.
[30] Ye. Bodyanskiy, V. Kolodyazhniy, and A. Stephan, "Recursive fuzzy clustering algorithms", in *Proc. 10th East West Fuzzy Colloquium*, 2002, pp. 276-283.
[31] Ye. Gorshkov, V. Kolodyazhniy, and Ye. Bodyanskiy, "New recursive learning algorithms for fuzzy Kohonen clustering network", in *Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems*, Rapperswil, Switzerland, June 21-24, 2009, pp. 58-61.
[32] Ye. Bodyasnkiy, O. Tyshchenko, and D. Kopaliani, "An evolving neuro-fuzzy system for online fuzzy clustering", in *Proc. 10th Int. Sci. Techn. Conf. "Computer Science and Information Technologies (CSIT'2015)"*, 14-17 Sept. 2015, pp. 158-161.
[33] R. Krishnapuram and J.M. Keller, "A possibilistic approach to clustering", *Fuzzy Systems*, vol. 1, no. 2, 1993, pp. 98-110.
[34] X.L. Xie and G. Beni, "A validity measure for fuzzy clustering", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, 1991, pp. 841–847.
[35] R. Xu and D.C. Wunsch, "Clustering", *IEEE Press Series on Computational Intelligence*, Hoboken, NJ: John Wiley & Sons, Inc., 2009, 341 p.

