

# Joining Concept's Based Fuzzy Cognitive Map Model with Moving Window Technique for Time Series Modeling

Wladyslaw Homenda<sup>1</sup>, Agnieszka Jastrzebska<sup>1</sup>, and Witold Pedrycz<sup>2,3</sup>

<sup>1</sup> Faculty of Mathematics and Information Science, Warsaw University of Technology  
ul. Koszykowa 75, 00-662 Warsaw, Poland

<sup>2</sup> Systems Research Institute, Polish Academy of Sciences,  
ul. Newelska 6, 01-447 Warsaw, Poland

<sup>3</sup> Department of Electrical & Computer Engineering, University of Alberta,  
Edmonton T6R 2G7 AB Canada

{homenda,A.Jastrzebska}@mini.pw.edu.pl, wpedrycz@ualberta.ca

**Abstract.** In the article we present a technique for time series modeling that joins concepts based Fuzzy Cognitive Map design with moving window approach. Proposed method first extracts concepts that generalize the underlying time series. Next, we form a map that consists of several layers representing consecutive time points. In each layer we place concepts obtained in the previous step. Fuzzified time series is passed to the map according to the moving window scheme. We investigate two most important aspects of this procedure: division into concepts and window size and their influence on model's accuracy. Firstly, we show that extraction of concepts plays a big role. Fitted models have low errors. Unfortunately, it is not always possible to extract appropriate number of concepts. The choice of the number of concepts is a compromise between model size and accuracy. Secondly, we show that increasing window size improves modeling accuracy.

## 1 Introduction

Time series modeling has been a vital area of research for many years now. Time series are basic stochastic processes, extremely important and common. Beside classical approaches, there are plenty alternative time series modeling methods, among which we find Fuzzy Cognitive Maps.

Fuzzy Cognitive Maps allow to process knowledge on an abstract, conceptual level, which is far more understandable and interpretable for human beings. Fuzzy Cognitive Maps are directed graphs-based models, where nodes are concepts that represent phenomena, for example high unemployment, high crime rate, low incomes, and weighted arcs represent relationships between concepts.

The major advantage of FCM-based models is their human-centered knowledge representation interface. Therefore, time series modeling with Fuzzy Cognitive

Maps may not beat well-researched classical approaches in terms of accuracy, but FCMs offer superior practical features.

In this article authors continue presentation of their research on Fuzzy Cognitive Maps-based time series modeling. The basics of our methodology have been presented in [3]. Cited paper provides information about a simple moving window FCM-based model, but processed information is crisp, scaled to  $[0, 1]$ , no concepts are designed and discussed. The objective of this paper is to continue with a presentation of our model. We stress and discuss the conceptual representation of time series for the model. Concept's based time series representation is joined with moving window approach. Introduced methods and ideas are illustrated with a series of experiments on synthetic time series.

The remainder of this paper is organized as follows. Section 2 covers literature review. In Section 3 we discuss proposed approach to time series modeling and prediction. In Section 4 we present a series of experiments. Section 5 concludes the article.

## 2 Literature Review

Fuzzy Cognitive Maps (FCMs) are abstract, graph-based knowledge representation models introduced in 1986 by B. Kosko in [4]. Time series modeling is one of many successful applications' areas of FCMs.

There are two major streams in research on time series modeling with FCMs. First, has been initiated in 2008 by W. Stach, L. Kurgan and W. Pedrycz in [7]. Named authors decided to design FCM's nodes as concepts describing the character of the value of input signal  $a(t)$  and its difference  $\delta a(t)$ .

Second approach is rooted in classification. Examples of research in these stream are: [2], [5], and [6]. On the input to classification-oriented time series modeling methods are multivariate time series. Each node corresponds to a different variable.

In this article are we propose an alternative, original approach to time series modeling.

## 3 Methods

Fuzzy Cognitive Maps offer a noteworthy time series processing framework. FCMs operate on abstract level of concepts that represent aggregates of knowledge, or in other words knowledge granules. Proposed modeling method translates a numerical problem (the input sequence of scalars - the time series) into concepts and generates all necessary elements to train a Fuzzy Cognitive Map. Information passed to FCM's input is processed and FCM responses with concept-based time series models or predictions.

In brief, the proposed method for time series modeling with Fuzzy Cognitive Maps can be divided into following phases:

1. FCM design.
2. FCM training.
3. FCM exploration (time series modeling and prediction).

In the following subsections we introduce step-by-step the proposed approach.

### 3.1 Fuzzy Cognitive Map Design – Extraction of Concepts

Time series modeling with Fuzzy Cognitive Maps requires an algorithmically aided translation method from numerical problem to abstract concepts.

The only exception from this rule is when a time series is reported in a form of concepts, not in a form of crisp values. This is very rare, because typically time series are collected with the use of crisp measurement systems and represented with, for example money units, distance units, velocity units, etc. Let us denote such crisp input scalar time series as a sequence of numbers:  $a_0, a_1, \dots, a_M$ .

The first task then is to elevate a scalar time series to concepts, which will become nodes in the designed FCM.

At the beginning, model designer has to decide on the number of concepts that will generalize the underlying time series. Concepts may be represented with any granular knowledge representation scheme. In this article we use fuzzy sets. For even more human-centered design, each concept has a linguistic variable, which describes the information that it aggregates. Proposed concepts should represent values of the underlying experimental data (the time series). Therefore, examples of concepts are: Small values, Moderate values, etc. Concepts in this study are realized with fuzzy sets. Thus, time series data points will be represented with degrees of membership to the fuzzy sets of proposed concepts.

The number of concepts, let us denote it as  $u$ , defines the level of model's specificity. The more concepts we decide on, the more specific model we get. The fewer concepts we have, the more general model we get. The balance between specificity and generality affects accuracy. Model's accuracy is typically measured with discrepancies between actual, observed phenomena and modeled values. As an indicator for model's accuracy in this study we use Mean Squared Error, which is a very common statistics. More specific models have lower errors, than general models. The other aspect of specificity/generality conflict is the ease of interpretation. Models based on fewer concepts are less complex and therefore easier to interpret and to apply. In this light, in all experiments presented in this paper we use 3 concepts with following linguistic terms: Small, Moderate, High, abbreviated as S, M and H respectively. We have chosen the value of 3, which is enough to clearly illustrate method's properties.

Ideally, the number of concepts should match the experimental data and be as small as possible at the same time. This topic is out of the scope of this paper, we are currently working on an algorithmic approach to a selection of appropriate number of concepts.

To facilitate an efficient extraction of concepts we apply well-known fuzzy c-means algorithm, [1]. On the input to fuzzy c-means we pass the scalar time series and the desired number of concepts-clusters. On the output of fuzzy c-means we

receive proposed cluster's centers and degrees of membership of all time series points to the proposed clusters. By assumption, the levels of belongingness are evaluated as a number from the  $[0, 1]$  interval, while cluster's centers are best found representatives of the underlying data set.

### 3.2 Fuzzy Cognitive Map Design – Moving Window Technique

In the subsequent step of Fuzzy Cognitive Map design it is necessary to collect all elements required for Fuzzy Cognitive Map training. The initial design, described in the previous subsection, resulted in a scratch of the FCM. Let us remind, that after the first steps of the method we have extracted  $u$  concepts that generalize the underlying time series. Concepts are realized with fuzzy sets, obtained with the use of fuzzy c-means. To calculate the level of membership for an  $a_i$ -th observation to  $j$ -th concept the standard Fuzzy C-Means objective function is used:

$$z_{ij} = \frac{1}{\sum_{k=1}^u \left( \frac{\|a_i - v_j\|}{\|a_i - v_k\|} \right)^{2/(m-1)}} \quad (1)$$

where  $u$  is the number of concepts,  $m$  is the fuzzification coefficient ( $m > 1$ ) and  $\|\cdot\|$  is the Euclidean distance function,  $a_i$  is  $i$ -th time series data point,  $v_j$  is  $j$ -th concept.

As a result, each  $i$ -th time series data point is now fuzzified into  $u$  concepts. We can denote elevated granular time series as:  $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_M$ , where  $\mathbf{a}_i = [z_{i1}, z_{i2}, \dots, z_{iu}]$ , and  $u$  is the number of concepts.

At this point we join the discussed concept's extraction procedure with moving window-based time series representation.

The moving-window technique analyzes data points by creating a series of different subsets of the full data set. The idea of moving window is perhaps most commonly applied and acknowledged in moving averages method. Moving window method for time series representation creates different sequences of consecutive data points from the time series. There are two parameters for the moving window method for time series representation: window size and window step size. Window size is the number of data points in each window. Window step size is the tempo at which the window moves. Window step size is typically equal 1, and in this case if current sequence starts with  $i$ -th observation from the time series, then the next window starts with  $i+1$ -th, then  $i+2$ -th, and so on. If the window step size is equal  $p$ , then if current sequence starts with  $i$ -th observation from the time series, then the next window starts with  $i+p$ -th, then  $i+2 \cdot p$ -th, and so on. In the experiments in this paper we always assume window step size equals 1.

The most important parameter in this technique is window size. In the experiment's section of this article we investigate the relation between window size and FCM-based model's accuracy.

### 3.3 Fuzzy Cognitive Map Learning

Moving window representation joined with extracted concepts provide necessary data to train the FCM.

At this point, let us briefly introduce necessary formalisms for Fuzzy Cognitive Maps.

The crux of each Fuzzy Cognitive Map is its weights matrix  $\mathbf{W}$ , which size is  $n \times n$ .  $n$  is the number of nodes in the FCM. The weights matrix contains values of all connections between nodes in the map. A single weight is denoted as  $w_{ij}$ , and  $w_{ij} \in [-1, 1]$  with exception for elements above r-diagonal, which are set to 0. Weights that are fixed to 0 represent relationships from future to past. This is conceptually impossible, hence we set them to 0. We will elaborate on this topic later in this subsection.

FCM learning is based on activations  $\mathbf{X}$  and goals  $\mathbf{G}$ . Goals are actual, reported states of phenomena. FCM responses are denoted as  $\mathbf{Y}$ . Map responses model phenomena, hence they should be as close to goals as possible. Activations', responses' and goals' matrices size is  $n \times N$ , where  $N$  is the number of training observations. Single activation is denoted as  $x_{ji}$ , single map response is denoted as  $y_{ij}$ , and single goal is denoted as  $g_{ij}$ , they all are real numbers from the  $[0, 1]$  interval.

FCM exploration is performed according to the formula:

$$\mathbf{Y} = f(\mathbf{W} \cdot \mathbf{X}) \quad (2)$$

where  $f$  is a sigmoid transformation function:

$$f(t) = \frac{1}{1 + \exp(-\tau t)} \quad (3)$$

The value of parameter  $\tau$  was set to 5 based on literature review.

The shape of weights matrix  $\mathbf{W}$  is obtained by a learning procedure. The goal is to adjust weights' matrix so that differences between FCM responses  $\mathbf{Y}$  and goals  $\mathbf{G}$  are the smallest. Typically in the literature, and also in our experiments, we minimize the Mean Squared Error:

$$MSE = \frac{1}{n \cdot N} \cdot \sum_{j=1}^N \sum_{i=1}^n (y_{ij} - g_{ij})^2 \quad (4)$$

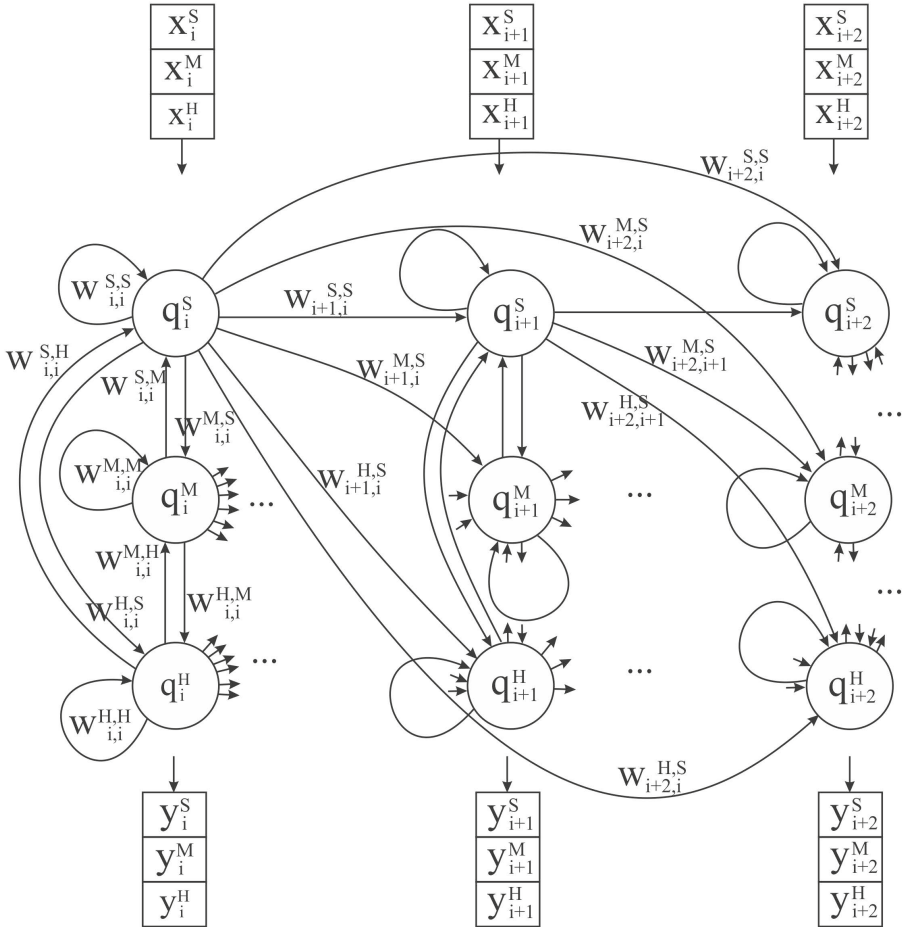
In our experiments we select first 70% of data points for FCM training. We call this dataset 'train'. Consecutive 30% data points are for one-step-ahead predictions only. We call this dataset 'test'. Test dataset is not involved in FCM training and thus it is used to confirm quality of models proposed by our FCMs.

Moving window representation of fuzzified time series becomes activations and goals. Observations falling to an  $i$ -th window become  $i$ -th activations vector:  $\mathbf{x}_i = [\mathbf{a}_{i+1}, \mathbf{a}_{i+2}, \dots, \mathbf{a}_{i+r}]$ , where  $r$  is window size. Goals are activations shifted one observation forward:  $\mathbf{g}_i = [\mathbf{a}_{i+2}, \mathbf{a}_{i+3}, \dots, \mathbf{a}_{i+r+1}]$ . The actual number of elements in such vectors is  $u \cdot r$ . In our experiments it is then  $3 \cdot r$ .

Nodes in FCM correspond to consecutive concepts that fall to window of size  $u \cdot r$ . The number of nodes  $n = u \cdot r$ ;  $u = 3$ , so:

- first triple of nodes corresponds to  $i$ -th crisp observation,
- second triple of nodes corresponds to  $i+1$ -th crisp observation,
- ...
- last triple of nodes corresponds to  $i+r$ -th crisp observation.

Note, that the triples of nodes are ordered and they represent consecutive time points. One may visualize it as layers.



**Fig. 1.** Generic schema of FCM with window size  $r = 3$ , division for  $u = 3$  concepts for time series modeling

Figure 1 illustrates the discussed scheme for window of size  $r = 3$  and  $u = 3$ . FCM is a collection of nodes, denoted as  $q_i^S, q_i^M, q_i^H, \dots, q_{i+2}^S, q_{i+2}^M, q_{i+2}^H$ . Notice, nodes form layers, each with 3 concepts. S, M, H abbreviate concepts' names.

First triple of nodes in Figure 1 corresponds to  $i$ -th time point, second triple to  $i+1$ -th time point, etc.

To improve readability we have not drawn all weights. Missing connections are represented with short arrows. Arrows are directed towards respective nodes. Activations ( $x$ ) and map responses ( $y$ ) are symbolically in boxes.

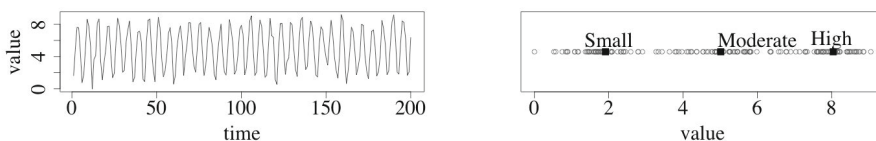
There are no connections from future to past. Hence, weights matrix is a block lower triangular matrix. Block lower triangular matrix is a combination of lower triangular matrix with  $r$ -diagonal matrix. For example if window size  $r = 3$ , then lower block triangular matrix is a combination of lower triangular matrix and a tridiagonal matrix. Such special block matrix visually forms steps. Steps' height is 1, steps' width is  $u$ .

FCM training can be conveniently preformed with a search metaheuristics, such as PSO. In experiments presented in this article we use PSO implementation from R package 'pso' with all default parameters listed under: [8]. After we have learned the FCM's weights matrix we can use it to model time series.

### 3.4 Illustration on an Example of a Synthetic Time Series

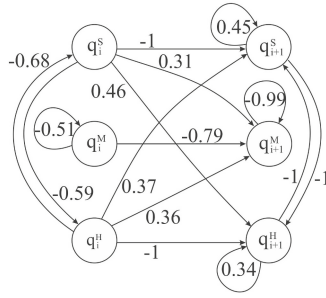
Let us illustrate the described procedure on an example of a synthetic time series. The time series was constructed by replicating a sequence 2, 5, 8, 8, 5, 2 500 times, what gave total 3000 observations. We have distorted the sequence with a random value drawn from normal distribution with mean equals 0 and standard deviation 0.7. The time series has a period equals 6 and no trends. Left plot in Figure 2 illustrates first 200 observations from this synthetic time series.

With the use of fuzzy  $c$ -means we extract 3 concepts to represent the underlying data set. Concepts' centers are in: 1.91, 5.01, and 8.04. Right plot of Figure 2 illustrates extracted concepts with first 200 data points of the synthetic time series in the background.



**Fig. 2.** Synthetic time series based on sequence 2, 5, 8, 8,5, 2. Left: first 200 data points in 2-dimens. space of time and values. Right: first 200 data points in 1-dimens. plot with extracted 3 concepts representing Small, Moderate and High values.

For the illustration let us assume window size  $r = 2$ .  $u = 3$ , so the number of nodes in the map is  $n = 2 \cdot 3$ . Figure 3 illustrates learned FCM. We have removed all weights weaker than 0.3 to improve visibility. Removing weak weights from the map may allow to simplify generated model, reduce dimensionality and improve



**Fig. 3.** Trained FCM with  $n = 6$  nodes for the  $(2, 5, 8, 8, 5, 2)$  synthetic time series

readability. By analogy to the removal of weights we can remove nodes that weakly influence other nodes. An example of the weakest node in FCM from Figure 3 is node denoted as  $q_{i+1}^S$ , which sum of absolute values of outgoing weights is 0.45.

Table 1 gathers activations and goals for this FCM. It is easy to notice that goals are activations shifted forward by 1. Also, each activation and goal is a collection of  $r$  triples of membership values to concepts described as Small, Moderate, and High.

The outcomes of modeling and predictions for this time series, among others, are discussed in the next section.

### 4 Experiments

In this Section we present a series of experiments on several synthetic time series.

The considered time series were built on three sets of numbers of cardinality 3 and 5:  $\{2, 5, 8\}$ ,  $\{2, 6, 8\}$ , and  $\{1, 3, 5, 7, 9\}$ . Cardinalities correspond to the number of real concepts, that have a support in the underlying data. The length

**Table 1.** The matrices of activations  $\mathbf{x}_i$  and goals  $\mathbf{g}_i$  for the synthetic time series, window size  $r = 2$ , number of concepts  $u = 3$ , thus map size  $n = 2 \cdot 3 = 6$

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\dots$	$\mathbf{x}_{N-1}$	$\mathbf{x}_N$	$\mathbf{g}_1$	$\mathbf{g}_2$	$\mathbf{g}_3$	$\dots$	$\mathbf{g}_{N-1}$	$\mathbf{g}_N$
0.78	0.04	0.00	$\dots$	0.00	0.00	0.04	0.00	0.00	$\dots$	0.00	0.01
0.03	0.10	0.98	$\dots$	0.98	0.98	0.10	0.98	0.98	$\dots$	0.98	0.01
0.19	0.87	0.02	$\dots$	0.01	0.02	0.87	0.02	0.02	$\dots$	0.02	0.99
0.04	0.00	0.00	$\dots$	0.00	0.01	0.00	0.00	0.00	$\dots$	0.01	0.88
0.10	0.98	0.97	$\dots$	0.98	0.01	0.98	0.97	0.97	$\dots$	0.01	0.03
0.87	0.02	0.02	$\dots$	0.02	0.99	0.02	0.02	0.02	$\dots$	0.99	0.09



**Table 2.** Summary of synthetic time series constructions

time series	period	number of concepts
268	3	3
258852	6	3
258582	6	3
225825558822885	15	3
268682826286286	15	3
15739	5	5
1573993751	10	5
1573971593	10	5
153791377195395	15	5
157393975117359	15	5

of each generated dataset was 3000. Datasets were distorted by adding to the original sequence random values drawn from the normal probability distribution with mean equals 0 and standard deviation equals 0.7. Synthetic time series have different periods. Summary of properties of generated time series is in Table 2.

Objective of the experiment's section is to investigate properties of the proposed time series modeling technique. We take closer look at the following issues:

- relation between window size ( $r$ ) and accuracy,
- influence of appropriate selection of the number of concepts ( $u$ ) on accuracy.

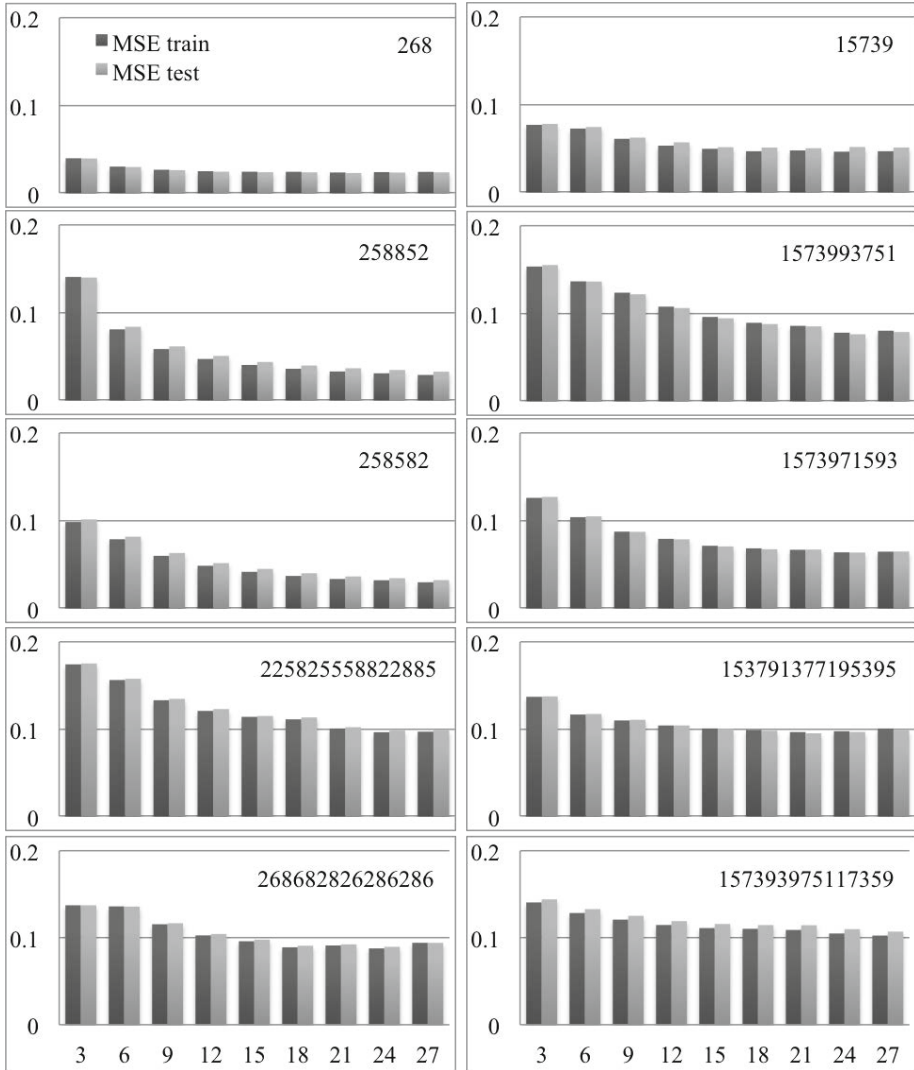
For each artificial time series we have built 9 FCMs for various values of window size:  $r = 1, 2, \dots, 9$ . In each case the number of concepts was  $u = 3$ . Thus, the sizes of tested maps were  $n = 3, 6, \dots, 27$  respectively.

With such selection of different time series we were able to investigate quality of built models. For each FCM we have repeated the procedure 3 times to verify if the model was stable. To compare the results we use Mean Squared Error both on train and test data sets.

Figure 4 presents an overview of the results. Plots illustrate MSE on train and test data sets for synthetic time series from Table 2. Left column is for time series based on three elements:  $\{2, 5, 8\}$  and  $\{2, 6, 8\}$ . Right column of Figure 4 is for synthetic time series based on 5 digits:  $\{1, 3, 5, 7, 9\}$ .

Figure 4 confirms that the proposed scheme's performance largely depends on characteristics of time series and on the window size. Note, that MSE for test does not differ much from MSE on train data set.

If the division into concepts directly fits time series, results are most accurate. For example, time series constructed on (2,6,8) sequence is exactly matching the division into  $u = 3$  concepts. In this case both period is equal 3, and the number



**Fig. 4.** MSE on train and test data for map sizes  $n = 3, 6, \dots, 27$  (OX axis). FCMs were constructed for  $u = 3$  concepts and window sizes  $r = 1, 2, \dots, 9$ . In the top right corner of each plot there is a base sequence of the corresponding synthetic time series.

of real concepts is equal 3. This is the time series that fits the most to our experiment scheme and we expect that in this case the results are the best. Indeed, in this case MSE on train and test are the smallest - in top left corner of Figure 4 bars are the lowest.

Other time series are less fitted to the division into  $u = 3$  concepts. Therefore, errors are higher. Time series synthesized based on  $(2,5,8,8,5,2)$  and  $(2,5,8,5,8,2)$  sequences are in partial accordance with division into  $u = 3$  concepts. These

are second and third plots in left column of Figure 4. In these cases period is 6 and the number of real concepts is 3. Observe, that errors are higher than for the perfect fit of a time series based on (2,6,8), but smaller than in other cases. In contrast, fourth and fifth plots in left column of Figure 4 are for time series based on 3 real concepts and period equals 15. Here the error is higher, than for the cases above.

Similarly, in the right column of Figure 4 one can observe, that the number of clusters that we split data into is important. For a fixed number of concepts, the more complex time series we discuss, the higher the MSE. Experiments prove, that results are the worst for time series that are the furthest from perfect fit.

Second objective of the conducted experiments was to investigate the influence of window size. It turned out that the larger the window size, the smaller the MSE. The decrease of MSE is not linear though. Observe, in all plots that there is an inflection point, which we can indicate by calculating the tempo of model's improvement. This is why we do not need to aim at the highest possible window size, but we shall accommodate at a value that provides us with a reasonable map size and accuracy. The larger the window size, the bigger the map. Big models are inconvenient to train, to visualize and to interpret. Hence, depending on application one may decide, for example, to set window size to  $r = 4$ , what in almost all cases in Figure 4 is a convenient balance between size of the map and accuracy.

The second conclusion is that the improvement in modeling accuracy gained by increasing the window size depends on the character of the time series. For models with concepts fitted to the time series, like in the case of the (2,6,8)-based time series, increasing the window size moderately influences accuracy. MSE drops when we increase the window size, but the change is smaller than for time series that do not fit the model this well. In the case of (2,6,8)-based time series the best model (accurate and simple) is for window size  $r = 2$ . Unfortunately, it is very unlikely that in real-world time series we would be able to propose concepts that match the time series this well. Real-world data is not as regular as synthetic ones. Therefore, increasing window size is a valid approach to improve accuracy of proposed model.

## 5 Conclusion

Proposed time series modeling approach is based on Fuzzy Cognitive Maps that represent concepts arranged in  $r$  consecutive time points. One may perceive such structure as layers of concepts. In each layer we have  $u$  concepts that generalize values of the underlying data set. Proposed modeling scheme provides a clear, structured concept-based model for time series.

We have investigated two major parameters of the proposed method: the number of concepts and the number of layers. We have shown that both elements largely influence the MSE between model and actual values. We have shown that in general, when we increase window size, we improve accuracy. Second, we have shown, that if proposed division into concepts match the real time series

characteristics, we get better results. The biggest indifference to the increase of the window size is for models that fit the data to the greatest extent.

Unfortunately, typically real-world data is not as regular as synthetic data sets. This is why we have conducted meticulous study on the influence of window sizes on MSE for several synthetic data sets with the number of concepts that not exactly match time series. Experiments show that there is an inflection point, after which increasing window size does not influence the MSE this much.

In future research we will focus on finding an algorithmic approach to determine the best set of concepts to generalize the underlying dataset.

**Acknowledgment.** The research is partially supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from The European Union within the Innovative Economy Operational Programme (2007- 2013) and European Regional Development Fund.

The research is partially supported by the National Science Center, grant No 2011/01/B/ST6/06478.

## References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. Froelich, W., Papageorgiou, E.I.: Extended Evolutionary Learning of Fuzzy Cognitive Maps for the Prediction of Multivariate Time-Series. In: Fuzzy Cognitive Maps for Applied Sciences and Engineering, pp. 121–131 (2014)
3. Homenda, W., Jastrzebska, A., Pedrycz, W.: Modeling Time Series with Fuzzy Cognitive Maps. In: Proc. of WCCI 2014 (in press, 2014)
4. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies 7, 65–75 (1986)
5. Lu, W., Yang, J., Liu, X.: The Linguistic Forecasting of Time Series based on Fuzzy Cognitive Maps. In: Proc. of IFSA/NAFIPS, pp. 649–654 (2013)
6. Song, H.J., Miao, C.Y., Wuyts, R., Shen, Z.Q., D’Hondt, M.: An Extension to Fuzzy Cognitive Maps for Classification and Prediction. IEEE Transactions on Fuzzy Systems 19(1), 116–135 (2011)
7. Stach, W., Kurgan, L., Pedrycz, W.: Numerical and Linguistic Prediction of Time Series. IEEE Transactions on Fuzzy Systems 16(1), 61–72 (2008)
8. <http://cran.r-project.org/web/packages/pso/pso.pdf>