Linguistic Signal Prediction with the use of Fuzzy Cognitive Maps

Wojciech Stach, Lukasz Kurgan, and Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta T6G 2V4, Canada
{wstach, lkurgan, pedrycz}@ece.ualberta.ca

Abstract

This paper proposes a new framework for signal prediction realized at linguistic level. It uses Fuzzy Cognitive Maps approach and a genetic algorithm based learning method. Fuzzy Cognitive Maps are very convenient and powerful technique of modeling and analysis of dynamic systems. Recently introduced learning method that uses real-coded genetic algorithm approach opened new possibilities of their applications, which are explored in this paper. The developed method learns model of a given continuous signal and predicts its future values as a set of linguistic terms. The theoretical design of the proposed method is supported by comprehensive experiments, carried out for synthetic and real-word data. The results demonstrate high quality of the developed method,

1. Introduction

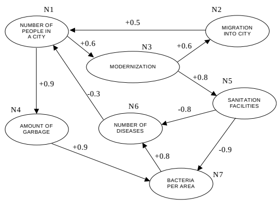
Given the complexity and omnipresence of signal processing (occurring in numerous problems of filtering, prediction, control) we are faced with continuous challenges of building more advanced algorithms that are in rapport with the reality. Prediction problems call for new approaches capable of addressing aspects of nonlinearity and nonstationarity that comes inherently associated with the phenomena under consideration. [9]. Many different approaches to signal representation exist. Mainly, they come from linear system theory [7], stochastic process theory [14] and are usually typical for the black-box methodology [6], and dynamical system analysis [8].

In this paper, we are focused on qualitative modeling, analysis and prediction. This means, that we describe given signal in qualitative terms, i.e. linguistic description, instead of quantitative ones, i.e. numerical. We combine fuzzy modeling, fuzzy cognitive maps and a recently introduced genetic learning algorithm in order to build a system that is able to model and predict signals on linguistic level. Reference [15] is a valuable introduction to fuzzy system modeling, which introduces an idea of distributed fuzzy modeling, whereas Sections 1.1 and 1.2 of this paper describe briefly fuzzy cognitive maps and explain the idea of its genetic learning using RCGA algorithm.

1.1. Fuzzy Cognitive Maps: Introductory Remarks

Fuzzy cognitive maps (FCMs) are simple, yet powerful tool for modeling and simulation of dynamic systems. They were originally introduced by Kosko [11] as an extension of cognitive maps [2].

The main advantage of FCMs lies in their straightforward graph representation, which consists of nodes connected by edges. Nodes correspond to concepts or variables within given domain of application, whereas directed edges reflect mutual relationship between concepts. Each edge is associated with a weight value from the range [-1,1] that expresses both the type and strength of given relationship. Negative value indicates prohibitory effect that source concept exerts on the destination one, Positive value indicates a promoting effect. The zero value denotes no causal relationship between two concepts. The absolute value of the weight corresponds to different fuzzy levels of relationships' strength. The graph representation can be equivalently denoted by a square matrix, called *connection matrix*. It accumulates all weight values for edges between corresponding concepts. Figure 1 shows an example of FCM model that concerns public city health issues [12].



	N1	N2	N3	N4	N5	N6	N7
N1	0	0	0.6	0.9	0	0	0
N2	0.5	0	0	0	0	0	0
N3	0	0.6	0	0	0.8	0	0
N4	0	0	0	0	0	0	0.9
N5	0	0	0	0	0	-0.8	-0.9
N6	-0.3	0	0	0	0	0	0
N7	0	0	0	0	0	0.8	0

Figure 1. Sample FCM model and its connection matrix

During simulation, FCM iteratively calculates its state that is represented by a *state vector*, which consists of all nodes values at given iteration. Value of each node is determined based on values of nodes that exert influence on the given node, i.e. nodes that are connected to this node. These values are multiplied by corresponding weight values and the sum of these products is taken as the input to a *transformation function.* The purpose of using this function is to normalize the node value, usually to the range [0,1]. As a result, each node can be defined as active (value of 1), inactive (value of 0), or active to a certain degree (value between 0 and 1). The non-linear transformation function makes possible to comparisons between nodes at the expense of quantitative analysis. Three popular transformation functions are bivalent, trivalent, and logistic. Depending on such transformation, several patterns of dynamic behavior are possible [10].

Despite of their simplicity, FCM models have been successfully applied to many different research and industrial areas. Examples of such specific applications include medical diagnosis [22], analysis of electrical circuits [21], analysis of failure modes effects [16], fault management in distributed network environment [13], modeling of software development project [17] [20], and many others.

1.2. Genetic Learning of FCM with RCGA Algorithm

The advantage of simplicity of development FCM models has a downside of subjectivity of the model and problems with unbiased assessment of its accuracy, since this process in most cases is based on expert(s) beliefs [1]. Given the set of nodes, the main difficulty is to accurately establish the weights. This issue can be solved by learning methods that allow for establishing FCM model from raw (historical) data. The novel approach to tackle this problem using real-coded genetic algorithm (RCGA) has been introduced recently [19] [18]. In this paper, the FCM models are used as a part of linguistic signal prediction system. The FCM models are automatically generated from a set of fuzzy signals.

Figure 2 shows the high-level diagram of RCGA learning method.

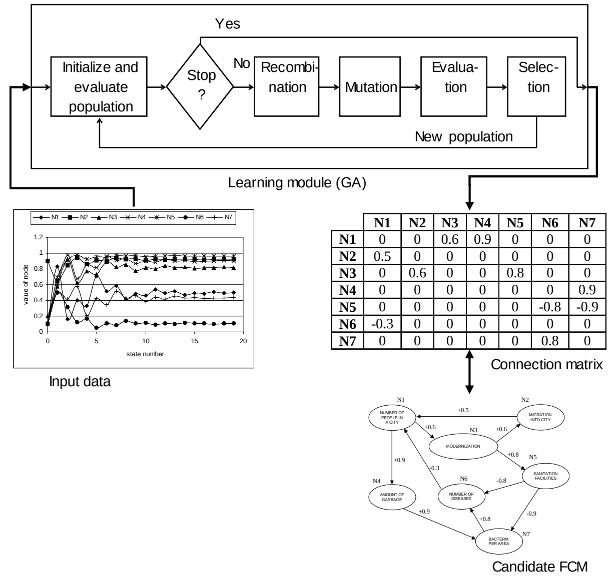


Figure 2. High-level diagram of the RCGA learning method

This is fully automated method that does not involve any human interference. Based on historical data given as a time series, called *input data*, it establishes the FCM model, *candidate FCM*, which is able to mimic the data. The core of this method is real-coded genetic algorithm, which is an extension to genetic algorithm [4] [5]. The fundamental difference concerns *chromosome* representation. In this case, each chromosome consists of floating point numbers that correspond to the problem variables. Accordingly, the genetic operators are redefined. This approach is suitable when tackling optimization problems of parameters with variables in continuous domains.

The RCGA approach uses the input data to find the variables. In this paper, the input data are set of fuzzy representations of given signal and have form of time series. The learning objective is to find the FCM model that produces the same state vector sequence (input data). The idea behind it is to have model that generalizes the input data, and is able to make signal prediction.

The chromosome representation is based on the FCM feature that any model can be fully described by its connection matrix, i.e. by N*N variables, where N refers to the number of concepts. In consequence, chromosome has the following structure $\hat{\mathbf{E}} = [e_{11}, e_{12}, e_{13},..., e_{1N}, e_{21}, e_{22}, e_{23},..., e_{2N},..., e_{NN}]^T$, where e_{ij} determines the value of an edge weight from i^{th} to j^{th} concept node. The role of *fitness function* is to assess the quality of the model represented by its connection matrix. It takes advantage of a specific aspect of FCM theory, i.e. that the current state of FCM model **C**(t+1) depends only on the system state in the immediately preceding iteration **C**(t). Assuming that the input data length is *K*, this feature allows for rearranging the given learning data set into the form of *K-1* different pairs by grouping each two adjacent state vectors

$$C(t) \rightarrow C(t+1) \ \forall t = 0,...,K-1$$

In each pair, C(t) is called *initial vector*, and C(t+1) is called *system response*. The fitness function is based on error between system response from input data set and from candidate FCM for all {initial vector, system response} pairs, which is calculated as follows:

$$Error = \frac{1}{(\mathrm{K}-1)\cdot \mathrm{N}} \sum_{t=1}^{K-1} \sum_{n=1}^{N} \left| C_n(t) - \hat{C}_n(t) \right|^2$$

where $\mathbf{C}(t) = [C_1(t), C_2(t), ..., C_n(t)]$ is a given system response for $\mathbf{C}(t-1)$ initial vector, $\hat{\mathbf{C}}(t) = [\hat{C}_1(t), \hat{C}_2(t), ..., \hat{C}_n(t)]$ is system response of candidate FCM for $\mathbf{C}(t-1)$ initial vector.

Fitness function is defined as:

$$h(Error) = \frac{1}{10000 \cdot Error + 1}$$

Other RCGA parameters, which are based on [18], include: *recombination method* (single-point crossover), *mutation method* (randomly chosen from random mutation, non-uniform mutation, and Mühlenbein's mutation), *selection method* (randomly chosen from roulette wheel and tournament), *probability of recombination* equal to 0.9, *probability of mutation* equal to 0.5, *population_size* equal to 100 chromosomes, *max_generation* size equal to 10000, and *max_fitness* value equal to 0.999.

2. Framework of Proposed Prediction Method

The proposed method's goal is linguistic prediction of signal, in terms of its amplitude and change. It uses fuzzy sets theory to convert the input signal into linguistic description. After fuzzification, given signal expressed in fuzzy terms is used to learn FCM model, which can be used for prediction. High-level diagram of the proposed approach is shown in Figure 3. The connectors (arrows) illustrate the typical routine that leads to establish FCM model. The test phase, i.e. prediction, is marked by dashed line.

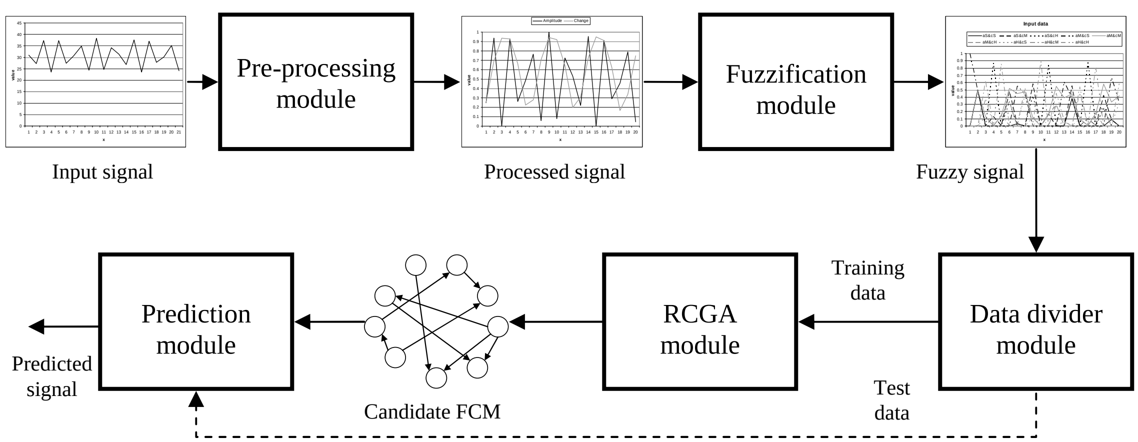


Figure 3. High-level diagram of the proposed prediction method

Proposed system architecture includes five main elements.

The role of **pre-processing module** is twofold. First, it accepts a raw signal, called *amplitude*, and determines its changes; we will be referring to such signal as a, *change*. In other words, this signal describes the input signal changes in time. Second, the role of this module is to normalize both signals. The normalization is done linearly and it brings the original signals to the unit interval. In case of change signal, this procedure is slightly different. Based on minimum and maximum value of input signal, the maximum change value is determined. The change signal is linearly scaled with respect to this maximum value, i.e. the set of all change signal value is a subset of [0,1]. This procedure allows avoiding artificial enlargements of small signal changes, which may happen when all changes for a given signal are small.

Fuzzification module represents both signals, i.e. amplitude and change, in terms of fuzzy sets. Based on given linguistic descriptors that are fuzzy sets defined by their membership functions, this module calculates membership values for each point of the two signals. As a result, the two input time series are transformed into N_1+N_2 time series, in which N_1 corresponds to number of linguistic descriptors of amplitude signal, and N_2 to change signal, respectively.

Data divider module (optional) separates data that are used for learning FCM, i.e. *training data*, from those that are used for testing model accuracy, i.e. *test data* according to given parameters.

RCGA module applies the real-coded genetic algorithm learning method in order to establish candidate FCM model from given dataset that mimic these data, see Section 1.2 for details. Set of nodes correspond to all possible combinations of signals linguistic descriptions, e.g. Node 1 – amplitude *Small* and change *High*.

Prediction module simulates the candidate FCM model and chooses predicted linguistic description of signal in terms of amplitude and change based on fuzzy operations. By changing maximum number of iterations in simulation FCM one obtains prediction of different length, i.e. different number of steps ahead.

3. Experimental Setup

The goal of the experiments is to assess quality of the proposed signal prediction system. The tests are carried out with this method and are divided into two groups: tests performed with *synthetic*, and with *real-life* data. The former group uses synthetic input data, which are generated from given type of function with random parameters. The latter one uses real-life input data, i.e. time series that come from a real system. Each input data set is divided into training and test set. The first one is exploited to learn FCM model that is able to generalize the data. Subsequently, this model is used to prediction.

All experiments are performed with three linguistic descriptors, i.e. *Small*, *Medium*, and *High*, for both amplitude and change signals in fuzzification module. All descriptors are defined as fuzzy sets with triangular membership functions that overlap at the value of 0.5, see Figure 4.

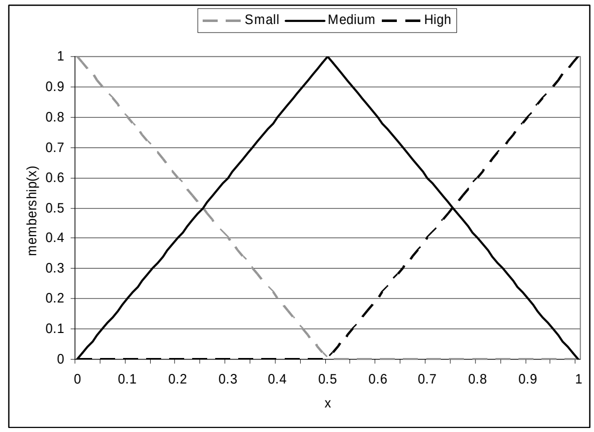


Figure 4. Membership functions of the linguistic terms

Consequently, the number of nodes of candidate FCM model is equal to 9, which corresponds to a number of all possible linguistic combinations of both amplitude and change signals descriptions at given point. Each node represents one linguistic description combination.

Prediction module carries out prediction uses simple *max* operator, i.e. it chooses the highest value among all nodes and predicts the linguistic description that is associated with this node.

Since the quality of learning FCM models with RCGA approach was reported to be dependent on input data length [19], experiments with varying length of training data are performed. For each type of input signal 10 independent experiments, each using different input signal of the same type, are performed and the average values and standard deviations are reported.

3.1. Setup for Experiments with Synthetic Data

Two sets of input data are generated. The first one is based on hyperbolic function, whereas the second one uses sinus function.

The input data for the hyperbolic function is generated based on the following formula $h_1(x) = \frac{1}{ax+b}$, where values for two variables a and b are chosen randomly as a floating point value from [0,10] interval. Similarly, the sinus function is generated by

$h_2(x) = a\sin(bx + c)$, where the three variables a, b and c are random floating point values from [0,10] range

Thus, in each case two sets consisting of 10 time series are generated. Considering the length of data used for experiments with real-life data, see Section 3.2, the length of each time series were set to 36. Sample input signals to the system in both cases are shown in Figure 5.

3.2. Setup for Experiments with Real-life Data

The data set of IBM common stock closing prices: daily, May 17, 1961 – November 2, 1962 [3] is used for this part of experiments. This set consists of 369 observations, see Figure 6. After removing last 9 observations, the data set was divided evenly into ten subsets. Each of them is used separately for experiments.

4. Analysis of experimental results

Figure 7 presents sample plots of signals obtained in different stages of prediction process. A signal generated from sinus type function was taken as example. Figure 7a shows the input signal, whereas Figure 7b depicts processed signal. Figures 7c and 7d plot fuzzy signals from original input and from simulation of candidate FCM, respectively. The abbreviations located on plots legend are as follows. Letter *a* refers to *amplitude* signal, letter *b* to *change* signal, whereas letters *S*, *M*, and *H* correspond to different linguistic signal descriptors, i.e. *Small*, *Medium*, and *High*, respectively. For example, notation *aM&cH* means that amplitude signal is medium, and the change signal is high.

The results from Figure 7 show that candidate FCM is able to mimic the given fuzzy signal. Even though a certain level of inaccuracy is observable, trends and dependency between signals are preserved. Moreover, one has to take into account that the FCM model in this case should mimic the "behavior" of fuzzy signal rather than match its individual values. This avoids overfitting the training data, which could have negative impact on prediction accuracy.

The overall results obtained from all experiments are shown in

Table 1. The three sections of the table correspond to different types of input signal, i.e. hyperbolic, sinus, and IBM stock closing prices. Each section has four rows that stores results for varying training data length, i.e. 5, 10, 15 and 20. Columns correspond to number of predicted steps. Each cell includes two values. The first value is the average accuracy of prediction, whereas the second value in smaller font is the corresponding standard deviation. Figure 8 shows relationships between prediction accuracy and the number of prediction steps for each dataset and each length of input data.

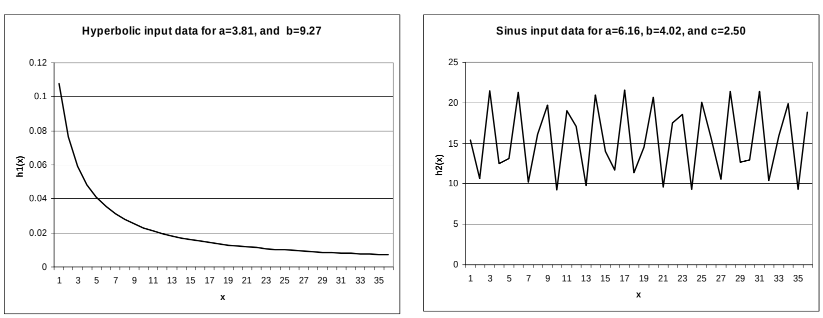


Figure 5. Sample input signal

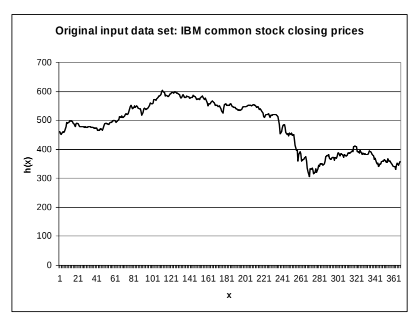
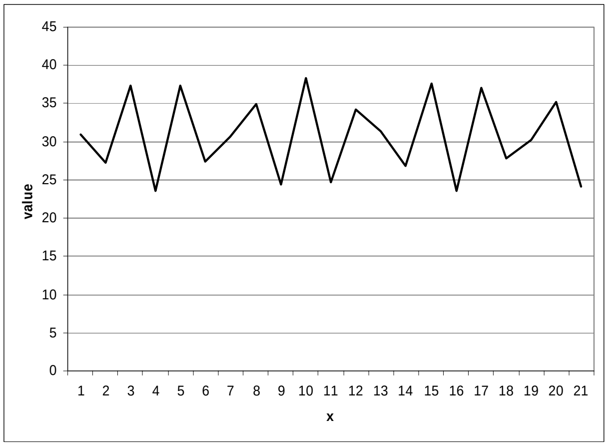
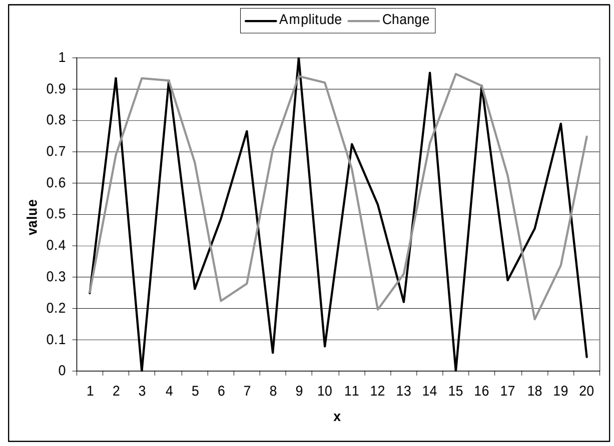


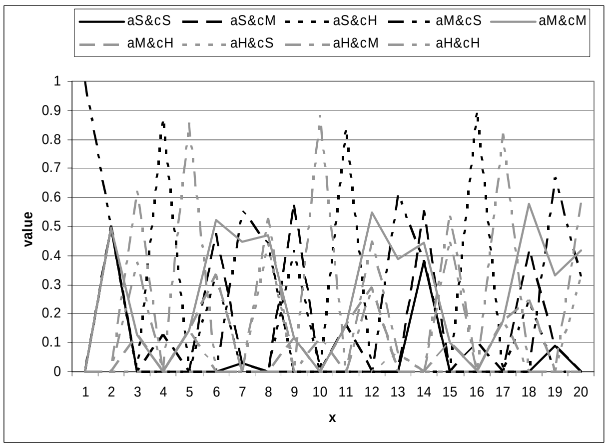
Figure 6. IBM stock prices data



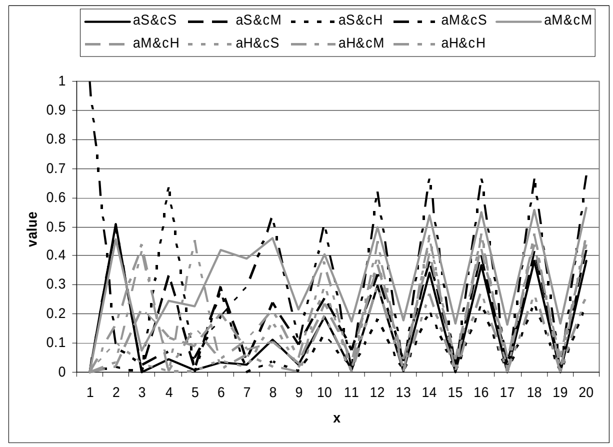
a) input signal



b) processed signal



c) fuzzy signal



d) fuzzy signal from candidate FCM simulation

Figure 7. Sample simulation result

Table 1. Experimental results

	Training data length	Number of prediction steps									
		1	2	3	4	5	6	7	8	9	10
hyperbolic	5	80.00±40.00	85.00±22.91	86.67±22.09	87.50±23.04	86.00±23.75	86.67±24.47	85.71±25.55	86.25±25.88	85.56±26.78	86.00±26.91
	10	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
	15	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
	20	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
sinus	5	60.00±48.99	45.00±41.53	33.33±33.33	30.00±33.17	26.00±25.38	25.00±20.05	22.86±18.28	20.00±16.00	17.78±14.21	19.00±13.75
	10	50.00±50.00	55.00±41.53	46.67±37.11	37.50±32.11	34.00±31.05	30.00±29.61	27.14±24.27	25.00±20.15	22.22±17.89	23.00±17.35
	15	80.00±40.00	80.00±33.17	70.00±31.43	60.00±32.02	54.00±28.35	45.00±23.62	40.00±21.93	40.00±20.00	40.00±19.97	38.00±18.87
	20	100.00±0.00	80.00±24.49	73.33±24.94	62.50±27.95	56.00±26.53	50.00±24.72	45.71±24.58	45.00±25.73	43.33±25.55	43.22±25.58
IBM stock	5	30.00±45.83	30.00±40.00	36.67±37.85	37.50±39.13	34.00±36.93	35.00±35.31	31.43±32.45	28.75±30.64	26.67±27.75	26.00±24.98
	10	40.00±48.99	45.00±35.00	50.00±37.26	50.00±38.73	46.00±36.93	40.00±32.65	37.14±28.71	38.75±24.64	37.78±23.41	37.55±24.02
	15	70.00±45.83	55.00±41.53	46.67±33.99	45.00±26.93	42.00±26.00	40.00±27.07	37.14±27.24	35.00±27.84	33.33±28.53	32.00±29.26
	20	100.00±0.00	85.00±22.91	73.33±29.05	70.00±31.22	68.00±33.70	68.33±32.86	65.71±33.32	63.75±35.10	60.00±32.65	55.00±34.00

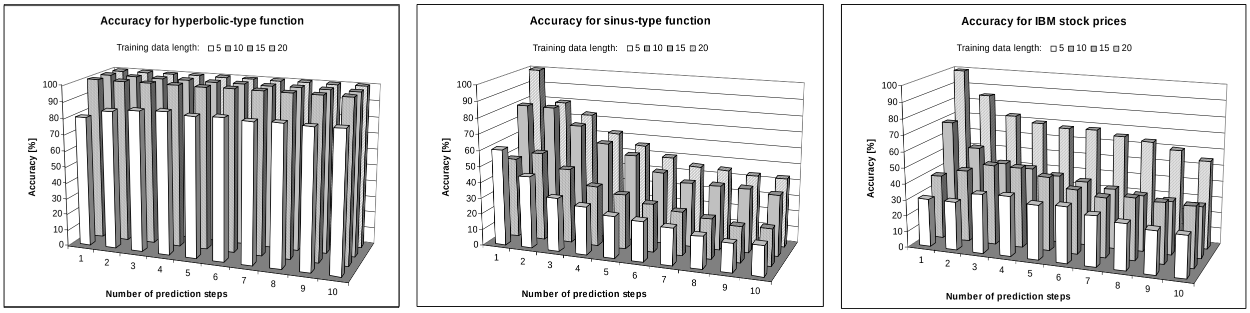


Figure 8. Average accuracy of prediction

We note that the quality of prediction for the first type of signal, i.e. hyperbolic one, is very high regardless of training data length and number of prediction step. This fact stems from simplicity and regularity of this type of signal.

However, two other cases are more interesting, since the input signals are more irregular. It can be seen that the accuracy of prediction depends on training data length and is higher for longer learning data set. In other words, the more information about the signal the system has, the more accurate model can be built, which agrees with our conclusions shown in [19]. For training data length equal to 5, the accuracy is in the 30% range and does not change significantly for bigger number of prediction steps. In order to put this number into perspective, we note that the baseline for prediction accuracy in this particular prediction problem, computed as a randomly chosen prediction, is approximately 11%. By increasing the training data length to 20, accuracy of 100% is obtained for one step prediction. This number slowly decreases along with increasing the prediction steps number until 40%-60% for training data length equal to 20. Considering the baseline, this result has very good quality showing applicability of the proposed methods to perform linguistic term prediction tasks.

5. Conclusions and Future Work

The paper introduces the framework for signal prediction on linguistic level using Fuzzy Cognitive Maps. Proposed approach incorporates recently introduced FCM learning method using genetic algorithm's based, RCGA algorithm.

Performed experiments are promising and show high quality of the proposed method. Prediction accuracy rate is very high in cases when the training signal length is long enough and the number of prediction steps is relatively low. The quality of prediction decreases gradually with the shorter input data length and the bigger number of prediction steps.

Future work will be focused on improving the prediction accuracy. One interesting research direction could be modification signal fuzzy representation, i.e. number of linguistic descriptors and/or their membership functions. The other area of exploration will be oriented towards application of this system to predict signal values instead of its linguistic descriptions, which involves adding defuzzification module.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

[1] J. Aguilar, "A Survey About Fuzzy Cognitive Maps Papers," *International Journal of Computational Cognition*, 3:2, 27-33, 2005

[2] R. Axelrod, *Structure of Decision: The Cognitive Maps of Political Elites*, *Princeton University Press*, 1976

[3] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel, *Time Series Analysis: Forecasting and Control*, *Prentice Hall*, 3rd Edition, 1994

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, *Addison-Wesley*, 1989

[5] F. Herrera, M. Lozano, and J.L. Verdegay, "Tackling Real-coded Genetic Algorithms: Operators and Tools for Behavioural Analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265-319, 1998

[6] A. Juditski, Q. Zhang, B. Delyon, P.Y. Clorennee, and A. Benvniste, "Wavelets in Identification: Wavelets, Splines, Neurons, Fuzzies,: How Good for Identification?", *INRIA reports*, No 2315, 1994

[7] T. Kailath, *Linear Systems*, *Prentice Hall*, 1980

[8] D. Kaplan, and L. Glass, *Understanding Nonlinear Dynamics, Springer Verlag*, 1995

[9] A. Karkanis, D. Karras, and B. Mertzios, "1-D Signal Prediction Using Wavelets", *Proc. of the 5th International Conference on Systems, Signals and Image Processing*, pp.20-23, Zagreb, Croatia, 1998

[10] M. Khan, and M. Quaddus, “Group Decision Support Using Fuzzy Cognitive Maps for Causal Reasoning”, *Group Decision and Negotiation Journal*, vol. 13, no. 5, 2004, in print

[11] B. Kosko, “Fuzzy Cognitive Maps”, *International Journal of Man-Machine Studies*, vol. 24, pp. 65-75, 1986

[12] K.C. Lee, W.J. Lee, O.B. Kwon, J.H. Han, and P.I. Yu, “Strategic Planning Simulation Based on Fuzzy Cognitive Map Knowledge and Differential Game”, *Simulation*, vol. 75, no. 5, pp. 316-327, 1998

[13] T.D. Ndousse, and T. Okuda, “Computational Intelligence for Distributed Fault Management in Networks Using Fuzzy Cognitive Maps,” *Proc. of IEEE Int. Conf on Communications Converging Techn. for Tomorrow’s Application*, 1558-1562, 1996

[14] A. Papoulis, *Probability, Random Variables and Stochastic Processes, McGraw Hill*, 1991

[15] W. Pedrycz, P.C.F. Lam, and A.F. Rocha, “Distributed Fuzzy System Modeling”, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 769-780

[16] C.E. Pelaez, and J.B. Bowles, “Applying Fuzzy Cognitive Maps Knowledge Representation to Failure Modes Effects Analysis,” *Proc. of the IEEE Annual Symposium on Reliability and Maintainability*, pp. 450-456, 1995

[17] W. Stach, and L. Kurgan, “Modeling software development project using fuzzy cognitive maps,” *Proc. of the 4th ASERC Workshop on Quantitative and Soft Software Engineering (QSSE'04)*, pp. 55-60, 2004

[18] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, “Genetic Learning of Fuzzy Cognitive Maps”, *Fuzzy Sets and Systems*, accepted, 2005

[19] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, “Learning Fuzzy Cognitive Maps with Required Precision Using Genetic Algorithm Approach,” *Electronics Letters*, vol. 40, no. 24, pp. 1519-1520, 2004

[20] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, “Parallel Fuzzy Cognitive Maps as a Tool for Modeling Software Development Project,” *Proc. of the 2004 North American Fuzzy Information Processing Society Conference (NAFIPS’04)*, pp. 28-33, Banff, AB, 2004

[21] M.A. Styblinski, and B.D. Meyer, “Signal Flow Graphs vs. Fuzzy Cognitive Maps in Application to Qualitative Circuit Analysis,” *International J. of Man-Machine Studies*, vol. 35, pp. 175-186, 1991

[22] R. Taber, “Knowledge Processing with Fuzzy Cognitive Maps,” *Expert Systems with Applications*, vol. 2, pp. 83-87, 1991