A SURVEY OF FUZZY COGNITIVE MAP LEARNING METHODS

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

1. Introduction

Fuzzy Cognitive Maps (FCMs) were originally introduced by Kosko [11] in 1986 as an extension of cognitive maps. They are a convenient modeling tool, usually categorized as a neuro-fuzzy method, for modeling and simulation of dynamic systems. One of their main advantages is an ability to incorporate and adapt human knowledge [18].

There are many techniques that can be used for modeling and analysis of dynamic systems. Generally speaking, they may be divided in to two groups such as quantitative and qualitative techniques [22]. The former one encompasses all quantitative methods that target both well-understood systems, (e.g., mathematical programming techniques of operation research) as well as those that ae less understood, e.g. statistically based data mining methods. The main restrictions of quantitative approaches originate from the fact that they require substantial effort and specialized knowledge from the outside of application domain in order to develop a correct model. In addition, some complex nonlinear systems cannot be modeled in this way. In a nutshell, quantitative modeling in some cases is difficult, costly, or even impossible [1]. The latter, alternative group includes qualitative approaches, which are free from the above restrictions. Modeling dynamic systems with the use of FCMs falls into this group. It is characterized by simplicity of both model representation and its execution. Furthermore, FCMs can easily incorporate human knowledge and adapt to a given domain.

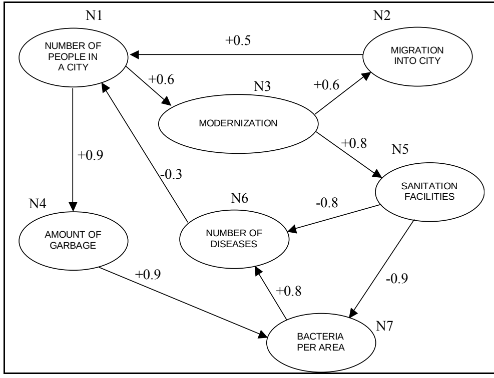
Applications of FCMs cover a wide range of research and industrial areas, such as electrical engineering, medicine, political science, international relations, military science, history, supervisory systems, etc. Examples of specific applications include diagnosis of diseases [30], analysis of electrical circuits [27], analysis of failure modes effects [19], fault management in distributed network environment [15], modeling and analysis of business performance indicators [8], modeling of supervisory systems [29], modeling of software development project [22] [24], modeling of plant control [6], modeling of political affairs in South Africa [10], modeling of virtual worlds [3], and protein sequence analysis [26]. According to literature research, a vast majority of FCM models were developed solely on the basis of expert(s) knowledge from a given domain [1]. This development technique takes advantage of FCM model representation, which is a simple graph, and model transparency. Expert-based development of FCM models was performed due to the lack of learning mechanisms that would allow for automated development of the models. As a result, FCM models were affected by subjectivity of human belief(s). In order to eliminate this inconvenience, several learning approaches have been recently proposed.

The remainder of this chapter is organized as follows. The first section delivers a brief introduction to FCMs. It includes historical background and elaborates on some working principles. The next section outlines and reviews techniques used in the development of FCMs. A short description of commonly used manual building of the FCMs is included, which is then followed by a presentation of different automated learning approaches. The last section presents some conclusions

2. History and Background

Cognitive maps were initially introduced by Robert Axelrod in 1976 and applied in political science [2]. They model a given system as a set of concepts and cause-effect relationships among them, which can be threefold: positive, negative, or neutral. Cognitive maps have a very simple representation consisting of a directed graph. Graph's nodes correspond to relevant concepts or variables in a given domain, whereas directed edges express mutual relationships between them. Type of relationship is determined by a sign that is associated with the edges. Positive sign stands for positive (promoting) type of relationship, negative sign for negative (inhibiting) one, and no edge means that concepts are neutral with each other. Positive relationship between two concepts describes a situation in which the source concept exerts promoting effect on the target one. This means that increase in source's value will lead to increase in target's value. Analogically, negative relationship expresses inhibitory effect, i.e. increase in source's value leads to decrease in target's value. No connection between two concepts means that they are independent. However, it turned out that modeling complex systems with the use of cognitive maps is infeasible, mainly due to insufficient representation of relationships.

Ten years later, in 1986, Kosko introduced fuzzy cognitive maps, which were extension to cognitive maps [11]. The most significant enhancement concerns relationships representation, which were fuzzified. This means that their description is enriched by numerical value instead of using only sign. This allows to applying varying degrees of causal influence. Relationship strengths are commonly normalized to the range [-1,1] [1]. Value of -1 corresponds to the strongest negative, whereas of +1 to the strongest positive relationship. Other values are used to express different fuzzy levels of influence. As stated above, a FCM model is represented by a graph. The difference, when compared with cognitive maps, is that each directed edge is associated with a number that expresses strength of a given relationship. Alternatively, model can be presented by a square matrix, called *connection matrix*. Each cell of this matrix stores a value of corresponding relationship. Commonly used convention is to place source nodes in rows and target nodes in columns. An example FCM graph, which describes model of city health issues [14] together with the connection matrix are shown in Figure 1.



	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 graph and its connection matrix

Once FCM model has been developed, it can be used to complete simulations by utilizing its execution model. The state of such model is determined by the values of all its concepts at a given time instant (iteration). Each value represents a degree to which the corresponding concept is active. Given the FCM consisting of *N* nodes, during every iteration the model's state is fully described by N-dimensional *state vector*. Values of concepts change as simulation goes on are governed by the following formula:

$$C_j(t+1) = f\left(\sum_{i=1}^{N} e_{ij} C_i(t)\right) \tag{1}$$

where $C_i(t)$ is the value of i^{th} node at the t^{th} iteration, e_{ij} is the edge weight (relationship strength) from the concept C_i to the concept C_j, *t* is the corresponding iteration, N is the number of concepts, and f is the transformation (transfer) function. Many researchers use a constraint where none of the concepts have feedback, i.e. $e_{ii}=0$ for i=1,…,N.

At the starting point of simulation one has to set the initial values of the state vector., Successive state vectors are calculated iteratively using (1). A transformation function is used to normalize concepts' values to a certain range. For most models reported in literature this range is [0,1]*. The values reflect degree of activation of a given concept. By applying a nonlinear transformation function quantitative analyses are being lost, but comparison of activation levels for different concepts is possible. Most commonly used examples are listed below:

- bivalent

$$f(x) = \begin{cases} 0, & x \le 0 \\ 1, & x > 0 \end{cases} \tag{2}$$

- trivalent

$$f(x) = \begin{cases} -1, & x \le -0.5 \\ 0, & -0.5 < x < 0.5 \\ 1, & x \ge 0.5 \end{cases} \tag{3}$$

- logistic signal

* Some researchers use [-1,1] range

$$f(x) = \frac{1}{1+e^{-Cx}} \qquad (4)$$

where C is a parameter used to determine the degree of fuzzification of the function[†].

Possible simulation outcomes and scenarios depend directly on applied type of transformation function. *Discrete-output* functions, e.g. (2) or (3), lead the simulation into either *hidden pattern* or *fixed-point attractor*. The former term refers to a situation, in which state vector becomes fixed at some iteration. The latter one describes a scenario, in which system keeps cycling between a fixed number of states. When the transformation function is *continuous-output* type, e.g. (4), it might result in *chaotic attractor*. This means that system produces different state vectors for successive iterations. Sample simulation outcome, which depicts hidden pattern, is shown in Figure 2.

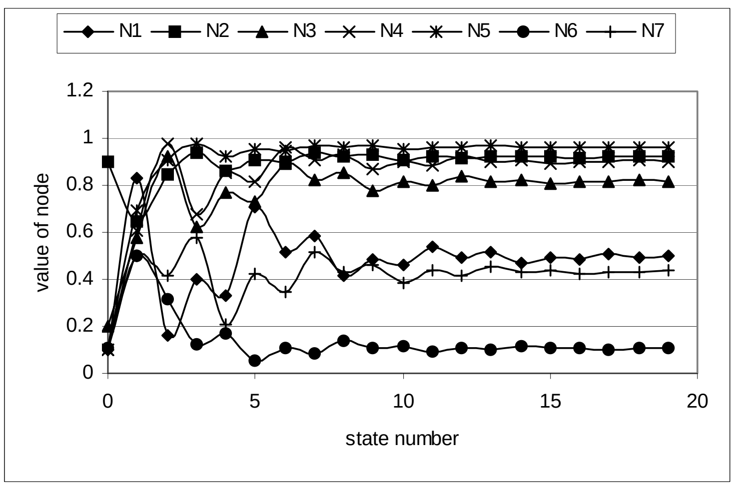


Figure 2 Sample simulation result

3. Development of FCM Model

There are two mainstreams of techniques for creation of FCM models. The first group covers techniques that exploit only human (expert) knowledge. We refer to them later as *manual methods*. For a long time this was practically the only way for establishing FCM models, mainly because of lack of automated or semi-automated approaches that would support this process. Recently several attempts were made to develop *computational methods*. They aim to substitute the expert and learn the model's structure from historical data.

The next subsection provides a short outline of manual methods. The disadvantages of these methods and motivation for automated methods are presented. This is followed by a survey of existing computational methods that allow learning FCM from data.

† In many practical applications it is equal to 5

3.1 Manual Methods for Development of FCM Models

Standard method of development of FCM models is based on expert knowledge in the area of application. One or more experts that are involved in this process design and implement a model manually based on their best knowledge and understanding of the modeled system. This approach usually consists of several steps [10]:

1. Identification of key domain issues or concepts.
2. Identification of causal relationships among these concepts.
3. Estimation of causal relationships strengths.

All of these steps are crucial and have to be performed carefully in order to develop a reliable model. The initial sketch of FCM is completed by the two initial steps, i.e. identification of concepts and relationships among them. The most convenient way to achieve this goal is by using pencil and paper to draw a corresponding graph. A set of nodes connected by directed edges is the outcome from this part. These two are relatively simple and straightforward compared to complexity of the third step. Estimation of causal relationships strengths, on the other hand, is very difficult because of relatively large number of numerical values that can be associated with each relationship. This number can be limited by imposing some restrictions, which reduce the possible number of combinations. Nevertheless, it has strong negative impact on model's accuracy, so a reasonable trade-off is desired. In practice, this is usually performed according to the following procedure [10][28]:

1. The influence of a concept on another between each pair of concepts is determined as "negative", "positive" or "neutral"
2. All relationships are expressed in fuzzy terms, e.g. very weak, weak, medium, strong and very strong
3. All established this way fuzzy expressions are mapped to numerical values, most frequently to the range from 0 to 1, for example

very weak	→	0.2
weak	→	0.4
medium	→	0.6
strong	→	0.8
very strong	→	1

Developing FCM model might be supported by analytical procedures, e.g. Analytical Hierarchy Process [20], during the assessment of fuzzy numerical values [21].

One of the advantages of FCM modeling is easiness of experts knowledge aggregation, i.e. a group of experts instead of one person can work on the model, which improves reliability of the final model [10]. Normally, each expert works on his or her FCM model separately and, later on, all models are combined together. Merging of the models can be performed using several different procedures [12], which are outside of the scope of this chapter.

Although the manual procedures for developing FCM are well-established, they have several drawbacks:

- they require expert knowledge, which has to be supported by knowledge of FCM methodology. Since the number of possible connections among concepts increases quadratically with increase of the number of concepts‡, expressing complex systems that consists of large number of nodes is often very difficult or even impossible to perform by a human. This may result in simplifications, which eventually translate into inaccuracy or incorrectness. The development process often requires many iterations and simulations before a suitable model is established. In case of group development, the quality of the final model can be improved by varying impact of a given expert model on the final model based on credibility of a particular expert. However, it requires additional parameters, such as credibility coefficients of each individual expert, which complicates the FCM development task.
- manual methods for development FCM models have also a major disadvantage of relying on human knowledge. It is very difficult to assess the model's accuracy in an unbiased way. What is more, even if there are available historical data to justify the model's quality, obtaining appropriate model that mimics the data requires laborious effort, which is performed by drawing and simulating successive models.

3.2 Automated and Semi-automated Methods for Learning of FSM Models

Problems associated with manual development of FCMs encourage researchers to work on automated or semi-automated computational methods for learning FCM structure, i.e. casual relationships (edges), and their strengths (weights) using historical data. Semi-automated methods still require a relatively limited human intervention, whereas fully automated approaches are able to compute a FCM model solely based on historical data, i.e. without any human interference.

The following paragraphs summarize the learning approaches that have been applied to FCMs. Literature research indicates that a number of algorithms for learning FCM model structure have been recently proposed. In general two main learning paradigms are used: Hebbian learning and genetic algorithms.

Dickerson and Kosko proposed simple Differential Hebbian Learning law (DHL) to be applied to learning FCMs [4]. This method is based on the law expressed by equation (5), which correlates changes of causal concepts.

$$\dot{e}_{ij} = -e_{ij} + \dot{C}_i \dot{C}_j \qquad (5)$$

where $\dot{e}_{ij}$ is the change of weight between concept i^{th} and jth e_{ij} is the current value of this weight, and $\dot{C}_i$ $\dot{C}_j$ are changes in concepts i^{th} and j^{th} values, respectively.

The learning process iteratively updates values of weights of all edges from the FCM graph until the desired structure is found. Considering that value of ΔC_i, which is defined as difference between i^{th} concept values in two successive states, ranges between -1 and 1, the C_i and C_j concept values increase or decrease at the same only when

‡ this number is equal to N^2, where N – number of concepts

$\Delta C_i \Delta C_j > 0$. If $\Delta C_i \Delta C_j < 0$ then one of the concept values decreases while the other one increases. In general, the weights of outgoing edges for a given concept node are modified when the corresponding concept value changes. The weights are updated according to the following formula:

$$e_{ij}(t+1) = \begin{cases} e_{ij}(t) + c_t \left[\Delta C_i \Delta C_j - e_{ij}(t) \right] & if\ \Delta C_i \neq 0 \\ e_{ij}(t) & if\ \Delta C_i = 0 \end{cases} \tag{6}$$

where e_{ij} denotes the weight of the edge between concepts C_i and C_j, ΔC_i represents the change in the C_i concept's activation value, t is the iteration number, and c_t is decreasing learning coefficient, e.g.

$$c_t = 0.1\left[1 - \frac{t}{1.1N}\right] \tag{7}$$

where t is the current iteration number, and the parameter N should be chosen to ensure the learning coefficient c_t never becomes negative. It is usually equal to the number of iterations or generations of observed states used for learning.

The results of experiments performed using this learning method were inconclusive. The main problem in this type of learning is that weights measure the causal-effect strength between two concepts C_i, C_j, and thus take into consideration only these two concepts. Moreover, it turned out that the learning process is highly sensitive to the order of data presentation.

In 2002, Vazquez presented an extension to DHL algorithm by introducing new rules to update edge values [31], see formula (8). This new algorithm was called Balanced Differential Algorithm (BDA).

$$e_{ij}(t+1) = \begin{cases} e_{ij}(t) + \dfrac{C_i}{N} & if\ i = j \\ e_{ij}(t) + c_t \left[\dfrac{\dfrac{\Delta C_i}{\Delta C_j}}{\sum\limits_{\substack{k=1 \\ \Delta C_i \Delta C_k > 0}}^{N} \dfrac{\Delta C_i}{\Delta C_k}} - e_{ij}(t) \right] & if\ i \neq j\ and\ \Delta C_i \Delta C_j > 0 \\ e_{ij}(t) + c_t \left[\dfrac{-\dfrac{\Delta C_i}{\Delta C_j}}{\sum\limits_{\substack{k=1 \\ \Delta C_i \Delta C_k < 0}}^{N} \dfrac{\Delta C_i}{\Delta C_k}} - e_{ij}(t) \right] & if\ i \neq j\ and\ \Delta C_i \Delta C_j < 0 \end{cases} \tag{8}$$

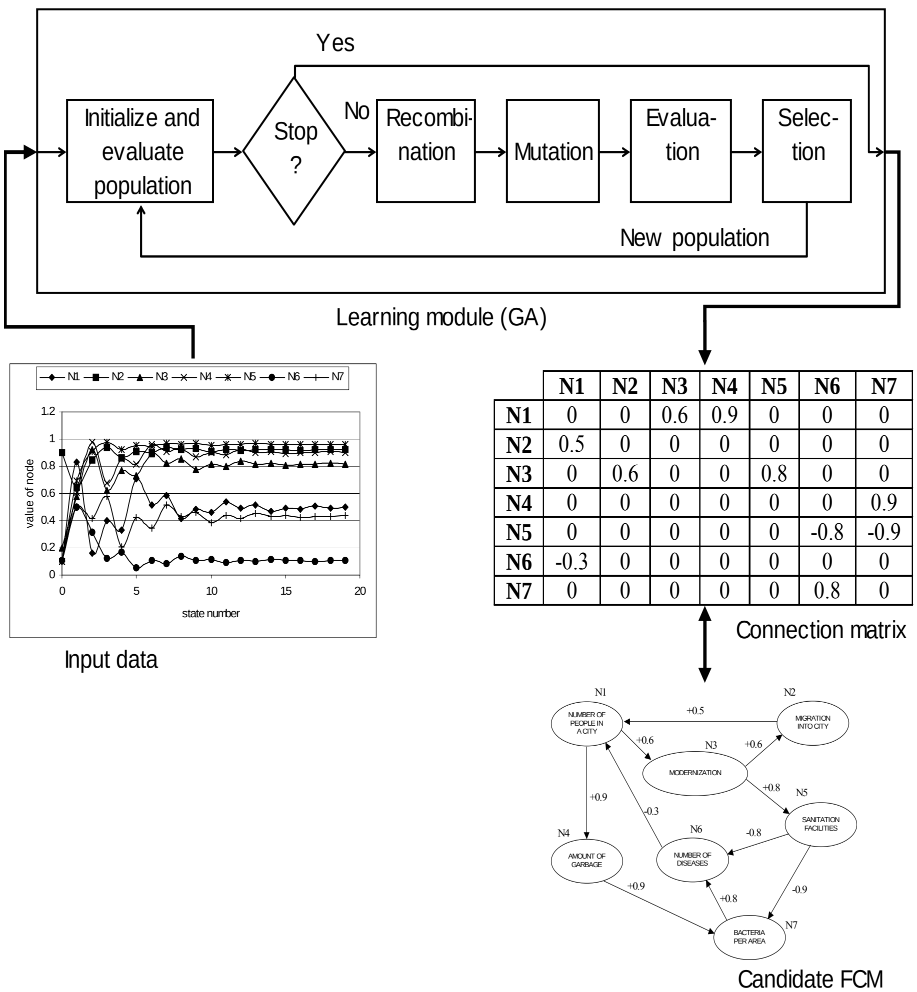
The new algorithm eliminates the limitation of DHL method where weight update for an edge connecting two concepts (nodes) is dependent only on the values of these two concepts. In BDA, during the learning process weights are updated taking into account all the concept values that change at the same time. This means, that formula for calculating $e_{ij}(t+1)$ takes into consideration not only changes ΔC_i and ΔC_j, but changes in all other concepts if they occur at the same iteration and in the same direction. The BDA algorithm was applied to learn structure of FCM models, which use bivalent transformation function, based on a historical data consisting of a sequence of state vectors. The goal was to develop FCM that is able to generate identical sequence of state vectors given the same initial state vector. The comparison results that are included in [31] proved that it improved learning quality compared to DHL method. However, proposed learning method was applied only to FCMs with binary concept values, which significantly restricts its application areas.

Another method based on Hebbian learning was proposed in 2003. Papageorgiou et al. developed an algorithm, called Nonlinear Hebbian Learning (NHL), to learn structure of FCMs [17]. The core of this method is a nonlinear extension to the basic Hebbian rule. This is semi-automated approach, since it requires initial human intervention. The main idea behind this method is to update weights associated only with edges that are initially suggested by expert(s), i.e. non-zero weights. Additionally, the experts have to indicate sign of each non-zero weight according to its physical interpretation. Weight values are updated synchronously, yet they have fixed signs for the entire learning process. As a result, the NHL algorithm allows obtaining model that retains structure, which is enforced by the expert(s), but at the same it requires human intervention before the learning process starts.

Active Hebbian Algorithm (AHL) introduced by Papageorgiu et al. in 2004 is the next attempt to help in FCM development [16]. This approach introduces and exploits the task of determination of the sequence of activation concepts. Expert(s) determines the desired set of concepts, initial structure and the interconnections of the FCM structure. In addition, they identify the sequence of activation concepts. A seven-step AHL procedure, which is based on Hebbian learning theory, is iteratively used to adjust the model (weights) to satisfy required stopping criteria. Mathematical formulation, implementation and analysis of AHL supported by examples can be found in [16]. The main disadvantage of this approach is that it still requires human intervention.

The other branch of computational methods for learning FCM structure involves application of genetic algorithms. In 2001, Koulouriotis et al. applied the Genetic Strategy (GS) to learn FCM's model structure, i.e. weights of relationships, from data [13]. In this method, the learning process is based on a collection of input/output pairs, which are called examples. Particular values of inputs and outputs depend on the designer's choice. Inputs are defined as the initial state vector values, whereas outputs are final state vector values, i.e. values of state vector after the FCM simulation terminates. For each learning session, the designer provides a set of input/output pairs, which requires historical data consisting of multiple sequences of state vectors. The learning algorithm computes structure of a FCM that is able to generate state vector sequences that transform the input vectors into the output vectors. Its main drawback is the need for multiple state vector sequences, which might be difficult to obtain for many real-life problems.

Parsopoulos et al. in 2003 applied Particle Swarm Optimization (PSO) method, which belongs to the class of Swarm Intelligence algorithms, to learn FCM structure based on a historical data consisting of a sequence of state vectors that leads to a desired fixed-point attractor state [18]. PSO is a population based algorithm, which goal is to perform a search by maintaining and transforming a population of individuals. This method improves the quality of resulting FCM model by minimizing an objective function. The function incorporates human knowledge by adequate constraints, which guarantee that relationships within the model will retain the physical meaning defined by expert(s). An example of application this method to industrial control problem is presented in [18].



	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 3 High-level diagram of the RCGA learning method

Another state-of-the-art learning method for FCMs, introduced by Stach et al. in 2005 [25], applies real-coded genetic algorithm (RCGA) to develop FCM model from a set of historical data. The core of this approach is learning module, which exploits RCGA to find FCM structure that is capable to mimic given historical data. Figure 3 shows a high-level diagram of this method.

The RCGA learning method is fully automated. Based on historical data given as a time series (called input data) it establishes the FCM model (called candidate FCM), which is able to mimic the data. This approach is very flexible in terms of input data: it can use either one time series or multiple sets of concepts values over successive iterations. The central part of this method is real-coded genetic algorithm, which is a floating-point extension to genetic algorithm [5] [7]. The extension mainly concerns chromosome representation. In this case, each chromosome consists of floating point numbers that correspond to the problem variables, which makes this approach suitable for optimization problems with continuous variables. The RCGA learning approach was intensively tested in [25], and experiments proved its effectiveness and high quality. The follow-up of this work presented in [23] includes analysis of RCGA learning quality depending on the available size of historical data. It shows that given input data of sufficient size, the method can generate FCM models that are identical to models developed by domain experts. Also, it shows that increasing size of input data improves accuracy of learning and that insufficient size of input data may result in poor quality of learning. In the latter case multiple different models that mimic input data of small size can be generated, and most of them fail to provide accurate results for experiments with new initial conditions, which were unseen during learning.

A different learning objective for FCM was presented by Khan and Chong in 2003 [9]. Instead of learning structure of FCM model, their aim was to find the initial state vector (initial conditions) that leads a given model to the desired fixed-point attractor or limit cycle. Genetic algorithms approach was used as a core of this method. This type of analysis might be especially useful to support decision-making processes.

Table 1 summarizes discussed learning approaches [25]. The comparison is made based on several factors, such as the learning goal, involvement of a domain expert, input historical data, and learning strategy type.

In summary, research on learning FCM models from data has resulted in a number of alternative approaches. One group of methods is aimed at providing a supplement tool that would help expert(s) to develop accurate model based on his or her knowledge about modeled system. Algorithms from the other group are oriented toward eliminating human from the entire development process, i.e. only historical data are necessary to establish FCM model. Initially proposed methodologies took advantage of Hebbian learning, whereas recently genetic algorithms gain the momentum.

Table 1 Overview of learning approaches applied to FCMs

Algorithm	Reference	Learning goal 1)	Requires human intervention	Type of data used 2)	Learning type
DHL	[4]	Connection matrix	No	Single	Hebbian
BDA	[31]	Connection matrix	No	Single	modified Hebbian
NHL	[17]	Connection matrix	Yes&No2)	Single	modified Hebbian
AHL	[16]	Connection matrix	Yes&No2)	Single	modified Hebbian
GS	[13]	Connection matrix	No	Multiple	Genetic
PSO	[18]	Connection matrix	No	Multiple	Swarm
RCGA	[25]	Connection matrix	No	Single	Genetic
GA	[9]	Initial vector	N/A	N/A	Genetic

1) *Single* – historical data consisting of one sequence of state vectors, *Multiple* – historical data consisting of several sequences of state vectors for different initial conditions

2) Initial human intervention is necessary but later when applying the algorithm there is no human intervention needed

4. Summary and Conclusions

In the recent years, FCMs have gained a well-deserved research interest. Numerous examples of successful applications of this technique in many different research and industrial areas demonstrate the usefulness of the concept. The unquestionable advantages of FCMs lie in the simplicity and adaptability to a certain application domain. However, it seems that their further development and is somewhat constrained by deficiencies that are present in their underlying theoretical framework. Disadvantages related to the manual development recently encouraged researchers to work on semi-automated or automated tools for learning FCM models from historical data.

Two main directions of research related to learning FCM have been proposed so far. The first group includes approaches that are based on Hebbian learning rule. A differential Hebbian learning algorithm was the first one applied to learn FCM. It served as a cornerstone for a number of algorithms that are based on a modification of the learning rule that it introduced. The other direction exploits genetic algorithm based methods. Several attempts with different algorithms, including genetic algorithm, particle swarm, and most recently real-coded genetic algorithm, have been performed so far. The results are very promising and encouraging further research and applications.

Even though the first step towards automation of development fuzzy cognitive map from data was done, there are still problems that need to be overcome. The major challenge is the scalability of learning methods for FCM. None of the above approaches has been tested and reported as capable to handle models consisted of a dozen or so or a few dozen of concepts. They were applied only to relatively small models, i.e. consisting of up to 10 nodes. Another issue is the physical interpretation of learned model. When constructed by an expert, all relationships can be explained and justified by the designer(s) on the basis of his or her knowledge. This is one of the key advantages of this modeling technique. Fully automated learning methods often generate models that have relationships directions and/or values hard to explain in a justifiable way, although they provide accurate simulation results.

5. References

[1] Aguilar J., A Survey about Fuzzy Cognitive Maps Papers (Invited Paper), *International Journal of Computational Cognition,* vol. 3, no. 2, pp. 27-33, 2005

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

[3] Dickerson J., A., Kosko B., Fuzzy Virtual Worlds, *Artificial Intelligence Expert*, vol. 7, pp. 25-31, 1994

[4] Dickerson J., A., Kosko B., Virtual Worlds as Fuzzy Cognitive Maps, *Presence*, vol. 3, no. 2, pp. 173-189, 1994

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

[6] Gotoh K., Murakami J., Yamaguchi T., Yamanaka Y., Application of Fuzzy Cognitive Maps to Supporting for Plant Control, *Proceedings of the SICE Joint Symposium of Fifteenth Systems Symposium and Tenth Knowledge Engineering Symposium*, pp. 99-104, 1989

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

[8] Kardaras D., Mentzas G., Using Fuzzy Cognitive Maps to Model and Analyse Business Performance Assessment, *Advances in Industrial Engineering Applications and Practice II, J. Chen, and A. Mital, (Eds)*, pp. 63-68, 1997

[9] Khan M., S., Chong A., Fuzzy Cognitive Map Analysis with Genetic Algorithm, *Proceedings of the 1st Indian International Conference on Artificial Intelligence (IICAI-03)*, 2003

[10] Khan, M., Quaddus, M., Group Decision Support using Fuzzy Cognitive Maps for Causal Reasoning, *Group Decision and Negotiation Journal*, vol. 13, no. 5, pp. 463-480, 2004

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

[12] Kosko B., Hidden Patterns in Combined and Adaptive Knowledge Networks, *International Journal of Approximate Reasoning,* vol. 2, pp. 377-393, 1988

[13] Koulouriotis D., E., Diakoulakis I., E., Emiris D., M., Learning Fuzzy Cognitive Maps using Evolution Strategies: a Novel Schema for Modeling and Simulating High-Level Behavior, *IEEE Congress on Evolutionary Computation (CEC2001)*, pp. 364-371, 2001

[14] Lee K., C., Lee W., J., Kwon O., B., Han J., H., Yu P., I., Strategic Planning Simulation Based on Fuzzy Cognitive Map Knowledge and Differential Game, *Simulation*, vol. 71, no. 5, pp. 316-327, 1998

[15] Ndousse T., D., Okuda T., Computational Intelligence for Distributed Fault Management in Networks Using Fuzzy Cognitive Maps, *Proceedings of the IEEE International Conference on Communications Converging Technologies for Tomorrow's Application*, pp. 1558-1562, 1996

[16] Papageorgiou E., Stylios C., D., Groumpos P., P., Active Hebbian Learning Algorithm to Train Fuzzy Cognitive Maps, *International Journal of Approximate Reasoning*, vol. 37, no. 3, pp. 219-249, 2004

[17] Papageorgiou E., Stylios C., D., Groumpos P., P., Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule, *Australian Conference on Artificial Intelligence*, pp. 256-268, 2003

[18] Parsopoulos, K., E., Papageorgiou, E., I., Groumpos, P., P., Vrahatis, M., N., A First Study of Fuzzy Cognitive Maps Learning Using Particle Swarm Optimization, *Proceedings of the IEEE 2003 Congress on Evolutionary Computation*, pp. 1440-1447, 2003

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

[20] Saaty T., L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, 1980

[21] Schneider M., Shnaider E., Kandel A. Chew G., Automatic Construction of FCMs, *Fuzzy Sets and Systems*, vol. 93, no. 2, pp. 161-172, 1998

[22] Stach W., Kurgan L., Modeling Software Development Project using Fuzzy Cognitive Maps, *Proceedings of the 4th ASERC Workshop on Quantitative and Soft Software Engineering (QSSE'04)*, pp. 55-60, 2004

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

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

[25] Stach, W., Kurgan, L., Pedrycz, W., and Reformat, M., Genetic Learning of Fuzzy Cognitive Maps, *Fuzzy Sets and Systems*, in print, 2005

[26] Stach, W., Kurgan, L., and Ruan, J., Fuzzy Cognitive Maps Based Analysis of Hydrophobic Scale for Protein Secondary Content and Structure Prediction, *Neural Network World*, special issue on *Biologically Inspired Computing and Computers in Biology*, submitted, 2005

[27] Styblinski M., A., Meyer B., D., Signal Flow Graphs versus Fuzzy Cognitive Maps in Application to Qualitative Circuit Analysis, *International Journal of Man-Machine Studies*, vol. 35, pp.175-186, 1991

[28] Stylios C., D., and Groumpos P., P., Fuzzy Cognitive Map in Modeling Supervisory Control Systems, *Journal of Intelligent and Fuzzy Systems*, vol. 8, no. 2, pp. 83-98, 2000

[29] Stylios C., D., and Groumpos P., P., The Challenge of Modelling Supervisory Systems using Fuzzy Cognitive Maps, *Journal of Intelligent Manufacturing*, vol. 9, no. 4, pp. 339-345, 1998

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

[31] Vazquez A., *A Balanced Differential Learning algorithm in Fuzzy Cognitive Maps*, Technical Report, Departament de Llenguatges I Sistemes Informatics, Universitat Politecnica de Catalunya (UPC), 2002