Brill Academic Publishers
P.O. Box 9000, 2300 PA Leiden,
The Netherlands

Lecture Series on Computer and Computational Sciences
Volume 1, 2005, pp. 1-4

Fuzzy Cognitive Maps Learning using Memetic Algorithms

Y.G. Petalas[1], E.I. Papageorgiou[2], K.E. Parsopoulos[1], P.P. Groumpos[2], and M.N. Vrahatis[1]

[1]Computational Intelligence Laboratory, Department of Mathematics, University of Patras Artificial Intelligence Center(UPAIRC) University of Patras, GR-26110 Patras, Greece.

[2]Department of Electrical and Computer Engineering, Laboratory of Automation and Robotics, University of Patras, GR-26500 Patras, Greece.

Abstract: Memetic Algorithms (MAs) are proposed for learning in Fuzzy Cognitive Maps (FCMs). MAs are hybrid search schemes, which combine a global optimization algorithm and a local search one. FCM's learning is accomplished through the optimization of an objective function with respect to the weights of the FCM. MAs are used to solve this optimization task. The proposed approach is applied to a well-established process control problem in industry and the results are promising.

Keywords: Fuzzy Cognitive Maps, Memetic Algorithms, Particle Swarm Optimization, Local Search

Mathematics Subject Classification: 03B52 90C30

1 Introduction

Fuzzy Cognitive Maps (FCMs) are a soft computing methodology developed by Kosko as an expansion of cognitive maps which are widely used to represent social scientific knowledge [1]. They belong to the class of neuro–fuzzy systems, which are able to incorporate human knowledge and adapt it through learning procedures. FCMs are designed by experts through an interactive procedure of knowledge acquisition, and they have a wide field of application, including modeling of complex and intelligent systems, decision analysis, and extend graph behavior analysis. They have also been used for planning and decision–making in the fields of international relations and social systems modeling, as well as in management science, operations research and organizational behavior [1, 2].

An FCM consists of nodes–concepts, C_i, $i = 1, \ldots, N$, where N is the total number of concepts. Each node–concept represents a key–factor of the system, and it is characterized by a value $A_i \in [0, 1]$, $\quad i = 1, \ldots, N$. The concepts are interconnected with weighted arcs, which imply the relations among them. Each interconnection between two concepts C_i and C_j has a weight W_{ij}, which is proportional to the strength of the causal link between C_i and C_j. At each step, the value, A_i, of the concept C_i is influenced by the values of the concepts–nodes connected to it, and it is updated according to the scheme [2]: $A_i(k+1) = f\left(A_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{n} W_{ji}A_j(k)\right)$, where k stands for the iteration counter; and W_{ji} is the weight of the arc connecting the concept C_j to the concept C_i. The function f is the sigmoid function.

A few algorithms have been proposed for FCM learning [3]. The main task of the learning procedure is to find a setting of the FCM's weights, that leads the FCM to a desired steady state. This is achieved through the minimization of a properly defined objective function.

We propose a new approach for FCM learning that is based on Memetic Algorithms (MAs) [4]. MAs are hybrid search schemes, integrating an evolutionary algorithm and a local search method, and they have been used with success in many difficult optimization problems. Their efficiency can be attributed to the exploitation of the advantages of both the global and the local search schemes. Global search algorithms can explore the whole search space but they are not efficient in locating the optimum of the objective function with high accuracy. On the other hand, local search methods can compute the optimum with high accuracy if they are initialized in its basis of attraction, but they are prone to getting stuck to local minima. MAs are used for the determination of optimum weight matrices for the system through the minimization of a properly defined objective function [3].

The rest of the paper is organized as follows: In Section 2, the proposed learning algorithm is presented, while results from the application of the proposed method in a industrial control problem are reported in Section 3. Section 4 concludes the paper.

2 The Proposed Approach

The main goal in FCM learning is to determine the values of the weights of the FCM that produce a desired behavior of the system. The determination of the weights is of major significance and it contributes towards the establishment of FCMs as a robust methodology. The desired behavior of the system is characterized by output concept values that lie within desired bounds prespecified by the experts.

The computation of the FCM's weights is accomplished through the minimization of a problem–dependent objective function. For this purpose, the following objective function was employed [3]:

$$F(W) = \sum_{i=1}^{m} H\left(A_{\mathrm{out}_i}^{\min} - A_{\mathrm{out}_i}\right)\left|A_{\mathrm{out}_i}^{\min} - A_{\mathrm{out}_i}\right| + \sum_{i=1}^{m} H\left(A_{\mathrm{out}_i} - A_{\mathrm{out}_i}^{\max}\right)\left|A_{\mathrm{out}_i}^{\max} - A_{\mathrm{out}_i}\right|, \qquad (1)$$

where H is the well–known Heaviside function and $A_{\mathrm{out}_i}^{\min}$, $A_{\mathrm{out}_i}^{\max}$, are bounds of the output concepts' values. This function has been used with success in the past, combined with a swarm intelligence algorithm, for FCM learning [3]. In the proposed approach, we use MAs to solve this optimization problem.

The proposed MA, called MemeticPSO (MPSO), consists of the Particle Swarm Optimization (PSO) algorithm as the global search component, and the Hooke and Jeeves (HJ) algorithm as local search component. PSO is a stochastic global optimization algorithm and it has been applied successfully for FCM's learning [3]. More specifically, it belongs to the class of *swarm intelligence* algorithms, which are inspired from the social dynamics and emergent behavior that arise in socially organized colonies. A brief description of PSO is provided below.

Assume a D–dimensional search space, $S \subset \mathbb{R}^D$, and a swarm consisting of N particles. Let $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^\top \in S$, be the i–th particle and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^\top \in S$, be its velocity. Let also the best previous position (i.e., the position that has the lowest function value) encountered by the i–th particle in S be denoted by $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^\top$. Assume g_i to be the index of the particle that attained the best previous position among all the particles in the neighborhood of the i–th particle, and G to be the iteration counter. Then, the swarm is manipulated by the equations [5]:

$$\begin{aligned} V_i(G+1) &= \chi\Big[V_i(G) + c_1\, r_1\big(P_i(G) - X_i(G)\big) + c_2\, r_2\big(P_{g_i}(G) - X_i(G)\big)\Big], && (2)\\ X_i(G+1) &= X_i(G) + V_i(G+1), && (3) \end{aligned}$$

Table 1: Pseudo code for the Memetic algorithm.

Input: N , χ, c_1, c_2, $w_{\min}$, $w_{\max}$ (lower & upper bounds)	
Step 1	**Set** $t = 0$.
Step 2	**Initialize** $w_i(t), v_i(t) \in [w_{\min}, w_{\max}]$, $p_i(t) \leftarrow w_i(t)$, $i = 1, \ldots, N$.
Step 3	**Evaluate** $F(w_i(t))$. **Determine** the indices g_i, $i = 1, \ldots, N$.
Step 4	**While** (stopping criterion is not satisfied) **Do**
Step 5	**Update** the velocities $v_i(t+1)$, $i = 1, \ldots, N$, according to Eq. (2).
Step 6	**Set** $w_i(t+1) = w_i(t) + v_i(t+1)$, $i = 1, \ldots, N$.
Step 7	**Constrain** each particle w_i in $[w_{\min}, w_{\max}]$.
Step 8	**Evaluate** $f(w_i(t+1))$, $i = 1, \ldots, N$.
Step 9	**If** $f(w_i(t+1)) < f(p_i(t))$ **Then** $p_i(t+1) \leftarrow w_i(t+1)$
	Else $p_i(t+1) \leftarrow p_i(t)$.
Step 10	**Update** the indices g_i.
Step 11	**While** (local search is applied) **Do**
Step 12	**Choose** a best position, $p_q(t+1)$, $q \in \{1, \ldots, N\}$.
Step 13	**Apply** local search on $p_q(t+1)$ and obtain a new solution y.
Step 14	**If** $F(y) < F(p_q(t+1))$ **Then** $p_q(t+1) \leftarrow y$.
Step 15	**End While**
Step 16	**Set** $t = t + 1$.
Step 17	**End While**

where $i = 1, \ldots, N$; χ is a parameter called *constriction coefficient*; c_1 and c_2 are two parameters called *cognitive* and *social* parameters, respectively; and r_1, r_2, are random vectors with components uniformly distributed within $[0, 1]$ (all vector operations in Eqs. (2) and (3) are assumed to be performed componentwise).

HJ is a direct search algorithm that uses function evaluations solely, without computing any derivative information [6]. Therefore, it can be applied in problems with non–differentiable or discontinuous objective functions. A pseudocode of the proposed methodology is provided in Table 1, where F denotes the objective function, and w denotes the matrix W of the FCM's weights, represented as a vector that contains its rows in turn.

3 Experimental Results

The proposed method has been applied to the industrial control problem investigated in [3]. The ranges of the weights implied by the fuzzy regions, as they were suggested by experts, were: $-0.50 \leqslant W_{12} \leqslant -0.30$, $-0.40 \leqslant W_{13} \leqslant -0.20$, $0.20 \leqslant W_{15} \leqslant 0.40$, $0.30 \leqslant W_{21} \leqslant 0.40$, $0.40 \leqslant W_{31} \leqslant 0.50$, $-1.0 \leqslant W_{41} \leqslant -0.80$, $0.50 \leqslant W_{52} \leqslant 0.70$, $0.20 \leqslant W_{54} \leqslant 0.40$. Since the consideration of all eight constraints on the weights prohibits the detection of a suboptimal matrix, some of the constraints were omitted. More specifically, the constraints for the weights W_{15}, W_{52}, and W_{54}, for which, the experts' suggestions regarding their values varied widely, were omitted. The corresponding weights were allowed to assume values in the range $[-1, 0]$ or $[0, 1]$, in order to avoid physically meaningless weight matrices [3]. The results obtained through MPSO were compared with that of PSO that are reported in [3]. We performed 100 independent experiments. The error goal for the optimization problem was set to 10^{-8}, and the swarm size of MPSO was set to 20. The HJ algorithm was applied on the best particle of the swarm with probability 0.05 at each iteration. In all cases, the local version of PSO was used, with neighborhood size 3. The performance of PSO and MPSO is analyzed statistically in Table 2, with respect to the required number of function evaluations. In Table 3, statistics regarding the weights are reported. The results suggest that MPSO is a very promising approach for FCM learning.

Table 2: Statistical analysis for the function evaluations required by PSO and MPSO.

	Max	**Min**	**Mean**	**Stdev**
PSO	760	240	491.20	104.16
MPSO	1071	40	315.51	155.07

Table 3: Statistical analysis for the weights obtained with PSO and MPSO.

	PSO				MPSO			
	Max	**Min**	**Mean**	**Stdev**	**Max**	**Min**	**Mean**	**Stdev**
W_{12}	-0.3000	-0.5000	-0.3369	0.0655	-0.3000	-0.5000	-0.3444	0.0705
W_{13}	-0.2000	-0.3068	-0.2100	0.0231	-0.2000	-0.3000	-0.2165	0.0306
W_{15}	1.0000	0.7163	0.902348	0.0900	1.0000	0.7166	0.9323	0.0789
W_{21}	0.4000	0.3903	0.399828	0.0011	0.4000	0.3991	0.3999	0.0000
W_{31}	0.5000	0.4843	0.4998	0.0015	0.5000	0.5000	0.5000	0.0000
W_{41}	-0.8000	-0.8000	-0.8000	0.0000	-0.8000	-0.8000	-0.8000	0.0000
W_{52}	1.0000	0.8272	0.9309	0.0619	1.0000	0.8224	0.9525	0.0594
W_{54}	0.1591	0.1000	0.1064	0.0136	0.1614	0.1000	0.1068	0.0138

4 Conclusions

A new learning algorithm, which is based on MAs, was proposed for determining the weight matrix of an FCM. MPSO proved to be very efficient, providing promising results. Future work will include the application of MAs on more complex problems, as well as the investigation of different memetic schemes for FCM learning.

Acknowledgment

This work was supported by the ”Pythagoras II” research grant co funded by the European Social Fund and National Resources.

References

[1] B. Kosko. Fuzzy cognitive maps. *Int. J. Man–Machine Studies*, **24** 65–75(1986).

[2] B. Kosko. *Fuzzy Engineering*. Prentice Hall, New York, 1997.

[3] E. I. Papageorgiou, K.E Parsopoulos, C.D. Stylios, P. P. Groumpos, and M. N. Vrahatis. Fuzzy cognitive maps learning using particle swarm optimization. *Journal of Intelligent Information Systems*, 2004. accepted for publication.

[4] D. Corne, M. Dorigo, and F. Glover. *New Ideas in Optimization*. McGraw-Hill, 1999.

[5] M. Clerc and J. Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.*, **6(1)** 58–73(2002).

[6] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, **8** 212–229(1961).