Differential Hebbian Learning in Fuzzy Cognitive Maps: A Methodological View in the Decision Support Perspective

Alex Chong, M. Shamim Khan, Tom Gedeon
School of Information Technology
Murdoch University

School of Information Technology, Murdoch University, Perth, WA 6150
E-mail: alexchong_@hotmail.com, {s.khan, t.gedeon} @murdoch.edu.au

Abstract

Differential Hebbian Learning (DHL) was proposed by Kosko as an unsupervised learning scheme for a Fuzzy Cognitive Map (FCM). DHL can be used with a sequence of state vectors to adapt the causal link strengths of an FCM. However, it does not guarantee learning of the sequence by the FCM. While the possible use of DHL in automating the development of FCMs has been addressed, there has been no known attempt to outline concrete procedures for the use of DHL for this purpose. In an attempt to devise an effective DHL scheme, a formal methodology is proposed for using DHL in the development of FCMs in a decision support context. The four steps in the methodology are: (1) Creation of a crisp cognitive map; (2) Identification of groups of events that occur simultaneously or sequentially; (3) Event sequence encoding using DHL; (4) Revision of the trained FCM. Feasibility of the proposed methodology is demonstrated with an example involving a dynamic system with feedback based on a real-life scenario.

1 Introduction

Cognitive maps are signed directed graphs designed to aid decision making [1]. The nodes in these digraphs represent concepts or variables relevant to a given domain. The causal links between these concepts are represented by the edges. The edges are directed and signed to show the direction and nature of influence. An edge directed from any node *A* to another node *B*, can be positive to indicate a promoting effect, or negative to represent an inhibitory effect of *A* upon *B*. Cognitive maps can be pictured as a form of signed directed graph. Figure 1 shows a cognitive map used to represent a scenario involving some issues in public health.

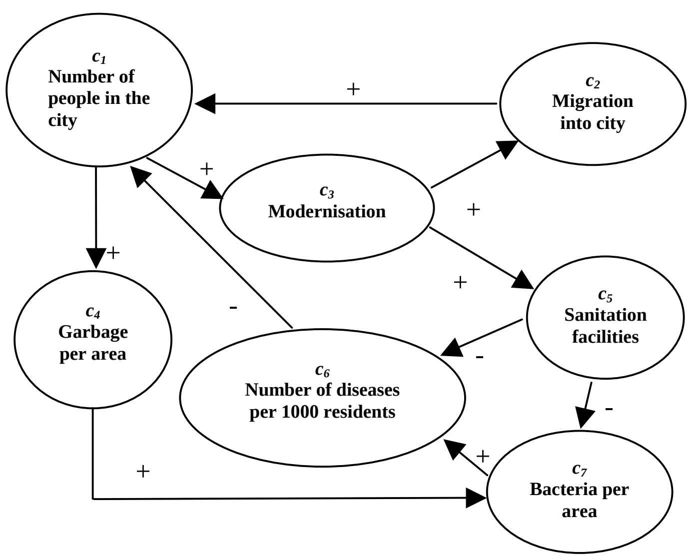


Figure 1. A cognitive map concerning some public health issues [9].

The main objective of building a cognitive map around a problem is to be able to predict the outcome by letting the relevant issues interact with one another. These predictions can be used for finding out whether a decision made by someone is consistent with the whole collection of stated causal assertions.

In a cognitive map, the effect of a node *A* on another node *B*, linked directly or indirectly to it, is given by the number of negative edges forming the path between the two nodes. The effect is positive if the path has an even number of negative edges, and negative otherwise. It is possible for more than one such path to exist. If the effects from these paths is a mix of positive and negative influences, the map is said to have an imbalance and the net effect of node *A* on node *B* is indeterminate. This calls for the assignment of some sort of weight to each inter-node causal link, and a framework for evaluating combined effects using these numerically weighted edges. Fuzzy cognitive maps (FCM) [2] were proposed as an extension of cognitive maps to provide such a framework. FCMs are an improvement on cognitive maps with the addition of four significant characteristics:

(1) Edges between nodes can take on numeric values representing degrees of causality.
(2) FCMs can model complex real-life scenarios, which are dynamic systems that evolve with time through feedback. The effect of change in a concept node affects other nodes, which in turn can affect the node initiating the change and so on. In this respect, FCMs may be viewed as recurrent artificial neural networks.
(3) The knowledge-base stored in an FCM can be augmented by combining a number of FCMs using rules of fuzzy union.
(4) Like artificial neural networks, FCMs can be adaptive, ie, the causal link strengths (initially specified by human experts) can be fine-tuned through learning.

The edge strength between two nodes c_i and c_j may be denoted by e_{ij}, with $e_{ij}, \in$ [-1,1]. Values –1 and 1 represent respectively full negative and full positive causality, zero denotes no causal effects and all other values correspond to different fuzzy levels of causal effects. In general, an FCM is described by a *edge or connection matrix E* whose elements are the connection strengths (or weights) e_{ij}. The element in the i^{th} row and j^{th} column of matrix *E* represents the connection strength of the link directed out of node c_i and into c_j . If the value of this link takes on discrete values in the set {-1, 0, 1}, it is called a simple FCM (example in Figure 3). The concept values of nodes $c_1, c_2, \ldots , c_n$ (where *n* is the number of concepts in the problem domain) together represent the state vector *C*.

An FCM state vector at any point in time gives a snapshot of events (concepts) in the scenario being modelled. In the example FCM shown in Figure 2, node c_2 relates to the 2nd component of the state vector and the state [0 1 0 0 0 0 0] indicates the event "freeway congestion" has happened. To let the system evolve, the state vector *C* is passed repeatedly through the FCM connection matrix *E*. This involves multiplying *C* by *E*, and then transforming the result as follows:

$$C(t + 1) \quad = \quad F[C(t) \,.\, E] \qquad\qquad (1)$$

where $C(t)$ is the state vector of concepts at some discrete time *t*, *F* is a nonlinear transformation function, and *E* is the FCM connection matrix.

With a hard-limiting transformation function, the FCM reaches either one of two states after a number of passes. It stabilises to a fixed pattern of node values - the so-called *hidden pattern* or *fixed-point attractor*. Alternatively, it keeps cycling between a number of fixed states - known as the *limit cycle*. In a decision support environment, a fixed point attractor can provide straightforward answers to causal “what if” questions. The equilibrium state can be used to predict the future state of the system being modelled by the FCM for a particular initial state. A limit cycle provides the user with a deterministic behaviour of the real-life situation being modelled. It allows the prediction of a cycle of events that the system will eventually find itself in.

The focus of this paper is on the adaptive aspect of FCMs mentioned above. The sections below outline the Differential Hebbian Learning (DHL) rule as applied to FCMs before proposing a formal method for its effective application.

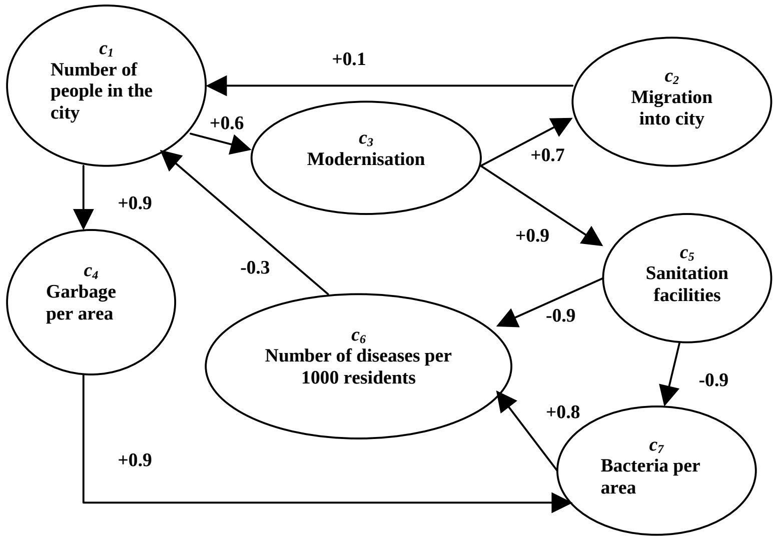


Figure 2. Fuzzified version of the cognitive map [14] in figure 1.

2 Fuzzy Cognitive Maps and Differential Hebbian Learning

Kosko [3],[4] discusses the use of Differential Hebbian Learning (DHL) as a form of unsupervised learning for FCMs. The DHL law correlates the changes of two concepts. If concept *A* and concept *B* move in the same direction (eg. B increases when A increases), the edge strength between the two concepts is increased; otherwise, the edge strength is decreased.

The training data resembles a sequence of state vectors. Training is done by going through each state vector and modifying the FCM matrix based on the DHL law. At each time step *t*, the value for e_{ij}, the edge linking concept *i* and concept *j*, is given by the discrete version of the DHL law:

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

Where ΔC_i is the change in concept *i* and $\Delta C_i(t) = C_i(t) - C_i(t-1)$. The learning coefficient μ_t decreases slowly over time. Kosko and Dickerson [5] uses

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

The constant *N* ensures the learning coefficient C_t never becomes negative.

2.1 Limitations of Differential Hebbian Learning

DHL can be used to make an FCM adapt causal link strengths as part of an unsupervised training with a sequence of state vectors (eg. a limit cycle). However, there is no guarantee that DHL will encode the sequence into the FCM. Kosko and Dickerson [5] explores the potential of DHL by creating a bivalent simple FCM, which is fed with stimulus vectors to obtain some limit cycles. Using these limit cycles as training data, DHL is then used to create a new FCM. They showed that the new FCM tends to learn the same attractors (ie.

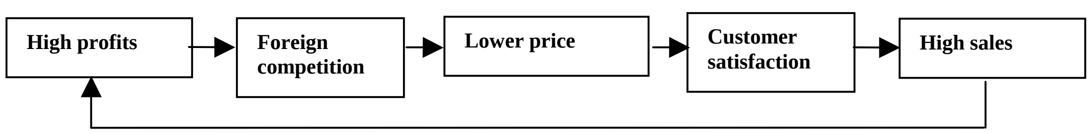
limit cycles) as those in the initial FCM. However, these results are only approximate and there is no formal proof to back them [6]. Kosko [4] concludes that tools are needed to force a limit cycle into the FCM dynamics.

At the time of writing, most of the research on DHL focuses on examining its underlying workings and dynamics. An extensive review of literature suggests the lack of practical application of DHL to real-world problems. Neither Kosko nor any of the other authors [7],[8],[9],[10], who addresses the possible use of DHL in automating the development of FCMs, outlines any concrete procedure for using DHL in this regard. The objective of this paper is to propose a formal methodology for using DHL to automate the creation of FCMs. Due to the limitations of DHL discussed above, the usage of DHL is not suited for applications where accuracy is a critical issue (eg. applications of FCMs to model control systems). This discussion is therefore limited to the development of FCMs to be used as a decision support tool.

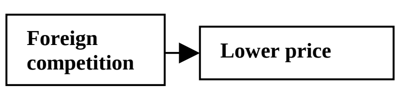
3 A Methodology for Differential Hebbian Learning

Theoretically, DHL can be used to infer an FCM based on a given set of training data. This could be the way to automate the FCM creation process, if not for the limitation discussed in the previous section. DHL can encode the training data, but it does not guarantee correct encoding all the time. Therefore, this approach may lead to the creation of an FCM that represents the problem domain poorly. A novel DHL methodology for use in the FCM creation process is described below with a view to overcoming this limitation. The idea is not to use DHL as an FCM creation mechanism, but to use it as a tool to assist the FCM creation process. The steps in this formal methodology are as follows:

1. *Create a crisp cognitive map*. A crisp cognitive map is one whose edge values are in the interval [-1,0,1] (0 for no connection between concepts).
2. *Identify groups of events that happen simultaneously, or in a sequence.* In this step, a group of events in the problem domain that normally occur either simultaneously or sequentially is identified. As an example taken from the automobile industry, a sequence of events can be as complex as:



Or as simple as



3. *Encode sequence.* Differential Hebbian Learning is used to encode the event sequences identified in step 2 into the crisp cognitive map created in step 1.
4. *Revise the trained FCM.* The final step involves detailed examination of the trained FCM by the domain expert to identify and correct any flaws.

In step 1, an FCM is created to represent the domain expert's understanding of her domain. The use of DHL serves as a tool to build on this initial FCM. The deliverable of step 3 is an FCM with modified edge strengths encoding the domain expert's beliefs as well as causal knowledge gained through DHL. In the succeeding step, the domain experts are given a final chance to verify and improve the trained FCM. The verification process eliminates any undesirable causal knowledge that may have become encoded into the FCM during the DHL process.

3.1 Application of the methodology

To demonstrate the feasibility of the proposed methodology, its use to automate the creation of an FCM is presented next. The example given below is based on a sample FCM drawn from the literature. Tsadiras and

Margaritis [11] use an FCM to model a freeway during rush hour. This FCM has been chosen for its ease of understanding due to the familiar nature of the problem it models. It is recreated here with some modification to illustrate the DHL methodology. The behaviour of the final FCM generated is studied and the effectiveness of the methodology is assessed.

Step1: Crisp cognitive map creation

The crisp cognitive map produced in this step is shown in figure 2.

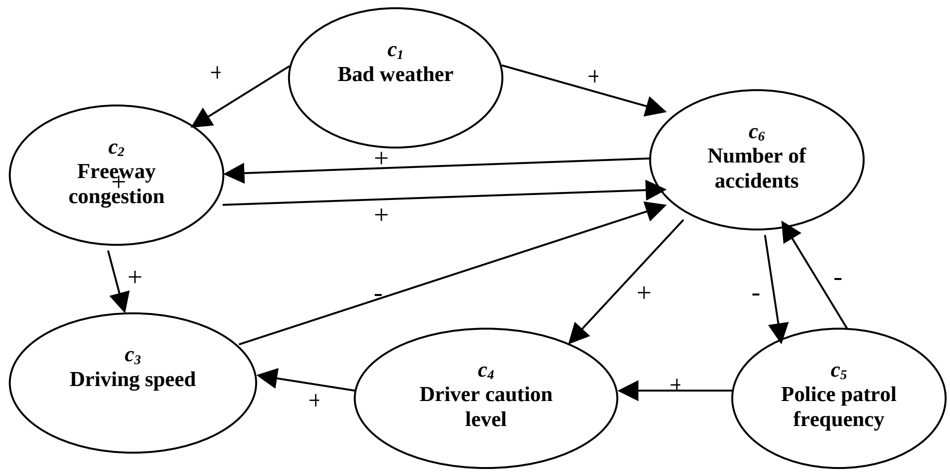


Figure 3: A simple FCM modelling a freeway.

The FCM matrix is as follow.

$$\mathbf{E} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & 0 \end{pmatrix}$$

Using a threshold of 0.5 as the nonlinear transfer function in (1), the stimulus vector **C1** [1 1 0 0 0 0] representing the events *bad weather* and *freeway congestion*, leads to the following sequence:

C1 × **E** = [0 1 1 0 0 2] → **C2** = [1 1 1 0 0 1]
C2 × **E** = [0 2 2 1 -1 1] → **C3** = [1 1 1 1 0 1]
C3 × **E** = [0 2 2 1 -1 1] → **C3** = [1 1 1 1 0 1]

The nodes *bad weather* (c_1) and *freeway congestion* (c_2) are kept on all the time, as they represent the so-called policy events whose effects are the subject of the study. An equilibrium state **C3** [1 1 1 1 0 1] is reached by the FCM after two passes. From the decision support point of view, this represents the inference that bad weather and freeway congestion eventually lead to increased driver caution level, decreased driving speed as well as increased number of accidents.

Step2: Identification of simultaneous and sequential events

This phase involves analysing the problem domain with the aid of historical data as well as the domain expert's experience of the occurrence of events. In case of the example freeway scenario, it is not uncommon to have traffic jams and police patrol on rainy days. The cause and effect relationships between weather, traffic jam, and police patrol may not be known with certainty, but one does observe the three events happening together most of the time. Accordingly, this set of events is a candidate for encoding into the crisp cognitive map. Given the observation that the rain, freeway congestion and police patrol can form a set of simultaneous events following on from no (significant) events taking place previously, the following sequence of events should be encoded

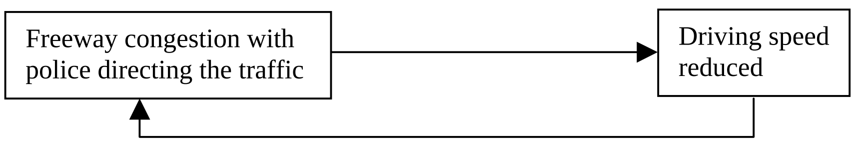
$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{array} \qquad \text{(seq1)}$$

The second row of the sequence has the concept bad weather, freeway congestion and police patrol turned on, implying that these three events occur simultaneously.

To identify another set of simultaneous events, let us consider the event of an accident on the freeway. Two events that one associates naturally as cooccurrences with this event are a traffic jam and police presence. Thus, the second event sequence can be encoded as follows:

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{array} \qquad \text{(seq2)}$$

Finally, we encode a sequence of events that happen from time to time. The sight of police directing traffic during freeway congestion is followed by a reduction in driving speed. Below is a causal view of the sequence.



Unlike the previous two examples that involved several events happening at the same time, the events in this case happen sequentially. The following is the corresponding sequence of state vectors:

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{array} \qquad \text{(seq3)}$$

Like the previous examples, we start with a state vector initialised to zeros. Next, the concepts *freeway congestion* and *police patrol frequency* are turned on. Note that these two concepts stay "on" for the rest of the time steps. In encoding event sequences, a node turned on should stay on all the time. It gets turned off only if we believe there is a special reason to do so. For example, we may believe that at a particular time step, concept node *B* is turned on. If node *B* has a negative effect on node *A* that has been previously turned on, we may want to turn node *A* off.

Step3: Encoding Sequences with Differential Hebbian Learning

To encode the event sequences, we use the following learning coefficient:

$$\mu_t = 0.2\left[1 - \frac{t}{1.1N}\right]$$

After encoding the sequence (seq1), we get the following edge matrix:

$$\mathbf{E} = \begin{pmatrix} 0.20 & 1.00 & 0.00 & 0.00 & 0.20 & 0.80 \\ 0.20 & 0.20 & 0.80 & 0.00 & 0.20 & 0.80 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.20 & 0.20 & 0.00 & 0.80 & 0.20 & -0.80 \\ 0.00 & 1.00 & 0.00 & 1.00 & -1.00 & 0.00 \end{pmatrix}$$

It may be observed that new links have been generated after the training. For example, nodes *bad weather* (c_1) and *freeway congestion* (c_2) now have links to patrol frequency. This is in agreement differential Hebbian learning which strengthens the link between two nodes if both nodes change in the same direction. In the freeway example, it is always observed that bad weather, freeway congestion and police patrol happen simultaneously, thus there should be links established between them (edge strength changed from zero to non-zero).

After encoding the sequence (seq2) into the FCM, we get the updated edge matrix

$$\mathbf{E} = \begin{pmatrix} 0.20 & 1.00 & 0.00 & 0.00 & 0.20 & 0.80 \\ 0.16 & 0.36 & 0.64 & 0.00 & 0.36 & 0.84 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.16 & 0.36 & 0.00 & 0.64 & 0.36 & -0.44 \\ 0.00 & 1.00 & 0.00 & 0.80 & -0.60 & 0.20 \end{pmatrix}$$

As may be observed in the matrix above, the -1 edge strength from node *accident* (c_6) to node *police patrol frequency* (c_5) stipulated initially has decreased. The question that may now be raised is whether the negative link between accident and police patrol in the initial FCM was appropriate. When there is an accident on the road, we usually see the police at work. This suggest that *accident* should have had a positive rather than negative link to *police patrol frequency*. It is worth noting the role DHL has played in this instance in highlighting parts of the FCM that may have been formulated incorrectly during the initial FCM building process. After training the FCM with sequence (seq3), we get the following edge matrix

$$\mathbf{E} = \begin{pmatrix} 0.20 & 1.00 & 0.00 & 0.00 & 0.20 & 0.80 \\ 0.13 & 0.49 & 0.51 & 0.00 & 0.49 & 0.67 \\ 0.00 & 0.00 & 0.15 & 0.00 & 0.00 & -0.85 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.13 & 0.49 & 0.00 & 0.51 & 0.49 & -0.35 \\ 0.00 & 1.00 & 0.00 & 0.80 & -0.60 & 0.20 \end{pmatrix}$$

Step4: Trained FCM revision

The last phase involves examining the trained FCM to identify and correct any flaws. In this phase, the domain expert is asked to identify any apparent flaws within the FCM. Adjustments may be made to the FCM if necessary. Continuing with the freeway example, the following improvements in the FCM are suggested.

1. All diagonal values should be set to zero to prevent self-feedback.
2. No concepts should affect the weather. Accordingly, all values in the first column of **E** representing effects on node *bad weather* due to all other nodes should be set to zero.

The final version of the FCM edge matrix is as follows:

$$\mathbf{E} = \begin{pmatrix} 0.00 & 1.00 & 0.00 & 0.00 & 0.20 & 0.80 \\ 0.00 & 0.00 & 0.51 & 0.00 & 0.49 & 0.67 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.85 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.49 & 0.00 & 0.51 & 0.00 & -0.35 \\ 0.00 & 1.00 & 0.00 & 0.80 & -0.60 & 0.00 \end{pmatrix}$$

The application of the stimulus vector **C1** [1 1 0 0 0 0] (representing events *bad weather* and *freeway congestion*) now leads to the following sequence of FCM states:

$\mathbf{C1}\times\mathbf{E} = \mathbf{C2} =$ [1 1 1 0 1 1]
$\mathbf{C2}\times\mathbf{E} = \mathbf{C3} =$ [1 1 1 1 0 0]
$\mathbf{C3}\times\mathbf{E} = \mathbf{C2} =$ [1 1 1 0 1 1]

The above sequence shows the FCM stabilising to a limit cycle of length 1 (C2-C3-C2), which is interpreted as follows: the incidence of bad weather (c_1) and freeway congestion (c_2) leads to a situation where driving speed stays reduced (c_3), but driver caution level (c_4), police patrol frequency (c_5) and incidence of accidents (c_6) keep fluctuating.

4 Conclusion

The idea of using differential Hebbian learning to implement adaptive FCMs has been around for some time, but no known attempts have been made to formalise its application. A methodology for using DHL to automate part of the FCM creation process in a decision support context has been proposed. The feasibility of the methodology has been demonstrated by applying it to develop an FCM, which models a real-life scenario. Encoding of selected event sequences into the FCM has resulted in a fine-tuned FCM whose node interconnection weights give a more accurate representation of the causal relationships in the problem domain. As a result, of training with the proposed DHL methodology, the fuzzified FCM has been found to form new causal links between some concepts while weakening some existing links. For the example FCM used in this study, both these adaptive changes and the resulting behaviour of the FCM agree with an intuitive understanding of the problem domain.

One of the criticisms directed at FCMs is that they require human experts to come up with fuzzy causal relationship values, which may sometimes be inaccurate. DHL can alleviate this problem by allowing the creation of less error-prone simple FCMs with crisp causal links, which are then fuzzified through an adaptive learning process. The applicability of the proposed DHL methodology to real-world industrial problems remains the subject of further investigations.

References:

[1] Axelrod, R., “Structure of Decision”, Princeton University Press, US, 1976.
[2] Kosko, B., "Fuzzy Cognitive Maps", Int. J. Man-Machine Studies, Vol.24, pp.65-75, 1986.
[3] Kosko, B., “Neural Networks and Fuzzy Systems”, Prentice Hall, Englewood Cliffs, NJ, 1992.
[4] Kosko, B., “Fuzzy Engineering”, Prentice Hall, US, 1997.
[5] Kosko, B., and Dickerson, J., “Virtual Worlds as Fuzzy Cognitive Maps”, Presence, 3(2), 173 – 189, 1994.
[6] Kosko, B., Dickerson, J., “Fuzzy Virtual World”, AI Expert, 9(7), 25-31, 1994.
[7] Han, J. H., Lee, K. C., Lee, W., Yu, P., Kwon, O., B., “Strategic Planning Simulation Based on Fuzzy Cognitive Map Knowledge and Differential Game”, Simulation, 71(5), 316-327, 1998.
[8] Margaritis, K. G., Tsadiras, A. K., “Cognitive Mapping and Certainty Neuron”, Information Sciences, 101, 109-130, 1997.
[9] Margaritis, K. G., Tsadiras, A. K., “Using Fuzzy Cognitive Maps for Decision Making”, 3rd Balkan Conference on O.R., 1-14, 1995.
[10] Lee, K., C., “A Fuzzy Cognitive Map-Based Bi-Directional Inference Mechanism: An Application to Stock Investment Analysis”, Intelligent Systems In Accounting, Finance and Management, 6, 41-57, 1997.
[11] Tsadiras, A., K., Margaritis, K., G., “On The Convergence Properties and Extensions of Fuzzy Cognitive Maps”, HERMIS, 2, 1994.