Appl. Sci. **2015**, *5*, 1399-1411; doi:10.3390/app5041399

OPEN ACCESS

applied sciences

ISSN 2076-3417

www.mdpi.com/journal/applsci

Article

NHL and RCGA Based Multi-Relational Fuzzy Cognitive Map Modeling for Complex Systems

Zhen Peng [1,*], Lifeng Wu [2] and Zhenguo Chen [3]

[1] Information Management Department, Beijing Institute of Petrochemical Technology, Beijing 100029, China

[2] College of Information Engineering, Capital Normal University, Beijing 100048, China; E-Mail: wooleef@gmail.com

[3] Computer Department, North China Institute of Science and Technology, East Yanjiao, Beijing 101601, China; E-Mail: 6638752@163.com

* Author to whom correspondence should be addressed; E-Mail: zhenpeng@bipt.edu.cn; Tel.: +86-10-60228013.

Academic Editor: Christian Dawson

Received: 7 October 2015 / Accepted: 13 November 2015 / Published: 27 November 2015

Abstract: In order to model multi-dimensions and multi-granularities oriented complex systems, this paper firstly proposes a kind of multi-relational Fuzzy Cognitive Map (FCM) to simulate the multi-relational system and its auto construct algorithm integrating Nonlinear Hebbian Learning (NHL) and Real Code Genetic Algorithm (RCGA). The multi-relational FCM fits to model the complex system with multi-dimensions and multi-granularities. The auto construct algorithm can learn the multi-relational FCM from multi-relational data resources to eliminate human intervention. The Multi-Relational Data Mining (MRDM) algorithm integrates multi-instance oriented NHL and RCGA of FCM. NHL is extended to mine the causal relationships between coarse-granularity concept and its fined-granularity concepts driven by multi-instances in the multi-relational system. RCGA is used to establish high-quality high-level FCM driven by data. The multi-relational FCM and the integrating algorithm have been applied in complex system of Mutagenesis. The experiment demonstrates not only that they get better classification accuracy, but it also shows the causal relationships among the concepts of the system.

Keywords: fuzzy cognitive map (FCM); multi-relational data mining (MRDM); nonlinear Hebbian learning (NHL); real code genetic algorithm (RCGA); complex system

1. Introduction

The aim of the paper is to auto simulate complex systems with multi-dimensions and multi-granularities driven by multi-relational data resources for better classification and causal relationships of a system.

Multi-Relational Data Mining (MRDM) [1,2] is able to discover knowledge directly from multi-relational data tables, not through connection and aggregation of multiple relational data into a single data. Multi-relationship data mining can effectively prevent the problems of information loss, statistical deviation and low efficiency, *etc.* These methods [3–5] such as CrossMine, MI-MRNBC and Graph-NB are fitting for multi-relational data mining, but cannot obtain causality in a multi-relational system.

In 1986, FCM [6–8] is introduced by Kosko, suggesting the use of fuzzy causal functions taking numbers in [−1, 1] in concept maps. FCM, as a kind of graph model, combines some aspects from fuzzy logic, neural networks and other techniques, and is fitting for modeling system from data resources. Compared to other techniques, FCM exhibits a number of highly appealing properties. In particular, FCM can directly show the multi-relationships between different concepts and the inference is easy and intuitive. FCM learning algorithms use learning algorithms to establish models from historical data (simulations of concept values), which have been widely used applied to various fields [9–13] of society, engineering, medicine, environmental science, *etc*. However, none of them consider the data characteristics of multi-relationships.

Multi-relational FCM discussed in the paper refers to two-levels FCM. There is one FCM of each dimension in low-level. There is only one FCM in high-level. The state value of each concept in high-level actually is a summary evaluation of low-level FCM in the dimension, which is inferred based on the weight vector, obtained by multi-instances oriented NHL, in low-level FCM. RCGA of high-level FCM aims to mine high-level FCM based on summary evaluations of low-level FCMs for high-quality classification and causality. Thus, the proposed multi-relational FCM and the integrating algorithm seem a rather realistic approach to solve the complex model.

The remainder of this paper is organized as follows. Section 2 describes FCM, existing learning algorithms of FCM, and problems to solve. Then, Section 3 proposes the multi-relational FCM and integrating NHL and RCGA based multi-relational FCM learning algorithm. In Section 4, the experiment and its results are represented and analyzed. Finally, we briefly conclude this paper in Section 5.

2. Backgrounds

2.1. Fuzzy Cognitive Map (FCM)

A Fuzzy Cognitive Map F in Figure 1 shows a relationship system, which is a 4-tuple (C, W, A, f) mathematically, where

- $C = \{C_1, C_2, \ldots, C_N\}$ is a set of N concepts forming the nodes of a graph.
- $W: (C_i, C_j) \rightarrow w_{ij}$ is a function associating w_{ij} with a pair of concepts, with w_{ij} equal to the weight of edge directed from C_i to C_j, where $w_{ij} \in [-1, 1]$. Thus, $W(NN)$ is a connection matrix.

If there is positive causality between concepts C_i and C_j, then $w_{ij} > 0$, which means an increase of the value of concept C_i will cause an increase of the value of concept C_j and a decrease of the value of C_i will lead to a decrease of the value of C_j.

If there is inverse causality between the two concepts, then $w_{ij} < 0$, which represents an increase of the value of concept C_i will cause a decrease of the value of the second concept and a decrease of the value of concept C_i will cause an increase of the value of the concept C_j.

If there is no relationship between the two concepts, then $w_{ij} = 0$.

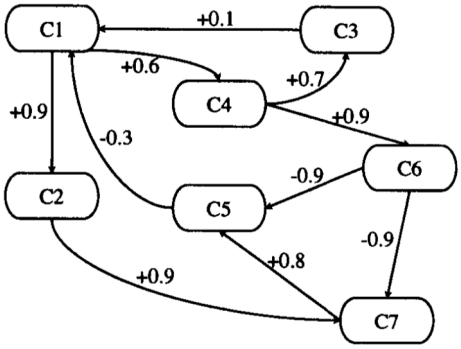

Figure 1. A simple fuzzy cognitive map.

- A: $C_i \rightarrow A_i(t)$ is a function that associates each concept C_i with the sequence of its activation degrees such as for $t \epsilon T$, $A_i(t) \epsilon L$ given its activation degree at the moment t. $A(0) \epsilon L^T$ indicates the initial vector and specifies initial values of all concept nodes and $A(t) \epsilon L^T$ is a state vector at certain iteration t.
- f is a transformation or activation function, which includes recurring relationship on $t \geq 0$ between $A(t + 1)$ and $A(t)$.

$$A_j(t+1) = f\left(A_j(t) + \sum_{\substack{i \neq j \\ i \in S}} A_i(t) w_{ij}\right) \tag{1}$$

The transformation function of FCM is with memory at previous moment shown as Equation (1), which is used to infer the state values of concepts in FCM. It limits the weighted sum to the range [0, 1] or [−1, 1]. The three most commonly used transformation (bivalent, trivalent, logistic) functions are shown below.

i. Bivalent

$$f(x) = \begin{cases} 0, x \leq 0 \\ 1, x > 0 \end{cases}$$

ii. Trivalent

$$f(x) = \begin{cases} -1, x \leq -0.5 \\ 0, -0.5 \leq x \leq 0.5 \\ 1, x \geq 0.5 \end{cases}$$

iii. Logistic

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

FCM can be used to perform simulation of an interconnected system. The vector $A(t)$ in FCM specifies state values of all concepts (nodes) in the t iteration. An FCM has a number of successive state iterations.

The state value of a concept is calculated by the preceding iteration of concepts states, which exert influence on the given node.

2.2. FCM Learning Algorithms

FCM learning algorithm is a kind of automated learning method to establish FCM model from data resources. There are two classes of FCM learning algorithms, Hebbian based learning and evolved based learning. The former are Hebbian based algorithms [14–16], mainly including NHL (Nonlinear Hebbian Learning), DD-NHL (Data-Driven Nonlinear Hebbian Learning) and AHL (Active Hebbian Learning). The differences of these algorithms are in the way of adjusting the edge weights. The latter are evolve based algorithms [17–20], which are composed of PSO (Particle Swarm Optimization), RCGA (Real Coded Genetic Algorithm), *etc.*

2.2.1. Nonlinear Hebbian Learning (NHL)

NHL is on the basis of the well-known Hebb's learning law, which is a kind of unsupervised learning algorithm. Considering the nonlinear output unit, given random pre-synaptic an input vector $\boldsymbol{x}$, weight vector $\boldsymbol{w}$, and output $z = f(\boldsymbol{w}^T \boldsymbol{x})$. The nonlinear activation function f is a sigmoid function. The criterion function J maximized by Hebb's rule may be written as Equation (2).

$$J = E[z^2] \tag{2}$$

An additional constraint such as $||\boldsymbol{w}|| = 1$ is necessary to stabilize the learning rule. A stochastic approximation solution is employed to the following nonlinear Hebbian learning rule as Equation (3).

$$\Delta w_{ji} = \eta_k z \frac{dz}{dy}(x_j - w_{ji} y_i) \tag{3}$$

Note that the nonlinear learning rules are seeking a set of weight parameters such that the outputs of the unit have the largest variance. The nonlinear unit constrains the output, ensuring it remains within a bounded range.

2.2.2. Real-Coded Genetic Algorithm (RCGA)

RCGA is a real-coded genetic algorithm to develop FCM connection matrix based on data resource. RCGA defines each chromosome as a floating-point vector. Each element in the vector is called gene. In case of the learning FCM with N node, each chromosome consists of $N(N-1)$ genes, which are floating point numbers from the range [−1, 1], defined as follows.

$$\boldsymbol{E} = [w_{12}, w_{13}, \ldots, w_{1N}, w_{21}, w_{23}, \ldots, w_{2N}, \ldots, w_{NN-1}]^{\mathrm{T}} \tag{4}$$

where w_{ij} is a weight value for an edge from i^{th} to j^{th} concept node. Each chromosome has to be decoded back into a candidate FCM. The number of chromosomes in a population is constant for each generation and it is specified by the *population_size* parameter.

The fitness function is calculated for each chromosome by computing the difference between system response generated using a FCM weights and a corresponding system response, which is known directly

from the data resource. The difference is computed across all $M-1$ initial vector/system response pairs, and for the same initial state vector. The measure of error is shown as Equation (5).

$$Error\_L_p = \alpha \sum_{t=1}^{M-1} \sum_{n=1}^{N} |A_n(t) - \widehat{A_n(t)}|^p \quad (5)$$

The parameter of α is used to normalize error rate, and p is 1, 2 or ∞. N is the number of concepts in FCM, and M is the number of iterations. The error measure can be used as the core of fitness function as Equation (6).

$$\text{Fitness function} = \frac{1}{a \times Error\_L_p + 1} \quad (6)$$

The fitness function is normalized to the (0, 1]. The parameter a can be set different value in different p condition.

The stopping condition of RCGA takes into consideration two possible scenarios of the learning process. One is the learning should be terminated when the fitness function value reaches a threshold value called *max_fitness*; the other is a maximum number of generations, named *max_generation*, has been reached. If the stopping conditions have not been reached, evolutionary operators and selection strategy need to be applied.

2.3. Problem Statements

In real world, a complex system has to have multi-dimensional groups with direct or indirect relationships, which generates multi-relational data. Moreover, each dimension maybe contains many concepts with different granularity relationships. For example, in a Mutagenesis system, there are three dimensions of atom, molecule and another atom, which are coarse-grained concepts; atype and charge are fine-grained concepts in the atom dimension; lumo, logp, indl and inda are fine-grained concepts in the molecule dimension. In two granularities, fine-grained concepts are on behalf of nodes in low-level and coarse-grained concepts represent hyper nodes. The coarse-grained concept of a dimension can be seen as a summary expression of its fine-grained concepts with multi-instances in low-level of the dimension.

Thus, problem to solve are: how to get summary evaluations in a dimension based on low-level data with multi-instances and how to mine high-level FCM from summary evaluations in dimensions for high-quality classification and causality.

3. Materials and Methods

3.1. Multi-Relational FCM

In order to better model the multi-relationships and coarse-grained concepts in the complex system, undoubtedly, a multi-relational FCM (Definition 1), extended from FCM, can represent the multi-relationship system. The multi-relational FCM can be divided into different groups and different levels. A group means a dimension. A level is a granularity. Coarse-grained concepts are upper-level nodes. Fine-grained concepts are low-level nodes. A coarse-grained concept in a dimension represents a FCM composed of fine-grained concepts, and is related with other coarse-grained concepts in other dimensions.

Definition 1. A multi-relational FCM with two-levels and n-dimensions is $U_n^2 = (C_n^2, W_n^2, A_n^2, f)$.

- C_n^2: $\{\{C_{1i}\}\ldots,\{C_{ji}\},\ldots\{C_{ni}\}\}$ is a set of concepts, $\{C_{ji}\}$ is on behalf of a coarse-grained concept in j dimension, and C_{ji} is i^{th} fine-grained concept in bottom-level of j^{th} dimension.
- W_n^2: $\{\{W_j\}, \{W_{ij}\}\}$. $<\{C_{ji}\}> \rightarrow W_j$ is a function associating W_j among j^{th} dimension, W_j:$\{w_{ij}\}$; $(<\{C_{1i}\}>\ldots, <\{C_{ji}\}>,\ldots<\{C_{ni}\}>) \rightarrow \{W_{ij}\}$ is function associating between coarse-grained concepts.
- A_n^2: $C_{ji} \rightarrow A_{ji}(t)$, $\{C_{ji}\} \rightarrow A_j(t)$. $A_j(t)$ is a function f at iteration t.
- f is a transformation function, which includes recurring relationship on $t \geq 0$ among $A_j(t + 1)$, $A_{ji}(t)$ and $A_i(t)$, *w*here $A_j(0)$ is referred out based on the weight vector W_j, got by multi-instances oriented NHL, in low-level FCM.

3.2. Multi-Instances Oriented NHL

In the multi-relational FCM, each concept represents a useful field name in data resource. In the multi-relationship, one field in main table corresponding to another table has some sub fields. The field in the main table points to a coarse-grained concept in low-level FCM and the fields in another table indicate the fine-grained concepts in the FCM. Thus, the low-level FCM can be used to model the multi-relationship.

One coarse-grained concept (C_j) in j^{th} dimension corresponds to some fine-grained concepts (C_{j1}, C_{j2}, . . . , C_{jn}). It becomes the key to the state value of get the weights between the coarse-grained concept and some fine-grained concepts in the low-level FCM. The weights express the causality relationship between fine-gain concepts and coarse-grained concept. The weights need to be learned from multi-instances. NHL can be extended to multi-instances oriented mining for the optimistic of the nonlinear units' weights in Figure 2. In the prerequisite, the state states of coarse-grained concepts in high-level FCM can be inferred.

Multi-instances oriented NHL has two constraints. First constraint is that it maximizes the mathematical expectation of A_j^2 of all multi-instances as Equation (7).

$$\text{maxmize } J = E[\sum_r A_j^2] \tag{7}$$

$$\text{subject to } ||\boldsymbol{w}|| = 1 \tag{8}$$

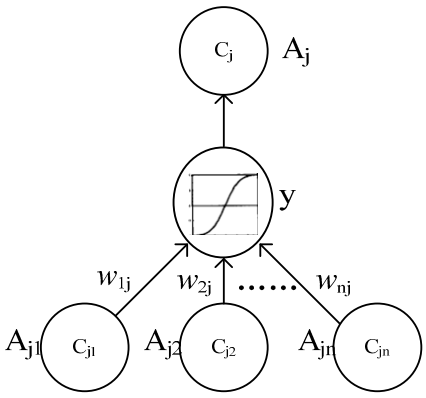

Figure 2. Nonlinear unit in Fuzzy Cognitive Map (FCM).

Second constraint is that the weight vector $\boldsymbol{w}$ has to be limited to stabilize the learning rule as Equation (8), which generates the following nonlinear Hebbian learning rule as Equation (9).

$$\Delta w_{ij}(t+1) = \eta A_j(A_{ji} - w_{ij}(t)A_j) \tag{9}$$

Accordingly, multi-instances oriented NHL is presented in the function of M_NHL. The execution phase of multi-instances oriented NHL (M_NHL) is consisted of the following steps:

Step 1: Random initialize the weight vector $\boldsymbol{w}_j(t)$, $t = 0$, $p = 0$ and input all instance $\{A_{ji}\}$

Step 2: Calculate the mathematical expectation of A_j^2 of all $\{A_{ji}\}$

Step 3: Set $t = t + 1$, repeat for each iteration step t:

3.1 Set $p = p + 1$, to the p^{th} instance:

3.1.1 Adjust $\boldsymbol{w}_j(t)$ matrix to $\{A_{ji}\}^p$ by Equation (9)

3.2 Calculate the A_j^2 to all $\{A_{ji}\}$ by Equation (10)

3.3 Determine whether A_L^2 is maximum or not at present

3.4 If A_j^2 is maximum, output the optimal $\boldsymbol{w}_j(t)$

Step 4: Return the final weight vector $\boldsymbol{w}_j(t)$

According to the literature [21], whether NHL clusters better or not is closely related to the activation function f. It means that if cumulative normal distribution function or approximate cumulative logic distribution function is chosen as activation function, output results will show a U-distribution and easily achieve better NHL clustering. In order to avoid integral calculation, the cumulative logic distribution, shown as Equation (11), is selected. The output values (nonlinear unit outputs) can be inferred by Equation (10), where A_{ij} is the input and w_{ji} is mined by M_NHL algorithm.

$$\mathrm{A_j(t)} = f(\sum_{\mathrm{i\neq j}} A_{ij}(\mathrm{t}-1)w_{ji}(\mathrm{t}-1)) \tag{10}$$

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{11}$$

3.3. NHL and RCGA Based Integrated Algorithm

RCGA is used to get the weight matrix of high-level FCM for high-quality classification and causality based on initial state values of coarse-grained concepts, which are got by M_NHL in Section 3.2.

$$W = [W_{12}, W_{13},\ldots,W_{1M},\ldots,W_{M1},\ldots,W_{M(M-1)}] \tag{12}$$

Each chromosome consists of $M(M-1)$ genes (see Equation (12)). M is the total number of hyper concepts or dimensions. The gene is a floating point number from the range $[-1, 1]$. W_{ij} specifies the weight between coarse-grained concept in i^{th} dimension and it in j^{th} dimension. Each chromosome can be decoded back into a high-level FCM.

When the *fitness* is more than *max_fitness* specified or the generation is more than *max_generation* specified, the procedure ends. The fitness function of RCGA is as follows (see Equation (13)).

$$fitness = \frac{1}{a\sum_{t=0}^{T}\sum_{j=1}^{M}\left(A_j(t) - \widehat{A_j}\right)^2 + 1} \tag{13}$$

where T is the number of iterations, $\widehat{A_j}$ is the actual output of j^{th} concept of system, $A_j(t)$ is simulated output at t iteration by computing.

In the procedure, the algorithm needs to call function M_NHL. Moreover, if the fitness and the number of iterations are not satisfied with the max, the next chromosome is created by select method, mutation method and recombination method. In our experiments, a simple one-point crossover, random mutation and roulette wheel selection are applied. The parameters are chosen and set as Table 1.

Table 1. The parameters in the integrated algorithm.

Parameters	Values	Meanings
probability of recombination	0.9	probability of single-point crossover
probability of mutation	0.5	probability of random mutation
population_size	50	the number of chromosomes
max_generation	500,000	a maximum number of generations
max_fitness	[0.6, 0.9]	fitness thresholds
a	1000	a parameter in Equation (13)

The execution procedure of the algorithm integrating RCGA and NHL consisted of the following steps:

Step 1: Initialize the parameters by the Table 1
Step 2: Randomly initialize population_size chromosomes, $g = 0$, $t = 0$
Step 3: Repeat for each dimension j:
    3.1 Calculate $A_j(t)$ by Equation (11) and $\mathbf{w}_j$ based on M_NHL
Step 4: Repeat for each chromosome:
    4.1 Calculate the *fitness* by $A_j(t)$ and Equation (13)
Step 5: Get max of the *fitness* and the W
Step 6: if max of *fitness* not more than *max_fitness* and g not more than *max_generation*
    6.1 Select chromosomes by roulette wheel selection
    6.2 Recombination the chromosomes by single-point crossover
    6.3 Random mutation to the chromosomes by the probability
    6.4 Set $t = i + 1$, Repeat for each chromosome:
        6.4.1 Calculate $A_j(t)$ by Equation (1)
    6.5 Set $g = g + 1$, go to Step 5
Step 7: The W is the optimal chromosome.

The criterion in Equation (14) is defined as a normalized average error between corresponding concept values at each iteration between the two states. The error is used to define the accuracy of the algorithm simulating FCM.

$$error = \frac{1}{M \times T} \sum_{t=0}^{T} \sum_{j=1}^{M} |A_j(t) - \hat{A}_j| \quad (14)$$

4. Results and Discussion

Experiments have been carried out using Mutagenesis describing molecular structure, which is a multi-relational dataset. The multi-relational system consists of five tables (relationships). They are Atom, Bond, Atom_1, Molatm and Mole, where Bond and Molatm play associations among other three tables. The class label is in Mole table. The Mutagenesis data describes 188 molecules falling in two classes, mutagenic (active) and non-mutagenic (inactive); 125 of these molecules are mutagenic.

A molecule is associated with multi-atoms through Molatm. An atom is associated with several atoms through Bond. So Mutagenesis can be expressed as a multi-relationship of three dimensions: Atom, Atom_1, and Mole. Each dimension has many fine-grained concepts, such as indl, inda, lumo, logp of molecular dimension.

There are three backgrounds of Mutagenesis shown in Table 2. The three multi-relational FCM structures as shown in Figures 3–5 are different in three different backgrounds. The dotted lines show high-level FCMs.

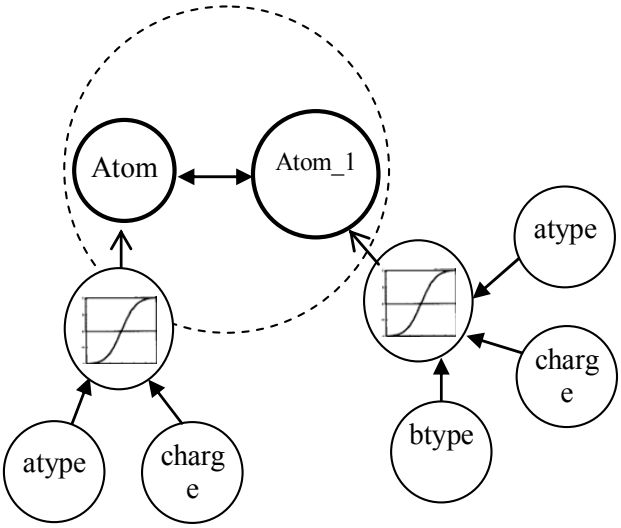

Figure 3. FCM structure of Mutagenesis in BK_0.

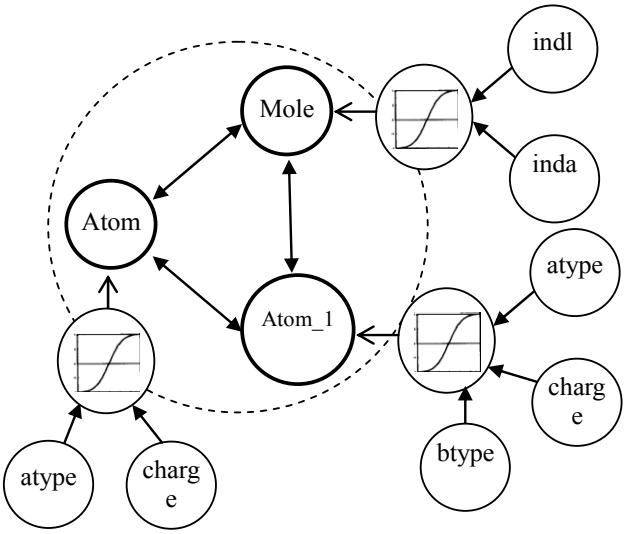

Figure 4. FCM structure of Mutagenesis in BK_1.

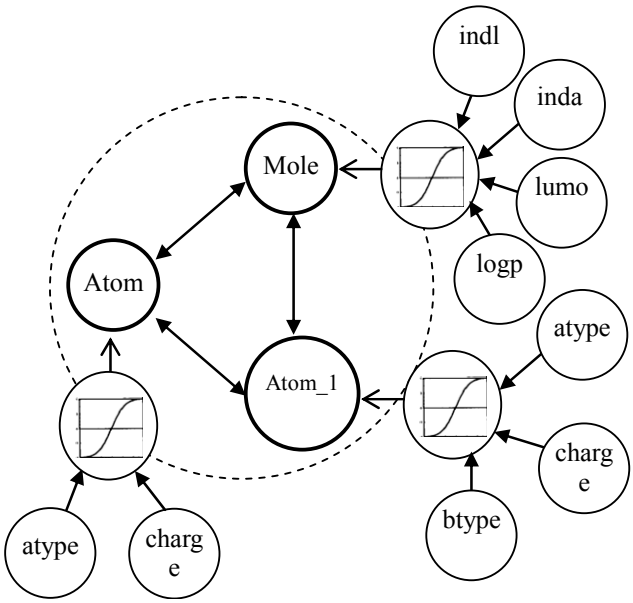

Figure 5. FCM structure of Mutagenesis in BK_2.

Table 2. Three kinds of background of Mutagenesis.

Background	Description
BK_0	Each compound includes the attributes of bond types, atom types, and partial charges on atoms
BK_1	Each compound includes indl and inda of mole besides those in BK_0
BK_2	Each compound includes all attributes that are logp and lumo of mole besides those in BK_1

The experiment is implemented for the class of molecular and the association weights among the three concepts (Atom, Atom_1 and Molecular) in the Mutagenesis.

For better operation efficiency in shorter runtime, an experiment, based on multi-relationship FCM and the integrated algorithm, has been carried out in the different fitness thresholds under three kinds of background. The learning runtimes are shown in the fitness thresholds from 0.6 to 0.9. From Figure 6, we can see that the changes of the runtime are not big under three kinds of background. The runtimes spent is changed. When the fitness threshold is at the interval of (0.65, 0.76], the operation takes less time.

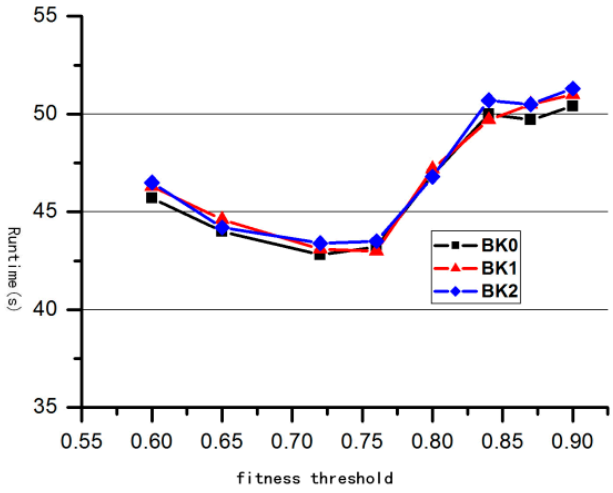

Figure 6. The learning runtime under the different fitness thresholds.

The classification results are compared in the *max_fitness* = 0.7 under three backgrounds. From the Table 3, the classification runtime of the integrated method is longer. The main reason for this is the costs in database access and FCM inference. The integrated method has better classification accuracy according to the label in Mole in three kinds of background knowledge; in particular, the accuracy rate is best in BK_1.

Table 3. Classification efficiency in different backgrounds.

Backgrounds	Runtime(s)	Accuracy (%)
BK_0	0.78	82.3%
BK_1	0.8	82.9%
BK_2	0.8	82.7%

And the method not only gets better classification, but also the association weights or causality for causal analysis of system, which is more than other methods. For example, the association matrix of high-level FCM of Mutagenesis in BK_2 is shown in Figure 7.

	Atom	**Atom_1**	**Mole**
Atom	0	0.294	0.6308
Atom_1	0.526	0	0.1801
Mole	0.2127	0.4758	0

Figure 7. Association matrix of high-level FCM of Mutagenesis in BK_2.

5. Conclusions

We construct a kind of multi-levels and multi-dimensions FCM to automatic model complex systems directly from multi-relational data resources. The multi-relational FCM include two levels and some dimensions. In the FCM, one concept in high-level has a summary evaluation in a dimension, which is inferred by the transformation function of low-level FCM. It has been solved that the weight vector in low-level FCM is learned by extended NHL from multi-instances for the inference. For getting better classification and causality, RCGA has been used in learning the association weights in high-level FCM. Moreover, the integrating algorithm of NHL and RCGA has been applied in the compounds of molecular of Mutagenesis, which obtains better accuracy and knowledge for causal analysis.

Acknowledgments

This research was supported by the Natural Science Foundation of Hebei Province (No. F2014508028), the National Natural Science Foundation of China (No. 61202027), the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (No.IDHT20150507), and the Scientific Research Base Development Program of the Beijing Municipal Commission of Education (TJSHG201310028014).

Author Contributions

Zhen Peng performed the multi-relational FCM model construction and the integrated algorithm. Zhen Peng, Lifeng Wu and Zhenguo Chen carried out experimental work. Zhen Peng prepared the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Han, J.W.; Kamber, M. *Concept and Technology of Data Mining*; Machinery Industry Press: Beijing, China, 2007; pp.373–383.
2. Spyropoulou, E.; de Bie, T.; Boley, M. Interesting pattern mining in multi-relational data. *Data Knowl. Eng*. **2014**, *28*, 808–849.
3. Yin, X.; Han, J.; Yang, J.; Yu, P.S. Efficient classification across multiple database relations: A CrossMine approach. *IEEE Trans. Knowl. Data Eng*. **2006**, *18*, 770–783.
4. Xu, G.; Yang, B.; Qin, Y.; Zhang, W. Multi-relational Naive Bayesian classifier based on mutual information. *J. Univ. Sci. Technol. Beijing* **2008**, *30*, 943–966.
5. Liu, H.; Yin, X.; Han, J. An efficient multi-relational naive Bayesian classifier based on semantic relationship graphs. In Proceedings of the 2005 ACM-SIGKDD Workshop on Multi-Relational Data Mining (KDD/MRDM05), Chicago, IL, USA, 21–24 August 2005.
6. Kosko, B. Fuzzy cognitive maps. *Int. J. Man Mach. Studs.* **1986**, *24*, 65–75.
7. Papageorgiou, E.I.; Salmeron, J.L. A review of fuzzy cognitive maps research during the last decade. *IEEE Trans. Fuzzy Syst*. **2013**, *12*, 66–79.
8. Groumpos, P.P. Large scale systems and fuzzy cognitive maps: A critical overview of challenges and research opportunities. *Annu. Rev. Control* **2014**, *38*, 93–102.
9. Zhang, Y. Modeling and Control of Dynamic Systems Based on Fuzzy Cognitive Maps. Ph.D. Thesis, Dalian University of Technology, Dalian, China, 2012.
10. Mago, V.K.; Bakker, L.; Papageorgiou, E.I.; Alimadad, A.; Borwein, P.; Dabbaghian, V. Fuzzy cognitive maps and cellular automata: An evolutionary approach for social systems modeling. *Appl. Soft Comput*. **2012**, *12*, 3771–3784.
11. Subramanian, J.; Karmegam, A.; Papageorgiou, E.; Papandrianos, N.I.; Vasukie, A. An integrated breast cancer risk assessment and management model based on fuzzy cognitive maps. *Comput. Methods Programs Biomed*. **2015**, *118*, 280–297.
12. Szwed, P.; Skrzynski, P. A new lightweight method for security risk assessment based on fuzzy cognitive maps. *Appl. Math. Comput. Sci*. **2014**, *24*, 213–225.
13. Peng, Z.; Peng, J.; Zhao, W.; Chen, Z. Research on FCM and NHL based High Order mining driven by big data. *Math. Probl. Eng*. **2015**, *2015*, doi:10.1155/2015/802505.
14. Papakostas, G.A.; Koulouriotis, D.E.; Polydoros, A.S.; Tourassis, V.D. Towards Hebbian learning of fuzzy cognitive maps in pattern classification problems. *Expert Syst. Appl*. **2012**, *39*, 10620–10629.

15. Salmeron, J.L.; Papageorgiou, E.I. Fuzzy grey cognitive maps and nonlinear Hebbian learning in process control. *Appl. Intell*. **2014**, *41*, 223–234.
16. Beena, P.; Ganguli, R. Structural damage detection using fuzzy cognitive maps and Hebbian learning. *Appl. Soft Comput.* **2011**, *11*, 1014–1020.
17. Kim, M.-C.; Kim, C.O.; Hong, S.R.; Kwon, I.-H. Forward-backward analysis of RFID-enabled supply chain using fuzzy cognitive map and genetic algorithm. *Expert Syst. Appl*. **2008**, *35*, 1166–1176.
18. Froelich, W.; Papageorgiou, E.I. Extended evolutionary learning of fuzzy cognitive maps for the prediction of multivariate time-series. *Fuzzy Cogn. Maps Appl. Sci. Eng*. **2014**, *2014*, 121–131.
19. Froelich, W.; Salmeron, J.L. Evolutionary learning of fuzzy grey cognitive maps for forecasting of multivariate, interval-valued time series. *Int. J. Approx. Reason*. **2014**, *55*, 1319–1335.
20. Wojciech, S.; Lukasz, K.; Witold, P. A divide and conquer method for learning large fuzzy cognitive maps. *Fuzzy Sets Syst*. **2010**, *161*, 2515–2532.
21. Sudjianto, A.; Hassoun, M.H. Statistical basis of nonlinear Hebbian learning and application to clustering. *Neural Netw*. **1995**, *8*, 707–715.