Inferring Fuzzy Cognitive Map Models for Gene Regulatory Networks from Gene Expression Data

Ye Chen
School of Electronics and Computing Systems
University of Cincinnati
Cincinnati, OH, United States
E-mail: chenye84@gmail.com

Lawrence J. Mazlack
Applied Computational Intelligence Laboratory
University of Cincinnati
Cincinnati, OH, United States
E-mail: mazlack@ucmail.uc.edu

Long J. Lu
Division of Biomedical Informatics
Cincinnati Children's Hospital Medical Center
Cincinnati, OH, United States
E-mail: Long.Lu@cchmc.org

***Abstract*—Gene Regulatory Networks (GRNs) represent the causal relations among the genes and provide insight on the cellular functions and the mechanism of the diseases. GRNs can be inferred from gene expression data by a number of algorithms, e.g. Boolean networks, Bayesian networks, and differential equations. While reliable inference of GRNs is still an open problem, new algorithms need to be developed. Fuzzy Cognitive Maps (FCMs) is used to represent GRNs in this paper. Most of the FCM learning algorithms are able to learn FCMs with less than 40 nodes. A new algorithm that is able to learn FCMs with more than 100 nodes is proposed. The proposed method is based on Ant Colony Optimization (ACO). A decomposed approach is proposed to reduce the dimension of the problem; therefore the FCM learning algorithm is more scalable (the dimension of the problem to be solved in one ACO run equals to the number of nodes or genes). The proposed approach is tested on data from DREAM project. The experiment results suggest the proposed approach outperforms several other algorithms.**

Keywords-fuzzy cognitive map; gene regulatory network; gene expression; ant colony optimization; learning algorithm

I. INTRODUCTION

Gene Regulatory Networks (GRNs) represent the causal relations among the genes. GRNs provide insight on cellular functions and the mechanism of diseases. The gene regulatory relations can be identified by experiments. However, experiment verification of the large number of possible gene regulatory relations is not time-efficient or cost-efficient. An efficient approach is to infer GRNs from observed gene expression data [1].

Varity of mathematic models have been developed to represent GRNs, e.g. Boolean networks [2], linear differential equations [3], Bayesian networks [4] and dynamic Bayesian networks [5]. Several studies compared the performance of GRN inference algorithms [6, 7].

DREAM project was initiated to assess the reverse engineering algorithms in computational biology [7]. The results from DREAM suggested the performance of most network inference algorithms was unsatisfactory.

The focus of this paper is developing a new GRN inference method based on Fuzzy Cognitive Maps (FCM).

FCMs [8] are graph models consist of nodes and weighted edges. The nodes represent concepts to be modeled. The edges represent the causal relations. FCM has several advantages. For example, FCMs can distinguish positive and negative relations; FCMs can represent recursive and nonlinear relations.

It is difficult to develop FCMs for GRNs from data automatically. Learning FCM from data can be formulated as an optimization problem. The objective function is the difference between the output of the learned FCM and the observed data. As the number of nodes increases, the dimension of the optimization problem, i.e. the number of weights to be optimized, increases quadratically and the number of candidate solutions increases exponentially. Many FCM learning algorithms were able to develop FCMs with less than 40 nodes [9].

ACOs solve optimization problems by simulating the ant's behaviors. An ACO algorithm [10] were proposed to learn FCMs with 40 nodes. The weights were discretized and then a standard ACO was used to learn FCMs from data. There are two deficits for the ACO approach in [10]:

- The weights of the FCMs are discretized and it will introduce discretization errors;
- The weight matrix is optimized as a whole; while in fact it can be decomposed (see Section III.B).

The performance of the algorithm can be improved by using an inherently continuous ACO algorithm ACO_R [11] and a decomposition approach. The improved algorithm can be used to infer GRNs.

The main contributions of this paper include:

- Developed a new FCM learning algorithm based on ACO_R and a decomposition approach;
- Proposed to represent GRNs by FCMs;
- Applied the proposed learning algorithm to learn the GRNs from gene expression data.

II. FUZZY COGNITIVE MAPS

Fig. 1 shows a FCM example. A FCM consists of N_N nodes and weighted edges between pairs of nodes $w_{ij} \in [-1,1]$. Every node represents a concept to be modeled. The values of the nodes are denoted as a vector

$$\mathbf{C} = \left[C_1, C_2, ..., C_{N_N} \right], \quad (1)$$

where $C_i \in [0,1]$ is the fuzzy membership values for the i-th nodes. Here, C_i represents the activation degree of gene i.

An edge with weight w_{ij} represents the causal relation from node i to node j. There are three types of causal

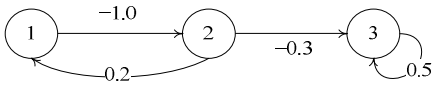

Figure 1. An FCM example.

relations: excitatory relations (w_{ij}>0), inhibitory relations (w_{ij}<0) and no causal relations (w_{ij}=0). The higher the absolute values of the weights the stronger the relations are. The relations are mathematically represented as:

$$C_i(t+1) = f\left(\sum_{j=1}^{N_N} w_{ji} C_i(t)\right), \quad (2)$$

where $C_i(t)$ is the activation degree of node i at time point t. $f(\cdot)$ is a transfer function defined as:

$$f(x) = 1/\left(1+e^{-\lambda x}\right), \quad (3)$$

where λ is a parameter. Larger λ indicate the output of the node is more sensitive to the input. λ is optimized along with the weights in this paper.

III. Proposed Approach

A. Problem Formulation

The input for the algorithm is several time series, which represents how gene expression levels change after a perturbation to the GRN. The objective is to find a weight matrix for the FCM that minimizes the error:

$$E = \frac{1}{\left(N_T - 1\right) N_N N_S} \sum_{\substack{2 \le t \le N_T \\ 1 \le n \le N_N \\ 1 \le s \le N_S}} \left(C_n(s,t) - \hat{C}_n(s,t)\right)^2, \quad (4)$$

where N_S is the number of time series, N_T is the number of time points in every time series, N_N is the number of nodes in the FCM, $C_n(s,t)$ is the observed gene expression value for gene n in the s-th time series at time point t, and $\hat{C}_n(s,t)$ is the estimated gene expression value based on the FCM under evaluation using the observed gene expression value from the previous time point, i.e.

$$\hat{C}_n(s,t) = f\left(\sum_{i=1}^{N_N} w_{in} C_n(s,t-1)\right). \quad (5)$$

B. Decomposed Approach

Equation (4) can be decomposed as follows:

$$E = \frac{1}{N_N} \sum_{n=1}^{N_N} E_n, \quad (6)$$

where E_n is the error function for node n. It is defined as:

$$E_n = \frac{1}{\left(N_T - 1\right) N_S} \sum_{\substack{2 \le t \le N_T \\ 1 \le s \le N_S}} \left(C_n(s,t) - \hat{C}_n(s,t)\right)^2, \quad (7)$$

According to (5), $\hat{C}_n(s,t)$ only depends on the n-th column of the weight matrix $\mathbf{W}_n$:

$$\mathbf{W}_n = \left[w_{1n}, w_{2n}, ..., w_{N_N n}\right]^{\mathrm{T}}. \quad (8)$$

Therefore E_n only depends on $\mathbf{W}_n$ too. E_n for each n can be minimized separately to determine $\mathbf{W}_n$. The final weight matrix can be obtained as

$$\mathbf{W} = \left[\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_{N_N}\right]. \quad (9)$$

Because GRNs are sparse networks, a penalty factor is added to E_n in order to force the algorithm to search for sparse networks. The new objective function is defined as

$$E_{pn} = E_n + p\sum_{i=1}^{N_N} \left|w_{in}\right|, \quad (10)$$

where penalty factor p is a parameter of the algorithm.

C. Ant Colony Optimization

An ACO algorithm named ACO_R [11] based on a Probability Density Function (PDF) is used to minimize the objective function (7). The PDF is updated in the iterative process of the ACO_R algorithm to reflect the relative goodness of the solutions. The ACO_R algorithm samples the solution space according to the PDF. The solutions around the good solutions that have been found by the ACO_R will have a higher probability of being chosen. In the next iteration, the top solutions found in the previous iteration are used to form a new PDF and the probability intensity around the new found good solutions will be increased. This is a positive feedback process. More solutions will be drawn around the top solutions; however, at the same time, there is a probability of jumping out of local optimal, because the PDF is a positive number for all the values in the optimization domain.

1) Pheromone Intensity Model

ACO_R algorithm represents the pheromone intensities by Gaussian kernels, which is weighted sum of several Gaussian functions. There is a Gaussian kernel for every dimension of the optimization problem. For FCM learning problems, the variables to be optimized in one optimizing run are $\mathbf{W}_n$ and λ. Therefore the dimension of the problem is N_D=N_N+1. The i-th Gaussian kernel is defined as:

$$G_i(x) = \sum_{j=1}^{N_G} \omega_j g_{ij}(x), \quad (11)$$

where N_G is the number of Gaussian functions in the Gaussian kernel, $g_{ij}(x)$ is the j-th Gaussian function that is used to compose the i-th Gaussian kernel $G_i(x)$, ω_j is the weight for the Gaussian function $g_{ij}(x)$. $g_{ij}(x)$ is defined as:

$$g_{ij}(x) = \frac{1}{\sigma_{ij}\sqrt{2\pi}} \exp\left\{-\frac{\left(x-\mu_{ij}\right)^2}{2\sigma_{ij}^2}\right\}, \quad (12)$$

where μ_{ij} and σ_{ij} is the mean and standard deviation for the Gaussian function $g_{ij}(x)$ respectively.

2) Determine the Parameters in the Gaussian Kernels

In order to establish the Gaussian kernels defined in (11), the weights ω_j, the mean values μ_{ij}, and the standard deviations σ_{ij} need to be determined according to the top solutions already found by ACO_R.

The top N_K solutions are kept in an archive and sorted according to their qualities, i.e. E_n defined in (7). For minimization problems, solutions with small objective function value rank before the ones with large value. The solutions with high ranks have a high impact on the PDF.

ω_j is determined by the rank of the solution:

$$\omega_j = \frac{1}{qN_K\sqrt{2\pi}} \exp\left\{-\frac{(j-1)^2}{2q^2N_K^2}\right\}, \quad (13)$$

where j is the rank of the solution, q is a parameter. The impact of best solution will be larger if q is smaller.

μ_{ij} equals to the value of the i-th dimension of the j-th best solution. σ_{ij} is determined by the top solutions as:

$$\sigma_{ij} = \xi \sum_{k=1}^{N_K} \frac{|s_{ik} - s_{ij}|}{N_K - 1}, \quad (14)$$

where s_{ij} is the value of the i-th dimension of the j-th best solution. A higher ξ leads to a lower convergence speed.

There are 3 parameters need to be determined by the users of ACO_R: number of top solutions N_K, the impact of the best solution q and the coefficient for the standard deviation ξ. They are usually determined by experiments.

3) Build Solutions

The solutions are built by sampling the PDF. The ants choose the value for the i-th dimension of the solution by sampling $G_i(x)$. Instead of directly sampling $G_i(x)$, an equivalent but computationally simpler two-step method is used. First, a Gaussian function $g_{ij}(x)$ is chosen randomly according to the probability defined in (15) and then $g_{ij}(x)$ is sampled based on Box-Muller method.

$$p_j = \omega_j \Big/ \sum_{k=1}^{N_K} \omega_k \quad (15)$$

4) The Optimization Process

The solution archive is first initialized with uniformly distributed random values and then evaluated and sorted.

ACO_R runs in an iterative way. There are N_A ants building solutions simultaneously. In every iteration, the parameters of $G_i(x)$ are calculated based on the solutions in the archive. Then every ant builds a solution by sampling $G_i(x)$. The ants are evaluated by calculating the objective function value of their corresponding solutions. The N_K solutions in the archive and the N_A solutions generated by the ants are sorted in the decreasing order of the objective function values. Only the top N_K solutions are kept in the solution archive. These steps run iteratively until the algorithm cannot find better solutions in several iterations. The solution ranks the first in the archive is the result.

D. Inferring Gene Regulatory Network

The gene expression data is first fuzzified. As the nodes in the FCM are defined as the activation status of the genes, the degree of activation is directly set to the normalized value of the gene expression data.

The gene expression data is used as the input. ACO_R will be applied N_N times. Every time ACO_R optimizes the vector $\mathbf{W}_n$ and the parameter λ_n. Finally, the weight vectors are combined into the weight matrix for the FCM.

Because the problem is decomposed into small ones and solved by ACO_R, this approach is called ACO_{RD}.

IV. Experimental Methods and Results

Two sets of experiments are performed. The first set of experiments is performed to compare ACO_{RD} with ACO. Test data is generated from random target FCMs. The FCM learning algorithm is supposed to recover the target FCM by using the data generated from it. The second set of experiments is performed to compare GRN inference based on FCM with other GRN inferences algorithms. Test data comes from the DREAM-4 project.

A. Performance Measures

For the first set of experiments, the performance of the FCM learning algorithms is measured by the average difference between the target FCM weights and the learned weights (see [10] for detail definition).

For the comparison of GRN inference algorithms, the performance measure is based on Receiver Operating Characteristics (ROC) and Precision-Recall (PR) [7].

B. Parameter Settings

There are 6 parameters need to be set. They are chosen based on experiments. ξ and q are searched in the interval [0, 1] with a step size of 0.1. N_K and N_A are searched in the interval [10, 100] with a step size of 10. Four values, 0.0, 0.1, 0.2 and 0.3, for the penalty factor p are tried. The number of iteration N_{iter} for the ACO algorithm is set to 15000, the same as in [10] for comparison purpose. The final settings are ξ=0.6, q=0.6, N_K=50, N_A=100, N_{iter}=15000, and p=0.2 for 100-gene experiments, and p=0.0 for the other experiments.

Every experiment is repeated 10 times with different random seeds.

C. Data Sets

In the first set of experiments, the test data is randomly generated from target FCMs using the same method as described in [10]. In this way, the results are comparable.

For the second set of experiments, data from DREAM-4 challenge 3 is used. There are time-series for five 10-gene networks and five 100-gene networks available. The topologies of these networks are extracted from known GRNs in *E. coli* and *S. cerevisiae*. The gene expression data is simulated using stochastic differential equations. 11 time points are sampled under network every perturbation and 10 time points after the perturbations are removed. 5 different perturbations are applied to every 10-gene network. Therefore there are 105 time points per network for 10-gene networks. 10 different perturbations are applied to every 100-gene network; therefore there are 210 time points in 100-gene networks.

D. Results

1) Comparison of the FCM Learning Algorithms

The comparison is based on the same experimental method as described in [10]. The target FCMs with different number of nodes and different connection densities (i.e. the number of connections divided by N_N^2) are used. From the experiment results shown in Fig. 2, we may notice the proposed ACO_{RD} algorithm performs significantly better than ACO algorithm in all the experiments in terms of model errors. The model error of the ACO_{RD} algorithm is approaching 0.4 as the number of nodes increases. A model error of 0.4 is not a good result,

but it is still 20% better than the results of ACO. The poor result is mainly because there are only 5 time sequences with 4 time points per sequence. If there are more data available, the model error can be less. We have performed an experiment on 40 time sequences with 10 time points per sequence and obtained a model error of 0.12 for ACO_{RD} and 0.45 for ACO.

2) Comparison of the GRN Inference Algorithms

The AUROC and AUPR of the proposed algorithm are compared with Ordinary Differential Equations (ODE) approach and two different Dynamic Bayesian Network (DBN) approaches. The results are shown in Fig. 3. (Note: the results for ODE and DBNs are reported in [6].)

For the inference of 10-gene networks, the FCM approach performs better than ODE in all the five networks in terms of both AUROC and AUPR; however, the FCM is better than the two DBN approach in only several cases, e.g. the AUROC for FCM is better than DBN1 and DBN2 in network 1, the AUPR for FCM is better than DBN1 and DBN2 in network 1 and 2.

For the inference of 100-gene networks, the FCM performs better than all the other three algorithms in terms of both AUROC and AUPR.

V. CONCLUSION

A new approach for building Gene Regulatory Networks (GRNs) based on Fuzzy Cognitive Maps (FCMs) and Ant Colony Optimization (ACO) is proposed.

Gene expression data is represented as fuzzy variables. This approach avoids discretizing the expression data. Therefore the dynamics of the genes can be represented more accurately. The relations among genes are modeled by fuzzy relations in FCMs. The FCM approach is able to reflect the nonlinear dynamics of the GRNs.

A learning algorithm based on an ACO for continuous domain problems is proposed to learn FCMs. The problem of optimizing the whole weight matrix is decomposed into small problems of optimizing one column of the weight matrix. Experiments are performed to show that the new algorithm outperforms the previous ACO algorithm.

The learning algorithm is applied to the GRN inference problem. The algorithm is tested on 10 DREAM-4 data sets. The results suggest that the proposed algorithm outperforms an ordinary differential equations approach and two dynamic Bayesian network approach in some of the 10-gene network problems and all of the 100-gene network problems.

Future work need to be done to further improve the proposed algorithm and apply the proposed algorithm to the GRN inference problems of larger scales.

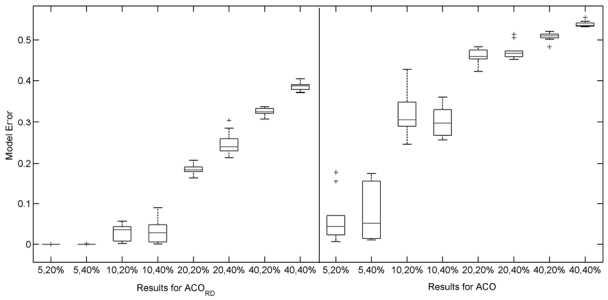

Figure 3. Comparison of model errors of the FCMs generated by ACO_{RD} and ACO; the numbers of nodes of the target FCMs range from 5 to 40 and map densities range from 20% to 40%.

REFERENCES

[1] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," Nat Rev Mol Cell Biol, vol. 9, pp. 770-780, 2008.

[2] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Random Boolean network models and the yeast transcriptional network," Proceedings of the National Academy of Sciences, vol. 100, pp. 14796-14799, December 9 2003.

[3] A. de la Fuente, P. Brazhnik, and P. Mendes, "Linking the genes: inferring quantitative gene networks from microarray data," Trends in Genetics, vol. 18, pp. 395-398, 2002.

[4] K. Haseong and E. Gelenbe, "Reconstruction of Large-Scale Gene Regulatory Networks Using Bayesian Model Averaging," in Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on, 2011, pp. 202-207.

[5] S. Y. Kim, S. Imoto, and S. Miyano, "Inferring gene networks from time series microarray data using dynamic Bayesian networks," Briefings in Bioinformatics, vol. 4, pp. 228-235, September 1 2003.

[6] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," Interface Focus, vol. 1, pp. 857-870, December 6 2011.

[7] G. Stolovitzky, R. J. Prill, and A. Califano, "Lessons from the DREAM2 Challenges," Annals of the New York Academy of Sciences, vol. 1158, pp. 159-195, 2009.

[8] B. Kosko, "Fuzzy cognitive maps," International Journal of Man-Machine Studies, vol. 24, pp. 65-75, 1986.

[9] E. I. Papageorgiou, "Learning Algorithms for Fuzzy Cognitive Maps - A Review Study," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, pp. 150-163, 2012.

[10] Y. Chen, L. J. Mazlack, and L. J. Lu, "Learning fuzzy cognitive maps from data by ant colony optimization," presented at the 14th annual conference on Genetic and Evolutionary Computation (GECCO), 2012, pp. 9-16.

[11] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," European Journal of Operational Research, vol. 185, pp. 1155-1173, 2008.

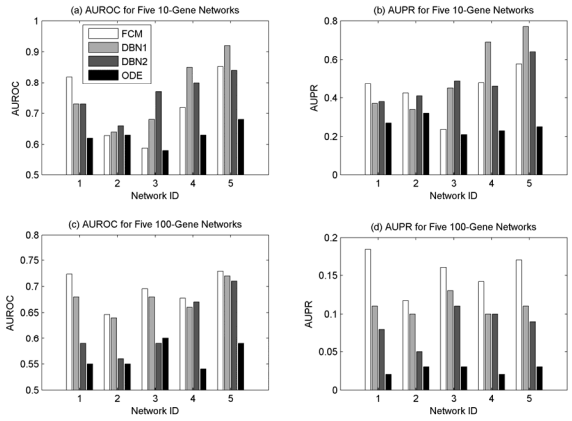

Figure 2. Comparison of GRN inference algorithms: fuzzy cognitive map (FCM), dynamic Bayesian networks without hidden states (DBN1), dynamic Bayesian networks with hidden states (DBN2) and Ordinary differential equations (ODE).