The Second International Symposium on Optimization and Systems Biology (OSB'08)
Lijiang, China, October 31– November 3, 2008

A Heuristic Method for Generating Probabilistic Boolean Networks from a Prescribed Transition Probability Matrix

Wai-Ki Ching*[1] Xi Chen[1] Nam-Kiu Tsing[1]
Ho-Yin Leung[1]

[1]Advanced Modeling and Applied Computing Laboratory, Department of Mathematics,
The University of Hong Kong, Hong Kong.
wching@hkusua.hku.hk, dlkcissy@hotmail.com, nktsing@hku.hk, obliging@hkusua.hku.hk

Abstract Probabilistic Boolean Networks (PBNs) have received much attention for modeling genetic regulatory networks. In this paper, we propose efficient algorithms for constructing a probabilistic Boolean network when its transition probability matrix is given. This is an important inverse problem in network inference from steady-state data, as most microarray data sets are assumed to be obtained from sampling the steady-state.

Keywords Boolean Networks; Probabilistic Boolean Networks; Inverse Problem; Transition Probability Matrix.

1 Introduction

The study of mathematical models and efficient numerical algorithms for the regulatory interactions among DNA, RNA, proteins and small molecules is an important issue in systems biology [9]. There have been many formalisms proposed in the literature to study genetic regulatory networks such as Bayesian networks [14], Boolean networks (BNs) [10, 11, 12, 13], multivariate Markov chain model [2], regression model [23], Probabilistic Boolean Networks (PBNs) [16, 17, 18, 19]. Interested readers are referred to the reviews in [7, 20].

BN and its extension PBN have received much attention as they are able to capture the switching behavior of the biological process [9]. BN was first introduced by Kauffman [10, 11, 12, 13]. Reviews on the good wills of BN can be found in [9, 21]. In a BN, the gene expression states are quantized to only two levels: on and off (represented as 1 and 0). The target gene is predicted by several genes called its input genes via a Boolean function. When the input genes and the Boolean functions are given, then we say that a BN is defined. A BN is a deterministic model and the only randomness comes from its initial state. Given an initial state, the BN will eventually enter into a set of state(s) called attractor cycle. Due to the facts that genetic regulation process exhibits uncertainty and microarray data sets have errors due to experimental noise in the complex measurement processes, BNs have been extended to PBNs (probabilistic model). The idea can be

described as follows. For each gene, there can be more than one Boolean function and selection probabilities are assigned to the Boolean functions. The dynamics (transitions) of a PBN can be studied by using Markov chains [16, 19]. The model parameters can be estimated by the statistical method Coefficient of Determination (COD) [8].

In a PBN, the network behavior is characterized by its steady-state probability distribution. One can understand a genetic network and identify the influence of different genes via such a network. An iterative method, namely power method has been used to compute the steady-state probability distribution with an efficient construction of the transition probability matrix [22]. Matrix approximation method has been also proposed in [3] to get an approximation of the steady-state probability distribution efficiently. In fact, it is possible to control some genes in a network so as to drive the whole network into a desirable steady-state probability distribution. Therapeutic gene intervention or gene control policy [4, 6, 17, 19, 23] can therefore be developed and studied. Pal, et al. [15] have presented two algorithms to solve the problem of finding attractors constituting a BN. Such problems are important to network inference from steady-state data, as most microarray data sets are assumed to be obtained from sampling the steady-state.

The remainder of the paper is structured as follows. Section 2 gives a brief review on BN and PBN. In Section 3, we present the inverse problem with the efficient algorithms for constructing a PBN. Some numerical examples are also given to demonstrate the proposed algorithms. Finally concluding remarks are given in Section 4.

2 A Review on Boolean Networks and Probabilistic Boolean Networks

A Boolean Network (BN) $G(V,F)$ actually consists of a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$. Define $v_i(t)$ to be the state (0 or 1) of the vertex v_i at time t and $(f_i : \{0,1\}^n \to \{0,1\})$, a list of Boolean functions : $F = \{f_1, f_2, \ldots, f_n\}$. The rules of the regulatory interactions among the genes are then represented by

$$v_i(t+1) = f_i(\mathbf{v}(t)), \quad i = 1,2,\ldots,n \tag{1}$$

where $\mathbf{v}(t) = (v_1(t), v_2(t), \ldots, v_n(t))^T$ is called the Gene Activity Profile (GAP). The GAP can take any possible forms (states) from the set

$$S = \{(v_1, v_2, \ldots, v_n)^T : v_i \in \{0,1\}\} \tag{2}$$

and thus totally there are 2^n possible states. It is known that eventually the BN will enter into a cycle and stay there forever [1, 10, 11]. The cycles actually can have biological significance [9] such as states of cell proliferation.

The following is an example of a BN of two genes with the truth table being given in Table 2.1. The transition probability matrix (Boolean network matrix) of the 2-gene BN is then given by

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3}$$

Table 1: The Truth Table.

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	0	0
2	0	1	1	0
3	1	0	0	1
4	1	1	1	0

Since the network is a deterministic one, each column in A has only one non-zero element and the column sum is one. We remark that there is an one-to-one relation between a BN and its corresponding BN matrix.

To overcome the deterministic rigidity of a BN, extension to a probabilistic setting is natural. To extend the concepts of a BN to a stochastic model, for each vertex v_i in a PBN, instead of having only one Boolean function as in BN, there are a number of Boolean functions (predictor functions) $f_j^{(i)}(j = 1,2,\ldots,l(i))$ to be chosen for determining the state of gene v_i. The probability of choosing $f_j^{(i)}$ as the predictor function is

$$c_j^{(i)}, 0 \le c_j^{(i)} \le 1 \quad \text{and} \quad \sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad \text{for} \quad i = 1,2,\ldots,n. \tag{4}$$

The probability $c_j^{(i)}$ can be estimated by using the statistical method, namely Coefficient of Determination (COD) [8] with real gene expression data sets.

Let f_j be the jth possible realization,

$$f_j = (f_{j_1}^{(1)}, f_{j_2}^{(2)}, \ldots, f_{j_n}^{(n)}), \quad 1 \le j_i \le l(i), \quad i = 1,2,\ldots,n$$

where $l(i) \le 2^{2^n}$ is the total number of possible Boolean functions of gene i. Then in an independent PBN (the selection of the Boolean function for each gene is assumed to be independent), the probability of choosing the corresponding BN is given by

$$q_j = \prod_{i=1}^{n} c_{j_i}^{(i)}, \quad j = 1,2,\ldots,N. \tag{5}$$

Therefore there are at most

$$N = \prod_{i=1}^{n} l(i) \tag{6}$$

different possible realizations of BNs. We note that the transition process among the states in the set S is a Markov chain process. Let $\mathbf{a}$ and $\mathbf{b}$ be any two column vectors in the set S. Then the transition probability

$$\begin{aligned} & \text{Prob}\,\{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}\} \\ = & \sum_{j=1}^{N} \text{Prob}\,\{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}, \text{the } j\text{th network is selected}\} \cdot q_j. \end{aligned}$$

The transition probability matrix A of the PBN (Markov chain) can then be obtained by computing the above probabilities for all the possible states in the set S in (2). In fact, it can be shown that the transition probability matrix A can be written as the sum of the Boolean network matrices A_i ([3]):

$$A = \sum_{i=1}^{N} q_i A_i \tag{7}$$

where q_i is the probability of choosing the BN having the BN matrix A_i. Here we will focus on estimating q_i when A and A_i are given. We remark that the selection probabilities $c_{i_j}^{(i)}$ in (5) can also be estimated when the estimated value of q_i is available [5].

3 The Inverse Problem and the Heuristic Algorithms

In this section, we first describe the inverse problem of constructing a PBN from a given transition probability matrix A and a set of Boolean networks $\{A_i\}$. We then present the heuristic algorithm.

We are interested in getting the parameters $q_i, i = 1,2,\ldots,N$ when A is given. Since the problem size is huge and A is usually very sparse. Here we assume that each column of A has m non-zero entries. In this case, we have $N = m^{2^n}$ and we can order $A_1, A_2, \cdots, A_{m^{2^n}}$ systematically. We note that q_i and A_i are non-negative and there are only $m \cdot 2^n$ non-zero entries in A. Thus we have $m \cdot 2^n$ equations for m^{2^n} unknowns. The problem is large and indetermined.

3.1 Algorithm I

In the following, we propose a simple and fast algorithm for constructing a PBN from the given transition probability matrix A. In particular the complexity of the algorithm is $O(m2^n)$ and we have

$$A = \sum_{i=1}^{M} q_i A_i \quad \text{where} \quad M \leq m2^n.$$

Algorithm I

Step 0: Set $R_1 = A$; $k = 0$
Step 1: $k := k+1$
Step 2: Choose the smallest non-zero entry q_k from R_k. Then for each of the other columns, find the largest entry. All the entries are bigger then or equal to q_k. Suppose the concerned entries are given by $[R_k]_{k_1,1}, [R_k]_{k_2,2}, \cdots, [R_k]_{k_{2^n},2^n}$ Then we define the following Boolean network matrix: $A_k = [\mathbf{e}_{k_1,1}, \cdots, \mathbf{e}_{k_{2^n},2^n}]$. Here $\mathbf{e}_{j,i}$ is the unit column vector whose jth entry is 1 for $i = 1,\ldots,2^n$.
Step 3: $R_{k+1} = R_k - q_k A_k$
Step 4: If R_{k+1} is the zero matrix then go to Step 5 otherwise go to Step 1.
Step 5: $M = k$ and $A = \sum_{i=1}^{M} q_k A_k$.

We then demonstrate the above algorithm by a simple example. The transition probability matrix of a PBN is given by

$$A = \begin{pmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \end{pmatrix}.$$

We would like to find a linear combination of Boolean network matrices constituting the matrix A. Here a Boolean network matrix is a matrix having 0 or 1 as its entries and each column it has only one non-zero entry.

We choose the smallest non-zero element in A and it is 0.2 from column two. We then take this out from column two. For the other columns, we are going to subtract this from the largest entry of each column. We have

$$A = \begin{pmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \end{pmatrix} = 0.2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.4 \\ 0.8 & 0.4 \end{pmatrix} \equiv 0.2A_1 + R_2.$$

We then do the same operations to R_2 and get

$$R_2 = \begin{pmatrix} 0.0 & 0.4 \\ 0.8 & 0.4 \end{pmatrix} = 0.4 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.0 \\ 0.4 & 0.4 \end{pmatrix} \equiv 0.4A_2 + R_3.$$

Finally we have

$$R_3 = 0.4 \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \equiv 0.4A_3.$$

Therefore we have $A = 0.2A_1 + 0.4A_2 + 0.4A_3$.

The following theorem tells us that the algorithm is very efficient.

Theorem 1: If each column of A has at most m non-zero entries, then Algorithm I will terminate in at most $m2^n$ iterations or $M \leq m2^n$.

Proof. We note that there are at most $m2^n$ non-zero entries in A. Each time from Step 2 to Step 3, (from R_k to R_{k+1}), the number of non-zero entries decreases by at least one and each of the column sum will decrease by the same amount q_k. Thus we conclude that the algorithm will terminate by at most $m2^n$ iterations. □

3.2 Algorithm II: A Modified Algorithm

The disadvantage of the algorithm is that it may only give a solution. This is because in Step 2, for each of the column, we choose the position with the largest entry to form part of a Boolean network matrix and deduct the value q_k. However, in fact, one can choose any one of the non-zero entries and proceed with the algorithm. Here we introduce a probabilistic approach to modify Step 2. Instead of choosing the largest values, suppose for the ith column there are m non-zero entries $R_{1i}, R_{2i}, \ldots, R_{mi}$. Then we assume the probability of choosing R_{ji}

$$\frac{R_{ji}}{R_{1i} + R_{2i} + \cdots + R_{mi}}. \tag{8}$$

We may further define a measure of goodness of the solution q_i by its entropy as follows:

$$-\sum_{i=1}^{M} q_i \log q_i. \tag{9}$$

The modified algorithm can then be run for a number of times and to get the best solution in the sense of (9).

N	m	Algorithm I	Algorithm II
8	4	3.0267	3.1238
16	4	2.3428	2.5021
32	4	4.2682	4.3618
64	4	4.9935	5.0296

Table 2: Comparison of Entropy of the Two Algorithms.

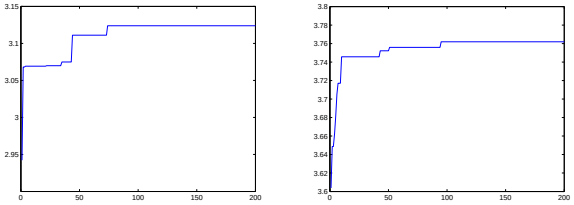

Figure 1: The Entropy for the Case of $N = 8$ and $K = 4$ (Left) and the Case of $N = 16$ and $K = 4$ (Right).

In the following, we conduct some numerical examples. In the examples the transition probability matrices are generated randomly for the cases of $N = 8, 16, 32, 64$. with the maximum number of non-zero entries in each column $m = 4$. We adopt entropy objective function in (9) as a measurement of the solution obtained. Table II gives the comparison of entropy between Algorithm I and Algorithm II (best of 200 runs). The figures report the best entropy obtained in running Algorithm II 200 times.

4 Concluding Remarks

We propose efficient algorithms for constructing a Probabilistic Boolean Network (PBN) when its transition probability matrix is given. The followings are further research issues: (i) proposing new measurement of the goodness of the solutions. (ii) more efficient and effective algorithm for construction of PBNs.

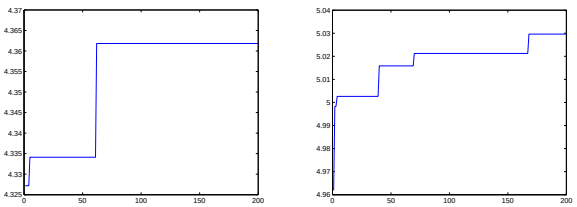

Figure 2: The Entropy for the Case of $N = 32$ and $K = 4$ (Left) and the Case of $N = 64$ and $K = 4$ (Right).

Acknowledges

Research support in part by HKRGC Grant 7017/07P and HKU CRCG Grants and HKU strategic theme grant on computational sciences.

References

[1] T. Akutsu, M. Hayasida, W, Ching and M. Ng. *Control of Boolean Networks: Hardness Results and Algorithms for Tree Structured Networks*, Journal of Theoretical Biology, 244: 670-679, 2007.

[2] W. Ching, E. Fung, M. Ng and T. Akutsu. *On Construction of Stochastic Genetic Networks Based on Gene Expression Sequences*, International Journal of Neural Systems, 15: 297-310, 2005.

[3] W. Ching, S. Zhang, M. Ng and T. Akutsu. *An Approximation Method for Solving the Steady-state Probability Distribution of Probabilistic Boolean Networks*, Bioinformatics, 23: 1511-1518, 2007.

[4] W. Ching, S. Zhang, Y. Jiao, T. Akutsu and A. Wong. *Optimal Finite-Horizon Control for Probabilistic Boolean Networks with Hard Constraints*, The International Symposium on Optimization and Systems Biology (OSB 2007), Lecture Notes in Operations Research, 2007.

[5] W. Ching, S. Zhang, X. Chen and N. Tsing. *On Construction of PBNs from a Prescribed Stationary Distribution*, submitted, 2008.

[6] A. Datta, A. Choudhary, M. Bitter, and E. R. Dougherty. *External Control in Markovian Genetic Regulatory Networks*, Machine Learning, 52 : 169-191, 2003.

[7] H. de Jong. *Modeling and Simulation of Genetic Regulatory Systems: A Literature Review*, J. Comput. Biol., 9: 69-103, 2002.

[8] E. Dougherty, S. Kim and Y. Chen. *Coefficient of Determination in Nonlinear Signal Processing*, Signal Processing, 80: 2219-2235, 2000.

[9] S. Huang and D.E. Ingber. *Shape-dependent Control of Cell Growth, Differentiation, and Apoptosis: Switching Between Attractors in Cell Regulatory Networks*, Exp. Cell Res., 261: 91-103, 2000.

[10] S. Kauffman. *Metabolic Stability and Epigenesis in Randomly Constructed Gene Nets*, J. Theoret. Biol., 22: 437-467, 1969.

[11] S. Kauffman. *Homeostasis and Differentiation in Random Genetic Control Networks*, Nature, 224: 177-178, 1969.

[12] S. Kauffman. *The Large Scale Structure and Dynamics of Genetic Control Circuits: An Ensemble Approach*, J. Theoret. Biol., 44: 167-190, 1974.

[13] S. Kauffman. *The Origins of Order: Self-organization and Selection in Evolution*, New York: Oxford Univ. Press, 1993.

[14] S. Kim, S. Imoto and S. Miyano. *Dynamic Bayesian Network and Nonparametric Regression for Nonlinear Modeling of Gene Networks from time Series Gene Expression Data*, Proc. 1st Computational Methods in Systems Biology, Lecture Note in Computer Science, 2602: 104-113, 2003.

[15] R. Pal, I. Ivanov, A. Datta, M. Bittner and E. Dougherty. *Generating Boolean Networks with a Prescribed Attractor Structure*, Bioinformatics, 21: 4021-4025, 2005.

[16] I. Shmulevich, E. Dougherty, S. Kim and W. Zhang. *Probabilistic Boolean Networks: A Rule-based Uncertainty Model for Gene Regulatory Networks*, Bioinformatics, 18: 261-274, 2002.

[17] I. Shmulevich, E. Dougherty, S. Kim and W. Zhang. *Control of Stationary Behavior in Probabilistic Boolean Networks by Means of Structural Intervention*, Journal of Biological Systems, 10: 431-445, 2002.

[18] I. Shmulevich, E. Dougherty, S. Kim and W. Zhang. *From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks*, Proceedings of the IEEE, 90: 1778-1792, 2002.

[19] I. Shmulevich and E. Dougherty. *Genomic Signal Processing*, Princeton University Press, U.S. 2007.

[20] P. Smolen, D. Baxter and J. Byrne. *Mathematical Modeling of Gene Network*, Neuron, 26: 567-580, 2000.

[21] R. Somogyi and C. Sniegoski. *Modeling the Complexity of Gene Networks: Understanding Multigenic and Pleiotropic Regulation*, Complexity, 1: 45-63, 1996.

[22] S. Zhang, W. Ching, M. Ng and T. Akutsu. *Simulation Study in Probabilistic Boolean Network Models for Genetic Regulatory Networks*, Journal of Data Mining and Bioinformatics, 1 :217-240, 2007.

[23] S. Zhang, W. Ching, N. Tsing, H. Leung and D. Guo. *A Multiple Regression Approach for Building Genetic Networks*, Proceedings of the International Conference on BioMedical Engineering and Informatics (BMEI2008) Sanya, China (in CD-ROM), 2008.