Title	Generating probabilistic Boolean networks from a prescribed stationary distribution
Author(s)	Zhang, SQ; Ching, WK; Chen, X; Tsing, NK
Citation	Information Sciences, 2010, v. 180 n. 13, p. 2560-2570
Issued Date	2010
URL	http://hdl.handle.net/10722/75296
Rights	Information Sciences. Copyright © Elsevier Inc.

Generating Probabilistic Boolean Networks from a Prescribed Stationary Distribution

Shu-Qin Zhang [*] Wai-Ki Ching [†] Xi Chen [‡] Nam-Kiu Tsing [§]

October 25, 2010

Abstract

Modeling gene regulation is an important problem in genomic research. Boolean networks (BN) and its generalization Probabilistic Boolean networks (PBNs) have been proposed to model genetic regulatory interactions. BN is a deterministic model while PBN is a stochastic model. In a PBN, on one hand, its stationary distribution gives important information about the long-run behavior of the network. On the other hand, one may be interested in system synthesis which requires the construction of networks from the observed stationary distribution. This results in an inverse problem which is ill-posed and challenging. Because there may be many networks or no network having the given properties and the size of the inverse problem is huge. In this paper, we consider the problem of constructing PBNs from a given stationary distribution and a set of given Boolean Networks (BNs). We first formulate the inverse problem as a constrained least squares problem. We then propose a heuristic method based on Conjugate Gradient (CG) algorithm, an iterative method, to solve the resulting least squares problem. We also introduce an estimation method for the parameters of the PBNs. Numerical examples are then given to demonstrate the effectiveness of the proposed methods.

Key Words: Boolean networks, genetic networks, inverse problem, probabilistic Boolean networks, stationary distribution.

[*]School of Mathematical Sciences, Fudan University, Shanghai, China. Email: zhangs@fudan.edu.cn

[†]Corresponding author. Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: wching@hkusua.hku.hk.

[‡]Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. Email:dlkcissy@hotmail.com

[§]Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: nktsing@hku.hk.

1 Introduction

Building mathematical models from microarray data sets and developing efficient numerical algorithms for studying the regulatory interactions among DNA, RNA, proteins and small molecules are hot topics in bioinformatics [5, 17]. There have been many formalisms proposed in the literature to study genetic regulatory networks such as Bayesian networks [22, 38], Boolean networks (BNs) [19, 20], correlation-based methods [26], multivariate Markov chain model [6], regression model [41], evolutionary models [2], Probabilistic Boolean Networks (PBNs) [31, 32, 33, 35]. Reviews on many other mathematical models can be found in [13, 37].

Among all these models, BN and its extension PBN have received much attention as they can capture the "switching behavior" of a biological process [17]. 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 of all the genes are given, we say that a BN is defined. We remark that complex network analysis can be applied to BNs [36]. Furthermore, applications of BNs to complex systems and solid earth geophysics can be found in [16].

There many methods available for inferring a BN. A general nonlinear framework for inferring the associations between the expressions of genes via Coefficient of Determination (COD) in multivariate expression arrays is given in [21]. The coefficient there measures the degree to which the expression levels of an observed gene set can be applied to improve the prediction of the expression level of the target gene relative to the best prediction in the absence of observations. Later inferring a BN is formulated as a consistency and best-fit extension problem [24]. More recent works related to BN construction can be found in [15, 27]. In [15], an universal minimum description length based method is proposed where the description length is derived from a universal normalized maximum likelihood model. The search space is reduced by an implementable analogue of Kolmogorov's structure function. In [27], Liu et al. analyzed the expected inference error relative to deviations in the networks' dynamic regime from the assumption of criticality. By taking into account the criticality via a penalty term in the inference procedure the prediction accuracy can be improved.

A BN is a deterministic model, the only randomness comes from its initial state. Given an initial state, the BN will eventually enter into a set of state(s) called the attractor cycles. The attractor cycles have significant biological meanings [17]. Since genetic regulation process exhibits uncertainty and microarray data sets used to infer the model have errors due to experimental noise in the complex measurement processes, it is more realistic to consider a stochastic model, the Probabilistic Boolean network (PBN). To extend BNs to PBNs, for each gene, there can be more than one Boolean function (a set of Boolean functions with probabilities assigning to them). The dynamics (transitions) of a PBN can be described by a Markov chain [31, 35].

For the inference of a PBN, the Boolean functions, the predictor sets and the selection probabilities of the Boolean functions can be obtained using the methods proposed in [14, 21, 25]. Given a PBN, assuming the underlying Markov chain is irreducible, its long-run behavior is characterized by its stationary distribution. The stationary distribution gives the first-order statistical information of a PBN and one can understand a genetic network, and identify the influence of different genes in such a network. Recently an iterative method, power method in conjunction with an efficient construction method for the transition probability matrix has been proposed to compute the stationary distribution [40]. Later a matrix approximation method has been proposed in [7] to get an approximation of the stationary distribution. Furthermore, it is possible to control some genes in a network so as to drive the whole network into a desirable stationary distribution. Therapeutic gene intervention or gene control policy [8, 11, 12, 32, 35, 41] can therefore be developed and studied.

In this paper, we focus on the construction of PBNs based on a given stationary distribution and a given set of BNs. This is an inverse problem of large size and it is ill-posed which means that there can be many networks or even no network having the desirable properties. A modified Conjugate Gradient (CG) method is employed to solve the problem. In fact, for the BN case, Pal, et al. [29] have presented two algorithms to solve the inverse problem of finding attractors constituting a BN. Such problems are very important to network inference from steady-state data, as most microarray data sets are assumed to be obtained from sampling the steady-state [29]. However, to our best knowledge, the PBN case has not been addressed in literature.

The remainder of the paper is structured as follows. In Section 2, we give a brief review on both BNs and PBNs. Section 3 presents the inverse problem of PBNs and the modified CG method for solving the problem. In Section 4, numerical examples are given to demonstrate the effectiveness of the proposed algorithm. Finally concluding remarks are given to address further research issues in Section 5.

2 A Review on Boolean Networks and Probabilistic Boolean Networks

A BN $G(V, F)$ consists of a set of vertices (genes)

$$V = \{v_1, v_2, \ldots, v_n\}$$

and a list of Boolean functions

$$F = \{f_1, f_2, \ldots, f_n\}.$$

We define $v_i(t)$ to be the state (0 or 1) (not expressed or expressed) of the vertex v_i at time t, and $f_i : \{0,1\}^n \to \{0,1\}$, to represent the rules of the regulatory interactions among the genes:

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

where

$$\mathbf{v}(t) = (v_1(t), v_2(t), \ldots, v_n(t))^T$$

is called the Gene Activity Profile (GAP). Here $\mathbf{x}^T$ is the transpose of the vector $\mathbf{x}$. The GAP can take any possible form (states) from the set

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

and thus totally there are 2^n possible states.

The following is an example of a BN of three genes. We give the truth table of the BN in Table 1.

Table 1

State	$v_1(t)$	$v_2(t)$	$v_3(t)$	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$
1	0	0	0	0	0	0
2	0	0	1	1	1	0
3	0	1	0	1	0	1
4	0	1	1	1	1	0
5	1	0	0	0	1	0
6	1	0	1	0	1	1
7	1	1	0	0	0	1
8	1	1	1	1	0	0

From the truth table above, we observe if the current network state is $(0,0,0)$ (State 1) then in the next time step, the network state will be again $(0,0,0)$. This means $(0,0,0) \leftrightarrow (0,0,0)$ is a cycle of period one. If the current network state is $(0,0,1)$ (State 2) then in the next time step, the network state will be $(1,1,0)$. While if the current network state is $(1,1,0)$ (State 7) then in the next time step, the network state will be $(0,0,1)$ (State 2). Thus $(0,0,1) \leftrightarrow (1,1,0)$ is a cycle of period two. The transition probability matrix of the 3-gene BN is then given by

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

Here each column has only one non-zero element and it is one. This is a special class of column stochastic matrices. We remark that for each given BN, its transition probability matrix is unique up to the ordering of the states. This representation of a BN is useful

for the extension to PBN.

Since a BN is a deterministic model, to overcome this deterministic rigidity, 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 and usually $l(i)$ cannot be very large. The probability of choosing $f_j^{(i)}$ as the predictor function is $c_j^{(i)}$,

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

The probability $c_j^{(i)}$ can be estimated by using the method of Coefficient of Determination (COD) [14] with real gene expression data sets.

We let f_j be the jth possible realization, where

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

Suppose that the selection of the Boolean function f_{j_i} for each gene i is an independent process, then the probability of choosing the corresponding BN with Boolean functions $(f_{j_1}, f_{j_2}, \ldots, f_{j_n})$ is given by

$$q_{j_1 j_2 \cdots j_n} = \prod_{i=1}^{n} c_{j_i}^{(i)}.$$

There are at most

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

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

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

Here we let

$$q_j = q_{j_1 j_2 \cdots j_n} \quad \text{and} \quad j = j_1 + \sum_{i=2}^{n} \left((j_i - 1)(\prod_{k=1}^{i-1} l(k)) \right).$$

We will then use both of them when there is no confusion. By letting **a** and **b** take all the possible states in S, one can get the transition probability matrix of the Markov chain (or the PBN). The transition probability matrix can be written as ([7])

$$A = \sum_{j=1}^{N} q_j A_j. \tag{2}$$

Here A_j is the corresponding transition matrix of the jth BN (see Equation (1) for instance) and q_j is the probability of choosing the jth BN. We note that there are at most $N2^n$ nonzero entries in the transition probability matrix A and this means the matrix is sparse, i.e., having a lot of zero entries.

We remark that there are several different kinds of PBNs. The instantaneously random PBN described above is the simplest one. Random gene perturbation can be added to a PBN to stabilize the network. It is the description of the random inputs from the outside due to external stimuli. The effect of the random gene perturbation is to make the genes flip from state 1 to state 0 or vice versa. This makes the underlying Markov chain of the PBN ergodic [34]. The instantaneously random PBN can also be extended to the context-sensitive PBN [30]. For simplicity of discussion, here we focus on the inverse problem of instantaneously random PBNs.

3 The Inverse Problem and the Modified Conjugate Gradient Method

In this section, we first present the inverse problem of building a PBN and we then present the modified Conjugate Gradient (CG) method for solving the inverse problem.

3.1 The Inverse Problem

Suppose that the possible BNs constituting the PBN are known and they are denoted by $(A_1, A_2, \ldots, A_N)$ and the steady-state behavior of the PBN, the stationary distribution **p**

can also be observed. Then we have

$$A\mathbf{p} = \mathbf{p} \quad \text{and} \quad A = \sum_{j=1}^{N} q_j A_j.$$

The inverse problem here is to get the parameters $q_j, j = 1, 2, \ldots, N$. Now we let the matrix

$$V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$$

where

$$\mathbf{v}_j = A_j\mathbf{p} \quad \text{and} \quad \mathbf{q} = (q_1, q_2, \ldots, q_N)^T.$$

Then one possible way to get q_j is to consider the following minimization problem:

$$h(\mathbf{q}^*) = \min ||V\mathbf{q} - \mathbf{p}||_2^2 \tag{3}$$

subject to

$$0 \leq q_j \leq 1 \quad \text{and} \quad \sum_{j=1}^{N} q_j = 1.$$

3.2 The Modified Conjugate Gradient Method

We note that in practice the matrix V can be very large, it may not be possible to store the whole matrix and therefore one may seek for iterative method for solving the above minimization. One possible candidate is to consider the Conjugate Gradient (CG) method, see for instance [3, p. 470]. Given a symmetric positive definite $m \times m$ matrix H_m, a well-known and successful iterative method for solving the linear system $H_m\mathbf{x} = \mathbf{b}$ is the CG method. The convergence rate of this method depends on the spectrum of the matrix H_m. For example if the spectrum of H_m is contained in an interval, i.e. $\sigma(H_m) \subseteq [a, b]$ and $\mathbf{x}_i$ is the approximate solution obtained in the ith iteration, then the error $\mathbf{r}_i = \mathbf{b} - H_m\mathbf{x}_i$ is given by

$$\frac{||\mathbf{r}_i||_2}{||\mathbf{r}_0||_2} \leq 2\left(\frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}}\right)^i. \tag{4}$$

Generally speaking, it can be shown that CG method will converge in at most m steps [3], i.e., the number of iterations required for convergence is of $O(m)$.

We observe that

$$||V\mathbf{q}-\mathbf{p}||_2^2 = (V\mathbf{q}-\mathbf{p})^T(V\mathbf{q}-\mathbf{p})$$

and

$$(V\mathbf{q}-\mathbf{p})^T(V\mathbf{q}-\mathbf{p}) = \mathbf{q}^TV^TV\mathbf{q} - 2\mathbf{q}^TV^T\mathbf{p} + \mathbf{p}^T\mathbf{p}.$$

Thus the minimization problem (3) without constraints is equivalent to

$$\min_{\mathbf{q}}\{\mathbf{q}^TV^TV\mathbf{q} - 2\mathbf{q}^TV^T\mathbf{p}\}.$$

If V is a full rank matrix then V^TV is a symmetric positive definite matrix. The minimization problem without constraints is equivalent to solving

$$V^TV\mathbf{q} = V^T\mathbf{p}$$

with the CG method, see for instance [23]. We note that if there is a probability distribution $\mathbf{q}$ satisfying the equation $V\mathbf{q} = \mathbf{p}$ with $\mathbf{1}^T\mathbf{q} = 1$ and $\mathbf{0} \leq \mathbf{q} \leq \mathbf{1}$, then the CG method can yield the solution. To ensure that $\mathbf{1}^T\mathbf{q} = 1$, we add a row of $(w, w, \ldots, w)$ to the bottom of the matrix V and form the new matrix is $\bar{V}$. At the same time, we add an entry w at the end of the vector $\mathbf{p}$ to get a new vector $\bar{\mathbf{p}}$. Here w is a large positive number so as to ensure the constraint $\mathbf{1}^T\mathbf{q} = 1$ is active. Thus we consider the revised equation:

$$\bar{V}^T\bar{V}\mathbf{q} = \bar{V}^T\bar{\mathbf{p}}.$$

Since it may happen that there is no such a vector $\mathbf{q}$, the CG algorithm has to be modified to ensure the first constraint $\mathbf{0} \leq \mathbf{q} \leq \mathbf{1}$ is satisfied, see Appendix 6.1. The modification is to ensure that the constraint $\mathbf{0} \leq \mathbf{q} \leq \mathbf{1}$ has to be satisfied in each iteration step. We have to run the modified CG method for a number of times with different initial guesses to get the best solution in the sense of the smallest residual error in 2-norm. We remark that the main computational cost of the CG method comes from the matrix-vector multiplication which takes $O(N2^n)$ operations. Since the number of iteration for convergence is $O(N)$, if we are going to run the CG method for T times with T different initial guesses, then

the total complexity will be $O(TN^2 2^n)$.

3.3 Estimation of $c_j^{(i)}$

Once we get the estimates of the probabilities $q_{j_1 j_2 \cdots j_n}$, we may have N equations of the following form:

$$\prod_{i=1}^{n} c_{j_i}^{(i)} = q_{j_1 j_2 \cdots j_n}.$$

For ease of presentation of the analysis, in the following, we consider the special case that $l(i) = 2$ for $i = 1, 2, \ldots, n$. We remark that the techniques can be applied similarly to the cases $l(i) \geq 3$ for $i = 1, 2, \ldots, n$.

Now we have $c_2^{(i)} = 1 - c_1^{(i)}$ and $N = 2^n$. To estimate $c_1^{(k)}$, we note that for $j_k = 1, 2$, we have respectively

$$c_{j_1}^{(1)} \cdots c_{j_{k-1}}^{(k-1)} c_1^{(k)} c_{j_{k+1}}^{(k+1)} \cdots c_{j_n}^{(n)} = q_{j_1 \cdots j_{k-1} 1 j_{k+1} \cdots j_n}$$

and

$$c_{j_1}^{(1)} \cdots c_{j_{k-1}}^{(k-1)} (1 - c_1^{(k)}) c_{j_{k+1}}^{(k+1)} \cdots c_{j_n}^{(n)} = q_{j_1 \cdots j_{k-1} 2 j_{k+1} \cdots j_n}.$$

Therefore for any $j_1, \cdots, j_{k-1}, j_{k+1}, \cdots, j_n \in \{1, 2\}$, we have

$$\frac{c_1^{(k)}}{1 - c_1^{(k)}} = \frac{q_{j_1 \cdots j_{k-1} 1 j_{k+1} \cdots j_n}}{q_{j_1 \cdots j_{k-1} 2 j_{k+1} \cdots j_n}}. \tag{5}$$

or equivalently

$$c_1^{(k)} r_{j_1 \cdots j_n}^{(k)} - q_{j_1 \cdots j_{k-1} 1 j_{k+1} \cdots j_n} = 0$$

where

$$r_{j_1 \cdots j_n}^{(k)} = q_{j_1 \cdots j_{k-1} 1 j_{k+1} \cdots j_n} + q_{j_1 \cdots j_{k-1} 2 j_{k+1} \cdots j_n}. \tag{6}$$

Since there may not exist $c_1^{(k)}$ satisfying all the equations. One possible way to estimate $c_1^{(k)}$ is to consider the minimizer of the following functional

$$J(c_1^{(k)}) = \sum_{j_1, \cdots, j_{k-1}, j_{k+1}, \ldots, j_n \in \{1,2\}} \left(c_1^{(k)} r_{j_1 \cdots j_n}^{(k)} - q_{j_1 \cdots j_{k-1} 1 j_{k+1} \cdots j_n} \right)^2 \tag{7}$$

and $0 \leq c_1^{(k)} \leq 1$.

Since the objective function of the above minimization problem is of the form:

$$f(x) = \sum_{i=1}^{n} (a_i x_i - b_i)^2,$$

by solving

$$f'(x) = \sum_{i=1}^{n} 2a_i(a_i x_i - b_i) = 0$$

we obtain the optimal solution as follow:

$$x^* = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2}.$$

The minimizer of the problem (7) can be easily shown to be

$$c_1^{(k)*} = \frac{\sum_{j_1,\cdots,j_{k-1},j_{k+1},\ldots,j_n \in \{1,2\}} r^{(k)}_{j_1\cdots j_n} \times q_{j_1\cdots j_{k-1}1 j_{k+1}\cdots j_n}}{\sum_{j_1,\cdots,j_{k-1},j_{k+1},\ldots,j_n \in \{1,2\}} (r^{(k)}_{j_1\cdots j_n})^2}.$$

It is straightforward to show that $c_1^{(k)*} \in [0,1]$ by using (6). We then define

$$J(c_1^{(1)}, \ldots, c_1^{(n)}) = \sum_{k=1}^{n} J(c_1^{(k)}).$$

as a measure of the the fitness of the estimators. The smaller this value is, the better the estimators are. Thus one may use

$$J(c_1^{(1)*}, \ldots, c_1^{(n)*}) = \sum_{k=1}^{n} J(c_1^{(k)*}) \tag{8}$$

together with the optimal value $h(\mathbf{q}^*)$ in (3) to rank different PBNs obtained.

4 Numerical Examples

In this section, we present some numerical examples. We first give a numerical demonstration of our proposed algorithm in Example I. We then consider a frequently used example

of a 3-genes network proposed by Shmulevich, et al. [31].

4.1 Example I

In the first example, we consider a PBN with three genes $n = 3$. The truth table of the Boolean functions are given in Table 2.

Table 2

State	$(v_1(t), v_2(t), v_3(t))$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
1	(0,0,0)	0	0	0	1	0
2	(0,0,1)	1	0	1	1	0
3	(0,1,0)	1	0	0	0	1
4	(0,1,1)	1	1	1	1	0
5	(1,0,0)	0	1	1	0	0
6	(1,0,1)	0	1	1	0	1
7	(1,1,0)	0	1	0	0	1
8	(1,1,1)	1	1	0	1	0

Since Gene 1 and 3 have two possible Boolean functions, there are four possible BNs. The transition matrices of the BNs A_1, A_2, A_3, A_4 are given as follow:

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad A_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Suppose

$$(c_1^{(1)}, c_2^{(1)}) = (0.25, 0.75) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.60, 0.40) \tag{9}$$

then we have

$$(q_1, q_2, q_3, q_4) = (0.15, 0.10, 0.45, 0.30). \tag{10}$$

It is straightforward to check that the stationary distribution is given by

$$\mathbf{p} = (0.1394, 0.1394, 0.1002, 0.0127, 0.1574, 0.1271, 0.2570, 0.0667)^T.$$

Now we may assume that $\mathbf{p}$ is the observed stationary distribution and we wish to apply the modified CG method to find q_i and $c_j^{(i)}$ pretending that q_i and $c_j^{(i)}$ are not known. We let

$$\mathbf{v}_i = A_i \mathbf{p} \quad i = 1, 2, 3, 4$$

and get

$$V = \begin{pmatrix} 0.2570 & 0.1394 & 0.1002 & 0.1394 \\ 0.1394 & 0.2570 & 0.1394 & 0.1002 \\ 0.2845 & 0.1574 & 0.0000 & 0.1394 \\ 0.0000 & 0.1271 & 0.0000 & 0.0000 \\ 0.1002 & 0.0667 & 0.2570 & 0.0667 \\ 0.0667 & 0.1002 & 0.0667 & 0.2570 \\ 0.0000 & 0.1521 & 0.4239 & 0.1701 \\ 0.1521 & 0.0000 & 0.0127 & 0.1271 \end{pmatrix}.$$

Because $q_1 + q_2 + q_3 + q_4 = 1$, to include this constraint, one has to add one more row of

(w, w, w, w) to the bottom of the matrix V and add one more entry (w) to the bottom of the vector $\mathbf{p}$, i.e.

$$\bar{V} = \begin{pmatrix} 0.2570 & 0.1394 & 0.1002 & 0.1394 \\ 0.1394 & 0.2570 & 0.1394 & 0.1002 \\ 0.2845 & 0.1574 & 0.0000 & 0.1394 \\ 0.0000 & 0.1271 & 0.0000 & 0.0000 \\ 0.1002 & 0.0667 & 0.2570 & 0.0667 \\ 0.0667 & 0.1002 & 0.0667 & 0.2570 \\ 0.0000 & 0.1521 & 0.4239 & 0.1701 \\ 0.1521 & 0.0000 & 0.0127 & 0.1271 \\ w & w & w & w \end{pmatrix}$$

and

$$\bar{\mathbf{p}} = (0.1394, 0.1394, 0.1002, 0.0127, 0.1574, 0.1271, 0.2570, 0.0667, w)^T.$$

Using the modified CG method with $w = 100$, one can recover the solution in (10).

To obtain $c_1^{(1)}, c_2^{(1)}, c_1^{(2)}, c_2^{(2)}$ we have the following equations:

$$\begin{cases} c_1^{(1)} c_1^{(2)} & = & q_1 = 0.15 \\ c_1^{(1)} c_2^{(2)} = c_1^{(1)}(1 - c_1^{(2)}) & = & q_2 = 0.10 \\ c_2^{(1)} c_1^{(2)} & = & q_3 = 0.45 \\ c_2^{(1)} c_2^{(2)} = c_2^{(1)}(1 - c_1^{(2)}) & = & q_4 = 0.30. \end{cases}$$

Then we have

$$\frac{c_1^{(1)}}{1 - c_1^{(1)}} = \frac{0.10}{0.30} \quad \text{and} \quad \frac{c_1^{(1)}}{1 - c_1^{(1)}} = \frac{0.15}{0.45}.$$

and

$$\frac{c_1^{(2)}}{1 - c_1^{(2)}} = \frac{0.45}{0.30} \quad \text{and} \quad \frac{c_1^{(2)}}{1 - c_1^{(2)}} = \frac{0.15}{0.10}$$

Solving the above equations, we have the solutions as follow:

$$c_1^{(1)*} = \frac{1}{4}, \quad c_2^{(1)*} = \frac{3}{4}, \quad c_1^{(2)*} = \frac{3}{5}, \quad c_2^{(2)*} = \frac{2}{5}$$

same as (9). The sum of squares of errors $J(c_1^{(1)*}, c_1^{(2)*}) = 0$ in this example.

In general, in the case of n genes and each gene has two possible Boolean func-

tions, there are $(2^{2^n})^{2n}$ truth tables and for each truth table, there are $N = 2^n$ BNs. The computational cost for examining all the possible PBNs will be $O((2^{2^n})^{2n} T N^2 2^n) = O(T 2^{n2^{n+1}+3n})$. Thus to find the possible BNs, i.e., the matrices A_i is still a challenging problem for future research.

4.2 Example II

In this subsection, we consider a frequently used example of a 3-genes network proposed by Shmulevich, et al. [31]. The function sets $F = (F_1, F_2, F_3)$, where

$$F_1 = \{f_1^{(1)}, f_2^{(1)}\}, \quad F_2 = \{f_1^{(2)}\} \quad \text{and} \quad F_3 = \{f_1^{(3)}, f_2^{(3)}\}.$$

The functions and their selection probability are given in Table 3.

Table 3

State	$(v_1(t), v_2(t), v_3(t))$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
1	(0,0,0)	0	0	0	0	0
2	(0,0,1)	1	1	1	0	0
3	(0,1,0)	1	1	1	0	0
4	(0,1,1)	1	0	0	1	0
5	(1,0,0)	0	0	1	0	0
6	(1,0,1)	1	1	1	1	0
7	(1,1,0)	1	1	0	1	0
8	(1,1,1)	1	1	1	1	1
	$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

Since

$$(c_1^{(1)}, c_2^{(1)}) = (0.60, 0.40) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.50, 0.50)$$

then we have

$$(q_1, q_2, q_3, q_4) = (0.30, 0.20, 0.30, 0.20).$$

It is straightforward to check that the stationary distribution is given by

$$\mathbf{p} = (0.5063, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.4937)^T.$$

Now we may assume that $\mathbf{p}$ is the observed stationary distribution and we wish to apply CG method to find q_i and $c_j^{(i)}$ pretending that q_i and $c_j^{(i)}$ are not known.

Using our proposed algorithm, we can get many different solutions when using different randomly chosen initial guesses. For this example, we applied our algorithm 30 times and got 30 different solutions. The followings are three of the solutions obtained:

(1) $q = [0.3901, 0.2267, 0.3704, 0.0128]$

$$(c_1^{(1)}, c_2^{(1)}) = (0.7256, 0.2744) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.5520, 0.4480).$$

(2) $q = [0.2286, 0.2067, 0.3754, 0.1893]$

$$(c_1^{(1)}, c_2^{(1)}) = (0.6128, 0.3872) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.4216, 0.5784).$$

(3) $q = [0.1820, 0.2931, 0.2732, 0.2517]$

$$(c_1^{(1)}, c_2^{(1)}) = (0.4587, 0.5413) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.4812, 0.5188).$$

To further evaluate the solutions, one can apply equation (8) as a measure to rank these solutions. Then have $q^* = [0.3, 0.2, 0.3, 0.2]$ and

$$(c_1^{(1)}, c_2^{(1)}) = (0.60, 0.40) \quad \text{and} \quad (c_1^{(2)}, c_2^{(2)}) = (0.50, 0.50)$$

is one of the best solution as the objective function in (8) is 0 in this case.

Moreover, the computational time of our algorithm also depends on the initial guess. In general, it will spend 3 seconds for our algorithm running for 30 times.

5 Concluding Remarks

In this paper, we study the problem of constructing Probabilistic Boolean Networks (PBNs) from a given stationary distribution and a set of BNs. This is an inverse problem of large size. We have formulated the inverse problem as a constrained least squares problem and proposed a heuristic method based on Conjugate Gradient (CG) method to solve the resulting least squares problem.

The followings are some future research issues.

(i) One can also consider the problem of finding PBNs without a set of given BNs, in this case the problem size is huge. One possible way to tackle this huge problem is to consider heuristic methods like genetic algorithms [9]. Moreover, we may still obtained many PBNs, one further possible criteria for selecting PBNs is to consider maximizing the entropy rate of a Markov chain [4, 10]. The entropy rate of a Markov chain is defined as

$$-\sum_{i=1}^{N}\left(\pi_i \sum_{j=1}^{N} p_{ji}\log p_{ji}\right)$$

where π_i is the stationary probability that the Markov chain is in state i and p_{ji} is the one-step transition probability from state i to state j. This can be computed easily by using the observed stationary distribution and the recovered transition probability matrix.

(ii) The computational cost for examining all the possible PBNs using our proposed algorithms is huge, heuristic methods such as genetic algorithms [28] and particle swarm optimization methods [39] will be developed to solve the problem efficiently.

(iii) Extension of the proposed methods to the case that each gene has more than two Boolean functions.

(iv) It is interesting to study the same inverse problem for more general PBNs such as the context-sensitive PBNs.

6 Appendix

6.1 The Modified CG Method

Choose an initial guess of probability distribution $\mathbf{y}_0$;

$H = \bar{V}^T\bar{V}$;

$\mathbf{b} = \bar{V}^T\mathbf{p}$;

$\mathbf{r}_0 = \mathbf{b} - H\mathbf{y}_0$;

$k = 1$;

$\mathbf{p}_1 = \mathbf{r}_0$;

$\alpha_1 = \mathbf{r}_0^t\mathbf{r}_0/\mathbf{p}_1^t H\mathbf{p}_1$;

$\mathbf{y}_1 = \mathbf{y}_0 + \alpha_1\mathbf{p}_1$;

$\mathbf{r}_1 = \mathbf{r}_0 - \alpha_1 H\mathbf{p}_1$;

while $||\mathbf{r}_k||_2 > tolerance$,

$k = k + 1$;

$\beta_k = \mathbf{r}_{k-1}^t\mathbf{r}_{k-1}/\mathbf{r}_{k-2}^t\mathbf{r}_{k-2}$;

$\mathbf{p}_k = \mathbf{r}_{k-1} + \beta_k\mathbf{p}_{k-1}$;

$\alpha_k = \mathbf{r}_{k-1}^t\mathbf{r}_{k-1}/\mathbf{p}_k^t H\mathbf{p}_k$;

$\mathbf{z}_k = \mathbf{y}_{k-1} + \alpha_k\mathbf{p}_k$;

% % % % Additional Constraints % % % %

For $l = 1 : n$,

If $\mathbf{z}_k(l) > 1$ then $temp = (1 - \mathbf{y}_{k-1}(l))/\mathbf{p}_k(l)$;

If $\mathbf{z}_k(l) < 0$ then $temp = -\mathbf{y}_{k-1}(l)/\mathbf{p}_k(l)$;

If $temp < \alpha_k$ then $\alpha_k = temp$;

end;

% % % % % % % % % % % % % % % % % %

$\mathbf{y}_k = \mathbf{y}_{k-1} + \alpha_k\mathbf{p}_k$;

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k H\mathbf{p}_k$;

end;

$\mathbf{y} = \mathbf{y}_k$.

Acknowledgments

The authors would like to thank the three anonymous reviewers for their helpful comments and suggestions. Wai-Ki Ching is supported in part by RGC Grant 7017/07P, Hung Hing Ying Physical Research Fund and HKU CRGC Grants.

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 (2007) 670-679.

[2] S. Ando, E. Sakamoto, and H. Iba, Evolutionary modeling and inference of gene network, Information Sciences 145 (2-4) (2002) 237-259.

[3] O. Axelsson, Iterative solution methods, Cambridge University Press, Cambridge, UK, 1996.

[4] K. Burns, H. Demaree, A chance to learn: On matching probabilities to optimize utilities, Information Sciences 179 (11) (2009) 1599-1607.

[5] J. Celis, M. Kruhøfferm, I. Gromova, and C. Frederiksen, M. Østergaard and T. Ørntoft Gene expression profiling: Monitoring transcription and translation products using DNA microarrays and proteomics, FEBS Letters 480 (2000) 2-16.

[6] 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 (2005) 297-310.

[7] W. Ching, S. Zhang, M. Ng and T. Akutsu, An approximation method for solving the steady-state probability distributions of probabilistic Boolean networks, Bioinformatics 23 (2007) 1511-1518.

[8] W. Ching, S. Zhang, Y. Jiao, T. Akutsu and A. Wong, Optimal control policy for Probabilistic Boolean Networks with hard constraints, IET Systems Biology 3 (2009) 90-99.

[9] W. Ching, H. Leung, N. Tsing and S. Zhang, A genetic algorithm for optimal control of probabilistic Boolean networks, The Second International Symposium on Optimization and Systems Biology (OSB 2008), Lecture Notes in Operations Research 9, Series Editors: Ding-Zhu Du and Xiang-Sun Zhang, 9 (2008) 29-35.

[10] T. Cover and J. Thomas, Elements of information theory, John Wiley and Sons, Inc., 1991.

[11] A. Datta, A. Choudhary, M. Bitter and E. Dougherty, External control in Markovian genetic regulatory networks, Machine Learning 52 (2003) 169-191.

[12] A. Datta, A. Choudhary, M. Bitter and E. Dougherty, External control in Markovian genetic regulatory networks: the imperfect information case, Bioinformatics 20 (2004) 924-930.

[13] H. de Jong, Modeling and simulation of genetic regulatory systems: a literature review, Journal of Computational Biology 9 (2002) 69-103.

[14] E. Dougherty, S. Kim and Y. Chen, Coefficient of Determination in Nonlinear Signal Processing, Signal Processing 80 (2000) 2219-2235.

[15] J. Dougherty, I. Tabus, J. Astola, Inference of Gene Regulatory Networks Based on a Universal Minimum Description Length, EURASIP Journal on Bioinformatics and Systems Biology (2008) Article ID 482090.

[16] M. Ghila, I. Zaliapind, and B. Coluzzib, Boolean delay equations: A simple way of looking at complex systems, Physica D: Nonlinear Phenomena 237 (23) (2008) 2967-2986.

[17] S. Huang, Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery, J. Mol. Med. 77 (1999) 469-480.

[18] S. Huang and D. Ingber, Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks, Exp. Cell Res. 261 (2000) 91-103.

[19] S. Kauffman, Metabolic stability and epigenesis in randomly constructed gene nets, Journal of Theoretical Biology 22 (1969) 437-467.

[20] S. Kauffman, Homeostasis and differentiation in random genetic control networks, Nature 224 (1969) 177-178.

[21] S. Kim, E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, and J. M. Trent, General Nonlinear Framework for the Analysis of Gene Interaction via Multivariate Expression Arrays, Journal of Biomedical Optics 5(4) (2000) 411-424.

[22] 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 Notes in Computer Science 2602 (2003) 104-113.

[23] D. Kincaid and W. Cheney, Numerical analysis : mathematics of scientific computing, (3rd Edition), CA, USA, 2001.

[24] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, On Learning Gene Regulatory Networks Under the Boolean Network Model, Machine Learning 52 (2003) 147-167.

[25] P. Li, C. Zhang, E. J. Perkins, P. Gong, Y. Deng, Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks, BMC Bioinformatics 8(Suppl 7):S13 (2007).

[26] A. Lindløf, and B. Olsson, Could correlation-based methods be used to derive genetic association networks? Information Sciences 146 (2002) (1-4) 103-113.

[27] W. Liu, H. Lähdesmäki, E. R. Dougherty, I. Shmulevich, Inference of Boolean Networks using Sensitivity Regularization. EURASIP Journal on Bioinformatics and Systems Biology (2008) Article ID 780541, 12 pages.

[28] M. Lozano, F. Herrera and J. Cano, Replacement strategies to preserve useful diversity in steady-state genetic algorithms. Information Sciences 178(23) (2008) 4421-4433.

[29] R. Pal, I. Ivanov, A. Datta, M. Bittner and E. Dougherty, Generating Boolean networks with a prescribed attractor structure. Bioinformatics 21 (2005) 4021-4025.

[30] R. Pal, A. Datta, M. Bittner and E. Dougherty, Intervention in context-sensitive probabilistic Boolean networks, Bioinformatics 21 (2005) 1211-1218.

[31] I. Shmulevich, E. Dougherty, S. Kim and W. Zhang, Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks, Bioinformatics 18 (2002) 261-274.

[32] 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 (2002) 431-445.

[33] I. Shmulevich, E. Dougherty, S. Kim and W. Zhang, ¿From Boolean networks to probabilistic Boolean networks as models of genetic regulatory networks, Proceedings of the IEEE 90 (2002) 1778-1792.

[34] I. Shmulevich, E. Dougherty and W. Zhang, Gene perturbation and intervention in probabilistic Boolean networks, Bioinformatics 18 (2002) 1319-1331.

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

[36] A. Shreim, A. Berdahl, V. Sood, P. Grassberger, and M. Paczuski, Complex network analysis of state spaces for random Boolean networks, New J. Phys. 10 (2008) 013028 (17pages).

[37] P. Smolen, D. Baxter and J. Byrne, Mathematical modeling of gene network, Neuron 26 (2000) 567-580.

[38] D. Slezak, Degrees of conditional (in)dependence: A framework for approximate Bayesian networks and examples related to the rough set-based feature selection Information Sciences 179 (3) (2009) 197-209.

[39] Y. Wang and Y. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, Information Sciences 179 (2009) (12), 1944-1959.

[40] S. Zhang, W. Ching, M. Ng and T. Akutsu, Simulation study in probabilistic Boolean network models for genetic regulatory networks, International Journal of Data Mining and Bioinformatics 1 (2007) 217-240.

[41] 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, 2008.