On Construction of Probabilistic Boolean Networks

Wai-Ki CHING
Advanced Modeling and Applied Computing Laboratory
Department of Mathematics
The University of Hong Kong

Abstract

Modeling genetic networks is an important in problem genomic research. Boolean Net- work (BN) and its extension Probabilistic Boolean networks (PBN) have been proposed to model genetic regulatory interactions. In a PBN, its steady-state distribution gives very important information about the long-run behavior of the network. The construction of PBNs from a given transition probability matrix and a given set of BNs is an inverse problem of huge size. We propose a maximum entropy approach for the above problem. Newton's method in conjunction with conjugate gradient method is then applied to solving the inverse problem. We investigate the convergence rate of the proposed method. Numerical examples are also given to demonstrate the effectiveness of our proposed algorithm.

The Outline

(1) Boolean Networks and Probabilistic Boolean Networks.

(2) The Inverse Problem.

(3) The Maximum Entropy Approach.

(4) Numerical Experiments.

(5) Concluding Remarks.

1. Boolean Networks and Probabilistic Boolean Networks.

1.1 Boolean Networks

• In a BN, each **gene** is regarded as a **vertex** of the network and is then quantized into **two levels** only (**expressed: 1 or unexpressed: 0**) though the idea can be extended to the case of more than two levels.

• The **target gene** is predicted by several genes called its **input genes** via a **Boolean function**.

• If the input genes and the corresponding Boolean functions are given, a BN is said to defined and it can be considered as a **deterministic dynamical system**.

• The only **randomness** involved in the network is the **initial system state**.

1.1.1 An Example of a BN of Three Genes

$$v_i(t+1) = f^{(i)}(v_1(t), v_2(t), v_3(t)), \quad i = 1, 2, 3.$$

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

Table 1

$$(0,0,0) \to (0,1,1) \leftrightarrow (0,1,1),$$

$$(1,0,1) \to (1,0,0) \to (0,1,0) \to (1,1,0) \to (1,0,1),$$

$$(0,0,1) \to (1,0,1), \qquad (1,1,1) \to (1,1,0).$$

• The **transition probability matrix** of the 3-gene BN is then given by

$$A_3 = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

• We note that each column has **only one non-zero element** and **column sum is one**.

1.2 A Review on BNs

- A BN $G(V, F)$ actually consists of **a set of vertices**

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

We define $v_i(t)$ to be the state (0 or 1) of the vertex v_i at time t.

- There is also **a list of Boolean functions** ($f_i : \{0,1\}^n \to \{0,1\}$):

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

to represent the **rules of the regulatory interactions** among the genes:

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

where

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

is called the **Gene Activity Profile** (GAP).

• The GAP can take any possible form (states) from the set

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

and thus totally there are 2^n possible states.

• Since BN is a **deterministic model**, to overcome this deterministic rigidity, extension to a probabilistic setting is natural.

• Reasons for a **stochastic model**:
- The biological system has its **stochastic** nature;
- It is likely that regularity of genetic function and interaction known to exist is not due to **logical rules**, but rather to the intrinsic self-organizing stability of the dynamical system;
- The microarray data sets used to infer the network structure are usually not accurate because of the **experimental noise** in the complex measurement process.

1.3 Probabilistic Boolean Networks (PBNs)

• 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**)

$$\boxed{f_j^{(i)}(j = 1, 2, \ldots, l(i))}$$

to be chosen for determining the state of gene v_i.

• The probability of choosing $f_j^{(i)}$ as the **predictor function** is

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

- If we let f_j be the jth possible realization,

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

Thus in an **independent PBN** (the selection of the Boolean function for each gene is independent), the probability of choosing the j-th BN p_j is given by

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

- There are at most

$$\boxed{N = \prod_{i=1}^{n} l(i)} \tag{3}$$

different possible realizations of BNs.

• We note that the **transition process** among the states in the set S in (1) is a **Markov chain process**. Let $\mathbf{a}$ and $\mathbf{b}$ be any two column vectors (binary unit vector) in S. Then the transition probability

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

• By letting $\mathbf{a}$ and $\mathbf{b}$ take all the possible states in S, one can get the transition probability matrix for the process. The transition matrix can also be given by:

$$\boxed{A = p_1 A_1 + p_2 A_2 + \cdots + p_N A_N.}$$

Here A_j is the corresponding transition probability matrix of the j-th BN.

• There are at most $N2^n$ nonzero entries for the transition probability matrix A. This matrix can be very **sparse**.

2. The Inverse Problem

2.1 The Motivation

• We study the problem of constructing a PBN from a given stationary distribution.

• 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.

• This is **an inverse problem** of huge problem size. The inverse problem is **ill-posed**, meaning that there will be many networks or no network having the desirable properties.

• Ching et al. (2008), a modified Conjugate Gradient (CG) method has been proposed to give some possible solutions of PBNs. However, there are **infinitely** many possible PBNs and the algorithm ends up with **different PBNs** with **different initial guesses**.

• The problem can be decomposed into two parts.

• (I) Construct a **sparse** transition probability matrix from a given steady-state probability distribution.

-A mathematical formulation based on entropy rate theory has been proposed for (I) Ching et al. (2009).

• (II) **Construct a PBN based on a given sparse transition probability matrix and a set of BNs**.

• We will focus on this problem here.

2.2 The Formulation

• Suppose that the possible BNs constituting the PBN are known and their BN matrices are denoted by

$$\{A_1, A_2, \ldots, A_N\}.$$

• The sparse transition probability matrix is given and they are related as follows:

$$A = \sum_{i=1}^{N} q_i A_i. \tag{4}$$

• 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 at most 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 at most $m \cdot 2^n$ equations for m^{2^n} unknowns.

• To reconstruct the PBN, one possible way to get q_i is to consider the following minimization problem:

$$\min_{\mathbf{q}} \left\| A - \sum_{i=1}^{m^{2^n}} q_i A_i \right\|_F^2 \tag{5}$$

subject to

$$0 \le q_i \le 1 \quad \text{and} \quad \sum_{i=1}^{m^{2^n}} q_i = 1.$$

• Here $\|\cdot\|_F$ is the **Frobenius norm** of a matrix. Let us define a mapping F from the set of $l \times l$ square matrices to the set of $l^2 \times 1$ vector by

$$F\left(\begin{pmatrix} a_{11} & \cdots & a_{1l} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ a_{l1} & \cdots & a_{ll} \end{pmatrix}\right) = (a_{11}, \ldots, a_{l1}, a_{12}, \ldots, a_{l2}, \ldots, \ldots, a_{1l}, \ldots a_{ll})^T. \tag{6}$$

• If we let

$$U = [F(A_1), F(A_2), \ldots, F(A_{m^{2^n}})] \quad \text{and} \quad \mathbf{p} = F(A) \tag{7}$$

then (5) becomes

$$\min \|U\mathbf{q} - \mathbf{p}\|_2^2 \tag{8}$$

subject to

$$0 \leq q_i \leq 1 \quad \text{and} \quad \sum_{i=1}^{m^{2^n}} q_i = 1.$$

- Since

$$||U\mathbf{q} - \mathbf{p}||_2^2 = (U\mathbf{q} - \mathbf{p})^T(U\mathbf{q} - \mathbf{p}) \tag{9}$$

and

$$(U\mathbf{q} - \mathbf{p})^T(U\mathbf{q} - \mathbf{p}) = \mathbf{q}^T U^T U\mathbf{q} - 2\mathbf{q}^T U^T \mathbf{p} + \mathbf{p}^T \mathbf{p}. \tag{10}$$

- Thus the minimization problem (10) **without constraints** is equivalent to

$$\min_{\mathbf{q}}\{\mathbf{q}^T U^T U\mathbf{q} - 2\mathbf{q}^T U^T \mathbf{p}\}. \tag{11}$$

- The matrix $U^T U$ is a **symmetric positive semi-definite matrix**. The minimization problem without constraints is equivalent to solving

$$U^T U\mathbf{q} = U^T \mathbf{p} \tag{12}$$

with the **Conjugate Gradient (CG) method**.

• We note that if there is $\mathbf{q}$ satisfying the equation $U\mathbf{q} = \mathbf{p}$ with $\mathbf{1}^T\mathbf{q} = 1$ and $\mathbf{0} \leq \mathbf{q} \leq \mathbf{1}$. Then the CG method can yield a solution.

• To ensure that $\mathbf{1}^T\mathbf{q} = 1$, we add a row of $(1, 1, \ldots, 1)$ to the bottom of the matrix U and form a new matrix $\bar{U}$. At the same time, we add an entry 1 at the end of the vector $\mathbf{p}$ to get a new vector $\bar{\mathbf{p}}$. Thus we consider the **revised equation**:

$$\bar{U}^T\bar{U}\mathbf{q} = \bar{U}^T\bar{\mathbf{p}}. \tag{13}$$

• This method can give a solution of the inverse problem. But usually there are **too many solutions**. Extra constraints or criterion have to be introduced in order narrow down the set of solutions or even a unique solution.

3. The Maximum Entropy Approach

• One possible and reasonable approach is to consider the solution which gives the **largest entropy** as $\mathbf{q}$ itself can be considered as a probability distribution.

• This means we are to find $\mathbf{q}$ such that it maximizes

$$-\sum_{i=1}^{m^{2^n}} q_i \log(q_i). \tag{14}$$

• Similar method has been used by Wilson (1970) in traffic demand estimation in a transportation network and it has become more popular (Ching et al. 2004).

• We recall that for the inverse problem, we have $m \cdot 2^n$ equations for m^{2^n} unknowns. Thus one may have infinitely many solutions.

• Since $\mathbf{q}$ can be viewed as a probability distribution, one possible way to get a better choice of q_i is to consider maximizing the entropy of $\mathbf{q}$ subject to the given constraints, i.e., the following maximization problem:

$$\max_{\mathbf{q}} \left\{ \sum_{i=1}^{m^{2^n}} (-q_i \log q_i) \right\} \tag{15}$$

subject to

$$\bar{U}\mathbf{q} = \bar{\mathbf{p}} \quad \text{and} \quad 0 \leq q_i \quad i = 1, 2, \ldots, m^{2^n}.$$

• We remark that the constraints that $q_i \leq 1$ can be discarded as we required that

$$\sum_{i=1}^{m^{2^n}} q_i = 1 \quad \text{and} \quad 0 \leq q_i \quad i = 1, 2, \ldots, m^{2^n}.$$

• The dual problem of (15) is therefore of the type

$$\min_{\mathbf{y}} \max_{\mathbf{q}} L(\mathbf{q}, \mathbf{y}) \tag{16}$$

where $\mathbf{y}$ is the **multiplier** and $L(\cdot, \cdot)$ is the **Lagrangian function**

$$L(\mathbf{q}, \mathbf{y}) = \sum_{i=1}^{m^{2^n}} (-q_i \log q_i) + \mathbf{y}^T(\bar{\mathbf{p}} - \bar{U}\mathbf{q}). \tag{17}$$

• The optimal solution $\mathbf{q}^*(\mathbf{y})$ of the inner maximization problem of (16) solves the equations

$$\nabla_{q_i} L(\mathbf{q}, \mathbf{y}) = -\log q_i - 1 - \mathbf{y}^T \bar{U}_{\cdot i} = 0, \quad i = 1, 2, \ldots, m^{2^n}$$

and is thus of the form:

$$q_i^*(\mathbf{y}) = e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}}, \quad i = 1, 2, \ldots, m^{2^n} \tag{18}$$

where $\bar{U}_{\cdot i}$ is the ith column of the matrix $\bar{U}$.

• After substituting $\mathbf{q}^*(\mathbf{y})$ back into (17) the **dual problem** (16) can be simplified to

$$\min_{\mathbf{y}} \left\{ \sum_{i=1}^{m^{2^n}} e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}} + \mathbf{y}^T \bar{\mathbf{p}} \right\}. \tag{19}$$

• The solution of the primal problem (16) is obtained from the solution of the **dual problem** (18) through (19).

• Thus we have transformed a constrained maximization problem with m^{2^n} variables into an unconstrained minimization problem of $m \cdot 2^n + 1$ variables.

• We will then apply **Newton's method** in conjunction with **Conjugate Gradient (CG) method** to solving the dual problem.

4. Numerical Experiments

4.1 Newton's Method

• In the following, we will explain how Newton's method in conjunction with the conjugate gradient method can be used. To this end we denote by

$$f(\mathbf{y}) = \sum_{i=1}^{m^{2^n}} e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}} + \mathbf{y}^T \bar{\mathbf{p}} \tag{20}$$

the function to be minimized.

• The **gradient** and the **Hessian** of f are respectively of the forms:

$$\nabla f(\mathbf{y}) = -\bar{U}\mathbf{q}^*(\mathbf{y}) + \bar{\mathbf{p}} \tag{21}$$

and

$$\nabla^2 f(\mathbf{y}) = \bar{U} \cdot \mathrm{diag}(\mathbf{q}^*(\mathbf{y})) \cdot \bar{U}^T \tag{22}$$

where $\mathbf{q}^*(\mathbf{y})$ is as defined in (18) and $\mathrm{diag}(\mathbf{q}^*(\mathbf{y}))$ is the diagonal matrix with diagonal entries $(\mathbf{q}^*(\mathbf{y}))$.

Newton's Method

Choose starting point $\mathbf{y}_0 \in Im(\bar{U})$

$k = 1;$

while $||\nabla f(\mathbf{y}_k)||_2 > tolerance$

find $\mathbf{p}_k$ with $\nabla^2 f(\mathbf{y}_{k-1})\mathbf{p}_k = -\nabla f(\mathbf{y}_{k-1})$;

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

$k = k + 1;$

end.

• From (22), we observe that f is **strictly convex** on the subspace $Im(\bar{U})$.

• Newton's method will produce a sequence of points $\mathbf{y}_k$ according to the iteration $\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{p}_k$, where the Newton step $\mathbf{p}_k$ is the solution of the Hessian matrix system:

$$\nabla^2 f(\mathbf{y}_{k-1})\mathbf{p}_k = -\nabla f(\mathbf{y}_{k-1}). \tag{23}$$

• We note that $\nabla^2 f(\mathbf{y}_{k-1})$ is a **one-to-one mapping** of the concerned subspace onto itself.

• Moreover, from (21) $\nabla f(\mathbf{y}) \in Im(\bar{U})$ as $\bar{\mathbf{p}} \in Im(\bar{U})$. Hence, Equation (23) has an unique solution and therefore Newton's method for minimizing f is well defined.

• If we start with $\mathbf{y}_0 \in Im(\bar{U})$ the Newton sequence will remain in the subspace. Moreover, it will converge locally at a **quadratic rate**.

• To enforce global convergence one may wish to resort to line search or trust region techniques. However, we did not find this necessary in our computational experiments.

4.2 Conjugate Gradient Method

• In each iteration of the Newton's method, one has to solve the linear system of the form (23). We propose to solve the linear system (23) by Conjugate Gradient (CG) method.

• The convergence rate of CG method depends on the **effective condition number**

$$\frac{\lambda_1(\nabla^2 f(\mathbf{y}))}{\lambda_s(\nabla^2 f(\mathbf{y}))} \tag{24}$$

of $\nabla^2 f(\mathbf{y})$. Since $\nabla^2 f(\mathbf{y})$ is **singular** we have to consider the second smallest eigenvalue $\lambda_s(\nabla^2 f(\mathbf{y}))$.

Theorem : For the Hessian matrix $\nabla^2 f(\mathbf{y})$, we have

$$2^n \cdot e^{-2(m\cdot 2^n+1)\cdot\|y\|_\infty} \leq \frac{\lambda_1(\nabla^2 f(\mathbf{y}))}{\lambda_s(\nabla^2 f(\mathbf{y}))} \leq \left(\sqrt{2^n}+\sqrt{m}\right)^2 \cdot e^{2(m\cdot 2^n+1)\cdot\|y\|_\infty}.$$

• For Newton's method, we set the tolerance to be 10^{-7} while the tolerance of CG method is 10^{-10}.

Example 1. In the first example, we consider the case $n = 2$ and $m = 2$ and we suppose that the observed/estimated transition probability matrix of the PBN is given as follows:

$$A_{2,2} = \begin{pmatrix} 0.1 & 0.3 & 0.5 & 0.6 \\ 0.0 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.4 \end{pmatrix}. \tag{25}$$

• Then there are 16 possible BNs for constituting the PBN and they are listed below:

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} A_3 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_6 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} A_7 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_8 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_9 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{10} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} A_{11} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{12} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{13} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{14} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} A_{15} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{16} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

- Suppose we have

$$A = \sum_{i=1}^{16} q_i A_i$$

and the followings are the 8 equations governing q_i (cf. (7)):

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}\right)
\left(\begin{array}{c} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \\ q_{15} \\ q_{16} \end{array}\right)
=
\left(\begin{array}{c} 0.1 \\ 0.0 \\ 0.0 \\ 0.9 \\ \hline 0.3 \\ 0.7 \\ 0.0 \\ 0.0 \\ \hline 0.5 \\ 0.0 \\ 0.5 \\ 0.0 \\ \hline 0.6 \\ 0.0 \\ 0.0 \\ 0.4 \end{array}\right).$$

Fig. 1. The Probability Distribution q for the case of $A_{2,2}$.

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

Table 2: The Truth Table for A_{13}.

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

Table 3: The Truth Table for A_{14}.

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

Table 4 : The Truth Table for A_{15}.

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

Table 5 : The Truth Table for A_{16}.

Example 2. We then consider the case $n = 3$ and $m = 2$ and we suppose that the observed transition matrix of the PBN is given as follows:

$$A_{3,2} = \begin{pmatrix} 0.1 & 0.3 & 0.5 & 0.6 & 0.2 & 0.1 & 0.6 & 0.8 \\ 0.0 & 0.7 & 0.0 & 0.0 & 0.8 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.0 \end{pmatrix}.$$

• There are 256 possible BNs for constituting the PBN. The solution is shown in Figure 2. We note that the PBN is dominated (over 60%) by 25 BNs.

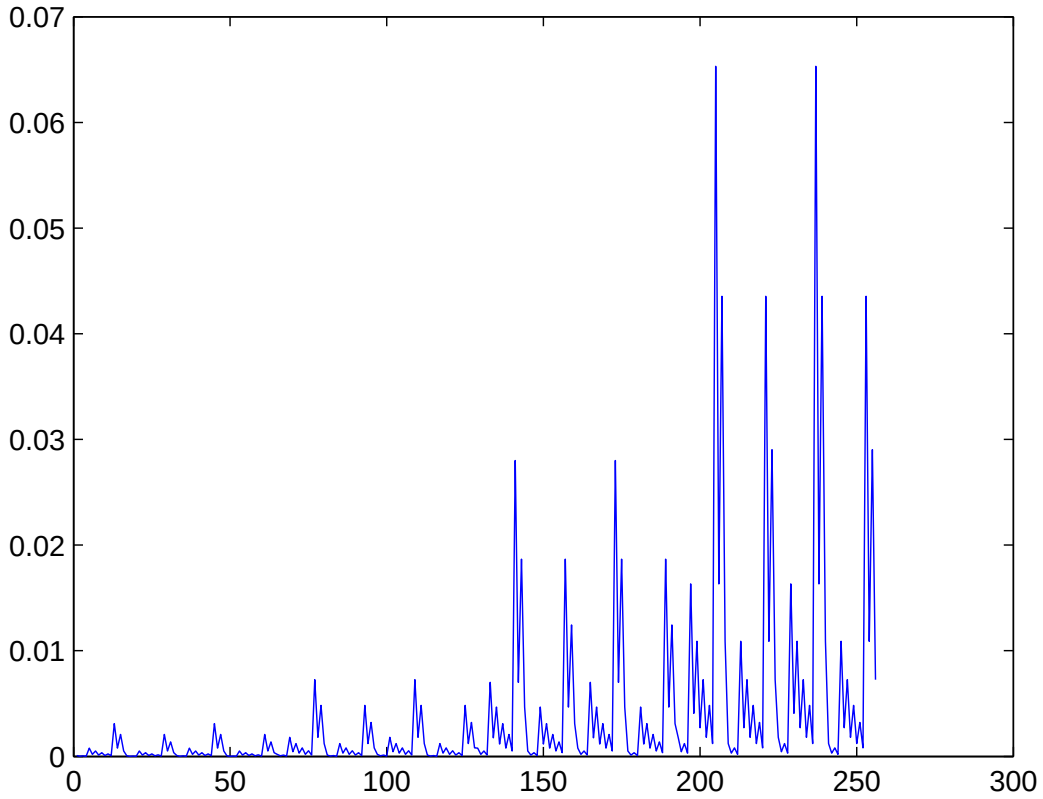

Fig. 2. The Probability Distribution q for the case of $A_{3,2}$.

• Finally, we present the number of Newton's iterations required for convergence and the average number of CG iterations in each Newton's iteration in the following table.

n	m	Number of BNs	Newton's Iterations	Average Number of CG Iterations
2	2	16	9	9
2	3	81	7	9
3	2	256	7	7
3	3	6561	11	13

Table 6 : Number of Iterations.

Example 3: A Three-gene Network with Small Perturbations

• In Shmulevich, et al. (2002), they proposed a PBN consisting of three genes $V = (x_1, x_2, x_3)$ and the function set $F = (F_1, F_2, F_3)$.

• Here $F_1 = \{f_1^{(1)}, f_2^{(1)}\}$, $F_2 = \{f_1^{(2)}\}$, and $F_3 = \{f_1^{(3)}, f_2^{(3)}\}$. The functions are given by the truth table below:

$x_1x_2x_3$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

Table 7 : Truth Table (Taken from Shmulevich et al.)

- The corresponding four BNs are given in the table below.

BN_1	1	7	7	6	3	8	6	8
BN_2	1	7	7	5	3	7	5	8
BN_3	1	7	7	2	3	8	6	8
BN_4	1	7	7	1	3	7	5	8

Table 8 : The Four BNs (The position of the non-zero entry in each column)

- The transition probability matrix is

$$A_{4,4} = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 1.0 & 1.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 1.0 \end{pmatrix}.$$

• We then consider adding some perturbations to the first two rows and the non-zeros entries of the transition probability as follows: $\delta(A_{4,4}) =$

$$\begin{pmatrix}
1.0-\delta & \delta & \delta & 0.2+\delta & \delta & \delta & \delta & \delta \\
\delta & \delta & \delta & 0.2+\delta & \delta & \delta & \delta & \delta \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0-2\delta & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.3-\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 \\
0.0 & 0.0 & 0.0 & 0.3-\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 \\
0.0 & 1.0-2\delta & 1.0-2\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5-\delta & 0.0 & 1.0-2\delta
\end{pmatrix}.$$

• For $\delta = 0.01, 0.02, 0.03$ and 0.04, using our algorithm, we obtain 16 major BNs. These BNs actually contribute around respectively 94%, 90%, 84% and 79% of the network. Moreover the four BNs (BN_1, BN_2, BN_3, BN_4) are included.

5. Concluding Remarks.

• We present the problem of constructing a PBN from a given transition probability matrix and a given set of BNs. It is an inverse problem of huge size.

• We propose a maximum entropy approach for solving the problem. Newton's method is then applied to solving the inverse problem with CG method for solving the Hessian matrix system. We also give a convergence rate analysis for the proposed method.

• The computational cost can be reduced significantly if further information about the set of possible BNs is given.

References

• Akutsu, T. et al. (2000) Inferring qualitative relations in genetic networks and metabolic arrays, *Bioinformatics*, **16**, 727-734.

• Akutsu, T. et al. (2007b) Control of Boolean networks: hardness results and algorithms for tree structured networks, *Journal of Theoretical Biology*, **244**, 670-79.

• Ching, W., Scholtes, S. and Zhang, S. (2004), Numerical Algorithms for Estimating Traffic Between Zones in a Network, Engineering Optimisation, **36**, 379-400.

• Ching, W. et al. (2005) On construction of stochastic genetic networks based on gene expression sequences, *International Journal of Neural Systems*, **15**, 297-310.

• Ching, W. et al. (2007) On Multi-dimensional Markov Chain Models, Pacific Journal of Optimization, **3** 235-243.

• Ching, W. and Ng, M. (2006) Markov Chains : models, algorithms and applications. International Series on Operations Research and Management Science, Springer, New York.

• Ching, W. et al. (2007a) An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks, *Bioinformatics*, **23** 1511-1518.

• Ching, W. et al. (2007b) Optimal finite-horizon control for probabilistic Boolean networks with hard constraints, *Proceedings of the International Symposium on Optimization and Systems Biology*.

• Ching, W., Zhang, S., Jiao, Y., Akutsu, T., Tsing, N. and Wong, A. (2009), Optimal Control Policy for Probabilistic Boolean Networks with Hard Constraints, IET on Systems Biology, **3** 90-99.

• Ching, W., Chen, X. and Tsing, N. (2009) Generating Probabilistic Boolean Networks from a Prescribed Transition Probability Matrix, IET on Systems Biology, **6** 453-464.

• Datta, A. et al. (2003) External control in Markovian genetic regulatory networks, *Machine Learning*, **52**, 169-91.

• Dougherty, E. R. et al. (2000) Coefficient of determination in nonlinear signal processing. *Signal Processing*, **80**, 2219-35.

• Faryabi, B. et al. (2008) On approximate stochastic control in genetic regulatory networks, *IET Systems Biology*, **6**, 361-368.

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

• De Jong, H. (2002) Modeling and simulation of genetic regulatory systems: A literature review, *J. Comp. Biol.*, **9**, 67-103.

• Kauffman, S. A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *Theoretical Biology*, **22**, 437-467.

• Kauffman, S. A. The origins of order: Self-organization and selection in evolution. New York: Oxford Univ. Press, 1993.

• Keller, A.D. (1994) Specifying Epigenetic States with Autoregulatory Transcription Factors. Journal of Theoretical Biology, **153**, 181-194.

• Ng, M. and Ching W. et al. (2006) A control model for Markovian genetic regulatory network. *Transactions on Computational Systems Biology*, Springer, **4070**, 36-48.

• Pal, R. et al. (2005) Intervention in context-sensitive probabilistic Boolean networks, *Bioinformatics*, **21 (7)**, 1211-1218.

• Pal, R. et al. (2006) Optimal Infinite-Horizon Control for Probabilistic Boolean networks. *IEEE Tran. Signal Processing*, **54** (6), 2375-2387.

• Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, **77**, 257-286.

• Shmulevich, I. et al. (2002a) Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks, *Bioinformatics*, **18**, 261-274.

• Shmulevich, I. et al. (2002b) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, *Proceedings of the IEEE*, **90**, 1778-1792.

• Shmulevich, I. et al. (2002c) Gene perturbation and intervention in probabilistic Boolean networks, *Bioinformatics*, **18**, 1319-1331.

• Shmulevich, I. et al. (2002d) Control of stationary behavior in probabilistic Boolean networks by means of structural intervention, *Biological Systems*, **10**, 431-46.

• Shmulevich, I. et al (2003) Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks, *Comparative and Functional Genomics*, **4**, 601-608.

• Smolen, P. et al. (2000) Mathematical modeling of gene network, *Neuron*, **26**, 567-580.

• Steggles, L. J. et al. (2007) Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach, *Bioinformatics*, **23**, 336-343.

• Thieffry, D. et al. (1998) From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in Escherichia coli, *BioEssays*, **20**(5), 433-40.

• Wilson, A. (1970), Entropy in Urban and Regional Modelling, Pion, London.

• Yeung, K. and Ruzzo, W. (2001) An Empirical Study on Principal Component Analysis for Clustering Gene Expression Data, Bioinformatics **17**, 763-774.

• Zhang, S. and Ching, W. et al. (2007a) Simulation study in probabilistic Boolean network models for genetic regulatory networks, *Journal of Data Mining and Bioinformatics*, **1**, 217-40.

• Zhang, S. and Ching, W. et al. (2007b) Algorithms for finding small attractors in Boolean networks, *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 20180.

• Zhang, S., Ching, W., Tsing, N., Leung, H. and Guo, D. (2010) A New Multiple Regression Approach for the Construction of Genetic Regulatory Networks, to appear in Journal of Artificial Intelligence in Medicine.