Abstract

Modeling genetic regulatory networks is an important problem in genomic research. Boolean Networks (BNs) and their extensions Probabilistic Boolean Networks (PBNs) have been proposed for modeling genetic regulatory interactions. In a PBN, its steady-state distribution gives very important information about the long-run behavior of the whole network. However, one is also interested in system synthesis which requires the construction of networks. The inverse problem is ill-posed and challenging, because there may be many networks or no network having the given properties, and the size of the problem is huge. The construction of PBNs from a given transition-probability matrix and a given set of BNs is an inverse problem of huge size. Here we propose a maximum entropy approach for the above problem. Newton's method in conjunction with the Conjugate Gradient (CG) 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 method.
The Bioinformatic project is a joint work with

- Prof. Tatsuya AKUTSU, Kyoto University.
- Prof. Kiyoko KINOSHITA, Soka University.
- Prof. Namkiu TSING, HKU.
- Prof. Xiaobo ZHOU, Cornell University.

and my group of students

- Dr. Shuqin ZHANG, Fudan University and Yale University.
- Dr. Limin LI, Xian Jiaotong University and Max Planck Institute.
- Yang CONG, Xi CHEN and Hao JIANG, HKU.
The Outline

(0) Motivations and Objectives.

(1) Boolean Networks and Probabilistic Boolean Networks.

(2) The Inverse Problem.

(3) The Maximum Entropy Approach.

(4) Numerical Experiments.
0. Motivations and Objectives.

- An important issue in **systems biology** is to model and understand the **mechanism** in which the cells execute and control a large number of operations for their **normal functions** and also the way in which they fail in **diseases** such as cancer (25000 genes in human Genome). Eventually to design some **control strategy** to avoid the **undesirable** state/situation.

- Mathematical models (A review by De Jong 2002):
  - **Boolean networks (BNs)** (Kaufman 1969)
  - Differential equations (Keller 1994)
  - **Probabilistic Boolean networks (PBNs)** (Shmulevich et al. 2002)
  - **Multivariate Markov chain model** (Ching et al. 2005)
  - Petri nets (Steggles et al. 2007) etc.

- Since genes exhibit “**switching behavior**”, **BNs and PBNs** models have received much attention.
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

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

<table>
<thead>
<tr>
<th>State</th>
<th>(v_1(t))</th>
<th>(v_2(t))</th>
<th>(v_3(t))</th>
<th>(f^{(1)})</th>
<th>(f^{(2)})</th>
<th>(f^{(3)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1

\[(0, 0, 0) \rightarrow (0, 1, 1) \leftrightarrow (0, 1, 1),\]

\[(1, 0, 1) \rightarrow (1, 0, 0) \rightarrow (0, 1, 0) \rightarrow (1, 1, 0) \rightarrow (1, 0, 1),\]

\[(0, 0, 1) \rightarrow (1, 0, 1), \quad (1, 1, 1) \rightarrow (1, 1, 0).\]
• The \textbf{transition probability matrix} of the 3-gene BN is then given by

\[
A_3 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.
\]

• We note that each column has \textbf{only one non-zero element} and \textbf{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 $(f_i : \{0, 1\}^n \rightarrow \{0, 1\})$:
  \[ F = \{f_1, f_2, \ldots, f_n\} \]
  to represent the rules of the regulatory interactions among the genes:
  \[ v_i(t + 1) = f_i(v(t)), \quad i = 1, 2, \ldots, n \]
  where
  \[ v(t) = (v_1(t), v_2(t), \ldots, v_n(t))^T \]
  is called the Gene Activity Profile (GAP).
• The GAP can take any possible form (states) from the set

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

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)

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

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

$$c_j^{(i)}, 0 \leq c_j^{(i)} \leq 1$$

and

$$\sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad \text{for} \quad i = 1, 2, \ldots, n.$$
• If we let $f_j$ be the $j$th possible realization,

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

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

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

• There are at most

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

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 $a$ and $b$ be any two column vectors (binary unit vector) in $S$. Then the transition probability

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

• By letting $a$ and $b$ 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:

$$
A = p_1A_1 + p_2A_2 + \cdots + p_NA_N.
$$

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 $A$. 

12
2. The Inverse Problem

2.1 The Motivation

• We study the problem of constructing a PBN from a given steady-state 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) Cong and 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

\[ \{A_1, A_2, \ldots, A_N\} \].

• Transition probability matrix is observed and they are related as follows:

\[ A = \sum_{i=1}^{N} q_i A_i. \]  

(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 $N = m^{2n}$ and we can order $A_1, A_2, \cdots, A_{m^{2n}}$ 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 $m^{2n}$ unknowns.

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

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

subject to

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

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

(6)

• If we let

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

(7)

then (5) becomes

$$\min \|Uq - p\|_2^2$$  

subject to

$$0 \leq q_i \leq 1 \quad \text{and} \quad \sum_{i=1}^{m^{2n}} q_i = 1.$$
• Since
\[ \|Uq - p\|_2^2 = (Uq - p)^T(Uq - p) \] (9)
and
\[ (Uq - p)^T(Uq - p) = q^T U^T U q - 2q^T U^T p + p^T p. \] (10)

• Thus the minimization problem (10) without constraints is equivalent to
\[
\min_q \{ q^T U^T U q - 2q^T U^T p \}. \] (11)

• The matrix \( U^T U \) is a symmetric positive semi-definite matrix. The minimization problem without constraints is equivalent to solving
\[ U^T U q = U^T p \] (12)
with the Conjugate Gradient (CG) method.
• We note that if there is \( q \) satisfying the equation \( Uq = p \) with \( 1^T q = 1 \) and \( 0 \leq q \leq 1 \). Then the CG method can yield a solution.

• To ensure that \( 1^T q = 1 \), 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 \( p \) to get a new vector \( \bar{p} \). Thus we consider the revised equation:

\[
\bar{U}^T \bar{U} q = \bar{U}^T \bar{p}.
\] (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 $q$ itself can be considered as a probability distribution.

• This means we are to find $q$ such that it maximizes

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

(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 $m^{2n}$ unknowns. Thus one may have infinitely many solutions.

• Since $q$ 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 $q$ subject to the given constraints, i.e., the following maximization problem:

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

subject to

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

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

$$\sum_{i=1}^{m^{2n}} q_i = 1 \quad \text{and} \quad 0 \leq q_i \quad i = 1, 2, \ldots, m^{2n}.$$
The dual problem of (15) is therefore of the type

$$\min_y \max_q L(q, y)$$

(16)

where $y$ is the multiplier and $L(\cdot, \cdot)$ is the Lagrangian function

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

(17)

The optimal solution $q^*(y)$ of the inner maximization problem of (16) solves the equations

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

and is thus of the form:

$$q^*_i(y) = e^{-1-y^T \bar{U}_i}, \quad i = 1, 2, \ldots, m^{2n}$$

(18)

where $\bar{U}_i$ is the $i$th column of the matrix $\bar{U}$. 
After substituting $q^*(y)$ back into (17) the **dual problem** (16) can be simplified to

$$\min_y \left\{ \sum_{i=1}^{m^2} e^{-1}y^T\bar{U}_i + y^T\bar{p} \right\}.$$  \hspace{1cm} (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 $m^{2n}$ 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

\[
 f(y) = \sum_{i=1}^{m^2 n} e^{-1} y^T \bar{U}_i + y^T \bar{p}
\]  

(20)

the function to be minimized.

• The gradient and the Hessian of \( f \) are respectively of the forms:

\[
 \nabla f(y) = -\bar{U} q^*(y) + \bar{p}
\]  

(21)

and

\[
 \nabla^2 f(y) = \bar{U} \cdot \text{diag}(q^*(y)) \cdot \bar{U}^T
\]  

(22)

where \( q^*(y) \) is as defined in (18) and \( \text{diag}(q^*(y)) \) is the diagonal matrix with diagonal entries \( (q^*(y)) \).
Newton’s Method

Choose starting point $y_0 \in Im(\bar{U})$

$$k = 1;$$

while $||\nabla f(y_k)||_2 > \text{tolerance}$

- find $p_k$ with $\nabla^2 f(y_{k-1})p_k = -\nabla f(y_{k-1});$
- set $y_k = y_{k-1} + p_k;$
- $k = k + 1;$

end.

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

- Newton’s method will produce a sequence of points $y_k$ according to the iteration $y_k = y_{k-1} + p_k,$ where the Newton step $p_k$ is the solution of the Hessian matrix system:

$$\nabla^2 f(y_{k-1})p_k = -\nabla f(y_{k-1}).$$  (23)
• We note that $\nabla^2 f(y_{k-1})$ is a one-to-one mapping of the concerned subspace onto itself.

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

• If we start with $y_0 \in \text{Im}(\bar{U})$ 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

\[ \frac{\lambda_1(\nabla^2 f(y))}{\lambda_s(\nabla^2 f(y))} \]  

(24)

of \( \nabla^2 f(y) \). Since \( \nabla^2 f(y) \) is singular we have to consider the second smallest eigenvalue \( \lambda_s(\nabla^2 f(y)) \).
**Theorem**: For the Hessian matrix $\nabla^2 f(y)$, we have

$$2^n \cdot e^{-2(m \cdot 2^n + 1) \cdot \|y\|_\infty} \leq \frac{\lambda_1(\nabla^2 f(y))}{\lambda_8(\nabla^2 f(y))} \leq (\sqrt{2^n} + \sqrt{m})^2 \cdot e^{2(m \cdot 2^n + 1) \cdot \|y\|_\infty}.$$ 

- 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:

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

(25)
• Then there are 16 possible BNs for constituting the PBN and they are listed below:

\[
\begin{align*}
A_1 &= \begin{pmatrix}
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
A_2 &= \begin{pmatrix}
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
A_3 &= \begin{pmatrix}
1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
A_4 &= \begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \\
A_5 &= \begin{pmatrix}
1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
A_6 &= \begin{pmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \\
A_7 &= \begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
A_8 &= \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \\
A_9 &= \begin{pmatrix}
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \\
A_{10} &= \begin{pmatrix}
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1
\end{pmatrix} \\
A_{11} &= \begin{pmatrix}
0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \\
A_{12} &= \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{pmatrix} \\
A_{13} &= \begin{pmatrix}
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \\
A_{14} &= \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1
\end{pmatrix} \\
A_{15} &= \begin{pmatrix}
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \\
A_{16} &= \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{pmatrix}.
\end{align*}
\]
• Suppose we have

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

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

\[
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
q_1 \\
q_2 \\
q_3 \\
q_4 \\
q_5 \\
q_6 \\
q_7 \\
q_8 \\
q_9 \\
q_{10} \\
q_{11} \\
q_{12} \\
q_{13} \\
q_{14} \\
q_{15} \\
q_{16}
\end{pmatrix} = \begin{pmatrix}
0.1 \\
0.0 \\
0.0 \\
0.9 \\
0.3 \\
0.7 \\
0.0 \\
0.0 \\
0.5 \\
0.0 \\
0.5 \\
0.0 \\
0.6 \\
0.0 \\
0.4
\end{pmatrix}.
\]
Fig. 1. The Probability Distribution $q$ for the case of $A_{2,2}$. 
<table>
<thead>
<tr>
<th>State</th>
<th>$v_1(t)$</th>
<th>$v_2(t)$</th>
<th>$f^{(1)}$</th>
<th>$f^{(2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>State</th>
<th>$v_1(t)$</th>
<th>$v_2(t)$</th>
<th>$f^{(1)}$</th>
<th>$f^{(2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>State</th>
<th>$v_1(t)$</th>
<th>$v_2(t)$</th>
<th>$f^{(1)}$</th>
<th>$f^{(2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>State</th>
<th>$v_1(t)$</th>
<th>$v_2(t)$</th>
<th>$f^{(1)}$</th>
<th>$f^{(2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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:

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

- There are 256 possible BNs for constituting the PBN. The solution is shown in Figure 5.2. 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.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$m$</th>
<th>Number of BNs</th>
<th>Newton’s Iterations</th>
<th>Average Number of CG Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>16</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>81</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>256</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6561</td>
<td>11</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 6: Number of Iterations.
References


• 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.


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


