

# On Construction of Sparse Probabilistic Boolean Networks from a Prescribed Transition Probability Matrix

Lu-Bin Cui<sup>1</sup>

Wen Li<sup>1</sup>

Wai-Ki Ching<sup>2</sup>

<sup>1</sup>School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China.  
Emails: hnzkc@163.com, liwen@scnu.edu.cn

<sup>2</sup>Advanced Modeling and Applied Computing Laboratory, Department of Mathematics,  
The University of Hong Kong, Hong Kong.  
Email: wching@hkusua.hku.hk

**Abstract** Probabilistic Boolean Networks (PBNs) are useful models for modeling genetic regulatory networks. In this paper, we propose efficient algorithms for constructing a sparse probabilistic Boolean network when its transition probability matrix and a set of possible Boolean networks are given. This is an interesting inverse problem in network inference and it is important in the sense that most microarray data sets are assumed to be obtained from sampling the steady-state.

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

## 1 Introduction

There are many formalisms and mathematical models proposed in the literature to study genetic regulatory networks such as Boolean networks (BNs) [4], regression model [9], Probabilistic Boolean Networks (PBNs) [5]. Among these models, BN and its extension PBN have received much attention. BN was first introduced by Kauffman [4]. In a BN, the gene expression states are quantized to only two levels: on and off (represented as 1 and 0). The target gene is predicted by several genes called its input genes via a Boolean function. When the input genes and the Boolean functions are given, then we say that a BN is defined. We remark that a BN is a deterministic model. Due to the facts that genetic regulation process exhibits uncertainty property and microarray data sets have errors due to experimental noise in the complex measurement processes, BNs have been extended to PBNs (stochastic models). In a PBN, for each gene, there can be more than one Boolean function and selection probabilities are assigned to the Boolean functions. The network dynamics of a PBN can be studied in a Markov chains framework [5]. In a PBN, the network behavior is characterized by its transition probability matrix. A matrix approximation method was proposed in [1] to get an approximation of the steady-state probability distribution efficiently. One can understand a genetic network and identify the influence of different genes via such a network [2]. Here we study an inverse problem in network inference from steady-state data. The problem is important in the sense that most

microarray data sets are assumed to be obtained from sampling the steady-state [3]. An entropy approach for this problem has been proposed in [3] and here we focus on getting sparse solution. We propose efficient algorithms for constructing a sparse PBN when its transition probability matrix and a set of possible BNs are given.

The remainder of the paper is organized as follows. Section 2 presents the inverse problem. In Section 3 we give efficient algorithms for constructing a sparse PBN. Numerical examples are also given to demonstrate the proposed algorithms in Section 4. Finally concluding remarks are given in Section 5.

## 2 The Inverse Problem of Construction of PBNs

Suppose we are given the transition probability matrix  $A$  of the PBN with the possible BNs constituting the PBN being given by  $\{A_1, A_2, \dots, A_N\}$ . That is to say

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

Here  $A_j$  is the corresponding transition probability matrix (Boolean matrix) of the  $j$ th BN and  $q_j$  is the probability of choosing the  $j$ th BN. Thus the larger the value of  $q_j$  is, the more important the corresponding BN will be. We are to estimate  $\mathbf{q} = (q_1, q_2, \dots, q_N)^T$  which is a probability distribution and it is also the weightings of the corresponding Boolean networks. The inverse problem here is to get the parameters  $q_j$ ,  $j = 1, 2, \dots, N$  and we require  $\mathbf{q}$  to be sparse so that we can get those dominant BNs. Now we let the matrix  $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N]$  where  $\mathbf{v}_j = A_j \mathbf{p}$ ,  $\mathbf{p}$  is the stationary distribution of the PBN, the steady-state behavior of the PBN. Then

$$A\mathbf{p} = \left( \sum_{j=1}^N q_j A_j \right) \mathbf{p} = \sum_{j=1}^N q_j (A_j \mathbf{p}) = \sum_{j=1}^N q_j \mathbf{v}_j = V\mathbf{q} = \mathbf{p}$$

Then one possible way to get  $q_j$  is to consider the following non-linear programming problem:

$$h(\mathbf{q}^*) = \min_{\mathbf{q}} \{ \|V\mathbf{q} - \mathbf{p}\|_2^2 + \lambda \|\mathbf{q}\|_\alpha^\alpha \} \quad (1)$$

subject to

$$\sum_{j=1}^N q_j = 1 \quad \text{and} \quad 0 \leq q_j \leq 1, j = 1, 2, \dots, N, \quad (2)$$

where  $\|\mathbf{q}\|_\alpha^\alpha = \sum_{j=1}^N |q_j|^\alpha$  and  $0 \leq \lambda \leq 1, 0 \leq \alpha \leq 1$ .

In the non-linear programming problems (1)-(2) there are two parameters  $\lambda$  and  $\alpha$ , where  $0 \leq \lambda \leq 1, 0 \leq \alpha \leq 1$ . Actually, there are two objectives in the non-linear programming problem (1)-(2). One is recovering the PBN, i.e.,  $\min\{\|V\mathbf{q} - \mathbf{p}\|_2^2\}$  and the other one is finding a sparse solution.

The motivation of the above formulation comes from [6, 7]. Xu et al. [6, 7] studied the  $L_{1/2}$  a regularization problem for variable selection and sparse reconstruction. Their problem takes the following form:  $\min\{\|H\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_\alpha^\alpha\}$ . They note when  $\alpha = 0$  the result obtained is the most sparse and  $\alpha = 1$  is less sparse than  $\alpha = 0$ . We note that

the main differences between our inverse problem and their problem is that we have the constraint (2). Moreover, when  $\alpha = 1$ , since  $\sum_{i=1}^N q_i = 1$  and  $q_i \geq 0$ , the second term of the objective function has no effect and we expect not to get good sparse solution. From our numerical examples in subsection 4.2, we notice that when  $\alpha$  is about 0.97 and  $\lambda$  is about 0.02, the result is better, because we recover the dominant BNs of PBN. But if  $\alpha$  is too small or  $\lambda$  is too large, the solution obtained from Lingo may have only one nonzero element, in this case there is only one BN and the BN cannot recover PBN and may be not the dominant BN of the PBN.

### 3 The algorithm

The non-linear programming problem (1)-(2) can be solved by feasible direction method using Lingo. The examples in this note were solved by Lingo 9 [8]. We denote the solution of the non-linear programming problem (1)-(2) by  $\tilde{\mathbf{q}} = (\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_N)$ . Similar to those in [6, 7], for variable selection, we only select the nonzero  $\tilde{q}_i$  corresponding BNs as the dominant BNs of the PBN, but the real  $q_i$  may be not  $\tilde{q}_i$ . For example, for some  $j \in \{1, 2, \dots, N\}$  the matrix  $B = A - \tilde{q}_j A_j$  may have negative elements but this is impossible in practice. In order to get an approximate real  $\mathbf{q}$  we should modify  $\tilde{\mathbf{q}}$ . We use the following dominant modified algorithm (DMA) to modify  $\tilde{\mathbf{q}}$ .

#### Algorithm (Dominant Modified Algorithm)

```

Let  $\bar{\mathbf{q}} = \tilde{\mathbf{q}}$ ;
qq=find( $\tilde{\mathbf{q}}$ );% find(x) is a function to get the index of the nonzero elements in x,
L=length(qq);
for  $i = 1 : L$ 
     $j = qq(i)$ ;  $B = A - \tilde{q}_j * A_j$ ;
    if( some elements of  $B$  are negative)
        let  $\gamma = \max\{| \text{negative elements of } B|\}$ ;
         $B = B + \gamma * A_j$ ;  $\bar{q}_j = \tilde{q}_j - \gamma$ ;
    else
         $B = B - A_j$ ;  $\bar{q}_j = \tilde{q}_j + 1$ ;
        if( some elements of  $B$  are negative)
            let  $\beta = \max\{| \text{negative elements of } B|\}$ ;
             $B = B + \beta * A_j$ ;  $\bar{q}_j = \tilde{q}_j - \beta$ ;
        endif
    endif
endfor
output  $\bar{\mathbf{q}}$ 

```

Here we demonstrate DMA algorithm by a simple example. Suppose the transition probability matrix of a PBN and the possible BNs constituting the PBN are given respectively by

$$A = \begin{pmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

One possibility is the following. If  $\tilde{q}_1 = 0.45$ , then

$$B = A - 0.45A_1 = \begin{pmatrix} -0.15 & 0.6 \\ 0.7 & -0.05 \end{pmatrix},$$

because matrix  $B$  has negative elements, so  $\gamma = \max\{|-0.15|, |-0.05|\} = 0.15$  and we update  $B$  by

$$B + \gamma A_1 = \begin{pmatrix} 0 & 0.6 \\ 0.7 & 0.1 \end{pmatrix}.$$

Thus we get  $\bar{q}_1 = 0.45 - \gamma = 0.3$ .

The purpose of the else part of the first if-clause in DMA is to modify the nonzero elements of  $\bar{q}$  such that it becomes as large as possible, since we hope that the nonzero  $\tilde{q}_i$  corresponding BNs is the dominant BNs of the PBN. The following example illustrates our purpose. If  $\tilde{q}_1 = 0.25$ , then

$$B = A - 0.25A_1 = \begin{pmatrix} 0.05 & 0.6 \\ 0.7 & 0.15 \end{pmatrix}.$$

Clearly  $B$  are nonnegative, then by the DMA algorithm we update  $B$  as follow:

$$B - A_1 = \begin{pmatrix} -0.95 & 0.6 \\ 0.7 & -0.85 \end{pmatrix},$$

and  $\bar{q}_1 = 0.25 + 1 = 1.25$ . Then  $\beta = \max\{|-0.95|, |-0.85|\} = 0.95$ , we update  $B$  again

$$B + 0.95A_1 = \begin{pmatrix} 0 & 0.6 \\ 0.7 & 0.1 \end{pmatrix},$$

so we get  $\bar{q}_1 = 1.25 - \beta = 0.3 > 0.25 = \tilde{q}_1$  and  $\bar{q}_1$  can't be larger, because the  $(1, 1)$  element of  $B$  is zero.

## 4 Numerical Examples

In this section we demonstrate our proposed algorithm with some numerical examples taken from [10].

### 4.1 Example 1

In the first example, we consider a PBN with three genes  $n = 3$ . The Boolean matrices  $A_1, A_2, A_3$  and  $A_4$  are given as follow:

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

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

$\lambda$	$\alpha$	$\tilde{q}_1$	$\tilde{q}_2$	$\tilde{q}_3$	$\tilde{q}_4$	$\lambda$	$\alpha$	$\tilde{q}_1$	$\tilde{q}_2$	$\tilde{q}_3$	$\tilde{q}_4$
0.5	0.5	0	0	0	1	0.3	0.9	0	0	0.43	0.57
0.5	0.6	0	0	0	1	0.1	0.7	0	0	0.43	0.57
0.5	0.7	0	0	0	1	0.1	0.8	0	0	0.44	0.56
0.5	0.8	0	0	0	1	0.05	0.5	0	0	0.44	0.56
0.5	0.9	0	0	0	1	0.05	0.8	0.15	0	0.48	0.37
0.5	0.95	0	0	0.43	0.57	0.05	0.9	0.18	0	0.49	0.33

Table 1: Solutions for Different  $\alpha$  and  $\lambda$ .

$\lambda$	$\alpha$	$\bar{q}_1$	$\bar{q}_2$	$\bar{q}_3$	$\bar{q}_4$
0.5	0.9	0	0	0	0.3
0.1	0.8	0	0	0.45	0.3
0.05	0.8	0.15	0	0.45	0.3
0.05	0.9	0.15	0	0.45	0.3

Table 2: New Solutions by Dominant Modified Algorithm.

Suppose the true  $\mathbf{q} = (0.15, 0.10, 0.45, 0.30)^T$ , then the transition probability matrix  $A$  of the PBN is given by

$$A = \begin{pmatrix} 0.4000 & 0 & 0.4500 & 0 & 0 & 0 & 0.1500 & 0 \\ 0.6000 & 0 & 0.3000 & 0 & 0 & 0 & 0.1000 & 0 \\ 0 & 0.3000 & 0 & 0 & 0.2500 & 0.1500 & 0 & 0 \\ 0 & 0.4500 & 0 & 0 & 0 & 0.1000 & 0 & 0 \\ 0 & 0 & 0.1500 & 0 & 0 & 0 & 0.4500 & 0.40 \\ 0 & 0 & 0.1000 & 0 & 0 & 0 & 0.3000 & 0.60 \\ 0 & 0.1000 & 0 & 0.4000 & 0.7500 & 0.4500 & 0 & 0 \\ 0 & 0.1500 & 0 & 0.6000 & 0 & 0.3000 & 0 & 0 \end{pmatrix},$$

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

$$\mathbf{p} = (0.3318, 0.3318, 0.2531, 0.1849, 0.4006, 0.3563, 0.5679, 0.2676)^T.$$

We may apply the feasible direction method using Lingo and DMA algorithm to find  $q_i$  pretending that  $q_i$  are not known. We use Lingo to solve the non-linear programming when  $\lambda$  and  $\alpha$  are different and the solutions are given in Table 1.

From the left of Table 1, we know when  $\lambda = 0.5$  and  $0.5 \leq \alpha \leq 0.9$ , there is only one nonzero in the solution and  $A_4$  is the dominated BN. When  $\lambda = 0.5$  and  $\alpha = 0.95$ , we get two dominated BNs, and the result is better. Compare the right of Table 1 with its left, we see that if  $\lambda$  is smaller  $\alpha$  may have more value to choose. Then we use the DMA algorithm to modify  $\tilde{\mathbf{q}}$ . The new solutions obtained  $\bar{\mathbf{q}}$  are given in Table 2. We note that when  $\alpha = 0.1$ ,  $\lambda = 0.8$  the dominated BNs  $A_3$  (45%),  $A_4$  (30%) matches with  $A_3$  ( $\bar{q}_3 = 45\%$ ) and  $A_4$  ( $\bar{q}_4 = 30\%$ ).

$\lambda$	$\alpha$	$m$	$\tilde{q}_1$	$\tilde{q}_2$	$\tilde{q}_3$	$\tilde{q}_4$	$\tilde{q}_5$	$\tilde{q}_6$	$\tilde{q}_7$	$\tilde{q}_8$
0.10	0.90	1	0	0	1	0	0	0	0	0
0.10	0.92	2	0.445	0	0	0	0.555	0	0	0
0.10	0.95	3	0.200	0	0.411	0	0.389	0	0	0
0.05	0.70	1	0	0	1	0	0	0	0	0
0.05	0.90	2	0.446	0	0	0	0.554	0	0	0
0.05	0.92	3	0.200	0	0.411	0	0.389	0	0	0
0.05	0.95	3	0.200	0	0.411	0	0.389	0	0	0
0.05	0.98	4	0	0	0.160	0.382	0.382	0.076	0	0
0.02	0.90	3	0.201	0	0.411	0	0.388	0	0	0
0.02	0.98	6	0.080	0.219	0.081	0.233	0.290	0	0	0.097
0.01	0.95	4	0	0	0.161	0.381	0.382	0.076	0	0
0.01	0.98	6	0.080	0.218	0.082	0.231	0.290	0	0	0.098

Table 3: Solutions for Different  $\alpha$  and  $\lambda$ .

### 4.2 Example 2

In Example 1  $N = 4$ , in this example we will consider the situation  $N = 8$ . Those  $A_1, A_2, A_3$  and  $A_4$  are the same as in Example 1. While  $A_5, A_6, A_7, A_8$  are given as follow:

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

$$A_7 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_8 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Suppose  $\mathbf{q} = (0.0709, 0.1789, 0.0541, 0.3253, 0.2965, 0.0363, 0.0297, 0.0082)^T$  is a randomly generated probability vector. The transition probability matrix of the PBN is

$$A = \begin{pmatrix} 0.5042 & 0.0082 & 0.0838 & 0 & 0 & 0.2965 & 0.0709 & 0 \\ 0.1250 & 0.0297 & 0.3335 & 0 & 0 & 0.0363 & 0.1789 & 0 \\ 0.0445 & 0.3253 & 0 & 0 & 0.5826 & 0.0709 & 0.2965 & 0 \\ 0.3262 & 0.0541 & 0 & 0 & 0 & 0.1789 & 0.0363 & 0 \\ 0 & 0.0363 & 0.3674 & 0.0445 & 0 & 0.0297 & 0.0541 & 0.5487 \\ 0 & 0.2965 & 0.2152 & 0.3262 & 0 & 0.0082 & 0.3253 & 0.4512 \\ 0 & 0.1789 & 0 & 0.5042 & 0.4173 & 0.0541 & 0.0297 & 0 \\ 0 & 0.0709 & 0 & 0.1250 & 0 & 0.3253 & 0.0082 & 0 \end{pmatrix}.$$

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

$$\mathbf{p} = (0.4107, 0.2862, 0.4403, 0.2463, 0.3297, 0.4708, 0.3488, 0.2071)^T.$$

The results of the non-linear programming when  $\lambda$  and  $\alpha$  are different which solved by Lingo are given in Table 3 (where  $m$  is the number of nonzero element of  $\tilde{\mathbf{q}}$ ). Then we use DMA algorithm to modify  $\tilde{\mathbf{q}}$ . The solutions  $\bar{\mathbf{q}}$  obtained are given in Table 4.

$\lambda$	$\alpha$	$m$	$\bar{q}_1$	$\bar{q}_2$	$\bar{q}_3$	$\bar{q}_4$	$\bar{q}_5$	$\bar{q}_6$	$\bar{q}_7$	$\bar{q}_8$
0.10	0.9	1	0	0	0.054	0	0	0	0	0
0.05	0.9	2	0.071	0	0	0	0.297	0	0	0
0.05	0.98	4	0	0	0.054	0.325	0.297	0.036	0	0
0.01	0.98	6	0.071	0.179	0.054	0.325	0.296	0	0	0.008

Table 4: New Solutions by Dominant Modified Algorithm.

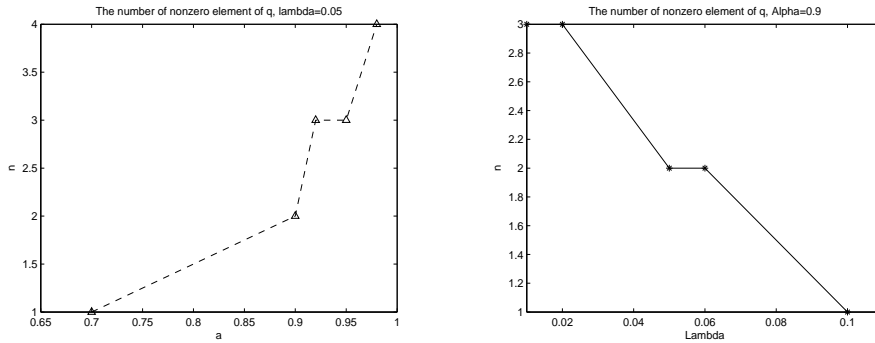


Figure 1: The number of nonzero elements of  $\tilde{\mathbf{q}}$ , when  $\lambda = 0.05$ (left),  $\alpha = 0.9$ (right)

From Tables 3 and 4 when  $\lambda = 0.1$  and  $\alpha = 0.9$  there is only one nonzero in  $\tilde{\mathbf{q}}$ ,  $\tilde{q}_3 = 1$ , after using DMA algorithm we get  $\bar{q}_3 = 0.0541$ . Although  $\bar{q}_3 = 0.0541$  recover the  $q_3 = 0.0541$ ,  $A_3$  is not the dominated BN. We note that as shown in Figure 1 and Table 3, on one hand, when  $\lambda$  is fixed, the number of nonzero elements of  $\tilde{\mathbf{q}}$  is increasing as  $\alpha$  is increased. On the other hand, when  $\alpha$  is fixed, the number of nonzero elements of  $\tilde{\mathbf{q}}$  is decreasing as  $\lambda$  is increased. We also do other examples and find that the number of nonzero elements in  $\tilde{\mathbf{q}}$  have the same change patterns. From the examples, we notice that if  $\lambda$  is too large,  $\tilde{\mathbf{q}}$  will be too sparse, the BN we got may not the dominant BN (or the probability of the BN is the dominant is small). So one should let  $\lambda$  small about 0.05 and then if we want to get a sparser  $\tilde{\mathbf{q}}$ , we could let  $\alpha$  to be 0.5, else let  $\alpha$  close to one.

### 5 Discussions

In this paper, we consider the sparse inverse problem of PBN. Firstly, we solve the non-linear program problem (1)-(2) by Lingo 9, and then propose a method to modify the solution of the non-linear programming problem. From numerical examples we discuss the problem of how to choose parameter  $\lambda$  and  $\alpha$ . Furthermore, Lingo 9 could find the local optimal solution of our examples and the complexity of DMA is  $O(N)$ .

The sparse inverse problem could also be considered as the following nonlinear programming problem:

$$h(\mathbf{q}^*) = \min\{\|\sum_{j=1}^N q_j A_j - A\|_2^2 + \lambda \|\mathbf{q}\|_\alpha^\alpha\} \tag{3}$$

subject to

$$\sum_{j=1}^N q_j = 1 \quad \text{and} \quad 0 \leq q_j \leq 1, j = 1, 2, \dots, N, \quad (4)$$

where  $0 \leq \lambda \leq 1, 0 \leq \alpha \leq 1$ . We shall study this in the future.

## Acknowledges

Research supported in part by HKRGC Grant No. 7017/07P, HKUCRGC Grants, HKU Strategy Research Theme fund on Computational Sciences, Hung Hing Ying Physical Research Sciences Research Grant, National Natural Science Foundation of China Grant No. 10971075, 10671077 and Guangdong Provincial Natural Science Grant No. 06025061, 9151063101000021.

## References

- [1] W. Ching, S. Zhang, M. Ng and T. Akutsu. *An Approximation Method for Solving the Steady-state Probability Distribution of Probabilistic Boolean Networks*, Bioinformatics, 23 (2007) 1511-1518.
- [2] W. Ching, S. Zhang, Y. Jiao, T. Akutsu and A. Wong. *Optimal Control Policy for Probabilistic Boolean Networks with Hard Constraints*, IET on Systems Biology, 3 (2009) 90-99.
- [3] W. Ching, X. Chen and N. Tsing. *Generating Probabilistic Boolean Networks from a Prescribed Transition Probability Matrix*, IET on Systems Biology, 6 (2009) 453-464.
- [4] S. Kauffman. *Homeostasis and Differentiation in Random Genetic Control Networks*, Nature, 224 (1969) 177-178.
- [5] 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.
- [6] Z. Xu, H. Zhang, Y. Wang and X. Chang.  *$L_{1/2}$  regularization*, Science in China Series F-Information Sciences, 40 (2010) 1-11.
- [7] Z. Xu, H. Zhang, Y. Wang, X. Chang, J. Wang and Y. Liang. *Variable selection and sparse reconstruction via  $L_{1/2}$  regularization*, (2010), submitted.
- [8] X. Yuan, D. Shao, S. Yu. *LINGO and Excel in mathematics modelling*, Science Press, (2007), (in Chinese).
- [9] S. Zhang, W. Ching, N. Tsing, H. Leung and D. Guo. *A New Multiple Regression Approach for the Construction of Genetic Regulatory Networks*, Journal of Artificial Intelligence in Medicine, 48 (2010) 153-160.
- [10] S. Zhang, W. Ching, X. Chen, N. Tsing. *Generating probabilistic Boolean networks from a prescribed stationary distribution*, Information Sciences, 180 (2010) 2560-2570.