JMLR Workshop and Conference Proceedings 6:237–248 NIPS 2008 workshop on causality

Reverse Engineering of Asynchronous Boolean Networks via Minimum Explanatory Set and Maximum Likelihood

Cheng Zheng ZZHENGCCHENG@PKU.EDU.CN
Yuanpei College
Peking University
Beijing, 100871, China

Zhi Geng ZGENG@MATH.PKU.EDU.CN
School of Mathematical Sciences
Peking University
Beijing, 100871, China

Editor: Isabelle Guyon, Dominik Janzing and Bernhard Schölkopf

Abstract

In this paper, we propose an approach for reconstructing asynchronous Boolean networks from observed data. We find the causal relationships in Boolean networks using an asynchronous evolution approach. In our approach, we first find a minimum explanatory set for a node to reduce complexity of candidate Boolean functions, and then we choose a Boolean function for the node based on the maximum likelihood. This approach is stimulated by the task SIGNET of the causal challenge #2 pot-luck (Jenkins, 2009). Besides the data set SIGNET, we also applied our approach to two other datasets to evaluate our approach: one is generated by Professor Isabelle Guyon and the other generated ourselves from the signal transduction network of Abscisic acid in guard cell.

Keywords: Boolean network, Reverse engineering, Structural learning

1. Introduction

Boolean network is a useful model in many applications, such as gene regulation networks (Thomas, 1973) and signal pathway (Davidich and Bornholdt, 2008). To simulate real biological networks better, various types of Boolean networks are developed, such as probabilistic Boolean networks (Shmulevich et al., 2002), and asynchronous Boolean networks (Kauffman et al., 2003; Harvey and Bossomaier, 1997). Because of the deficiency of knowledge about related chemical reaction speeds, physicists often use the asynchronous evolution approach to study the general dynamic characteristic of these networks (Chaves et al., 2005). By using the asynchronous approach, the relative timing of each reaction is chosen randomly for each update round, which is defined as the longest time for a node to respond to a change of its parent nodes. In studies of gene regulation networks, the environmental factors may change during the experiment process, and thus the reaction velocity may be affected by some external variables. Such problems can be treated by using an asynchronous Boolean network. Although many approaches have been developed for learning Boolean networks from steady states data,

which are equivalent to learning the synchronous networks (Ideker et al., 2000; Nam et al., 2006), there are a few of approaches for reconstructing asynchronous Boolean network (Jenkins, 2009; Saeed, 2009). In this paper, we propose an approach for reconstructing asynchronous Boolean networks.

In Section 2, we describe the model of asynchronous networks, and we briefly describe the format of the dataset generated from a guard cell signaling network given in Li et al. (2006). In Section 3, we discuss identification of asynchronous Boolean networks. In Section 4, we first give the definition of candidate explanatory sets and propose an algorithm for finding the minimum explanatory set. Then we propose a method for choosing a Boolean function of parent nodes for each node from all possible functions based on the maximum likelihood method. Algorithm complexity and simulation results are given in Section 5. Some possible improvement of our approach and the disadvantage of our approach are discussed in Section 6. The result for the challenge problem is given in Appendix B.

2. Model of Asynchronous Boolean Networks

A Boolean network is represented as a directed graph with N nodes where the orientation of the edges shows the causal relationship among variables. Suppose that the state of each node is determined by a Boolean regulatory function of the states of its parent nodes. For an asynchronous Boolean network, the order to update all nodes in every update round is random. The Boolean updating rules of an asynchronous network can be described as

$$\mathcal{S}_i^t = \mathcal{B}_i(\mathcal{S}_{i_1}^{t-\mathcal{I}(R_{i_1}^{t-1}>R_i^{t-1})}, \mathcal{S}_{i_2}^{t-\mathcal{I}(R_{i_2}^{t-1}>R_i^{t-1})}, \cdots, \mathcal{S}_{i_{n_i}}^{t-\mathcal{I}(R_{i_{n_i}}^{t-1}>R_i^{t-1})}) \tag{1}$$

for node $i = 1, \ldots, N$, where $\mathcal{S}_i^t$ denotes the state of node i in round t, $\mathcal{B}_i(\cdot)$ denotes the Boolean function for node i, R_i^t with the domain $\{1, \cdots, N\}$ denotes the update order of node i in round t, n_i is the number of parent nodes of node i, and $\mathcal{I}(A)$ is defined as

$$\mathcal{I}(A) = \begin{cases} 1, & A = True; \\ 0, & A = False. \end{cases} \tag{2}$$

We write the formula (1) simply as

$$\mathcal{S}_i^* = \mathcal{B}_i(\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \cdots, \mathcal{S}_{i_{n_i}}). \tag{3}$$

The observed data generated from different initial values are sequentially arranged in a matrix E with N columns and $(T+1)M$ rows, where T denotes the number of update rounds for every initial value and M denotes the number of initial values. The first $T+1$ rows are the first initial value and the data updated consequently in T rounds, the next $T+1$ rows are the data for the second initial value, and finally the last $T+1$ rows are the data for the Mth initial value.

3. Identifiability of Asynchronous Boolean Networks

For an asynchronous update rule, we can obtain more states of a network than a synchronous update rule. Thus more observed states can be used to reconstruct an asynchronous network. However, a disadvantage for an asynchronous network is that we cannot know from the data at which time point the state of a node is affected by the states of its parents. As shown in the following example, there is a network structure which cannot be identified from observed data if the data are generated from an asynchronous network, but which can be identified if the data are generated from a synchronous network.

Example 1 Consider the following two Boolean networks

$$\mathcal{S}_A^* = \mathcal{S}_B,\ \mathcal{S}_B^* = \neg\mathcal{S}_A,\ \mathcal{S}_C^* = (\mathcal{S}_A \wedge \mathcal{S}_B) \vee (\neg\mathcal{S}_A \wedge \neg\mathcal{S}_B),\ \mathcal{S}_D^* = \mathcal{S}_E,\ \mathcal{S}_E^* = \neg\mathcal{S}_D,$$

$$\mathcal{S}_A^* = \mathcal{S}_B,\ \mathcal{S}_B^* = \neg\mathcal{S}_A,\ \mathcal{S}_C^* = (\mathcal{S}_D \wedge \mathcal{S}_E) \vee (\neg\mathcal{S}_D \wedge \neg\mathcal{S}_E),\ \mathcal{S}_D^* = \mathcal{S}_E,\ \mathcal{S}_E^* = \neg\mathcal{S}_D.$$

The two networks cannot be distinguished if they are treated as asynchronous rules since $\mathcal{S}_C^t$ is independent of $(\mathcal{S}_A^t, \mathcal{S}_A^{t-1}, \mathcal{S}_B^t, \mathcal{S}_B^{t-1})$ and $(\mathcal{S}_D^t, \mathcal{S}_D^{t-1}, \mathcal{S}_E^t, \mathcal{S}_E^{t-1})$, which is shown in Appendix A. However, these two networks can easily be distinguished when they are treated as synchronous rules. It is because, for the initial value $(\mathcal{S}_A, \mathcal{S}_B, \mathcal{S}_C, \mathcal{S}_D, \mathcal{S}_E) = (1,1,0,1,0)$, the first network has the updated value $(1,0,1,0,0)$ after one update round, while the second network has the value $(1,0,0,0,0)$.

We guess that if an asynchronous network has a steady state for each initial value, then the network may be identifiable. The identification means that if we obtain enough initial states and enough update processes, we can discover the structure and Boolean rules for the whole network correctly.

4. Reconstruction of Asynchronous Boolean Networks

As shown in formula (1), the state of node i in round t can be affected only by the state of its parents in round t or round $t-1$. For each time point t, we combine the states of N nodes in both rounds t and $t-1$ into one row, and we get a data matrix with $2N$ columns and $T \times M$ rows for M initial values and T rounds. In this section, we propose an algorithm MESML for finding Boolean functions in which for a particular node, first the Minimum Explanatory Set is found, and then a Boolean function that has the Maximum Likelihood is chosen.

4.1 Find the minimum explanatory set

Ideker et al. (2000) proposed an algorithm for reconstructing a synchronous Boolean network. The network contains N nodes: $a_1, \cdots, a_N$, and each node a_i has a Boolean function f_i. Observed data are given in a matrix E', whose each row is for an individual and each column is for a variable. For every node a_i, assume that a Boolean function $a_i = f_i(a_{i_1}, \cdots, a_{i_{n_i}})$ always holds for all rows. Their algorithm has the following three steps. For every node a_i,

(i). consider every pair of rows (t_1, t_2) in E' such that the states of a_i differs. For each pair, find the set V_{t_1,t_2} of all other nodes whose states are different for these two rows, i.e., $V_{t_1,t_2} = \{j : \mathcal{S}_j^{t_1} \neq \mathcal{S}_j^{t_2}\}$;

(ii). find the minimum set of nodes V_{min} which intersects all $V_{j,k}$ obtained at Step 1, i.e., $V_{min} \cap V_{j,k} \neq \emptyset$; V_{min} is used to find a Boolean rule f_i;

(iii). find a truth table from the data. For a deterministic Boolean network, f_i can be obtained from the observed data of variables in V_{min}. For a combination of levels of variables in V_{min} that does not appear in the data, we cannot get the truth value of node a_i for this combination.

For asynchronous networks, it seems that a difference set V_{t_1,t_2} for node i at a pair (t_1, t_2) of time points could be defined similarly to the above case by comparing a variable at two consecutive time points simultaneously. That is, $V_{t_1,t_2} = \{j : \mathcal{S}_j^{t_1} \neq \mathcal{S}_j^{t_2}$ or $\mathcal{S}_j^{t_1-1} \neq \mathcal{S}_j^{t_2-1}\}$. In the following example, we show that such a difference set is improper.

Example 2 Consider the asynchronous Boolean network

$$\mathcal{S}_A^* = 1,\ \mathcal{S}_B^* = \neg\mathcal{S}_A,\ \mathcal{S}_C^* = \neg\mathcal{S}_B.$$

For the single value $(0,0,0)$ of $(\mathcal{S}_A^t, \mathcal{S}_B^t, \mathcal{S}_C^t)$, we may get the following two different data $(1,1,0)$ or $(1,1,1)$ after one update round. By the above definition of a difference set, we get for variable C that $V_{t_1,t_2} = \emptyset$ for these two data, and thus we cannot get V_{min}, which means that no variable can explain variable C.

To avoid the above mistake, we can revise the definition of the difference set for node i as $V_{t_1,t_2} = \{j : \mathcal{S}_j^{t_1} \neq \mathcal{S}_j^{t_2}$ or $\mathcal{S}_j^{t_1-1} \neq \mathcal{S}_j^{t_2-1}$ or $\mathcal{S}_j^{t_1} \neq \mathcal{S}_j^{t_1-1}\}$. By the revised definition, we get $V_{t_1,t_2} = \{A,B\}$ for Example 2. We say that V is an explanatory set for node i if $V \cap V_{t_1,t_2} \neq \emptyset$ for all pairs (t_1,t_2). In our approach, we find the minimum explanatory set V as a candidate set of parent nodes of node i. This is called the minimum set covering and can be calculated by the branch and bound technique (Nemhauser, 1988). By the revised difference set, it may be shown that the parent set of node i can be found by using an explanatory set. If a node has n parents, then the algorithm can stop before we search all sets with not more than n elements. Unfortunately, for an explanatory set V, we cannot ensure the existence of a Boolean function that can explain the whole data (see Example 3). Thus unlike the approach for synchronous networks, to decide whether an explanatory set is appropriate, we need to check the existence of a Boolean function which can explain the whole data. If there does not exist such a function, we try the next minimum explanatory set V that satisfies $V \cap V_{t_1,t_2} \neq \emptyset$ for all t_1 and t_2. We repeat this process until finding a Boolean function.

Example 3 This example illustrates that an explanatory set V may not ensure the existence of a Boolean function for explaining all data from an asynchronous network. Consider the following asynchronous network

$$S_A^* = 1,\ S_B^* = S_A,\ S_C^* = \neg S_A.$$

From it, we may get the following data of $(\mathcal{S}_A^t, \mathcal{S}_B^t, \mathcal{S}_C^t; \mathcal{S}_A^{t+1}, \mathcal{S}_B^{t+1}, \mathcal{S}_C^{t+1})$ at three update rounds

$$(1,1,0;1,1,0) \text{ for } t = t_1,\ (0,0,0;1,0,0) \text{ for } t = t_2,\ (0,1,0;1,0,1) \text{ for } t = t_3.$$

It is obvious that $\{B\}$ is an explanatory set of C. From the data at t_1 and t_2, we find that C is a constant 0 for $B = 0$ and 1, and thus we get the function $C = 0$. But for the data at t_3, we have $C_{t_3+1} = 1$. This means that there does not exist a Boolean function $S_C^* = \mathcal{B}_C(S_B)$.

4.2 Find a Boolean Function

To find a Boolean function, we propose the following process. Given the minimum explanatory set $V = \{j_1, j_2, \cdots, j_{n_i}\}$ for node i obtained with the approach presented in the previous subsection, first we extract those rows that satisfy $\mathcal{S}_{j_m}^{t-1} = \mathcal{S}_{j_m}^t$ for all m, and we can directly get a value of the Boolean function from these rows. We extract all different states and calculate a

log likelihood for each of possible functions $\mathcal{B}_i(\mathcal{S}_{j_1},\cdots,\mathcal{S}_{j_{n_i}})$

$$\begin{aligned}
&l[S_i^* = \mathcal{B}_i(\mathcal{S}_{j_1},\cdots,\mathcal{S}_{j_{n_i}}), data] \\
=&\sum_t \ln P[S_i^t = \mathcal{B}_i(\mathcal{S}_{j_1}^{t-\mathcal{I}(R_{j_1}^{t-1}>R_i^{t-1})},\cdots,\mathcal{S}_{j_{n_i}}^{t-\mathcal{I}(R_{j_{n_i}}^{t-1}>R_i^{t-1})})] \\
=&\sum_t \ln\{\sum_{(R_{j_1}^{t-1},\cdots,R_{j_{n_i}}^{t-1},R_i^{t-1})} [\mathcal{I}(S_i^t = \mathcal{B}_i(\mathcal{S}_{j_1}^{t-\mathcal{I}(\mathcal{R}_{j_1}^{t-1}>\mathcal{R}_i^{t-1})},\cdots,\mathcal{S}_{j_{n_i}}^{t-\mathcal{I}(\mathcal{R}_{j_{n_i}}^{t-1}>\mathcal{R}_i^{t-1})})) \\
&\times P(R_{j_1}^{t-1},\cdots,R_{j_{n_i}}^{t-1},R_i^{t-1})]\} \\
=&\sum_t \ln\{\sum_{(a_{j_1}^{t-1},\cdots,a_{j_{n_i}}^{t-1})} [\mathcal{I}(S_i^t = \mathcal{B}_i(\mathcal{S}_{j_1}^{t-a_{j_1}^{t-1}},\cdots,\mathcal{S}_{j_{n_i}}^{t-a_{j_{n_i}}^{t-1}}))P(a_{j_1}^{t-1},\cdots,a_{j_{n_i}}^{t-1})]\},
\end{aligned} \tag{4}$$

where $a_{j_k}^{t-1} = \mathcal{I}(R_{j_k}^{t-1} > R_i^{t-1})$, which has value 0 or 1. The maximum log likelihood equal to $-\infty$ implies that there is not any function $\mathcal{B}_i$ such that $\mathcal{S}_i^* = \mathcal{B}_i(\mathcal{S}_{j_1},\cdots,\mathcal{S}_{j_{n_i}})$. We find a function $\mathcal{B}_i$ which has the maximum log likelihood. Assume that $(R_1^t,\cdots,R_N^t)$ has a uniform distribution over all permutations of $\{1,2,\cdots,N\}$ for all t. Then we have

$$P(a_1^t,\cdots,a_{n_i}^t) = \frac{(\sum_{m=1}^{n_i} a_m^t)! \times (n_i - (\sum_{m=1}^{n_i} a_m^t))!}{(n_i+1)!}. \tag{5}$$

4.3 Algorithm MESML for finding Boolean functions

Let $\Omega = \{1,2,\ldots,(T+1)M\}$ denote a set of row indexes for the data matrix E. Let $\Omega_I = \{1,(T+1)+1,2(T+1),\ldots,(M-1)(T+1)+1\}$ denote the set of row indexes for M initial values, and let $\Omega_{NI} = \Omega \setminus \Omega_I$ denote the set of row indexes for non-initial data.

For each node $i \in \{1,\ldots,N\}$, we find a Boolean function $\mathcal{B}_i(V)$ as follows:

Step 1 If all non-initial values of variable i are constant, that is, $S_i^t = S_i^{t'}$ for all t and $t' \in \Omega_{NI}$, then return a constant function $\mathcal{B}_i(V) = S_i^t$.

Step 2 For each $t \neq t'$, calculate the difference set $V_{t,t'}$ if $S_i^t \neq S_i^{t'}$.

Step 3 Find a minimum set V which intersects all $V_{t,t'}$ obtained at Step 2, that is, $V \bigcap V_{t,t'} \neq \emptyset$ for all t and $t' \in \Omega_{NI}$.

Step 4 Find a Boolean function $\mathcal{B}_i(V)$ which maximizes the log likelihood $l[\mathcal{B}_i(V), data]$.

Step 5 If $l[\mathcal{B}_i(V), data] = -\infty$, then we cannot obtain a Boolean function of the current set V for the node i and go to Step 3 to try the next minimum set V.

Step 6 Otherwise, output the Boolean function $\mathcal{B}_i(V)$.

Example 4 This example illustrates the algorithm MESML. Consider the following Boolean network with $N = 3$ nodes A, B and C:

$$S_A^* = 1,\ S_B^* = S_A,\ S_C^* = \neg S_A.$$

Suppose that we have $M = 3$ initial vectors of $(\mathcal{S}_A,\mathcal{S}_B,\mathcal{S}_C)$: $(1,1,0)$, $(0,0,0)$ and $(0,1,0)$. After one update round for each initial vector, we get the following data of $(\mathcal{S}_A^t,\mathcal{S}_B^t,\mathcal{S}_C^t;\mathcal{S}_A^{t+1},$ $\mathcal{S}_B^{t+1},\mathcal{S}_C^{t+1})$:

(i). $(1,1,0;1,1,0)$ for the first initial value labeled as $t=1$,

(ii). $(0,0,0;1,0,0)$ for the second initial value labeled as $t=3$, and

(iii). $(0,1,0;1,0,1)$ for the third initial value labeled as $t=5$,

where the first three numbers in each bracket are an initial vector, and the last three numbers are the vector obtained after one update round. To clearly separate the initial values from the updated data, the data matrix E is revised by combining two rows into one as follows

$$\left(\begin{array}{ccc|ccc} \mathcal{S}_A^1 & \mathcal{S}_B^1 & \mathcal{S}_C^1 & \mathcal{S}_A^2 & \mathcal{S}_B^2 & \mathcal{S}_C^2 \\ \mathcal{S}_A^3 & \mathcal{S}_B^3 & \mathcal{S}_C^3 & \mathcal{S}_A^4 & \mathcal{S}_B^4 & \mathcal{S}_C^4 \\ \mathcal{S}_A^5 & \mathcal{S}_B^5 & \mathcal{S}_C^5 & \mathcal{S}_A^6 & \mathcal{S}_B^6 & \mathcal{S}_C^6 \end{array}\right) = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right),$$

where the left parts in the above matrixes are for the initial values and the right parts are for the data obtained after one update round. First we use the algorithm MESML to find the Boolean function for node A. At Step 1, since $\Omega_{NI}=\{2,4,6\}$ and $\mathcal{S}_A^2=\mathcal{S}_A^4=\mathcal{S}_A^6=1$, we output $S_A^*=1$.

Next we try to find a Boolean function for node B. Since $\mathcal{S}_B^2\neq\mathcal{S}_B^4$, the condition at Step 1 does not hold. At step 2, there are two pairs of (t,t'): (2,4) and (2,6) such that $\mathcal{S}_B^t\neq\mathcal{S}_B^{t'}$. We calculate the difference set for the pair $(2,4)$. Since $\mathcal{S}_A^1=\mathcal{S}_A^2=\mathcal{S}_A^4=1\neq 0=\mathcal{S}_A^3$ and $\mathcal{S}_C^1=\mathcal{S}_C^2=\mathcal{S}_C^3=\mathcal{S}_C^4=0$, by definition of difference set, we get $V_{2,4}=\{A\}$. Then we calculate the difference set for the pair $(2,6)$. Since $\mathcal{S}_A^1=\mathcal{S}_A^2=\mathcal{S}_A^6=1\neq 0=\mathcal{S}_A^5$ and $\mathcal{S}_C^1=\mathcal{S}_C^2=\mathcal{S}_C^5=0\neq 1=\mathcal{S}_C^6$, we get $V_{2,6}=\{A,C\}$. At Step 3 we find that for each set that has one element, only $\{A\}$ is an explanatory set. At Step 4, we find the truth table for A and B, and we can obtain the Boolean function $\mathcal{S}_B^*=\mathcal{S}_A$. At Step 6, we output the result.

Finally we want to find a Boolean function for node C. Since $\mathcal{S}_C^2\neq\mathcal{S}_C^6$, the condition at Step 1 does not hold. At step 2, there are also two pairs of (t,t'): (2,6) and (4,6) such that $\mathcal{S}_C^t\neq\mathcal{S}_C^{t'}$. First we calculate the difference set for the pair $(2,6)$. Since $\mathcal{S}_A^5=0\neq 1=\mathcal{S}_A^1=\mathcal{S}_A^2=\mathcal{S}_A^6$ and $\mathcal{S}_B^1=\mathcal{S}_B^2=\mathcal{S}_B^5=1\neq 0=\mathcal{S}_B^6$, we get $V_{2,6}=\{A,B\}$. Then we calculate the difference set for the pair $(4,6)$. Since $\mathcal{S}_A^4=\mathcal{S}_A^6=1\neq 0=\mathcal{S}_A^3=\mathcal{S}_A^5$ and $\mathcal{S}_B^3=\mathcal{S}_B^4=\mathcal{S}_B^6=0\neq 1=\mathcal{S}_B^5$, we have $V_{4,6}=\{A,B\}$. At Step 3, we find that for each set with a single element, both $\{A\}$ and $\{B\}$ are explanatory sets. At Step 4, we search the Boolean rules for them separately. As shown in Example 3, there does not exist a Boolean rule of $\{B\}$ that can explain the data, which means that $l[\mathcal{B}_C(\mathcal{S}_B),data]=-\infty$. So go back to Step 3 to try the next minimum set. Since there exists a boolean rule of A and C, we get $\mathcal{S}_C^*=\neg\mathcal{S}_A$ such that $l[\mathcal{B}_C(\mathcal{S}_B),data]\neq-\infty$, and then go to Step 6 for output.

5. Simulation and Algorithm Analysis

5.1 Simulation results

From the result obtained by our approach, we find that if the parents of a variable are found correctly, then the Boolean regulatory rule is always obtained correctly. Thus our method based on the maximum likelihood is an efficient way for selecting the Boolean regulatory rule. For our results, the average error rate defined by the challenge organizers is 0.14 for the original SIGNET dataset, 0.06 for our dataset and 0.05 for the dataset generated by Professor Guyon. We also find that the error rate increases as the the number of parent nodes increases. The dataset generated by Professor Guyon is used for further simulation, and the error rate, the number of real parent nodes and CPU time for various cases are shown in Table 1. All of our computations are performed on a computer with CPU 1.73 GHz and 1.00 GB RAM and the algorithm is implemented with R language. The CPU times do not include the data preprocess

and the preprocess takes about an hour. In our algorithm, we limit the maximum size of the set of parent nodes up to 4 since the CPU time for more than 5 parent nodes may take more than 30 days. For the last line in Table 1, the number of real parents is 5, but we limit the number of parents to be found up to 4. Thus the CPU time is almost the same as the case with 4 real parents, but the average error rate is much larger than the case with 4 real parents.

Number of real parents	Average error rate	CPU Time
0	0.000	0.1 seconds
1	0.000	10 seconds
2	0.063	10 minutes
3	0.150	12 hours
4	0.110	2.5 days
5	0.281	2.5 days

Table 1: Error rate and CPU time for the different number of parents of a node.

5.2 Algorithm Analysis

To find the minimum set, we need to calculate $V_{t,t'}$ for all t and $t' \in \Omega_{NI}$. The complexity is $2N(T \times M)^2$. If a node has K parents, the total computational complexity for finding the minimum explanatory set V is not more than $\sum_{k=1}^{K} C_N^k$ times. Given a minimum set V with K nodes, we need to calculate not more than 2^{2^K} log likelihoods. But if we search on the whole space of Boolean functions without using the minimum set, the complexity is about $\sum_{k=1}^{K}(2^{2^k} \times C_N^k)$. So our algorithm can greatly reduce the complexity for a large network with large N and K. However, when K and N are small but M and T are large, it takes much times to find the minimum set, and so we directly try all possible sets V with sizes equal to 1 and 2 without finding the minimum set.

The advantages of our algorithm are that our approach takes full use of the information in the data, especially the information on the dynamic evolution processes. Thus our algorithm can have a higher accuracy for learning structures and Boolean functions, especially for those sparse structures with circles and without steady states.

6. Discussion

In the algorithm MESML, we stop the process as long as we find a single minimum parent set which has a Boolean function that maximizes the likelihood and can explain the whole data set. If we do not consider the computational complexity, we should try all possible minimum parent sets to find a Boolean function which maximizes the likelihood or some other scores over all sets. To compare the models with different numbers of parent nodes, we can use AIC score or other scores for model selection. Pearson's χ^2 test can be used to check whether the Boolean regulatory function fits the observed data. The χ^2 test is described as follow. The hypotheses are

$$H_0 : \mathcal{S}_i^t = \mathcal{B}_i(\mathcal{S}_{j_1}^{t-\mathcal{I}(\mathcal{R}_{j_1}^{t-1}>\mathcal{R}_i^{t-1})}, \cdots, \mathcal{S}_{j_{n_i}}^{t-\mathcal{I}(\mathcal{R}_{j_{n_i}}^{t-1}>\mathcal{R}_i^{t-1})}),$$

$$H_1 : \mathcal{S}_i^t \neq \mathcal{B}_i(\mathcal{S}_{j_1}^{t-\mathcal{I}(\mathcal{R}_{j_1}^{t-1}>\mathcal{R}_i^{t-1})}, \cdots, \mathcal{S}_{j_{n_i}}^{t-\mathcal{I}(\mathcal{R}_{j_{n_i}}^{t-1}>\mathcal{R}_i^{t-1})}).$$

An asynchronous rule $\mathcal{S}_i^t = \mathcal{B}_i(\cdot)$ may take multiple values depending the orders of parent nodes before or after the order of node i. Under the assumption that the orders have a uniform dis-

tribution, we can obtain the distribution of S_i^t conditional on its parent nodes, which is given in the likelihood (4) (i.e., the argument of ln function). Thus we can test the hypothesis using Pearson's χ^2 statistic:

$$\chi^2 = \sum_{\lambda} T_{i\lambda} = \sum_{\lambda} \sum_{j=0}^{1} \frac{(O_{ij\lambda} - E_{ij\lambda})^2}{E_{ij\lambda}},$$

where λ denotes a state of the conditional set $(S_{j_1}^t, \cdots, S_{j_{n_i}}^t; S_{j_1}^{t-1}, \cdots, S_{j_{n_i}}^{t-1})$, $O_{ij\lambda}$ is an observed frequency

$$O_{ij\lambda} = \sum_t \mathcal{I}(S_i^t = j, (S_{j_1}^t, \cdots, S_{j_{n_i}}^t; S_{j_1}^{t-1}, \cdots, S_{j_{n_i}}^{t-1}) = \lambda),$$

$E_{ij\lambda}$ is the expectation

$$E_{ij\lambda} = n_\lambda P(S_i^t = j | (S_{j_1}^t, \cdots, S_{j_{n_i}}^t; S_{j_1}^{t-1}, \cdots, S_{j_{n_i}}^{t-1}) = \lambda; H_0),$$

which can be calculated by (4) and (5), and the total frequency is

$$n_\lambda = \sum_t \mathcal{I}((S_{j_1}^t, \cdots, S_{j_{n_i}}^t; S_{j_1}^{t-1}, \cdots, S_{j_{n_i}}^{t-1}) = \lambda).$$

The statistics $T_{i\lambda}$ for all states of the conditional set are mutually independent. The statistic χ^2 asymptotically has a χ^2 distribution with F degrees of freedom, where F is the number of different states of $(S_{j_1}^t, \cdots, S_{j_{n_i}}^t; S_{j_1}^{t-1}, \cdots, S_{j_{n_i}}^{t-1})$.

To evaluate the test, we did a simulation on the above test for checking whether a Boolean rule can be accepted or rejected correctly. Using the dataset of the task SIGNET, we obtain the results shown in Table 2. The significance level α for the test is 0.001. The test results illustrate that Pearson's χ^2 test can be used to improve the accuracy for learning Boolean rules.

Rule	Accept	Reject
True	34	2
False	1	6

Table 2: Simulation results of χ^2 tests for Boolean rules.

The deficiency of our approach is that the accuracy may decrease if the network contains too many hub nodes. We limit the number of parent nodes up to 4 and stop our algorithm after searching all sets with not more than four nodes even if no explanatory set is found. This is a tradeoff between time cost and the error rate.

In our approach, we use a conditional likelihood for a single node i instead of the full likelihood for all nodes to reduce the computational complexity. Thus our approach may not obtain the optimal result. Since we consider a Boolean regulatory rule for each node i separately, the update order we obtained may not be suitable for other Boolean regulatory rules although it maximizes the likelihood of the Boolean rule for node i. Below we give an example to illustrate this.

Example 5 Suppose that the observed data of $(\mathcal{S}_A^0, \mathcal{S}_B^0, \mathcal{S}_A^1, \mathcal{S}_B^1)$ are (1,1,0,0). First consider A only, and the rule $S_A^* = \neg S_B$ can explain the data under the order $(R_A^1 = 1, R_B^1 = 2)$. Next consider B only, and the rule $S_B^* = \neg S_A$ can also explain the data under another update order $(R_A^1 = 2, R_B^1 = 1)$. So it seems that the rules $S_A^* = \neg S_B$ and $S_B^* = \neg S_A$ could explain the data. However, from the rules, we find that possible data are either $(1,1,0,1)$ or $(1,1,1,0)$, which do not contain the observed data $(1,1,0,0)$. Thus we must add one more step after our algorithm to check whether all the Boolean rules we learned can explain the observed data.

Acknowledgments

We would like to thank the four reviewers for their valuable comments. We would appreciate I. Guyon and all the competition organizers for their encouragement and support to our work. This research was supported by NSFC (10771007, 10721403), 863 Project of China (2007AA01Z437), MSRA and MOE-Microsoft Key Laboratory of Statistics and Information Technology of Peking University.

References

M. Chaves, R. Albert, and E. Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *J Theor Biol*, 235:431–449, 2005.

M. Davidich and S. Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One.*, 3:e1672, 2008.

I. Harvey and T. Bossomaier. Time out of joint: Attractors in asynchronous random boolean networks. *Proc. Fourth European Conference on Artificial Life*, pages 67–75, 1997.

T. Ideker, V. Thorsson, and R. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. *Pacific Symp Biocomput*, 5:302–313, 2000.

J. Jenkins. Signet: Boolean rule determination for abscisic acid signaling. *Proc. Machine Learning Research*, 5(In this volume), 2009.

S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Random boolean network models and the yeast transcriptional network. *PNAS.USA*, 100:14796–14799, 2003.

S. Li, S. Assmann, and R. Albert. Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling. *PLOS Biol*, 4(10),e312: 1732–1748, 2006.

D. Nam, S. Seo, and S. Kim. An efficient top-down search algorithm for learning boolean networks of gene expression. *Machine Learning*, 65,1:229–245, 2006.

G. Nemhauser. *Integer and combinatorial optimization*. Wiley, New York, 1988.

M. Saeed. The use of bernoulli mixture models for identifying corners of a hypercube and extracting boolean rules from data. *Proc. Machine Learning Research*, 5(In this volume), 2009.

I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinfo.*, 18:261–274, 2002.

R. Thomas. Boolean formalization of genetic control circuits. *Theor Biol*, 42:563–585, 1973.

Appendix A. Proof of Example 1

Below we show the statement for Example 1: $\mathcal{S}_C^t$ is independent of $(\mathcal{S}_A^t, \mathcal{S}_A^{t-1}, \mathcal{S}_B^t, \mathcal{S}_B^{t-1})$. Since all information on the network is contained by the conditional probabilities $P(S_A^t, S_B^t, S_C^t | S_A^{t-1}, S_B^{t-1}, S_C^{t-1})$, we first calculate these probabilities, and then we can calculate the conditional probabilities $P(S_C^t | S_A^{t-1}, S_B^{t-1}, S_A^t, S_B^t)$. It is easy to get that $P(S_C^t = 1 | S_A^{t-1} = i, S_B^{t-1} =$

$j, S_A^t = k, S_B^t = l) = 0.5$ for all i, j, k and l =0 and 1. Thus we showed that $\mathcal{S}_C^t$ is independent of $(\mathcal{S}_A^t, \mathcal{S}_A^{t-1}, \mathcal{S}_B^t, \mathcal{S}_B^{t-1})$. By the symmetry of (A,B) and (D,E), we have that $\mathcal{S}_C^t$ is independent of $(\mathcal{S}_D^t, \mathcal{S}_D^{t-1}, \mathcal{S}_E^t, \mathcal{S}_E^{t-1})$.

Appendix B. Pot-luck challenge: FACT SHEET

Method:

- Preprocessing
  First, we combine the state vectors at two consequent time points into one vector. Then for each variable i, we find a difference set V_{t_1,t_2} of variables for a pair of two combined vectors.

- Possible Explanatory Set finding and Boolean Rule Finding
  Since all initial values for the variable ABA are 1 in the original SIGNET dataset, each node which has a single parent ABA will have a constant value, and then the node is determined incorrectly as a root node. Thus we assume that the five root nodes are known for the dataset. For our dataset and Isabelle's dataset, we have different initial values of root nodes, and thus we need not make this assumption. To reduce the computational complexity, we limit the number of parents of each variable not more than 4. For each variable i that is not a root, we first find a minimum explanatory set V that satisfies $V \cap V_{t,t'} \neq \emptyset$ for all t and t', and then we find the Boolean rule of the parent set V for the variable i based on the maximum likelihood. If there is no such Boolean rules, then we try the next minimum explanatory set and repeat this process until finding a Boolean rule.

- Post Process
  If all minimum sets with not more than 4 nodes cannot explain the data, then we select a set V and a Boolean rule that contradict with the data set at the least.

Results:

Dataset/Task	Score
SIGNET/original dataset generated by Jenkins	0.14
SIGNET/dataset generated by Isabelle	0.05
SIGNET/my dataset	0.06

Table 3: Average error rates for three data sets.

- Quantitative advantages (e.g., compact feature subset, simplicity, computational advantages) The computational complexity is too high to utilize fully observed data from the dynamic process of an asynchronous network. But it is easy to implement our algorithm since our algorithm uses conditional likelihoods and it treats nodes one-by-one. See Section 5.2 for the complexity analysis.

- Given a parent set of a node, its Boolean rule is found based on the maximum likelihood. We use the conditional likelihood for a single node to reduce the computational complexity. When the network is not too complexity (i.e., the number of parent nodes for each node is less than 5), our method can run fast and it may be improved by using

the full maximum likelihood of all nodes and using Pearson's χ^2 test or other model-selection scores to check whether a Boolean rule fits observed data well. Besides, our method takes full use of the information in the data, especially the information on the dynamic evolution processes. Using the information, we can discover many structures that cannot be found only from the information on steady states.

Now we briefly explain our implementation. First, we change the names of variables to be numerical style and transform the original data to a matrix. Then for each node, we use R program to process the preprocessed data to find its parent set and its truth table. Finally, we find a Boolean rule from the truth table. Contact us via email to ask for the code.