Proceedings of the 10th
World Congress on Intelligent Control and Automation
July 6-8, 2012, Beijing, China

Approximation of Boolean Networks*

Daizhan Cheng and Yin Zhao
Key Lab. of Systems and Control
Institute of Systems Science, Chinese Academy of Sciences
Beijing 100190, P.R. China
dcheng@iss.ac.cn, zhaoyin@amss.ac.cn

Jongrae Kim and Yunbo Zhao
Division of Biomedical Engineering
University of Glasgow
Glasgow G12 8QQ, UK
Jongrae.Kim@glasgow.ac.uk, yunbozhao@gmail.com

***Abstract*— The problem of approximation to large-scale Boolean networks is considered. First, we assume a large-scale Boolean network is aggregated into several sub-networks. Using the outputs(or inputs) of each sub-network as new state variables, a new simplified time-varying network is obtained. Then a time-invariant Boolean network is used to approximate each subsystem. Observed data are used to find the best approximating dynamic models. Finally, the aggregation method is investigated.**

***Index Terms*— Boolean network, Aggregation, Approximation, Modularity.**

I. PRELIMINARIES

Accompanying the development of the systems biology the interest in Boolean networks is rapidly increasing, because the Boolean network, introduced firstly by Kauffman [1], has been proved to be very powerful in modeling and quantitative description of cell regulation [1], [2], [3], [4], [5], [6]. There are two major obstacles in applying this model to real biological problems. The first one is that the dynamic model of Boolean networks is a logical process. Unlike quantity-based process, which can be described by differential equations or difference equations, there are less mathematical tools for logical process. Another difficulty lies on the computational complexity. In general a real molecular network has large number of nodes, it is very difficult, if not impossible, to produce a precise model to describe it and further analyze and control it. It has been pointed out that the computation complexity of Boolean network related problems are NP hard problem [7]. In fact, precise model can only be used to deal with some simple genic networks such as yeast cell network [8].

Recently, a new approach called the linear representation of Boolean networks was proposed to convert the dynamics of a Boolean network from its logical form into an algebraic form [9]. Under the algebraic form the dynamics of a Boolean network becomes a standard discrete-time dynamic systems, which involves no logical operators any more. Hence, the conventional tools, dealing with discrete-time systems, are applicable to Boolean networks. This approach has been successfully applied to solve several important problems about the structure analysis and synthesis of Boolean (control) networks [10].

To describe this approach, we first introduce some notations.

1) $\mathcal{M}_{m\times n}$: the set of $m\times n$ real matrices. When $m=n$ it is briefly denoted as $\mathcal{M}_n$;
2) $\mathrm{Col}(A)$ ($\mathrm{Row}(A)$): the set of columns of A, $\mathrm{Col}_i(A)$ ($\mathrm{Row}_i(A)$): the i-th column (row) of A;
3) δ_n^i: the i-th column of the identity matrix I_n;
4) $\mathcal{D}:=\{0,1\}$;
5) Δ_n: $\Delta_i=\mathrm{Col}(I_n)$;
6) $A\in\mathcal{M}_{m\times n}$ is called a logical matrix, if $\mathrm{Col}(A)\subset\Delta_m$, the set of $m\times n$ logical matrices is denoted by $\mathcal{L}_{m\times n}$;
7) A logical matrix $A=[\delta_m^{i_1}\ \delta_m^{i_2}\ \cdots\ \delta_m^{i_n}]$ is briefly denoted by $A=\delta_m[i_1,\cdots,i_n]$.
8) A matrix $A=(a_{i,j})\in\mathcal{M}_{m\times n}$ is called a Boolean matrix if its entries $a_{i,j}\in\mathcal{D}$, $i=1,\cdots,m$, $j=1,\cdots,n$. The set of $m\times n$ Boolean matrices is denoted by $\mathcal{B}_{m\times n}$.
9) Let $A=(a_{i,j})\in\mathcal{B}_{m\times n}$. The Hamming weight of A is the number of its nonzero entries, denoted by

$$\|A\|:=|\{a_{i,j}|a_{i,j}=1,\ 1\le i\le m,\ 1\le j\le n\}|.$$

The basic tool for this approach is the semi-tensor product of matrices [11], which is a generalization of the conventional matrix product, defined as follows.

Definition 1.1: Let $A\in\mathcal{M}_{m\times n}$ and $B\in\mathcal{M}_{p\times q}$. Assume the least common multiple of n and p is $t=\mathrm{lcm}\{n,p\}$. Then the semi-tensor product of A and B, denoted by $A\ltimes B$, is defined as

$$\left(A\otimes I_{t/n}\right)\left(B\otimes I_{t/p}\right). \tag{1}$$

Remark 1.2: Throughout this paper the matrix product is assumed to be the semi-tensor product. Since it is a generalization of conventional matrix product, the symbol $\ltimes$ is mostly omitted, unless we should like to emphasize it.

A logical variable x can take values from $\mathcal{D}$. A logical function $f(x_1,\cdots,x_n)$ is a mapping $f:\mathcal{D}^n\to\mathcal{D}$.

* Supported partly by NNSF 61074114, 60736022, and 60821091 of China.

978-1-4673-1398-8/12/$31.00 ©2012 IEEE

Identifying $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$, we have vector form $x \in \Delta_2$ and $f : \Delta_2^n \to \Delta_2$. In vector form, define $x = \ltimes_{i=1}^n x_i$, then we have the following fundamental result.

Proposition 1.3: Given logical function $f(x_1, \cdots, x_n)$. There exists a unique logical matrix, $M_f \in \mathcal{L}_{2\times 2^n}$, such that in vector form we have

$$f(x_1, \cdots, x_n) = M_f x. \tag{2}$$

Next, we consider a Boolean network. In general, the dynamics of a Boolean network can be expressed as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \cdots, x_n(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \cdots, x_n(t)), \end{cases} \tag{3}$$

where f_i, $i = 1, \cdots, n$ are logical functions.

Using Proposition 1.3, we can express (3) into its component-wise algebraic form as

$$\begin{cases} x_1(t+1) = L_1 x(t) \\ \vdots \\ x_n(t+1) = L_n x(t), \end{cases} \tag{4}$$

where $L_i \in \mathcal{L}_{2\times 2^n}$ is the structure matrix of f_i, $i = 1, \cdots, n$.

Multiplying the n equations in (4) together yields the algebraic form of (3) as [10]

$$x(t+1) = Lx(t), \tag{5}$$

where

$$L = L_1 * L_2 * \cdots * L_n,$$

"*" is the the Khatri-Rao product [12]. That is,

$$\mathrm{Col}_i(L) = \ltimes_{j=1}^n \mathrm{Col}_i(L_j), \quad i = 1, \cdots, n.$$

L is called the transition matrix of the network.

We give a simple example to illustrate it.

Example 1.4: Fig. 1 depicts a Boolean network.

Its dynamics is described as

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ x_2(t+1) = \neg x_1(t) \\ x_3(t+1) = x_2(t) \vee x_3(t). \end{cases} \tag{6}$$

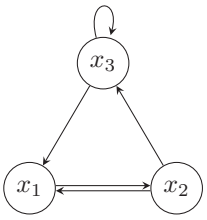

Fig. 1. A Boolean network

It is easily verified that the network (6) has its component-wise algebraic form as

$$\begin{cases} x_1(t+1) = \delta_2[1\ 2\ 2\ 2\ 1\ 2\ 2\ 2]x(t) \\ x_2(t+1) = \delta_2[2\ 2\ 2\ 2\ 1\ 1\ 1\ 1]x(t) \\ x_3(t+1) = \delta_2[1\ 1\ 1\ 2\ 1\ 1\ 1\ 2]x(t). \end{cases} \tag{7}$$

Multiplying three equations in (7) yields the algebraic form (5) of network (6) with its transition matrix matrix

$$L = \delta_8[3\ 7\ 7\ 8\ 1\ 5\ 5\ 6].$$

The algebraic form (5) of a Boolean network is very powerful in investigating its properties and its control designs. In fact, since the transition matrix is unique, we can design or manipulate a system by determine or change its transition matrix.

Under the framework of algebraic form, this paper aims at reducing the computational complexity of a large-scale Boolean network by approximating it by a simplified network. First, assume the network is aggregated, and time-invariant networks are used to approximate each aggregated sub-networks. Secondly, the modularity proposed in [13], [14] is used to aggregate the system.

The paper is organized as follows: Section 2 considers the expression of the simplified network, assume the network is aggregated. Using observed data, section 3 provides a approximation to each sub-network by a time-invariant Boolean network. Section 4 reviews the method to aggregate a network. Section 5 is a brief conclusion.

II. Transfer Matrix of Aggregated Networks

Consider a Boolean control network. Its dynamics can be expressed as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)); \end{cases}$$

$$y_j(t) = h_j(x_1(t), \cdots, x_n(t)), \quad j = 1, \cdots, p, \tag{8}$$

where x_i, $i = 1, \cdots, n$ are state variables, u_k, $k = 1, \cdots, m$ are inputs (or controls), and y_j, $j = 1, \cdots, p$ are outputs.

Using a similar technique as for Boolean network, we can get the algebraic form of Boolean control network (8) as It can be expressed into its algebraic form as

$$\begin{aligned} x(t+1) &= Lx(t)u(t) \\ y(t) &= Hx(t), \end{aligned} \tag{9}$$

where $x(t) = \ltimes_{i=1}^n x_i(t)$, $u(t) = \ltimes_{i=1}^m u_i(t)$, $y(t) = \ltimes_{i=1}^p y_i(t)$, $L \in \mathcal{L}_{2^n\times 2^{n+m}}$, and $H \in \mathcal{L}_{2^p\times 2^n}$.

For a control system people are mainly interested in the input-output mapping. Hence, a control system is commonly considered as a black box. (see in Fig. 2)

Fig. 2. A Boolean control network

Ignoring the internal variables "$x_i(x)$", we have the input-output dynamics as Then we have

$$y(t+1) = HLx(t)u(t) := \tilde{L}(t)u(t). \tag{10}$$

where

$$\tilde{L}(t) = HLx(t) \in \mathcal{L}_{2^p \times 2^m} \tag{11}$$

is the input-output transition matrix, which is a time-varying matrix.

In general, $p, m << n$. Hence, when we use input-output description, the size of the dynamics could be tremendously reduced. This is the motivation of our approach.

Consider Boolean network (3) again. Assume it is aggregated into s blocks. That is, there is a partition of states as

$$\{x_1, x_2, \cdots, x_n\} = \Sigma_1 \cup \Sigma_2 \cup \cdots \cup \Sigma_s.$$

Each block Σ_i has out-degree p_i, $i = 1, \cdots, s$, and we denote its outgoing variables by

$$y^i = \{y_1^i, y_2^i, \cdots, y_{p_i}^i\}, \quad i = 1, \cdots, s.$$

Fig. 3 depicts this aggregation.

Remark 2.1: 1) y_j^i is also a state variable. That is,

$$Y := \left\{ y_j^i \;\middle|\; i = 1, \cdots, s;\ j = 1, \cdots, p_i \right\} \subset X := \{x_1, x_2, \cdots, x_n\}.$$

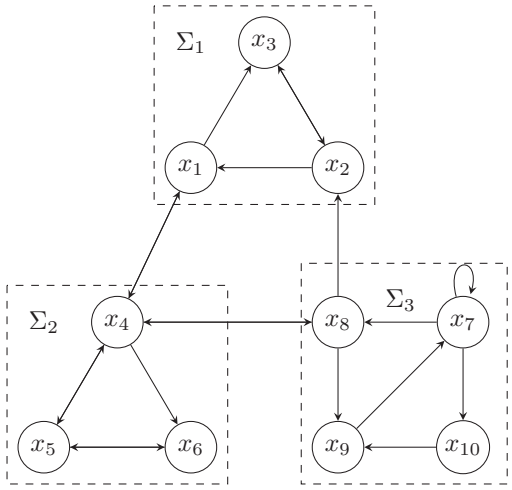

Fig. 3. Network Aggregation

2) Let $\{u_1^i, \cdots, u_{q_i}^i\}$ be the inputs to Σ_i. Since each output of a sub-system Σ_i is the input of another sub-system Σ_j and vise versa, we have

$$\left\{ u_j^i \;\middle|\; i = 1, \cdots, s;\ j = 1, \cdots, q_i \right\} = Y.$$

We call the variables in C the connecting variables. We aggregate the overall network in such a way that the number of connecting variables is as small as possible. Then we can have the dynamics about variables in C as follows.

$$\begin{cases} y^1(t+1) = \tilde{L}_1(t)u^1(t) \\ y^2(t+1) = \tilde{L}_2(t)u^2(t) \\ \vdots \\ y^s(t+1) = \tilde{L}_s(t)u^s(t), \end{cases} \tag{12}$$

where, to avoid notational mess, we also use y^i for $y^i = \ltimes_{j=1}^{p_i} y_j^i$, and y^i for $u^i = \ltimes_{j=1}^{q_i} u_j^i$. We call (12) the dynamics of the aggregated system. Since we aggregate the system in such a way that $|C| << n$, (10) has a much smaller size than the original system.

We use an example to illustrate the aggregated system.

Example 2.2: Consider the following network.

$$\begin{cases} x_1(t+1) = x_2(t) \vee x_4(t) \\ x_2(t+1) = x_3(t) \wedge x_8(t) \\ x_3(t+1) = x_1(t) \leftrightarrow x_2(t) \\ x_4(t+1) = (x_1(t) \vee x_5(t)) \rightarrow x_8(t) \\ x_5(t+1) = \neg x_4(t) \vee x_6(t) \\ x_6(t+1) = x_4(t) \wedge x_5(t) \\ x_7(t+1) = x_7(t) \bar{\vee} x_9(t) \\ x_8(t+1) = x_4(t) \leftrightarrow x_7(t)) \\ x_9(t+1) = x_8(t) \vee x_{10}(t) \\ x_{10}(t+1) = \neg x_7(t). \end{cases} \tag{13}$$

Now assume we aggregate the system into 3 subnetworks as:

$$\{x_1, x_2, x_3\} \in \Sigma_1, \quad \{x_4, x_5, x_6\} \in \Sigma_2, \quad \{x_7, x_8, x_9, x_{10}\} \in \Sigma_3$$

Now for outputs of each subsystem are

$$C = \left\{ y^1 = x_1,\ y^2 = x_4,\ y^3 = x_8 \right\}.$$

Then the system (13) can be expressed into its aggregated form as

$$\begin{cases} y^1(t+1) = \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 2]x_2(t)y^2(t)y^3(t) \\ \qquad\qquad := \tilde{L}_1 y^2(t)y^3(t) \\ y^2(t+1) = \delta_2[1\ 2\ 1\ 2\ 1\ 2\ 1\ 1]x_5(t)y^1(t)y^3(t) \\ \qquad\qquad := \tilde{L}_2 y^1(t)y^3(t) \\ y^3(t+1) = \delta_2[1\ 1\ 2\ 1]x_7(t)y^2(t) := \tilde{L}_3 y^2(t). \end{cases} \tag{14}$$

We refer to Fig. 4 for the description.

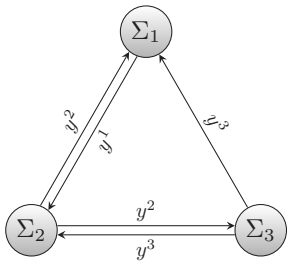

Fig. 4. Aggregated system

III. Model Approximation

Consider the dynamics (12) of the aggregated network. Now the problem is how to find the time-varying transition matrices $\tilde{L}_i$, $i = 1, \cdots, s$. We use Example 2.2 to show this.

Example 3.1: Consider Example 2.2. Choosing $X_0 = (1, 1, 0, 1, 1, 0, 1, 0, 1, 1)$, it is easy to calculate that for $T = 0, 1, 2, \cdots$, we have

$$\tilde{L}_1(t) = \begin{cases} \delta_2[1\ 1\ 1\ 1], & t = 8T,\ t = 8T+2 \\ \delta_2[1\ 1\ 1\ 2], & \text{Otherwise;} \end{cases} \tag{15}$$

$$\tilde{L}_2(t) = \begin{cases} \delta_2[1\ 2\ 1\ 2], & t = 8T,\ t = 8T+2, \\ & t = 8T+5,\ t = 8T+7 \\ \delta_2[1\ 1\ 1\ 2], & \text{Otherwise;} \end{cases} \tag{16}$$

$$\tilde{L}_3(t) = \begin{cases} \delta_2[1\ 1], & t = 8T,\ t = 8T+2, \\ & t = 8T+5,\ t = 8T+7 \\ \delta_2[2\ 1], & \text{Otherwise.} \end{cases} \tag{17}$$

In fact, since (13) has a unique attractor as

$$\begin{array}{ccccc} (1101101011) & \to & (1010010110) & \to & (0101101111) \\ \uparrow & & & & \downarrow \\ (1001101011) & & & & (1001010110) \\ \uparrow & & & & \downarrow \\ (1000010110) & \leftarrow & (0001101111) & \leftarrow & (1011010110) \end{array}$$

Then we know that for any initial state, after certain time (transient time), (15)-(17) are always true. (replacing t by $t + k_0$ for certain $k_0 > 0$.

But in general, it is very difficult, if not impossible, to find $L_i(t)$. In fact, for a large or huge network, it is even hard to know the dynamic model of the overall network. Then an approximation is necessary.

We consider a general time-varying Boolean network

$$\begin{cases} z_1(t+1) = f_1(t, z_1(t), \cdots, z_s(t)) \\ \vdots \\ z_s(t+1) = f_s(t, z_1(t), \cdots, z_s(t)), \end{cases} \tag{18}$$

where f_i, $i = 1, \cdots, s$, are time-varying logical functions.

In this sub-section, we consider to use a time-invariant model to approximate it. Consider the following model

$$\begin{cases} \hat{z}_1(t+1) = \hat{f}_1(\hat{z}_1(t), \cdots, \hat{z}_s(t)) \\ \vdots \\ \bar{z}_s(t+1) = \hat{f}_s(\hat{z}_1(t), \cdots, \hat{z}_s(t)). \end{cases} \tag{19}$$

Assume the time-varying Boolean network (18) is approximated by a time-invariant Boolean network (19).

Our purpose is to choose an approximation model, which makes the error probability as small as possible.

As aforementioned, for a large or huge network, it is very hard to know the dynamics of the overall network. So we assume we have some observed paired data as

$$D = \{(d_k, e_k) | k = 1, 2, \cdots, N\}, \tag{20}$$

where d_k and e_k are two observed data for two adjacent moments t and $t+1$ respectively. Precisely, there is a moment t such that (in vector form)

$$d_k = \ltimes_{i=1}^{s} z_i(t), \quad e_k = \ltimes_{i=1}^{s} z_i(t+1).$$

For statement ease, we denote

$$D_0 = \{d_k | (d_k, e_k) \in D\}. \tag{21}$$

Denote the algebraic form of (19) as

$$\hat{z}(t+1) = L\hat{z}(t), \tag{22}$$

where $L \in \mathcal{L}_{2^s \times 2^s}$. Then L is determined by

$$L^* = \arg \left\{ L \left| \min_{L \in \mathcal{L}_{2^s \times 2^s}} \sum_{z_t^\alpha \in D_0} \left(\left\| L z_t^\alpha \bar{\vee} z_{t+1}^\alpha \right\| \right) \right. \right\}. \tag{23}$$

In fact, a more precise approach is the component-wise approximation. Assume (19) has its component-wise algebraic form as

$$\hat{z}_i(t+1) = L_i \hat{z}(t), \quad i = 1, \cdots, s. \tag{24}$$

Then,

$$L_i^* = \arg \left\{ L_i \left| \min_{L_i \in \mathcal{L}_{2 \times 2^s}} \sum_{z_t^\alpha \in D_0} \left(\left\| L_i z_t^\alpha \bar{\vee} (z_{t+1}^\alpha)_i \right\| \right) \right. \right\}, \quad i = 1, \cdots, s. \tag{25}$$

We consider how to find L^*. (Since L_i^* can be found in the same way, we ignore it here.) Split D_0 into 2^s groups as

$$E_i = \left\{ d_k \in D_0 | d_k = \delta_{2^s}^i \right\}, \quad i = 1, \cdots, 2^s. \tag{26}$$

Assume

$$E_i \neq \emptyset, \quad i = 1, \cdots, 2^s. \tag{27}$$

If (27) is not satisfied, we say that the data D are not enough.

Split E_i into 2^s subsets as

$$E_i^j = \left\{ e_k = \delta_{2^s}^j | d_k \in E_i \right\}, \quad i, j = 1, \cdots, 2^s.$$

Then we set

$$j_i^* = \arg \left\{ j_i \left| \max_{1 \le j_i \le 2^s} \left| E_i^{j_i} \right| \right. \right\}. \tag{28}$$

Note that it is easy to see that j_i^* always exists, but it may not unique.

The following result is an immediate consequence of the construction.

Theorem 3.2: Let the structure matrix L^* of the best approximated model be as in (23) and j_i^* are calculated as in (28). Then

$$L^* = [j_1^* \ j_2^* \ \cdots \ j_{2^s}^*]. \tag{29}$$

Example 3.3: Consider the follows system

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ \vdots \\ x_{99}(t+1) = x_{100}(t) \wedge x_{101}(t) \\ x_{100}(t+1) = \neg (x_{101}(t) \wedge x_{102}(t)) \\ x_{101}(t+1) = \neg (x_{102}(t) \vee x_{103}(t)) \\ \vdots \\ x_{199}(t+1) = \neg (x_{200}(t) \vee x_{201}(t)) \\ x_{200}(t+1) = \neg (x_{201}(t) \vee x_{202}(t)) \\ x_{201}(t+1) = \neg (x_{202}(t) \to x_{203}(t)) \\ \vdots \\ x_{298}(t+1) = \neg (x_{299}(t) \to x_{300}(t)) \\ x_{299}(t+1) = x_{300}(t) \wedge x_1(t) \\ x_{300}(t+1) = \neg (x_1(t) \wedge x_2(t)). \end{cases} \tag{30}$$

Assume we aggregate it into three sub-systems as follows:

$$\begin{aligned} \Sigma_1 &= \{x_i | 1 \le i \le 100\}; \\ \Sigma_2 &= \{x_i | 101 \le i \le 200\}; \\ \Sigma_3 &= \{x_i | 201 \le i \le 300\}. \end{aligned}$$

Then the connecting elements are

$$\begin{aligned} C = \{ & y_1 = x_{101}, \ y_2 = x_{102}, \ y_3 = x_{201}, \\ & y_4 = x_{202}, \ y_5 = x_1, \ y_6 = x_2 \}. \end{aligned}$$

Setting $y^1 = y_1 y_2$, $y^2 = y_3 y_4$, $y^3 = y_5 y_6$, we have the following component-wise algebraic form as

$$\begin{cases} y^1(t+1) = L_1 y^2(t) \\ y^2(t+1) = L_2 y^3(t) \\ y^3(t+1) = L_3 y^1(t), \end{cases} \tag{31}$$

where $L_i \in \mathcal{L}_{2 \times 4}$, $i = 1, 2, 3$. Since the model (30) is known, we can choose $x_k(0)$ as the initial states arbitrary and calculate the corresponding $x_k(1)$. Then take the projection $x_k(0)|_C := d_k$ and $x_k(1)|_C := e_k$ as the observed data. Since every trajectory will enter a cycle, the data may not be enough if the cycle and the basin are not long enough. Thus, we can choose another initial state and finally add all $|E_i^j|$ of the data with different initial states.

TABLE I

OBSERVED DATA

i\$|E_i^j|$\j	1	2	3	4
1	4	2	1	17
2	775	764	360	912
3	691	800	356	930
4	3342	84876	2185	3885

a. E_i^j for L_1

i\$|E_i^j|$\j	1	2	3	4
1	0	311	312	62527
2	0	12	8	223
3	0	15	14	322
4	0	2444	2421	31291

b. E_i^j for L_2

i\$|E_i^j|$\j	1	2	3	4
1	307	7	5	4515
2	62303	205	302	23557
3	303	5	3	2619
4	313	7	6	5443

c.E_i^j for L_3

In this example, we choose the initial states randomly for 100 times, each time we observe 1000 data. The observed data is as in Table 1.

Thus, L_1, L_2, and L_3 can be determined as

$$\begin{cases} L_1 = \delta_4[4\ 4\ 4\ 2] \\ L_2 = \delta_4[4\ 4\ 4\ 4] \\ L_3 = \delta_4[4\ 1\ 4\ 4]. \end{cases}$$

IV. CONCLUSION

In this paper, we considered an aggregation-based approximation to large-scale Boolean network. If the network is aggregated to several sub-networks, we obtain a simplified network using the outputs(or inputs) of each sub-network as new state variables. However, it is difficult to find the precise expression of the simplified network. Then a time-invariant Boolean network is used to approximate the network, using the observed data. Finally, an aggregation method which maximizes the modularity is reviewed.

The results in this paper is very elementary and heuristic. But it might hint a very challenging and meaningful method for reducing the computation of Boolean network to make it applicable.

REFERENCES

[1] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," J. Theoretical Biology, vol. 22, pp. 437-467, 1969.

[2] S. Huang and I. Ingber, "Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks," Exp. Cell Res., vol. 261, pp. 91-103, 2000.
[3] S. Huang, "Regulation of cellular states in mammalian cells from a genomewide view," in Gene Regulation and Metabolism, J. Collado-Vodes and R. Hofestadt, Eds. Cambridge, MA: MIT Press, 181-220, 2002.
[4] E.H. Davidson, J.P. Rast, P. Oliveri, A. Ransick, C. Calestani and C.H. Yuh, et al., "A genomic regulatory network for development," Science, vol. 295, pp. 1669-1678, 2002.
[5] R. Albert and H.G. Othmer, "The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drospphila melanogaster," J. Theoretical Biology, vol. 223, no. 1, pp. 1-18, 2003.
[6] C. Farrow, J. Heidel, H. Maloney and J. Rogers, "Scalar equations for synchronous Boolean networks with biological applications," IEEE Trans. Neural Networks, vol. 15, no. 2, pp. 348-354, 2004.
[7] Q. Zhao, "A remark on 'Scalar Equations for synchronous Boolean Networks with biologicapplications' by C. Farrow, J. Heidel, J. Maloney, and J. Rogers," IEEE Trans. Neural Networks, vol. 16, no. 6, pp. 1715-1716, 2005.
[8] F. Li, T. Long, Y. Lu, Q. Ouyang and C. Tang, "The yeast cell-cycle network in robustly designed," PNAS, vol. 101, no. 14, pp. 4781-4786, 2004.
[9] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," IEEE Trans. Aut. Contr., vol. 55, no. 10, pp. 2251-2258, 2010.
[10] D. Cheng, H. Qi and Z. Li, Analysis and Control of Boolean Networks – A Semi-tensor Product Approach, London: Springer, 2011.
[11] D. Cheng, "Semi-tensor Product of Matrices and its Applications - A Survey," in Proc. ICCM 2007, vol. 3, pp. 641-668, 2007.
[12] L. Ljung and T. So'derstro'm, Theory and Practice of Recursive Identification, MA: MIT Press, 2000.
[13] M.E.J. Newman, "Modularity and community structure in networks," PNAS, vol. 103, no. 23, pp. 8577-8582, 2006.
[14] E.A. Leicht and M.E.J. Newman, "Community structure in directed metworks," Physical Review Letters, vol.100, pp. 118703, 2008.

APPENDIX

We review the method proposed in [13], [14] to split a directed network into two subnetworks: Assume the network consists of n nodes. Let A be the adjacent matrix, k_i^{in} the in-degree of i-th node, k_i^{out} the out-degree of i-th node, $s = [s_1, \cdots, s_n]$, where $s_i \in \{-1.1\}$, which corresponds a partition as:

$$G_1 = \{i | s_i = 1\}, \quad \text{and} \quad G_2 = \{i | s_i = -1\}.$$

Set

$$m := \sum_{i=1}^{n} k_i^{\text{out}} = \sum_{i=1}^{n} k_i^{\text{in}} = \sum_{i,j} A_{ij}.$$

Then we define the modularity as

$$\begin{aligned} Q(s) &= \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right) (s_i s_j + 1) \\ &= \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right) s_i s_j. \end{aligned} \tag{32}$$

The method proposed by [14] is to find the best s which maximize Q. Denote

$$B_{ij} = A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m}. \tag{33}$$

Then (32) can be expressed in a quadratic form as

$$Q(s) = s^T B s. \tag{34}$$

Since the graph is directed, B is not symmetric. However, to maximize (34) is equivalent to maximize

$$Q(s) = s^T (B + B^T) s, \tag{35}$$

and $B + B^T$ is manifestly symmetric, thus all of its eigenvalues are real.

To find the best split, we need to find the maximum eigenvalue μ_m of $s^T(B + B^T)s$. If $\mu_m > 0$, we can find its corresponding eigenvector u. Then set

$$s_i = \begin{cases} 1, & u_i \geq 0 \\ -1, & u_i < 0. \end{cases}$$

(When $\mu_m \leq 0$, the network is indivisible.) For the second division for sub-network G, we need to consider

$$\begin{aligned} \Delta Q &= \frac{1}{2m} \sum_{i,j \in G} \left[(B_{i,j} + B_{j,i}) - \delta_{ij} \sum_{k \in G} (B_{i,k} + B_{k,i}) \right] s_i s_j \\ &:= \frac{1}{2m} s^T (B^{(G)} + B^{(G)T}) s, \end{aligned} \tag{36}$$

where

$$B_{i,j}^{(G)} = B_{i,j} - \frac{1}{2} \delta_{ij} \sum_{k \in G} (B_{i,k} + B_{k,i}).$$

1. Approximation of boolean networks

Accession number: 20130415919393
Authors: Cheng, Daizhan (1); Zhao, Yin (1); Kim, Jongrae (2); Zhao, Yunbo (2)
Author affiliation: (1) Key Lab. of Systems and Control, Institute of Systems Science, Chinese Academy of Sciences, Beijing 100190, China; (2) Division of Biomedical Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom
Corresponding author: Cheng, D.(dcheng@iss.ac.cn)
Source title: Proceedings of the World Congress on Intelligent Control and Automation (WCICA)
Abbreviated source title: Proc. World Congr. Intelligent Control Autom. WCICA
Monograph title: WCICA 2012 - Proceedings of the 10th World Congress on Intelligent Control and Automation
Issue date: 2012
Publication year: 2012
Pages: 2280-2285
Article number: 6358254
Language: English
ISBN-13: 9781467313988
Document type: Conference article (CA)
Conference name: 10th World Congress on Intelligent Control and Automation, WCICA 2012
Conference date: July 6, 2012 - July 8, 2012
Conference location: Beijing, China
Conference code: 94938
Sponsor: Academy of Mathematics and Systems Science; IEEE Robotics and Automation Society; IEEE Control Systems Society; National Natural Science Foundation of China; Chinese Association of Automation
Publisher: Institute of Electrical and Electronics Engineers Inc., 445 Hoes Lane / P.O. Box 1331, Piscataway, NJ 08855-1331, United States
Abstract: The problem of approximation to large-scale Boolean networks is considered. First, we assume a large-scale Boolean network is aggregated into several sub-networks. Using the outputs(or inputs) of each sub-network as new state variables, a new simplified time-varying network is obtained. Then a time-invariant Boolean network is used to approximate each subsystem. Observed data are used to find the best approximating dynamic models. Finally, the aggregation method is investigated. © 2012 IEEE.
Number of references: 14
Main heading: Boolean algebra
Controlled terms: Agglomeration - Intelligent control
Uncontrolled terms: Aggregation methods - Approximation - Boolean Networks - Modularity - Observed data - State variables - Sub-network - Subnetworks - Time invariants
Classification code: 721.1 Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory - 731.1 Control Systems - 802.3 Chemical Operations
DOI: 10.1109/WCICA.2012.6358254
Database: Compendex

Data Provider: Engineering Village