Optimal Reconstruction of Noisy Dynamics and Selection Probabilities in Boolean Networks ⋆

Koichi Kobayashi [a], Yuhu Wu [b,⋆⋆],

[a] *Faculty of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan,*

[b] *Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, and School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China*

Abstract

In the analysis and control of complex systems, including gene regulatory networks, it is important to reconstruct a mathematical model from a priori information and noisy experimental data. A Boolean network (BN) is well known as a mathematical model of gene regulatory networks. Each state of BNs takes a binary value (0 or 1), and its update rule is given by a set of Boolean functions. In this paper, we consider the optimal reconstruction problem of finding a probabilistic BN consisting of the main dynamics and the noisy dynamics, by giving the main dynamics and the sample mean of the state obtained from noisy experimental data. In the proposed method, the selection probability of the main dynamics is maximized. We show that the optimal Boolean function of the noisy dynamics is a constant (0 or 1) map under no assumption on the structure of noisy dynamics. Finally, as a biological application, the reconstruction of a PBN model of the *lac* operon networks of *Escherichia* coli bacterium is addressed using the proposed approach.

Key words: Boolean network, Noisy dynamics, Optimal reconstruction, Gene regulatory network, Systems biology

1 Introduction

Modeling, analysis, and control of complex systems with nonlinear dynamics and complex constraints, such as gene regulatory networks, have attracted much attention. In such cases, it is important to simplify the relevant mathematical model. In the field of systems biology, the Boolean network (BN) model, which is a discrete-time discrete model, is frequently used (see, e.g., Akutsu (2018); Imani and Braga-Neto (2018); Kim et al. (2020)).

In a BN, each state takes a binary value (0 or 1). The time evolution of each state is expressed as Boolean functions. Many studies related to the analysis and control of BNs have been conducted (see, e.g., (Cheng et al., 2011a; Fornasini and Valcher, 2013; Yerudkar et al., 2020; Weiss et al., 2018; Zhong et al., 2020; Li et al., 2020a; Wu et al., 2021; Zhu et al., 2020)). As an extended model of BNs, a probabilistic BN (PBN) model is well known. In a PBN, the update rule for each state is randomly chosen from the candidates of Boolean functions (see, e.g., Shmulevich et al. (2002); Meng et al. (2018)). Owing to the noise present in gene regulatory networks, it is appropriate to use a PBN. In recent years, the analysis and synthesis of PBNs have been extensively investigated and exploited (see, e.g., Kobayashi and Hiraishi (2011); Trairatphisan et al. (2013); Toyoda and Wu (2021); Acernese et al. (2020)).

One important problem in data science is the extraction of information (e.g., the structure and the dynamics) from the experimental data. For BNs and PBNs, data-driven methods have been obtained (see, e.g., Cheng et al. (2011b); Akutsu and Melkman (2018); Melkman et al. (2017); Sun et al. (2020)). In data-driven methods, the intrinsic noise in the experimental data must be considered. From this viewpoint, the noisy dynamics has been studied (see, e.g., Ching and Tam (2017); Freilich et al. (2020)). A kind of parameterized set was constructed in Li et al. (2020b), for the robust stability and stabilization of Boolean networks with function perturbation.

In this paper, we focus on the reconstruction problem of the noisy dynamics in PBNs, when the main dynamics and the sample mean of the state, by the data driven approach, are given. The main dynamics is given based on a

⋆ This work was supported in part by JSPS KAKENHI Grant Numbers JP17K06486, JP19H02157, JP19H02158, JP21H04558 and the National Natural Science Foundation of China under Grant 61773090, 62173062. The material in this paper was not presented at any conference.

⋆⋆Corresponding author

Email addresses: k-kobaya@ssi.ist.hokudai.ac.jp (Koichi Kobayashi), wuyuhu@dlut.edu.cn (Yuhu Wu).

priori information. The sample mean of the state is generated from noisy experimental data (see Section 2 for further details). A previous study has proposed methods in the linear programming formulation that could solve the aforementioned problem (Kobayashi and Hiraishi, 2016; Umiji et al., 2019). In those works, the PBN was constructed as a combination of BNs with a certain selecting probability distribution. The candidates of update dynamics of PBN include the main dynamics, and the noisy dynamics, which is represented by multiple Boolean functions. Hence, the expression of noisy dynamics was complicated. Furthermore, the linear programming approach given in (Kobayashi and Hiraishi, 2016; Umiji et al., 2019) requires that the network structure of the noisy dynamics must be given in advance. In the approach, proposed in this work, the prior information about the network structure of the noisy dynamics is not required compared to the previous approach. Our approach is optimal owing to the following reasons: first, the proposed solution guarantees the maximum choosing probability of the main dynamics; second, it is proven that the noisy dynamics is equivalently represented as a constant mapping (0 or 1), which is a specific characteristic of perturbed BNs, and not available for general continuous or discrete dynamical systems. Although PBNs obtained by two existing methods (Kobayashi and Hiraishi, 2016; Umiji et al., 2019) and the proposed method are equivalent in the sense that the same main dynamics and the same sample mean of the state are used, the PBN obtained by the proposed method is optimal owing to the above two reasons.

To overcome the difficulty caused by the lack of network structure information of the noisy dynamics, the full representation structure matrix (FRSM) of the Boolean function is introduced. An algorithm for deriving the noisy dynamics and estimating the selection probability of the candidate dynamics is also proposed based on the minimal representation structure matrix (MRSM). The proposed method is demonstrated using a simple example and a *Lac* operon networks application (Veliz-Cuba and Stigler, 2011; Chen et al., 2018).

The main contributions of this paper are summarized as follows:

i) The optimal solution for the reconstruction problem of the noisy dynamics in PBNs is solved. We show that the noisy dynamics are equivalently represented as constant (0 or 1) maps under no assumption on network structure of noisy dynamics.

ii) A computationally efficient algorithm for optimal reconstruction of the noisy dynamics is proposed based on the network structure of the main dynamics.

This paper is organized as follows. In Section 2, a PBN and the problem studied are explained. In Section 3, the MRSM and FRSM are introduced, and the noisy dynamics is investigated. In Section 4, the proposed method is applied to a *lac* operon network. In Section 5, we provide the conclusion of this study.

Notation: Let $\mathcal{R}$ denote the set of real numbers. Let $\{0,1\}^n$ denote the set of n-dimensional vectors whose elements take either 0 or 1. For the matrix M, let $\mathrm{Col}_i(M)$ and $\mathrm{Row}_i(M)$ denote the i-th column and row of M, respectively. For the vector x, let $[x]_i$ denote the i-th element of x. Let I_n denote the n-dimensional identity matrix. For two matrices $A \in \mathcal{R}^{m\times n}$ and $B \in \mathcal{R}^{p\times q}$, the semi-tensor product (STP) of A and B is defined by $A \ltimes B = (A \otimes I_{c/n})(B \otimes I_{c/p})$, where $\otimes$ is the Kronecker product, and c is the least common multiple of n and p. For r vectors $y_1, y_2, \ldots, y_r$ and an ordered index set $\mathcal{J} = \{j_1, j_2, \ldots, j_s\} \subseteq \{1, 2, \ldots, r\}$, with $j_1 < j_2 < \cdots < j_s$ define $\ltimes_{j\in\mathcal{J}} y_j := y_{j_1} \ltimes y_{j_2} \ltimes \cdots \ltimes y_{j_s}$. For example, for r two-dimensional vectors $y_1, y_2, \ldots, y_r$ and $\mathcal{J} = \{1, 5\}$, we can obtain $\ltimes_{j\in\mathcal{J}} y_j = y_1 \ltimes y_5 = [[y_1]_1[y_5]_1, [y_1]_1[y_5]_2, [y_1]_2[y_5]_1, [y_1]_2[y_5]_2]^\top$.

The i-th column of I_n is denoted by δ_n^i. The set of δ_n^i is denoted by $\Delta_n = \{\delta_n^i, i = 1, 2, \ldots, n\}$. The matrix $A \in \{0,1\}^{m\times n}$ is called a logical matrix if each column of A is given by some element of Δ_m. The set of $m \times n$ logical matrices is denoted by $\mathcal{L}^{m\times n}$. The matrix $L = [\delta_n^{i_1}, \delta_n^{i_2}, \ldots, \delta_n^{i_s}]$ is simply denoted by $L = \delta_n[i_1, i_2, \ldots, i_s]$. Let $\mathbf{1}_n$ denote the n-dimensional column vector $[1, 1, \ldots, 1]^\top$.

2 Preliminaries and Problem Formulation

A Boolean network (BN), which is a directed network containing binary (Boolean) logical-valued state nodes $\mathcal{N} = \{1, 2, \ldots, n\}$, can be represented by

$$x_i(k+1) = f^{(i)}([x_j(k)]_{j\in\mathcal{N}^{(i)}}), \quad i \in \mathcal{N}, \tag{1}$$

where $x := [x_1, x_2, \ldots, x_n]^\top \in \{0,1\}^n$ is the state, and $k \geq 0$ is the update time. The set $\mathcal{N}^{(i)} \subseteq \mathcal{N}$ is the indegree index of the node i, and the update function $f^{(i)} : \{0,1\}^{|\mathcal{N}^{(i)}|} \to \{0,1\}$ is a Boolean function consisting of logical operators such as AND ($\wedge$), OR ($\vee$), and NOT ($\neg$) with minimal representation. If $x_i(k+1)$ is uniquely determined as 0 or 1, then $\mathcal{N}^{(i)} = \emptyset$.

Next, we explain a probabilistic Boolean network (PBN) (see, e.g., Shmulevich et al. (2002) for further details). In a PBN, the update rule of state x_i is regulated by one Boolean function, which is randomly selected from a set of Boolean functions. The dynamics of PBN is described as

$$x_i(k+1) = f^{(i)}(x(k)), \quad i \in \mathcal{N}, \tag{2}$$

with a probability of $f^{(i)}$ choosing $f_l^{(i)}$ is

$$c_l^{(i)} := \Pr\{f^{(i)} = f_l^{(i)}\}, \tag{3}$$

where

$$f_l^{(i)}\left([x_j(k)]_{j\in\mathcal{N}_l^{(i)}}\right), \quad l = 0, 1, \ldots, q(i), \tag{4}$$

denote the candidates of $f^{(i)}$. Then, $\sum_{l=0}^{q(i)} c_l^{(i)} = 1$ must be satisfied. Notice that if $q(i) = 0$ for each $i \in \mathcal{N}$, then the PBN (2)–(4) degenerate into a deterministic BN (1). Finally, for PBN (2)–(4), the in-degree index $\mathcal{N}^{(i)}$ of node $i \in \mathcal{N}$ is defined by $\mathcal{N}^{(i)} := \bigcup_{l=0}^{q(i)} \mathcal{N}_l^{(i)}$.

Complex systems, such as gene regulatory networks, are affected by noises. Therefore, we must consider both the main dynamics and the noisy dynamics. The main dynamics is given based on a priori information, and is dominant in a given system. Assume that the main dynamics is expressed by the following BN:

$$x_i(k+1) = f_0^{(i)}([x_j(k)]_{j\in\mathcal{N}^{(i)}}), \quad i \in \mathcal{N}. \tag{5}$$

To model the effect of the noise, we then introduce the noisy dynamics. Suppose that the noisy dynamics is represented by other BNs with some probability distribution, but is not given in advance. The noisy dynamics is derived from the main dynamics and the sample mean of the state. The sample mean of the current state, and the sample mean of the next state are denoted by $s_k = \left[s_k^{(1)}, s_k^{(2)}, \ldots, s_k^{(n)}\right]^\top$, and $s_{k+1} = \left[s_{k+1}^{(1)}, s_{k+1}^{(2)}, \ldots, s_{k+1}^{(n)}\right]^\top$, respectively, which are generated in the following way.

We sample (or observe) the BN process for enough large N times, and at each observing time (step) $k_1, k_2, \ldots, k_N$, with $k_i \le k_{i+1}, i = 1, 2, \ldots, N-1$, we collect a sequential pair observed state values, i.e., the current state $x(k_i) = [x_1(k_i), x_2(k_i), \ldots, x_n(k_i)]^\top$ and the next state $x(k_i+1) = [x_1(k_i+1), x_2(k_i+1), \ldots, x_n(k_i+1)]^\top$ of the BN. In this case, the sample mean $s_k^{(i)}$ of the i-th component of the current state, and the sample mean $s_{k+1}^{(i)}$ of the i-th component of the next state are respectively defined as [1].

$$s_k^{(i)} = \frac{1}{N}\sum_{j=1}^{N} x_i(k_j), \quad s_{k+1}^{(i)} = \frac{1}{N}\sum_{j=1}^{N} x_i(k_j+1), \quad i \in \mathcal{N}.$$

Thus, $s_k^{(i)}$ and $s_{k+1}^{(i)}$ are derived based on the long-time behavior, where subscript $k+1$ of $s_{k+1}^{(i)}$ mainly stresses that $s_{k+1}^{(i)}$ is the next state corresponding to the current state $s_k^{(i)}$. The values of $s_k^{(i)}$ and $s_{k+1}^{(i)}$ are not changed over time.

From the Law of Large Numbers, a fundamental result in statistics, the sample mean converges to the expected value when the sample size is sufficiently large (see, e.g, Ibe (2013)). Based on this fact, we assume $E[x_i(k)] = s_k^{(i)}$ and $E[x_i(k+1)] = s_{k+1}^{(i)}$ in our problem settings.

[1] In the case of the successive sampling case, i.e, continuously picking up the sample from an observed time series, one can deduce that $s_k^{(i)} \approx s_{k+1}^{(i)}$, with enough sufficient sampling. In the general case, we cannot guarantee that $s_k^{(i)} \approx s_{k+1}^{(i)}$.

Based on the above discussion, we formulate the optimal reconstruction problem of a BN with noisy dynamics. First, the following two assumptions are formally made for the system.

Assumption 2.1 *The main dynamics* (5) *is given.*

Assumption 2.2 *The sample means of the current state and the next state, i.e., $s_k^{(i)}$ and $s_{k+1}^{(i)}$ are given, respectively.*

Next, the optimal reconstruction problem is given as follows.

Problem 1 *Consider a BN with noisy dynamics given by a PBN. Under Assumption 2.1 and Assumption 2.2, for each $i \in \mathcal{N}$, find*

(i) *the number of candidates $q(i)$,*

(ii) *probabilities $c_l^{(i)}$, $l = 0, 1, \ldots, q(i)$,*

(iii) *Boolean functions $f_1^{(i)}, f_2^{(i)}, \ldots, f_{q(i)}^{(i)}$*

maximizing $c_0^{(i)}$ subject to $E[x_i(k+1)] = s_{k+1}^{(i)}$, $E[x_i(k)] = s_k^{(i)}$ and $c_0^{(i)} + \sum_{l=1}^{q(i)} c_l^{(i)} = 1$.

Similar to previous work such as (Kobayashi and Hiraishi, 2016; Umiji et al., 2019), $q(i)$ is not given, and is a decision variable. In other words, $q(i)$ cannot be specified in advance. In the above problem, the main dynamics and the noisy dynamics are separated. As it is desirable for the selection probability of the main dynamics to be higher, we consider maximizing $c_0^{(i)}$. In this sense, we call the optimal reconstruction problem. Moreover, we remark that there is no assumption for the noisy dynamics.

3 Main Results

First, we explain the outline of the matrix representation for PBNs (see, e.g., Li and Sun (2011); Toyoda and Wu (2021)), where the STP of the matrices is used. For any $x \in [0, 1]$, we define its corresponding vector expression as

$$\overline{x} := \begin{bmatrix} x \\ 1-x \end{bmatrix}. \tag{6}$$

Based on the above definition, the logical vectors δ_2^1, and δ_2^2 represent Boolean values 1, and 0, respectively. In addition, under expression (6), according to Theorem 3.2 of Cheng et al. (2011a), which points out that a logical function can be represented in an algebraic form, we have the following fact.

Lemma 3.1 *For any given logical function in the minimal representation form*

$$y = f([x_j]_{j\in\mathcal{M}}), \tag{7}$$

with $[x_j(k)]_{j\in\mathcal{M}} \in \{0,1\}^{|\mathcal{M}|}$, $\mathcal{M} \subset \mathcal{N}$, and $y \in \{0,1\}$, there exist logical matrices $A \in \mathcal{L}^{2\times 2^{|\mathcal{M}|}}$, and $\overline{A} \in \mathcal{L}^{2\times 2^n}$ such that (7) *can be rewritten in the following two multilinear forms as*

$$\overline{y} = A\left(\ltimes_{j\in\mathcal{M}} \overline{x}_j\right), \tag{8}$$

and

$$\overline{y} = \overline{A}\left(\ltimes_{j\in\mathcal{N}}\overline{x}_j\right). \qquad (9)$$

Matrix $A \in \mathcal{L}^{2\times 2^{|\mathcal{M}|}}$ in (8), and $\overline{A} \in \mathcal{L}^{2\times 2^n}$ in (9) are called the minimal representation structure matrix (MRSM), and the full representation structure matrix (FRSM) of the logical equation (7), respectively. Notice that the MRSM and FRSM of the logical function f are uniquely determined by each other. Specially we can give the following property by a simple calculation based on the STP technique inspired by Lemma 16 of Cheng (2014).

Proposition 3.1 *The* MRSM A*, and the* FRSM $\overline{A}$ *of the logical function* (7) *satisfy the following relations: if* $\mathcal{M} = \mathcal{N}$*, then* $\overline{A} = A$*, and if* $\mathcal{M} \neq \mathcal{N}$*, then*

$$\overline{A} = \begin{cases} A\left(\ltimes_{t\in\overline{\mathcal{M}}} D_r[2^{t-1},2]\right), & \text{if } 1 \notin \overline{\mathcal{M}}, \\ AD_f[2,2]\left(\ltimes_{t\in\overline{\mathcal{M}}} D_r[2^{t-1},2]\right), & \text{if } 1 \in \overline{\mathcal{M}}, \end{cases} \qquad (10)$$

where the dummy state set $\overline{\mathcal{M}}$ *of* f *is defined as the residual set* $\overline{\mathcal{M}} := \mathcal{N} \setminus \mathcal{M}$*, and* $D_f[p,q] = \mathbf{1}_p^\top \otimes I_q$ *and* $D_r[p,q] = I_p \otimes \mathbf{1}_q^\top$ *are dummy matrices.*

By this proposition, the relation between the MRSM and the FRSM is clarified. This proposition is used in the proof of Theorem 3.2.

Define the probability of the state $x_i(k)$ choosing 1 as

$$p_i(k) := \Pr\{x_i(k) = 1\}, \quad \forall i \in \mathcal{N}, \quad k \geq 0. \qquad (11)$$

Then, the evolution dynamics of the distribution $p(k) = [p_1(k), p_2(k), \ldots, p_n(k)]^\top$ of $x(k)$ is given by the following fact.

Lemma 3.2 *For a given PBN* (2)–(4)*, the distribution* $p(k)$ *of* $x(k)$ *satisfies the following equation:*

$$p_i(k+1) = \left(\delta_2^1\right)^\top \sum_{l=0}^{q(i)} c_l^{(i)} \overline{A}_l^{(i)} (\ltimes_{j\in\mathcal{N}}\overline{p}_j(k)), \qquad (12)$$

where $\overline{A}_l^{(i)} \in \mathcal{L}^{2\times 2^n}$ *is the FRSM of* $f_l^{(i)}$*,* $l = 0, 1, \ldots, q(i)$*,* $i \in \mathcal{N}$*.*

Proof. According to definition (11), we have the following equation:

$$p_i(k+1) = \sum_{j=1}^{2^n} P_j^{(i)} \Pr\{\overline{x}(k) = \delta_{2^n}^j\}, \qquad (13)$$

which follows from the law of total probability, where $P_j^{(i)}$ denotes the conditional probability of $x_i(k+1)$ given $x(k)$ as $P_j^{(i)} = \Pr\left\{x_i(k+1) = 1 \middle| \overline{x}(k) = \delta_{2^n}^j\right\}$. For any $i \in \mathcal{N}$, $j = 1, 2, \ldots, 2^n$, define

$$\Omega_j^{(i)} := \left\{l = 0, 1, \ldots, q(i) \middle| \delta_2^1 = \overline{A}_l^{(i)} \delta_{2^n}^j\right\}. \qquad (14)$$

Then $P_j^{(i)} = \sum_{l\in\Omega_j^{(i)}} c_l^{(i)}$. Recalling that $\overline{A}_l^{(i)} \in \mathcal{L}^{2\times 2^n}$, we get

$$\begin{cases} \delta_2^1 = \overline{A}_l^{(i)} \delta_{2^n}^j \iff (\delta_2^1)^\top \overline{A}_l^{(i)} \delta_{2^n}^j = 1, \\ \delta_2^1 \neq \overline{A}_l^{(i)} \delta_{2^n}^j \iff (\delta_2^1)^\top \overline{A}_l^{(i)} \delta_{2^n}^j = 0, \end{cases}$$

for each $i \in \mathcal{N}$, $l = 0, 1, \ldots, q(i)$, and $j = 1, 2, \ldots, 2^n$, which implies

$$P_j^{(i)} = \sum_{l\in\Omega_j^{(i)}} c_l^{(i)} = \sum_{l=1}^{q(i)} c_l^{(i)} (\delta_2^1)^\top \overline{A}_l^{(i)} \delta_{2^n}^j. \qquad (15)$$

By (11), we have $\overline{p}_i(k) = \left[\Pr\{\overline{x}_i(k) = \delta_2^1\}, \Pr\{\overline{x}_i(k) = \delta_2^2\}\right]^\top$, which implies that for any $j = 1, 2, \ldots, 2^n$,

$$\Pr\{\overline{x}(k) = \delta_{2^n}^j\} = (\delta_{2^n}^j)^\top \ltimes_{j\in\mathcal{N}} \overline{p}_j(k). \qquad (16)$$

Substituting the inequalities (15) and (16) into (13), we have

$$\begin{aligned} p_i(k+1) &= \sum_{j=1}^{2^n} \sum_{l=0}^{q(i)} c_l^{(i)} (\delta_2^1)^\top \overline{A}_l^{(i)} \delta_{2^n}^j (\delta_{2^n}^j)^\top \ltimes_{j\in\mathcal{N}} \overline{p}_j(k) \\ &= (\delta_2^1)^\top \sum_{l=0}^{q(i)} c_l^{(i)} \overline{A}_l^{(i)} \sum_{j=1}^{2^n} \delta_{2^n}^j (\delta_{2^n}^j)^\top \ltimes_{j\in\mathcal{N}} \overline{p}_j(k) \\ &= (\delta_2^1)^\top \sum_{l=0}^{q(i)} c_l^{(i)} \overline{A}_l^{(i)} \ltimes_{j\in\mathcal{N}} \overline{p}_j(k), \end{aligned}$$

by noticing $\sum_{j=1}^{2^n} \delta_{2^n}^j (\delta_{2^n}^j)^\top = I_{2^n}$, and complete the proof of Lemma 3.2. □

For any $i \in \mathcal{N}$, and $k \geq 0$, by observing

$$E[\overline{x}_i(k)] = \delta_2^1 p_i(k) + \delta_2^2 (1 - p_i(k)) = \overline{p}_i(k),$$

the expected value $E[\overline{x}_i(k+1)]$ of the next state is given by the following result, as a direct consequence of Lemma 3.2.

Corollary 3.1 *The evolution dynamics of expectation of state* $x(k)$ *of PBN* (2)–(4) *can be expressed by the following equation, for any* $i \in \mathcal{N}$*, and* $k > 0$*,*

$$E[\overline{x}_i(k+1)] = \sum_{l=0}^{q(i)} c_l^{(i)} \overline{A}_l^{(i)} \left(\ltimes_{j\in\mathcal{N}} E[\overline{x}_j(k)]\right). \qquad (17)$$

This representation is used in the proof of Theorem 3.1.

To further characterize the structure of logical matrix $A \in \mathcal{L}^{2\times 2^n}$, define index set $\mathcal{I}(A)$ for A as

$$\mathcal{I}(A) := \left\{i \in \{1, 2, \ldots, 2^n\} \middle| \mathrm{Col}_i(A) = \delta_2^1\right\}. \qquad (18)$$

Lemma 3.3 *For any subdynamics $x_i(k+1) = f^{(i)}([x_j(k)]_{j\in\mathcal{N}^{(i)}})$ of node i with FRSM $\overline{A}^{(i)}$,*

$$\mathcal{I}(\overline{A}^{(i)}) = \emptyset \Leftrightarrow f^{(i)} \equiv 0 \text{ and } \mathcal{N}^{(i)} = \emptyset, \quad (19)$$

$$\mathcal{I}(\overline{A}^{(i)}) = \{1, 2, \dots, 2^n\} \Leftrightarrow f^{(i)} \equiv 1 \text{ and } \mathcal{N}^{(i)} = \emptyset. \quad (20)$$

From this lemma, we can determine if f is a constant mapping using index set $\mathcal{I}(A)$. This lemma is used in the proof of Theorem 3.1.

Here, we present a simple example to illustrate the evolution dynamics (17) and index set $\mathcal{I}(A)$.

Example 3.1 *Consider a PBN with two states ($\mathcal{N} = \{1, 2\}$). Boolean functions and their selection probabilities are given as follows:*

$$f^{(1)} = \begin{cases} f_1^{(1)} = x_1(k) \wedge \neg x_2(k), & c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_1(k), & c_2^{(1)} = 0.2, \end{cases}$$

$$f^{(2)} = \begin{cases} f_1^{(2)} = x_1(k) \vee x_2(k), & c_1^{(2)} = 0.5, \\ f_2^{(2)} = x_1(k), & c_2^{(2)} = 0.3, \\ f_3^{(2)} = \neg x_2(k), & c_3^{(2)} = 0.2, \end{cases}$$

where $\mathcal{N}^{(1)} = \mathcal{N}^{(2)} = \{1, 2\}$, $q(1) = 1$, and $q(2) = 2$. Using the truth table, we derive the matrix representation for each Boolean function. As an example, consider the Boolean function $f_1^{(1)}$. From the truth table, we can obtain $f_1^{(1)} = 0$ if $(x_1, x_2) = (0, 0), (0, 1), (1, 1)$ and $f_1^{(1)} = 1$ if $(x_1, x_2) = (1, 0)$. Then, we can obtain MRSM $A_1^{(1)} = \delta_2[2, 1, 2, 2]$, with $\mathcal{I}(\overline{A}_1^{(1)}) = \{1\}$, where the first row of matrix $A_1^{(1)}$, $[0, 1, 0, 0]$, corresponds to the truth table of $f_1^{(1)}$. In this case, the MRSM and the FRSM are the same, i.e., $A_1^{(1)} = \overline{A}_1^{(1)}$ holds. Next, consider the Boolean function $f_2^{(1)}$. From the truth table, we can obtain MRSM $A_2^{(1)} = \delta_2[2, 1]$. Noting $\overline{\mathcal{M}} = \{2\}$, from Proposition 3.1, we can obtain the FRSM as follows:

$$\overline{A}_2^{(1)} = A_2^{(1)}(I_2 \otimes \mathbf{1}_2^\top) = \delta_2[2, 2, 1, 1],$$

with $\mathcal{I}(\overline{A}_2^{(1)}) = \{3, 4\}$. Similarly, we can obtain FRSM $\overline{A}_1^{(2)} = \delta_2[1, 1, 1, 2]$, $\overline{A}_2^{(2)} = \delta_2[1, 1, 2, 2]$, and $\overline{A}_3^{(2)} = \delta_2[2, 1, 2, 1]$. From Corollary 3.1, we can obtain the following matrix representation of the PBN:

$$E[\overline{x}_1(k+1)] = \begin{bmatrix} 0 & 0.8 & 0.2 & 0.2 \\ 1 & 0.2 & 0.8 & 0.8 \end{bmatrix} \left(\ltimes_{j\in\mathcal{N}} E[\overline{x}_j(k)]\right),$$

$$E[\overline{x}_2(k+1)] = \begin{bmatrix} 0.8 & 1 & 0.5 & 0.2 \\ 0.2 & 0 & 0.5 & 0.8 \end{bmatrix} \left(\ltimes_{j\in\mathcal{N}} E[\overline{x}_j(k)]\right).$$

For the given sample mean $s_k^{(j)}$ of the state x_i, $i \in \mathcal{N}$, define

$$Q_{\mathcal{N}} := \ltimes_{j\in\mathcal{N}} \overline{s}_k^{(j)}. \quad (21)$$

Then, noticing that $\overline{s}_k^{(j)} = [s_k^{(j)}, 1 - s_k^{(j)}]^\top$, and $s_k^{(j)} \in [0, 1]$ for each node $i \in \mathcal{N}$, we easily deduce that $\sum_{j=1}^{2^n} [Q_{\mathcal{N}}]_j = 1$, and $[Q_{\mathcal{N}}]_j \geq 0$, for any $j = 1, \dots, 2^n$. Thus, we have the following theorem as the main result.

Theorem 3.1 *A solution of Problem 1 is given by the following statements. For any $i \in \mathcal{N}$,*

(i) *If $s_{k+1}^{(i)} = \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j$, then $q(i) = 0$, with $c_0^{(i)} = 1$;*

(ii) *If $s_{k+1}^{(i)} < \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j$, then $q(i) = 1$, and $f_1^{(i)} \equiv 0$ with*

$$c_0^{(i)} = \left(\sum\nolimits_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j\right)^{-1} s_{k+1}^{(i)}; \quad (22)$$

(iii) *If $s_{k+1}^{(i)} > \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j$, then $q(i) = 1$, and $f_1^{(i)} \equiv 1$ with*

$$c_0^{(i)} = \left(1 - \sum\nolimits_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j\right)^{-1} (1 - s_{k+1}^{(i)}). \quad (23)$$

Proof. According to Corollary 3.1, Problem 1 can be rewritten as follows. Given $\overline{A}_0^{(i)} \in \mathcal{L}^{2\times 2^n}$, and $s_k^{(i)}, s_{k+1}^{(i)}, i \in \mathcal{N}$, find the number of candidates $q(i)$, probabilities $c_0^{(i)}, c_1^{(i)}, \dots, c_{q(i)}^{(i)}$, and matrices $\overline{A}_1^{(i)}, \overline{A}_2^{(i)}, \dots, \overline{A}_{q(i)}^{(i)}$ maximizing $c_0^{(i)}$ subject to the following condition, for each $i \in \mathcal{N}$,

$$\overline{s}_{k+1}^{(i)} = \left(c_0^{(i)}\overline{A}_0^{(i)} + \sum_{l=1}^{q(i)} c_l^{(i)}\overline{A}_l^{(i)}\right) Q_{\mathcal{N}}, \quad (24)$$

where $Q_{\mathcal{N}}$ is given by (21), and $\overline{A}_l^{(i)}$ is the FRSM of $f_l^{(i)}$, $l = 0, 1, \dots, q(i)$. By observing that $s_{k+1}^{(i)} = \mathrm{Row}_1\left(\overline{s}_{k+1}^{(i)}\right)$, (24) is equivalent to

$$s_{k+1}^{(i)} = \sum_{l=0}^{q(i)} c_l^{(i)} \mathrm{Row}_1\left(\overline{A}_l^{(i)}\right) Q_{\mathcal{N}} \quad (25)$$

$$= c_0^{(i)} \sum\nolimits_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j + \sum_{l=1}^{q(i)} c_l^{(i)} \sum\nolimits_{j\in\mathcal{I}(\overline{A}_l^{(i)})} [Q_{\mathcal{N}}]_j,$$

where the second equation follows from that all of $\overline{A}_l^{(i)}$, $l = 0, 1, \dots, q(i)$, $i \in \mathcal{N}$ are logical matrices in $\mathcal{L}^{2\times 2^n}$.

First, consider the case of $s_{k+1}^{(i)} = \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})} [Q_{\mathcal{N}}]_j$. From (25), we can obtain $c_0^{(i)} = 1$ is optimal, as a trivial solution. Hence, other candidates of Boolean functions are not required, which means there is no disturbance, and consequently $q(i)=0$.

Next, consider the case of $s_{k+1}^{(i)} < \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j$. In this case, $\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j \neq 0$ holds from $0\le s_{k+1}^{(i)}\le 1$. Then, from (25), we can obtain

$$c_0^{(i)} = \frac{s_{k+1}^{(i)}}{\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j} - \frac{\sum_{l=1}^{q(i)} c_l^{(i)} \sum_{j\in\mathcal{I}(\overline{A}_l^{(i)})}[Q_{\mathcal{N}}]_j}{\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j}.$$

$c_0^{(i)}$ reaches the maximal value when the second term of the right-hand side of the equation above is equal to zero, which implies we can choose $\mathcal{I}(\overline{A}_l^{(i)})=\emptyset$, for each $l\neq 0$. Then, according to Lemma 3.3, $q(i)=1$ with $f_1^{(i)}\equiv 0$. Hence, we can obtain the statement (ii).

Finally, consider the case of $s_{k+1}^{(i)} > \sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j$. We consider maximizing $c_0^{(i)}$ of (25) subject to $0\le c_0^{(i)}\le 1$. That is, $\sum_{l=1}^{q(i)} c_l^{(i)}$ must be minimized, since $c_0^{(i)} + \sum_{l=1}^{q(i)} c_l^{(i)} = 1$. To minimize $\sum_{l=1}^{q(i)} c_l^{(i)}$ subject to $0\le c_0^{(i)}\le 1$, the term $\sum_{j\in\mathcal{I}(\overline{A}_l^{(i)})}[Q_{\mathcal{N}}]_j$ of the second term of the right hand side of (25) must be maximized. Recalling $\sum_{j=1}^{2^n}[Q_{\mathcal{N}}]_j = 1$, we have $\sum_{j\in\mathcal{I}(\overline{A}_l^{(i)})}[Q_{\mathcal{N}}]_j$ reaches the maximum point 1, when $\mathcal{I}(\overline{A}^{(i)})=\{1,2,\dots,2^n\}$. Hence, applying again to Lemma 3.3, we have $f^{(i)}\equiv 1$ with $q(i)=1$. Then, (25) can be rewritten as

$$s_{k+1}^{(i)} = c_0^{(i)}\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j + (1-c_0^{(i)}),$$

which deduces (23). This completes the proof. □

Theorem 3.1 is used for analysis, based on full information of FRSM. We comment on the computational complexity for verifying Theorem 3.1. First, from the definition of STP, the computational complexity for deriving $Q_{\mathcal{N}}$ is $O(2^n)$. Next, for each $i\in\mathcal{N}$, consider computation of $\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j$. Since, in the extreme case of $f^{(i)}\equiv 1$ and $\mathcal{N}^{(i)}=\emptyset$, we have $|\mathcal{I}(\overline{A}_0^{(i)})| = 2^n$. Hence, considering the n loops, the computational complexity for verifying Theorem 3.1 is $O(2^n)+O(n2^n)=O(n2^n)$.

From the viewpoint of computation, it is desirable that the size of matrices is small. By combining Theorem 3.1 with Proposition 3.1, we know that for each node $i\in\mathcal{N}$, the indegree index $\mathcal{N}^{(i)}$ is uniquely determined by the main dynamics $f_0^{(i)}$. Hence, we propose Algorithm 1 as the algorithm with minimum calculations. One of the advantages of Algorithm 1 is that the proposed method can be used for comparatively large networks, because the computational complexity can be reduced tremendously, when using MRSM, instead of FRSM, especially in the case of that the number of in-degree indexes is much less than the number of nodes. In a similar way to the computational complexity of Theorem 3.1, the computational complexity of Algorithm 1 can be obtained as $O(2^{|\mathcal{N}^{(i)}|}) + O(n2^{|\mathcal{N}^{(i)}|}) = O(n2^{|\mathcal{N}^{(i)}|})$. Thus, using Algorithm 1, the computational complexity can be reduced depending on the number of in-degree indexes.

Algorithm 1 Optimal reconstruction of the noisy dynamics

Step 0. Initialization:
(i) Given main dynamics (5);
(ii) Collect $s_{k+1}^{(i)}, s_k^{(i)}, i\in\mathcal{N}$;
Step 1. For each $i\in\mathcal{N}$, compute $Q_{\mathcal{N}}^{(i)} = \ltimes_{j\in\mathcal{N}^{(i)}}\overline{s}^{(j)}(k)$;
Step 2. For each $i\in\mathcal{N}$,
(i) If $s_{k+1}^{(i)} = \sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j$, then $c_0^{(i)}=1$;
(ii) If $s_{k+1}^{(i)} < \sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j$, then $f_1(i)\equiv 0$ with

$$c_0^{(i)} = \left(\sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j\right)^{-1} s_{k+1}^{(i)};$$

(iii) If $s_{k+1}^{(i)} > \sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j$, then $f_1(i)\equiv 1$ with

$$c_0^{(i)} = \left(1-\sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j\right)^{-1}(1-s_{k+1}^{(i)}).$$

Remark 3.1 *In existing approaches such as Kobayashi and Hiraishi (2016); Umiji et al. (2019), multiple pairs of $s_{k+1}^{(i)}$ and $s_k^{(i)}$ are considered simultaneously. This requires solving a linear programming problem with the assumption of the network structure of noisy dynamics in advance. However, with the help of the separately characterization of evolution dynamics of selection probabilities given in Lemma 3.2, we can solve the reconstruction problem without this assumption, as provided by Theorem 3.1. From this theorem, we see that $q(i)$ may be given by at most $q(i)=1$. We also see that the noisy dynamics may be given by a constant mapping, and can be easily derived.*

Now, we state the following theorem, which provides the correctness of Algorithm 1.

Theorem 3.2 *The optimal $c_0^{(i)}$ in Theorem 3.1 is the same as that in Algorithm 1.*

Proof. According to Theorem 3.1, it is enough to prove that

$$\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j = \sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j. \tag{26}$$

As an extension of Lemma 16 of Cheng (2014), we claim that for any $x\in\mathcal{R}^p, y\in\mathcal{R}^q$, with $\sum_{i=1}^p x_i = 1$, and $\sum_{i=1}^q y_i = 1$, we have

$$D_f[p,q]\ltimes x\ltimes y = y, \text{and } D_r[p,q]\ltimes x\ltimes y = x. \tag{27}$$

In fact, for any $x\in\mathcal{R}^p$, with $\sum_{i=1}^p x_i = 1$, we get $D_f[p,q]\ltimes x = (\mathbf{1}_p^\top\otimes I_q)\ltimes x = I_q$, which implies

$D_f[p,q] \ltimes x \ltimes y = y$. Consequently, $D_r[p,q] \ltimes x \ltimes y = D_r[p,q]W_{[q,p]} \ltimes y \ltimes x = D_f[q,p] \ltimes y \ltimes x = x$, by applying the properties $W_{[q,p]} \ltimes y \ltimes x = y \ltimes x$, and $D_r[p,q]W_{[q,p]} = D_r[q,p]$ of the swap matrix $W_{[p,q]}$ defined by $[I_q \otimes \delta_p^1, I_q \otimes \delta_p^2, \ldots, I_q \otimes \delta_p^p]$ in Cheng et al. (2011a).

We assume that $\overline{\mathcal{N}}_0^{(i)} = \{j_1, j_2, \ldots, j_m\}$, and $\mathcal{N}_0^{(i)} = \{l_1, l_2, \ldots, l_{n-m}\}$. Without loss of generality, we also assume that $j_1 = 1$. Then, according to Proposition 3.1,

$$
\begin{aligned}
&\overline{A}_0^{(i)} Q_{\mathcal{N}} = \overline{A}_0^{(i)} \ltimes_{t\in\mathcal{N}} \overline{s}_k^{(t)} \\
&= AD_f[2,2]\left(\ltimes_{t\in\overline{\mathcal{N}}_0^{(i)}} D_r[2^{t-1},2]\right) \ltimes_{t\in\mathcal{N}} \overline{s}_k^{(t)} \\
&= AD_f[2,2]\left(\ltimes_{t\in\overline{\mathcal{N}}_0^{(i)}\setminus\{j_m\}} D_r[2^{t-1},2]\right) D_r[2^{j-1},2] \\
&\qquad\qquad (\ltimes_{t=1}^{j_m-1}\overline{s}_k^{(t)})\overline{s}_k^{(j)}(\ltimes_{t=j_m+1}^{n}\overline{s}_k^{(t)}) \\
&= AD_f[2,2]\left(\ltimes_{t\in\overline{\mathcal{N}}_0^{(i)}\setminus\{j_m\}} D_r[2^{t-1},2]\right) \\
&\qquad\qquad (\ltimes_{t=1}^{j_m-1}\overline{s}_k^{(t)})(\ltimes_{t=j_m+1}^{n}\overline{s}_k^{(t)}) \\
&= \cdots\cdots \\
&= AD_f[2,2]x_1(k)x_{l_2}(k) \ltimes_{t\in\mathcal{N}_0^{(i)}\setminus\{l_1\}} \overline{s}_k^{(t)} \\
&= A \ltimes_{t\in\mathcal{N}_0^{(i)}} \overline{s}^{(t)}(k) = AQ_{\mathcal{N}}^{(i)},
\end{aligned}
$$

where the equations in (27) are repeatedly applied. Thus, $\sum_{j\in\mathcal{I}(\overline{A}_0^{(i)})}[Q_{\mathcal{N}}]_j = \mathrm{Row}_1\left(\overline{A}_l^{(i)}Q_{\mathcal{N}}\right) = \mathrm{Row}_1\left(AQ_{\mathcal{N}}^{(i)}\right) = \sum_{j\in\mathcal{I}(A_0^{(i)})}\left[Q_{\mathcal{N}}^{(i)}\right]_j$ holds. □

Using a simple example, we demonstrate the above algorithm.

Example 3.2 *Consider a PBN with three states. Suppose that the main dynamics is given by*

$$
\begin{cases}
x_1(k+1) = \neg x_3(k), \\
x_2(k+1) = x_1(k) \wedge \neg x_3(k), \\
x_3(k+1) = x_1(k) \vee x_2(k).
\end{cases}
$$

From these Boolean functions, we see that $\mathcal{N}^{(1)} = \{3\}$, $\mathcal{N}^{(2)} = \{1,3\}$, and $\mathcal{N}^{(3)} = \{1,2\}$ hold. Suppose also that the sample mean for each i is given by $s_{k+1}^{(1)} = s_k^{(1)} = 0.8$, $s_{k+1}^{(2)} = s_k^{(2)} = 0.4$, and $s_{k+1}^{(3)} = s_k^{(3)} = 0.7$.

First, consider the case of x_1. From $Q_{\mathcal{N}}^{(1)} = \overline{s}^{(3)}(k)$, we can obtain $Q_{\mathcal{N}}^{(1)} = [0.7, 0.3]^\top$. The matrix $A_0^{(1)}$ can be derived as $A_0^{(1)} = \delta_2[2,1]$. From $\mathcal{I}(A_0^{(1)}) = \{2\}$, we can obtain $\sum_{j\in\mathcal{I}(A_0^{(1)})}\left[Q_{\mathcal{N}}^{(1)}\right]_j = 0.3$. From $s_{k+1}^{(1)} = 0.8 > 0.3$, condition (iii) in Step 2 of Algorithm 1 is satisfied. Hence, we can obtain $c_0^{(1)} = (1-0.8)/(1-0.3) = 0.2857$.

Next, consider the case of x_2. From $Q_{\mathcal{N}}^{(2)} = \overline{s}^{(1)}(k) \ltimes \overline{s}^{(3)}(k)$, we can obtain

$$
Q_{\mathcal{N}}^{(2)} = \begin{bmatrix} 0.8\cdot 0.7 \\ 0.8(1-0.7) \\ (1-0.8)0.7 \\ (1-0.8)(1-0.7) \end{bmatrix} = \begin{bmatrix} 0.56 \\ 0.24 \\ 0.14 \\ 0.06 \end{bmatrix}.
$$

The matrix $A_0^{(2)}$ can be derived as $A_0^{(2)} = \delta_2[2,1,2,2]$. From $\mathcal{I}(A_0^{(2)}) = \{2\}$, we can obtain $\sum_{j\in\mathcal{I}(A_0^{(2)})}\left[Q_{\mathcal{N}}^{(2)}\right]_j = 0.24$. From $s_{k+1}^{(2)} = 0.4 > 0.24$, the condition (iii) in Step 2 of Algorithm 1 is satisfied. Hence, we can obtain $c_0^{(2)} = (1-0.4)/(1-0.24) = 0.7895$.

Finally, consider the case of x_3. From $Q_{\mathcal{N}}^{(3)} = \overline{s}^{(1)}(k) \ltimes \overline{s}^{(2)}(k)$, we can obtain $Q_{\mathcal{N}}^{(3)} = [0.32, 0.48, 0.08, 0.12]^\top$. The matrix $A_0^{(3)}$ can be derived as $A_0^{(3)} = \delta_2[1,1,1,2]$. From $\mathcal{I}(A_0^{(2)}) = \{1,2,3\}$, we can obtain

$$
\sum\nolimits_{j\in\mathcal{I}(A_0^{(2)})}\left[Q_{\mathcal{N}}^{(2)}\right]_j = 0.32 + 0.48 + 0.08 = 0.88.
$$

From $s_{k+1}^{(3)} = 0.7 < 0.88$, the condition (ii) in Step 2 of Algorithm 1 is satisfied. Hence, we can obtain $c_0^{(3)} = 0.7/0.88 = 0.7955$. Thus, we can obtain the following PBN with the network graph illustrated by Fig. 1:

$$
\begin{aligned}
f^{(1)} &= \begin{cases} f_1^{(1)} = \neg x_3(k), & c_0^{(1)} = 0.2857, \\ f_2^{(1)} = 1, & c_1^{(1)} = 0.7143, \end{cases} \\
f^{(2)} &= \begin{cases} f_1^{(2)} = x_1(k) \wedge \neg x_3(k), & c_0^{(2)} = 0.7895, \\ f_2^{(2)} = 1, & c_1^{(2)} = 0.2105, \end{cases} \\
f^{(3)} &= \begin{cases} f_1^{(3)} = x_1(k) \vee x_2(k), & c_0^{(3)} = 0.7955, \\ f_2^{(3)} = 0, & c_1^{(3)} = 0.2045. \end{cases}
\end{aligned}
$$

From the obtained PBN, we can discuss the difference between the sample mean (i.e., data) and the mathematical model (i.e., the Boolean function).

We compare the proposed method with the existing approach, given in Kobayashi and Hiraishi (2016); Umiji et al. (2019). We focus on $f^{(2)}$. Using the method in

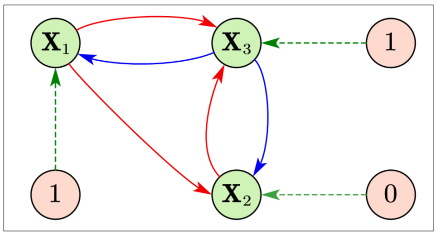

Fig. 1. Network graph of Example 3.2, where $\rightarrow$ and $\rightarrow$ express the activation and inhibition relationship, respectively, and dashed $\dashrightarrow$ indicates noise dynamics.

Kobayashi and Hiraishi (2016), we can obtain $q(2) = 4$ *and* $c_0^{(2)} = 0.4391$ *(see Umiji et al. (2019)). From* $q(2) = 4$*, we see that the number of the candidates of Boolean functions is redundant. Using the method in Umiji et al. (2019), we can obtain* $q(2) = 1$ *and* $c_0^{(2)} = 0.7143$ *(see Umiji et al. (2019)). Hence, the value of* $c_0^{(2)}$ *in the proposed method is larger than that in the method in Umiji et al. (2019). Thus, the proposed method is better than previously reported methods. PBNs obtained by three methods are equivalent in the sense that these are generated from the same main dynamics and the same sample mean (*$s_{k+1}^{(i)}$ *and* $s_k^{(i)}$*). In the proposed method, the probability* $c_0^{(i)}$ *of selecting the main dynamics is maximized. In this sense, a PBN obtained by the proposed method is optimally reconstructed.*

4 Application to *Lac* Operon Networks

In this section, the reconstruction of the PBN model of the *lac* operon, which contains the genes that control the transport and metabolism of lactose, is performed as an application to practical gene regulatory networks.

Fig. 2 shows a graphical representation of the *lac* operon network of the *Escherichia* coli bacterium. As investigated in Chen et al. (2018), the update logics of *lac* operon networks can be described by the following Boolean equations:

$$\begin{cases} f_M = C \wedge \neg R \wedge \neg R_m, \\ f_B = M, \\ f_R = \neg A \wedge \neg A_m, \\ f_A = L \wedge B, \\ f_L = P \wedge L_e \wedge \neg G_e, \\ f_P = M, \\ f_C = \neg G_e, \\ f_{R_m} = (\neg A \wedge \neg A_m) \vee R, \\ f_{A_m} = L \vee L_m, \\ f_{L_m} = ((L_{em} \wedge P) \vee L_e) \wedge \neg G_e, \end{cases} \tag{28}$$

where all the variables represent the concentration levels of the corresponding gene products; 1 denotes “present” or “high concentration” and 0 denote “absent” or “low (basal) concentration.” M denotes the *lac* mRNA, B is the β-galactosidase (LacZ), R is the repressor protein (LacI), A is the allolactose, L is the lactose, P is the transport protein (LacY; “*lac* permease”), and C is the cAMP-CAP protein complex. There exist three Boolean control variables, lactose L_e, L_{em} and glucose G_e. The subscript e represents *extracellular* concentration, and m represents at least medium concentration in (28). For more details of the biological justification of each update function of (28), see Veliz-Cuba and Stigler (2011).

For convenience, we rename Boolean variables $(M, B, R, A, L, P, C, R_m, A_m, L_m)$ as $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$, and change the parameters (L_e, L_{em}, G_e) to (u_1, u_2, u_3).

In this example, we assume that $u_i(k)$ is given by

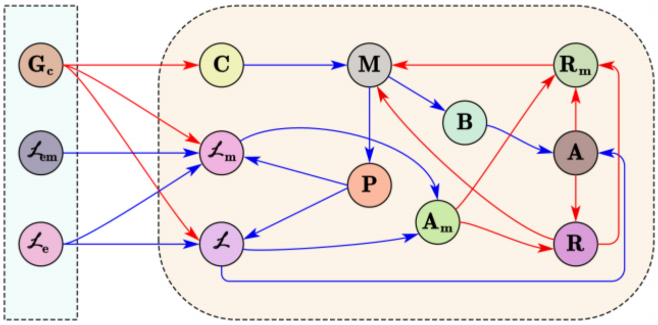

Fig. 2. Lac operon network in *Escherichia coli.*

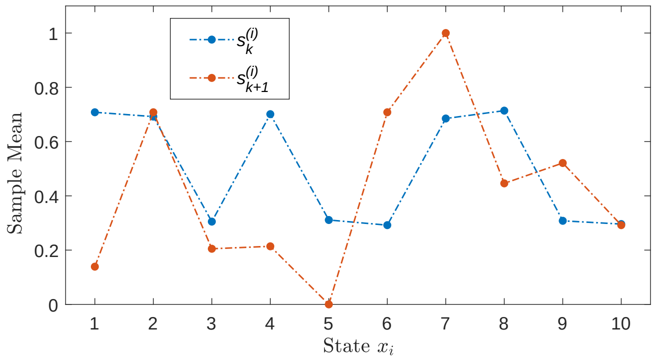

Fig. 3. Sample mean of the state.

$u_1(k) = u_3(k) = 0$ and $u_2(k) = 1$. Consider setting the state at time k as $x(k) = [1, 1, 0, 1, 0, 0, 1, 1, 0, 0]^\top$. We suppose that setting $x(k)$ is failed with the probability 0.3. According to this probability, 1000 samples of $x(k)$ were randomly generated. For each sample of $x(k)$, the next state $x(k+1)$ was calculated. From the obtained samples, $s_k^{(i)}$ and $s_{k+1}^{(i)}$ were generated as given in Table 1, see also Figure 3.

Next, we present the results of computation. Since both $x_5(k+1) = 0$ and $x_7(k+1) = 1$ hold, we do not need to investigate the dynamics of x_5 and x_7. In the cases of x_2, x_6, and x_{10}, the condition (i) in Step 2 of Algorithm 1 is satisfied. Hence, $q(2) = q(6) = q(10) = 0$ holds. Finally, we consider the dynamics of x_1, x_3, x_4,

Table 1
Sample mean of states for the *lac* operon network (28).

i	1	2	3	4	5	6	7	8	9	10
Sample mean $s_k^{(i)}$	0.7080	0.6920	0.3050	0.7010	0.3110	0.2920	0.6850	0.7140	0.3080	0.2960
Sample mean $s_{k+1}^{(i)}$	0.1390	0.7080	0.2050	0.2140	0	0.7080	1	0.4460	0.5210	0.2920

x_8, and x_9. Then, we can obtain

$$\begin{cases} f_1^{(1)} = x_7(t) \wedge \neg x_3(t) \wedge \neg x_8(t), & c_0^{(1)} = 0.8862, \\ f_2^{(1)} = 0, & c_1^{(1)} = 0.1138, \end{cases}$$

$$\begin{cases} f_1^{(3)} = \neg x_4(t) \wedge \neg x_9(t), & c_0^{(3)} = 0.9823, \\ f_2^{(3)} = 0, & c_1^{(3)} = 0.0177, \end{cases}$$

$$\begin{cases} f_1^{(4)} = x_5(t) \wedge x_2(t), & c_0^{(4)} = 0.9230, \\ f_2^{(4)} = 0, & c_1^{(4)} = 0.0770, \end{cases}$$

$$\begin{cases} f_1^{(8)} = (\neg x_4(t) \wedge \neg x_9(t)) \vee x_3(t), & c_0^{(8)} = 0.9828, \\ f_2^{(8)} = 1, & c_1^{(8)} = 0.0172, \end{cases}$$

$$\begin{cases} f_1^{(9)} = x_5(t) \vee x_{10}(t), & c_0^{(9)} = 0.9985, \\ f_2^{(9)} = 1, & c_1^{(9)} = 0.0015. \end{cases}$$

From these results, we see that for the data set used, the dynamics on x_1 is more sensitive to the noise than those on x_3, x_4, x_8, and x_9. Thus, we can obtain useful information from the proposed method.

5 Conclusion

In this paper, we proposed an optimal reconstruction method of the noisy dynamics in BNs. We showed that the noisy dynamics is derived as a constant under the assumption that the main dynamics and the sample mean of the state are given. As an application, we considered a *lac* operon network.

One of the future efforts is to apply the proposed method to experimental data sets. In this paper, we do not consider the distribution of the noise. Detailed discussion about the distribution is one of the future efforts.

References

Acernese, A., Yerudkar, A., Glielmo, L., Del Vecchio, C., 2020. Reinforcement learning approach to feedback stabilization problem of probabilistic Boolean control networks. IEEE Control Systems Letters 5, 337–342.

Akutsu, T., 2018. Algorithms for analysis, inference, and control of Boolean networks. World Scientific Singapore.

Akutsu, T., Melkman, A.A., 2018. Identification of the structure of a probabilistic Boolean network from samples including frequencies of outcomes. IEEE Transactions on Neural Networks and Learning Systems 30, 2383–2396.

Chen, S., Wu, Y., Macauley, M., Sun, X.M., 2018. Monostability and bistability of Boolean networks using semitensor products. IEEE Transactions on Control of Network Systems 6, 1379–1390.

Cheng, D., 2014. On finite potential games. Automatica 50, 1793–1801.

Cheng, D., Qi, H., Li, Z., 2011a. Analysis and control of Boolean networks: a semi-tensor product approach. London, U.K.: Springer.

Cheng, D., Qi, H., Li, Z., 2011b. Model construction of Boolean network via observed data. IEEE Transactions on Neural Networks 22, 525–536.

Ching, E.S., Tam, H., 2017. Reconstructing links in directed networks from noisy dynamics. Physical Review E 95, 010301.

Fornasini, E., Valcher, M.E., 2013. On the periodic trajectories of Boolean control networks. Automatica 49, 1506–1509.

Freilich, M.A., Rebolledo, R., Corcoran, D., Marquet, P.A., 2020. Reconstructing ecological networks with noisy dynamics. Proceedings of the Royal Society A 476, 20190739.

Ibe, O., 2013. Markov processes for stochastic modeling. Newnes.

Imani, M., Braga-Neto, U.M., 2018. Particle filters for partially-observed Boolean dynamical systems. Automatica 87, 238–250.

Kim, E., Ivanov, I., Dougherty, E.R., 2020. Network classification based on reducibility with respect to the stability of canalizing power of genes in a gene regulatory network – a Boolean network modeling perspective. IEEE/ACM Transactions on Computational Biology and Bioinformatics , published online DOI: 10.1109/TCBB.2020.3005313.

Kobayashi, K., Hiraishi, K., 2011. An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks. Automatica 47, 1260–1264.

Kobayashi, K., Hiraishi, K., 2016. Design of probabilistic Boolean networks based on network structure and steady-state probabilities. IEEE Transactions on Neural Networks and Learning Systems 28, 1966–1971.

Li, F., Sun, J., 2011. Controllability of probabilistic Boolean control networks. Automatica 47, 2765–2771.

Li, H., Wang, S., Li, X., Zhao, G., 2020a. Perturbation analysis for controllability of logical control networks. SIAM Journal on Control and Optimization 58, 3632–3657.

Li, H., Yang, X., Wang, S., 2020b. Robustness for stability and stabilization of boolean networks with stochastic function perturbations. IEEE Transactions on Automatic Control 66, 1231–1237.

Melkman, A.A., Cheng, X., Ching, W.K., Akutsu, T., 2017. Identifying a probabilistic Boolean threshold network from samples. IEEE Transactions on Neural

Networks and Learning Systems 29, 869–881.
Meng, M., Lam, J., Feng, J.E., Cheung, K.C., 2018. Stability and stabilization of Boolean networks with stochastic delays. IEEE Transactions on Automatic Control 64, 790–796.
Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18, 261–274.
Sun, J., AlMomani, A.A., Bollt, E., 2020. Data-driven learning of Boolean networks and functions by optimal causation entropy principle (bocse). arXiv preprint arXiv:2006.01023 .
Toyoda, M., Wu, Y., 2021. Mayer-type optimal control of probabilistic Boolean control network with uncertain selection probabilities. IEEE Transactions on Cybernetics 51, 3079–3092.
Trairatphisan, P., Mizera, A., Pang, J., Tantar, A.A., Schneider, J., Sauter, T., 2013. Recent development and biomedical applications of probabilistic Boolean networks. Cell communication and signaling 11, 46.
Umiji, K., Kobayashi, K., Yamashita, Y., 2019. Construction method of probabilistic Boolean networks based on imperfect information. Algorithms 12, 268.
Veliz-Cuba, A., Stigler, B., 2011. Boolean models can explain bistability in the lac operon. Journal of Computational Biology A Journal of Computational Molecular Cell Biology 18, 783–794.
Weiss, E., Margaliot, M., Even, G., 2018. Minimal controllability of conjunctive Boolean networks is np-complete. Automatica 92, 56–62.
Wu, Y., Guo, Y., Toyoda, M., 2021. Policy iteration approach to the infinite horizon average optimal control of probabilistic boolean networks. IEEE Transactions on Neural Networks and Learning Systems 32, 2910–2924.
Yerudkar, A., Del Vecchio, C., Glielmo, L., 2020. Feedback stabilization control design for switched Boolean control networks. Automatica 116, 108934.
Zhong, J., Li, B., Liu, Y., Lu, J., Gui, W., 2020. Steady-state design of large-dimensional Boolean networks. IEEE Transactions on Neural Networks and Learning Systems 32, 1149–1161.
Zhu, Q., Gao, Z., Liu, Y., Gui, W., 2020. Categorization problem on controllability of boolean control networks. IEEE Transactions On Automatic Control 66, 2297–2303.

Koichi Kobayashi received the B.E. degree in 1998 and the M.E. degree in 2000 from Hosei University, and the D.E. degree in 2007 from Tokyo Institute of Technology. From 2000 to 2004, he worked at Nippon Steel Corporation. From 2007 to 2015, he was an Assistant Professor at Japan Advanced Institute of Science and Technology. Since 2015, he has been an Associate Professor at the Graduate School of Information Science and Technology, Hokkaido University. His research interests include discrete event and hybrid systems.

Yuhu Wu received the Ph.D. degree in mathematics from the Harbin Institute of Technology, Harbin, China, in 2012. Since 2012, he has held an Assistant Professor position with the Harbin University of Science and Technology, Harbin. He held a Postdoctoral Research position with Sophia University, Tokyo, Japan, from 2012 to 2015. In 2015, he joined the School of Control Science and Engineering, Dalian University of Technology, Dalian, China, where he is currently a Full Professor. His research interests are related to optimization, and nonlinear control theory and applications of control to Boolean networks, automotive powertrain systems, and unmanned aerial vehicles.