2024 IEEE Conference on Control Technology and Applications (CCTA)
August 21-23, 2024. Newcastle upon Tyne, UK

Bayesian Optimization for State and Parameter Estimation of Dynamic Networks with Binary Space

Mohammad Alali and Mahdi Imani

2024 IEEE Conference on Control Technology and Applications (CCTA) | 979-8-3503-7094-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCTA60707.2024.10666595

Abstract— **This paper focuses on joint state and parameter estimation in partially observed Boolean dynamical systems (POBDS), a hidden Markov model tailored for modeling complex networks with binary state variables. The majority of current techniques for parameter estimation rely on computationally expensive gradient-based methods, which become intractable in most practical applications with large size of networks. We propose a gradient-free approach that uses Gaussian processes to model the expensive log-likelihood function and utilizes Bayesian optimization for efficient likelihood search over parameter space. Joint state estimation is also achieved alongside parameter estimation using the Boolean Kalman filter. The performance of the proposed method is demonstrated using gene regulatory networks observed through synthetic gene-expression data. The numerical results demonstrate the scalability and effectiveness of the proposed method in the joint estimation of the model parameters and genes' states.**

I. Introduction

Most real-world systems consist of multiple interacting elements, which are monitored using noisy data collected from various sensors [1], [2]. A special and important class of networked systems are those with binary state variables (i.e., nodes), such as gene regulatory networks [3]–[9], attack graphs [10], [11], sensor networks [12], [13], brain networks [14], and social networks [15]. Partially-observed Boolean dynamical systems (POBDS) model is designed to represent intricate networks with binary state variables [16], [17]. POBDS consolidates all existing Boolean network frameworks [18], [19]. Moreover, it possesses the capability to effectively incorporate arbitrary non-binary data without the necessity for ad-hoc binarization procedures. The solution to the state estimation problem for fully-known POBDS is investigated in [20]. However, the estimator requires full knowledge of the system model, which is often unknown or only partially known in complex practical networks.

The maximum likelihood (ML) and Bayesian estimators have also been developed for domains modeled by partially-known POBDS. These methods are applicable to small networks with finite parameter spaces [21], [22]. However, the huge computational cost associated with these approaches makes them intractable or inefficient in large systems or systems with large parameter spaces. More specifically, the computational cost of evaluating the likelihood/posterior grows exponentially with respect to the size of networks/systems; also, searching for the best set of parameters becomes extremely costly in domains with larger unknown parameters. These limit the application of existing methods, as well as conventional techniques, such as gradient-based maximum likelihood or expectation maximization methods [23]–[26]. All these methods rely on frequent evaluations of the log-likelihood (or its gradient), which makes their computations intractable in most realistic systems.

This paper addresses the challenges associated with joint state and parameter estimation of POBDS with significant uncertainty in their models. A Bayesian optimization approach is introduced to tackle the computationally challenging task of inferring the unknown parameters of these models. Initially, a Gaussian process (GP) regression is used as a surrogate model to represent the expensive-to-evaluate log-likelihood function over the parameter space. This surrogate model is Bayesian and offers a sample-efficient representation of the log-likelihood function. Subsequently, a sequential and sample-efficient search using Bayesian optimization is achieved by sequentially maximizing an acquisition function (defined over the surrogate model). This efficient search over the parameter space differentiates our proposed method from existing approaches, allowing for more scalable, accurate, and computationally efficient parameter estimation. Meanwhile, the joint state estimation, alongside parameter estimation, is achieved using the Boolean Kalman filtering theorem.

The application of the proposed method has been demonstrated using gene regulatory networks observed through a single time series of cDNA microarray data. This includes joint estimation of genes' states, as well as noise and other expression parameters associated with cDNA microarray data. Numerical experiments conducted over the p53-MDM2 network demonstrate the superiority of the proposed method compared to existing approaches.

II. Background and Problem Formulation

A. POBDS Model

The POBDS consists of state and measurement processes [16], [17]. The *state process* $\{\mathbf{x}_k; k = 0, 1, \dots T\}$ specifies the underlying dynamics of systems, which $\mathbf{x}_k = [\mathbf{x}_k(1), ..., \mathbf{x}_k(n)]^T$ represents the state values of n components of the system at time step k. We assume that each element of the state vector at every time step is drawn from the binary set $\{1, 0\}$. The states are not directly observable, rather a vector of measurements $\mathbf{y}_k$ is observed through the *measurement process*: $\{\mathbf{y}_k; k = 1, 2, \dots, T\}$. The POBDS

M. alali and M. Imani are with the Department of Electrical and Computer Engineering at Northeastern University. Emails: alali.m@northeastern.edu, m.imani@northeastern.edu

model is described as [27]:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{n}_k, \theta\right) \quad \text{(referred to as state process)} \\ \mathbf{y}_k &= \mathbf{h}\left(\mathbf{x}_k, \mathbf{v}_k, \theta\right) \quad \text{(referred to as measurement process)} \end{aligned} \tag{1}$$

for $k = 1, 2, \dots, T$. Here, θ represents a vector of unknown parameters drawn from the set Θ, $\mathbf{n}_k$ denotes the noise of transition at time step k, $\mathbf{f}$ is defined as the *network function*, and $\mathbf{h}$ refers to a function that maps the present state ($\mathbf{x}_k$) and observation noise ($\mathbf{v}_k$) to the measurement space. It is important to note that the noise processes $\mathbf{n}_k, \mathbf{v}_k; k = 1, 2, \dots$ are assumed to be "white" in nature, signifying that the noises at different time points are uncorrelated random variables. Furthermore, we assume that these noise processes are uncorrelated both with each other and with the initial state $\mathbf{x}_0$.

B. Maximum Likelihood State and Parameter Estimation

Let $\theta \in \Theta$ be a realization of the parameters of the POBDS model, where Θ represents the space of parameters. The parameter vector, θ, could consist of the state and measurement process parameters in (1). Let $\mathbf{y}_{1:T} = (\mathbf{y}_1, ..., \mathbf{y}_T)$ be the available measurements. The objective is to estimate the true parameters of the network according to the observed measurements. The maximum likelihood (ML) estimate of the parameters can be represented as:

$$\hat{\theta}^{\mathrm{ML}} = \underset{\theta \in \Theta}{\operatorname{argmax}} \log p(\mathbf{y}_{1:T} \mid \theta), \tag{2}$$

where $\log p(\mathbf{y}_{1:T} \mid \theta)$ is the logarithm of the likelihood function, i.e., $p(\mathbf{y}_{1:T} \mid \theta)$.

The state estimation of POBDS can be computed according to the ML estimate of the parameters as:

$$\hat{\mathbf{x}}_{k|T}^{\mathrm{ML}} = \underset{\hat{\mathbf{x}}_{k|T} \in \{0,1\}^n}{\operatorname{argmin}} \; \mathbb{E}\left[\|\hat{\mathbf{x}}_{k|T} - \mathbf{x}_k\|_2^2 \mid \mathbf{y}_{1:T}, \hat{\theta}^{\mathrm{ML}} \right], \tag{3}$$

for $k = 1, ..., T$, where $\hat{\mathbf{x}}_{k|T}$ represents the estimate of the system state at time state k given the information up to time step T, $\|.\|_2^2$ is the squared L_2 norm of the vector, and $\{1, 0\}^n$ is the set of all 2^n possible state estimators. The ML state estimator, $\hat{\mathbf{x}}_{k|T}^{\mathrm{ML}}$, in (3) yields the minimum mean-square error (MMSE) optimality related to the model ML estimate, i.e., $\hat{\theta}^{\mathrm{ML}}$. The expected error of state estimator, $C_{k|T}^{\mathrm{ML}}$, in (3) in terms of the conditional mean-square error (MSE) is derived as:

$$C_{k|T}^{\mathrm{ML}} = \mathbb{E}\left[\|\hat{\mathbf{x}}_{k|T}^{\mathrm{ML}} - \mathbf{x}_k\|_2^2 \mid \mathbf{y}_{1:T}, \hat{\theta}^{\mathrm{ML}} \right], \text{for } k = 1, ..., T, \tag{4}$$

where $C_{k|T}^{\mathrm{ML}}$ falls within the range of 0 to n. When $C_{k|T}^{\mathrm{ML}}$ is closer to the value of n, it indicates a greater estimation error, while a value close to 0 represents a more precise state estimation. In the following sections, our proposed approach for the computation of the ML estimate of state and parameters is described.

III. Proposed Maximum Likelihood Estimators

In this section, we introduce a framework for efficient and scalable solutions for the maximization problems in (2) and (3). Let $(\mathbf{x}^1, \dots, \mathbf{x}^{2^n})$ represent an arbitrary listing of all the possible state vectors (e.g., $\mathbf{x}^1 = [0, ..., 0]^T, \mathbf{x}^{2^n} = [1, ..., 1]^T$), and $A = \left[\mathbf{x}^1 \cdots \mathbf{x}^{2^n}\right]$ be a matrix of size $n \times 2^n$ which consists of all the the possible Boolean states. With the sequence of measurements denoted as $\mathbf{y}_{1:T}$, the subsequent state conditional distribution vectors related to $\theta \in \Theta$ are defined as:

$$\begin{aligned} \mathbf{\Pi}_{k|T}^{\theta}(j) &= p\left(\mathbf{x}_k = \mathbf{x}^j \mid \mathbf{y}_{1:T}, \theta\right), \\ \mathbf{\Pi}_{k|k}^{\theta}(j) &= p\left(\mathbf{x}_k = \mathbf{x}^j \mid \mathbf{y}_{1:k}, \theta\right), \\ \mathbf{\Pi}_{k|k-1}^{\theta}(j) &= p\left(\mathbf{x}_k = \mathbf{x}^j \mid \mathbf{y}_{1:k-1}, \theta\right), \end{aligned} \tag{5}$$

for $j = 1, \dots, 2^n$, and $k = 1, 2, \dots, T$. $\mathbf{\Pi}_{0|0}^{\theta}$ is also defined as the initial distribution of the states related to model θ.

The *transition matrix* (M_k^{θ}) of a Markov chain represented by the parameters θ is defined as the following $2^n \times 2^n$ matrix:

$$(M_k^{\theta})_{ij} = p(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}^j, \theta), \tag{6}$$

for $i, j = 1, \dots, 2^n$ and $\theta \in \Theta$. If all the parameters belong to the measurement process, the transition matrix in (6) is independent of θ.

Furthermore, when provided with an observation vector $\mathbf{y}_k$ at time step k, the *update matrix* $T_k^{\theta}(\mathbf{y}_k)$ corresponding to the parameter vector θ, is a $2^n \times 2^n$ diagonal matrix with the diagonal elements described as:

$$(T_k^{\theta}(\mathbf{y}_k))_{jj} = p\left(\mathbf{y}_k \mid \mathbf{x}_k = \mathbf{x}^j, \theta\right), \tag{7}$$

for $j = 1, \dots, 2^n$ and $\theta \in \Theta$, where $p(\cdot)$ refers to a probability mass function for discrete measurements or a probability density function for continuous measurements. Notice that the update matrix in (7) becomes independent of θ, if the parameters only belong to the state process.

Our goal is to formulate the matrix representation and provide an accurate approach for computing the log-likelihood function in equation (2). In that regard, the log-likelihood function is expressed as the following:

$$\begin{aligned} L(\theta) &= \log p(\mathbf{y}_{1:T} \mid \theta) \\ &= \sum_{k=1}^{T} \log p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \theta), \end{aligned} \tag{8}$$

where

$$\begin{aligned} p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \theta) &= \sum_{i=1}^{2^n} p(\mathbf{y}_k \mid \mathbf{x}_k = \mathbf{x}^i, \theta)\, p(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{y}_{1:k-1}, \theta) \\ &= \sum_{i=1}^{2^n} (T_k^{\theta}(\mathbf{y}_k))_{ii}\, \mathbf{\Pi}_{k|k-1}^{\theta}(i) \\ &= \|T_k^{\theta}(\mathbf{y}_k)\, \mathbf{\Pi}_{k|k-1}^{\theta}\|_1, \end{aligned} \tag{9}$$

where $\|.\|$ for vector $\mathbf{v}$ is defined as: $\|\mathbf{v}\|_1 = \sum_{i=1}^{2^n} \mathbf{v}(i)$.

The computation of state predictive posterior distribution, i.e., $\mathbf{\Pi}_{k|k-1}^{\theta}$, can be achieved recursively through [28]:

$$\mathbf{\Pi}_{k|k-1}^{\theta} = M_k^{\theta} \mathbf{\Pi}_{k-1|k-1}^{\theta}, \tag{10}$$

with the posterior update as:

$$\mathbf{\Pi}_{k|k}^{\theta} = \frac{T_k^{\theta}(\mathbf{y}_k)\, \mathbf{\Pi}_{k|k-1}^{\theta}}{\|T_k^{\theta}(\mathbf{y}_k)\, \mathbf{\Pi}_{k|k-1}^{\theta}\|_1}, \text{ for } k = 1, ..., T. \tag{11}$$

Replacing (9) into (8) leads to:

$$L(\theta) = \sum_{k=1}^{T} \log ||T_k^{\theta}(\mathbf{y}_k)\, \mathbf{\Pi}^{\theta}_{k|k-1}||_1. \tag{12}$$

Evaluating the log-likelihood function for any given $\theta \in \Theta$ is of order $O(2^{2n}T)$ due to the size of transition and update matrices involved. This complexity grows exponentially with the number of nodes in the network (i.e., Boolean variables), making the calculation of the log-likelihood extremely difficult for large networks. Meanwhile, the parameter space Θ in which the log-likelihood needs to be searched is often large in practice, preventing the application of the existing optimization techniques [24], [26] (e.g., gradient-based maximum likelihood or expectation maximization techniques), which rely on excessive log-likelihood evaluations.

Bayesian Optimization for ML Parameter Estimation: We assume that the parameter space Θ is a compact and continuous set. The goal is to find the maximizer of the log-likelihood function in (2) through a minimum number of computationally expensive log-likelihood evaluations. Toward this, we employ Gaussian process (GP) regression [29] to capture the log-likelihood function. The GP model offers a Bayesian and sample-efficient way to represent the log-likelihood function, which can be expressed as follows [30], [31]:

$$\mathcal{L}(\theta) = \mathcal{GP}\left(\mu(\theta), k(\theta, \theta)\right), \tag{13}$$

where $\mu(.)$ is the mean function, and $k(.,.)$ is the kernel function. The kernel function models the correlation of the log-likelihood function across the samples in the parameter space. We define the kernel function as:

$$k(\theta, \theta') = \sigma_f^2 \exp\left(- \sum_{i=1}^{|\theta|} \frac{(\theta(i) - \theta'(i))^2}{l_i} \right), \tag{14}$$

where $|\mathbf{v}|$ is the size of vector $\mathbf{v}$, θ and θ' are two arbitrary parameter vectors, l_i is the length-scale hyperparameter associated with the ith parameter, and σ_f^2 is called the scale factor hyperparameter. The aforementioned hyperparameters represent the correlation between the two parameter vectors. One option for the mean function $\mu(.)$ in (13), which characterizes the initial shape of the log-likelihood function across parameter space, is the constant mean function. Hyperparameters of the GP ($\mu(.)$, σ_f, and l_i) can be learned by optimizing the marginal likelihood function of the GP model at each iteration.

Let $\theta_{1:t}$ denote the initial t samples drawn from the space of parameters with the corresponding log-likelihood values $L_{1:t} = [L(\theta_1), ..., L(\theta_t)]$. Using this information, we can represent the posterior distribution of the GP model as follows:

$$\mathcal{L}(\theta) \mid \theta_{1:t}, L_{1:t} \sim \mathcal{N}(\mu_\theta^t, \Sigma_\theta^t), \tag{15}$$

where

$$\begin{aligned} \mu_\theta^t &= \mu(\theta) + \mathbf{K}_{(\theta, \theta_{1:t})}\, \mathbf{K}^{-1}_{(\theta_{1:t}, \theta_{1:t})}\, (L_{1:t} - \boldsymbol{\mu}(\theta_{1:t})), \\ \Sigma_\theta^t &= k_{(\theta,\theta)} - \mathbf{K}_{(\theta, \theta_{1:t})}\, \mathbf{K}^{-1}_{(\theta_{1:t}, \theta_{1:t})}\, \mathbf{K}^T_{(\theta, \theta_{1:t})}, \end{aligned} \tag{16}$$

and

$$\begin{aligned} \boldsymbol{\mu}(\theta_{1:t}) &= [\mu(\theta_1), ..., \mu(\theta_t)]^T, \\ \mathbf{K}_{(\boldsymbol{\Theta}, \boldsymbol{\Theta}')} &= \begin{bmatrix} k(\theta_1, \theta_1') & \dots & k(\theta_1, \theta_r') \\ \vdots & \ddots & \vdots \\ k(\theta_l, \theta_1') & \dots & k(\theta_l, \theta_r') \end{bmatrix}, \end{aligned} \tag{17}$$

for $\boldsymbol{\Theta} = \{\theta_1, ..., \theta_l\}$, $\boldsymbol{\Theta}' = \{\theta_1', ..., \theta_r'\}$.

By employing the posterior distribution of the GP model, a sample-efficient sequential optimization can be achieved through the following:

$$\theta_{t+1} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \alpha_t(\theta), \tag{18}$$

where $\alpha_t(\theta)$ is the acquisition function at iteration t, which relies on the constructed surrogate model predictions rather than the true log-likelihood function. Expected Improvement (EI) [32], [33] is a well-known acquisition function frequently used in Bayesian optimization. EI is defined as:

$$\begin{aligned} \alpha_t(\theta) = \left(\mu_\theta^t - L_{\max}^t\right) \Phi\left(\left(\mu_\theta^t - L_{\max}^t\right) / \sqrt{\Sigma_\theta^t} \right) \\ + \sqrt{\Sigma_\theta^t}\, \phi\left(\left(\mu_\theta^t - L_{\max}^t\right) / \sqrt{\Sigma_\theta^t} \right), \end{aligned} \tag{19}$$

where $\phi(.)$ and $\Phi(.)$ are the probability density function and cumulative density function of standard normal distribution, and $L_{\max}^t$ denotes the largest log-likelihood value up to the current iteration.

The sequential selection in (19) guarantees a balance between exploration and exploitation during the parameter estimation process [32]. The sequential selection process persists until either a predetermined number of likelihood evaluations have been completed or when consecutive iterations fail to reveal significant changes in the GP model's maximum value. Upon termination of the sequential search, the parameter vector that achieves the highest assessed log-likelihood value represents the ML estimate of the parameters. This can be formulated as:

$$\hat{\theta}^{\mathrm{ML}} := \theta_{m^*}, \text{ where } m^* = \underset{m \in \{1, ..., N\}}{\operatorname{argmax}}\, L_m, \tag{20}$$

where N stands for the total number of log-likelihood evaluations.

Boolean Kalman Smoother for ML State Estimation: In this part of the paper, we provide an algorithm for computation of the ML estimate of the state defined in (3). According to the Boolean Kalman smoother theorem [21], the exact solution for the optimal model-specific MMSE state estimation problem in (3) can be expressed as:

$$\hat{\mathbf{x}}_{k|T}^{\mathrm{ML}} = \overline{\mathbb{E}\left[\mathbf{x}_k \mid \mathbf{y}_{1:T}, \hat{\theta}^{\mathrm{ML}}\right]}, \text{ for } k = 1, ..., T, \tag{21}$$

where $\overline{\mathbf{v}} \in \{1, 0\}^n$ and $\overline{\mathbf{v}}(i) = \begin{cases} 0 & \text{if } \mathbf{v}(i) \le \frac{1}{2} \\ 1 & \text{if } \mathbf{v}(i) > \frac{1}{2} \end{cases}$ for $i = 1, \dots, n$. The expected error of the state estimator in (3) can also be obtained as:

$$C_{k|T}^{\mathrm{ML}} = \sum_{i=1}^{n} \min\Big\{ \mathbb{E}\Big[\mathbf{x}_k(i) \mid \mathbf{y}_{1:T}, \hat{\theta}^{\mathrm{ML}}\Big], \\ 1 - \mathbb{E}\Big[\mathbf{x}_k(i) \mid \mathbf{y}_{1:T}, \hat{\theta}^{\mathrm{ML}}\Big] \Big\}. \tag{22}$$

The expressions in (21) and (22) require the distribution of the state associated with the ML estimate of the parameters. In that regard, the backward probability distribution vector is defined as:

$$\mathbf{\Delta}_{k|s}^{\hat{\theta}^{\mathrm{ML}}}(i) = p(\mathbf{y}_{s+1:T} \mid \mathbf{x}_k = \mathbf{x}^i, \hat{\theta}^{\mathrm{ML}}),\ i = 1, \dots, 2^n, \tag{23}$$

for $s, k = 0, \dots T$. We also define $\mathbf{\Delta}_{T|T}^{\hat{\theta}^{\mathrm{ML}}} = \mathbf{1}_{2^n}$, where $\mathbf{1}_{2^n}$ represents a vector of size 2^n, consisting of elements that are all equal to 1. The smoothed posterior distribution can be then calculated as:

$$\mathbf{\Pi}_{k|T}^{\hat{\theta}^{\mathrm{ML}}} = \frac{\mathbf{\Pi}_{k|k-1}^{\hat{\theta}^{\mathrm{ML}}} \circ \mathbf{\Delta}_{k|k-1}^{\hat{\theta}^{\mathrm{ML}}}}{||\mathbf{\Pi}_{k|k-1}^{\hat{\theta}^{\mathrm{ML}}} \circ \mathbf{\Delta}_{k|k-1}^{\hat{\theta}^{\mathrm{ML}}}||_1},\ k = 1, \dots, T, \tag{24}$$

where $\circ$ is the Hadamard product. According to equations (21)-(24), the ML estimate of state and its expected error at time step k can be expressed as:

$$\begin{aligned} \hat{\mathbf{x}}_{k|T}^{\mathrm{ML}} &= \overline{A\,\mathbf{\Pi}_{k|T}^{\hat{\theta}^{\mathrm{ML}}}}, \\ C_{k|T}^{\mathrm{ML}} &= \left\| \min\left\{ A\,\mathbf{\Pi}_{k|T}^{\hat{\theta}^{\mathrm{ML}}}, \mathbf{1}_n - A\,\mathbf{\Pi}_{k|T}^{\hat{\theta}^{\mathrm{ML}}} \right\} \right\|_1, \end{aligned} \tag{25}$$

where $\mathbf{1}_n$ denotes a vector of length n where every element is equal to 1, and the minimum applies componentwise.

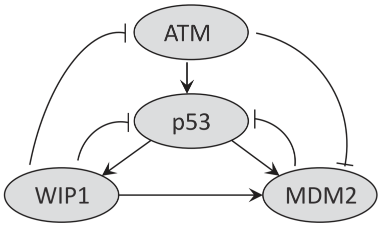

Fig. 1: p53-MDM2 network pathway model.

IV. Numerical Experiments

A. POBDS Model of Gene Regulatory Networks

Gene regulatory networks (GRNs) consist of a large number of interacting genes observed via gene-expression data. In GRNs, the POBDS state process in (1) is written as follows [27], [30]:

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{n}_k, \theta) = \overline{C\mathbf{x}_{k-1}} \oplus \mathbf{n}_k\,, \tag{26}$$

where "$\oplus$" is the componentwise module-2 addition. Assuming that the GRN consists of n genes, the connectivity matrix C is an $n \times n$ matrix in which the element in the ith row and jth column specifies the type of regulation from the jth component to the ith component. The positive and negative regulations are represented by $+1$ and -1, respectively, and 0 denotes no interaction between the two components. Note that the operator $\overline{\mathbf{v}}$ assigns 0 to the vector elements less than or equal to zero and assigns 1 to the others.

Further, $\mathbf{n}_k$ represents the process noise, where the bits in $\mathbf{n}_k$ are considered to be i.i.d. with $P(\mathbf{n}_k(i) = 1) = p$, for $i = 1, \dots, n$. $0 \le p \le \frac{1}{2}$ is a parameter that specifies the stochasticity in the state process, where $p = 0$ corresponds to no stochasticity and $p = \frac{1}{2}$ represents the largest possible stochasticity. In our numerical experiments, we assume that the parameter p is unknown, and its value is inferred through the available data.

The specific type of gene-expression data determines the characteristics of the POBDS observation model. This paper considers the following Gaussian model, which is often used for representing cDNA microarrays or live-cell imaging-based assays [34]:

$$\mathbf{y}_k(i) = m + \delta\mathbf{x}_k(i) + \mathbf{v}_k(i)\,,\ k = 1, 2, \dots, T, \tag{27}$$

for $i = 1, ..., n$. $\mathbf{v}_k(i) \sim \mathcal{N}(0, \sigma^2)$ represents a zero-mean Gaussian noise vector with uncorrelated elements, m denotes the baseline expression which refers to the "zero" states, δ is the differential expression value, and T represents the trajectory length of the gene-expression data. cDNA microarray technology has enabled scientists to study gene activity by placing DNA pieces from different genes onto a surface. Through the use of these arrays, researchers can analyze and compare the levels of gene expression in a given sample. In practice, the gene expression parameters in (26) and (27) are unknown and should be estimated according to the available data. In this section, we assume that $\theta = (p, m, \delta, \sigma)$ is the vector of parameters, and all four parameters should be estimated simultaneously according to the available data.

B. Results for p53-MDM2 Gene Regulatory Network

The proposed framework is applied to evaluate the performance of the widely known *p53-MDM2* GRN [16], [17], [35]–[37]. This GRN is tasked with encoding the p53 tumor suppressor protein in humans. Activation of the p53 gene is essential for cellular responses to diverse stress signals that may ultimately cause genome instability. Figure 1 provides a visual representation of the pathway diagram for the p53-MDM2 network, with suppressive interactions (-1) shown using blunt arrows and activating regulations (+1) represented by standard arrows. Considering $\mathbf{x}_k = (\mathit{ATM}, \mathit{p53}, \mathit{Wip1}, \mathit{MDM2})$ as the vector of states, the connectivity matrix can be expressed as:

$$C = \begin{bmatrix} 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & +1 \\ -1 & -1 & 0 & +1 \\ 0 & -1 & 0 & 0 \end{bmatrix}^T \tag{28}$$

In our experiments, the trajectory length T is considered to be 100. Further, the parameter vector consists of continuous elements $\theta = (p, m, \delta, \sigma)$, where $\theta^* = (0.01, 20, 30, 5)$ and $\Theta = [0, 0.2] \times [10, 32] \times [20, 42] \times [1, 21]$. This space of parameters, denoted by Θ, is very large. This large size makes it

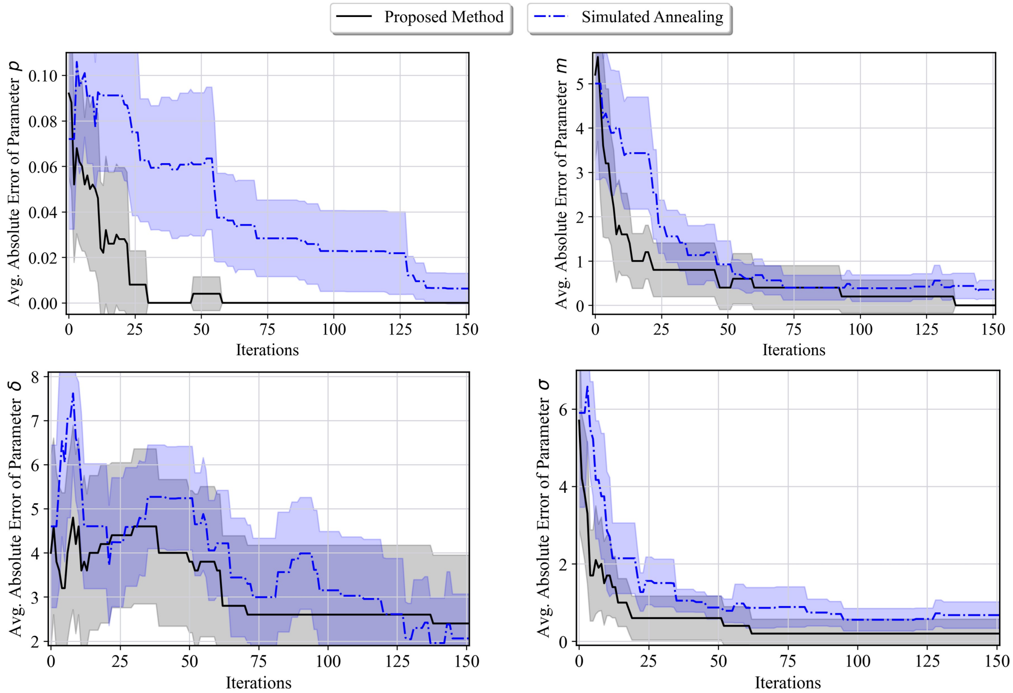

Fig. 2: Average absolute error of the p53-MDM2 network estimated parameters (p, m, δ, σ) from their true values obtained by the proposed method and simulated annealing.

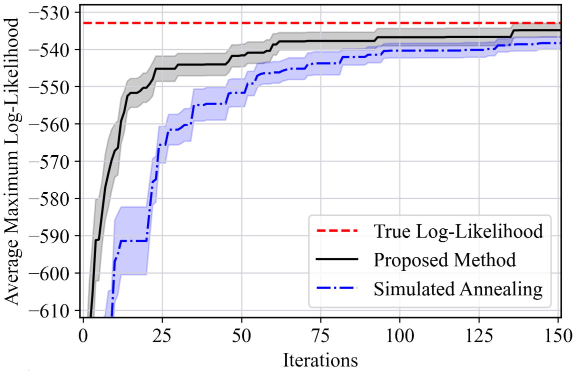

Fig. 3: Comparison of the average maximum log-likelihood progress of the proposed approach and simulated annealing.

impractical to use common optimization methods like those based on gradients or expectation maximization, as they require numerous evaluations of the computationally expensive log-likelihood function. Meanwhile, the computational complexity of the log-likelihood function's gradients poses another challenge in performing the gradient-based optimization approaches for the inference process. However, heuristic optimization methods such as Simulated Annealing [38] have been extensively used for efficiently optimizing continuous parameters across large parameter spaces, particularly in the context of biological networks [39], [40]. Hence, we have chosen Simulated Annealing as a comparison method with our proposed method. All the experiments are repeated 10 times, and the figures display the mean values and the 95% confidence intervals for all the results.

The proposed ML estimation method is employed using 100 gene-expression data (T = 100) over the p53 network for 150 iterations (i.e. log-likelihood evaluations). Notice that each iteration refers to a single log-likelihood evaluation. We also conduct additional experiments using the Simulated Annealing method for comparison purposes. Note that the Simulated Annealing parameters have been fine-tuned according to the current problem. The average maximum log-likelihood progress with respect to each iteration is represented in Figure 3. The red dashed line shows the log-likelihood of the true underlying parameters, i.e., θ^*. Further, the maximum log-likelihood values achieved by the proposed method and Simulated Annealing are indicated by the black and blue lines, respectively. One can see that in all the steps, on average, our proposed method has a better performance than Simulated Annealing. This shows how efficient our method is in searching the large parameter space, Θ.

We also obtained the absolute error of the estimated parameters from their true values for the previous scenario and presented it in Figure 2. One can notice that our estimation method reaches a smaller average error (closer to zero) in almost all the steps for all the parameters, which shows the superiority of the proposed method in comparison to Simulated Annealing. Furthermore, the estimates for each of the four parameters using three independent runs of our method are shown in Figure 4. The red line shows the true value of each parameter, and the black, blue, and green lines represent the estimates of our proposed method in the first, second, and third runs, respectively. We can see that except in the second and third runs for parameter δ, the estimates for the four parameters converge to the true values within the 150 iterations, which again shows the efficiency in terms of parameters despite the complexity of the problem.

Additionally, the ML state estimator has been performed alongside the ML parameter estimator. Considering that the trajectory length is T, we define the normalized state

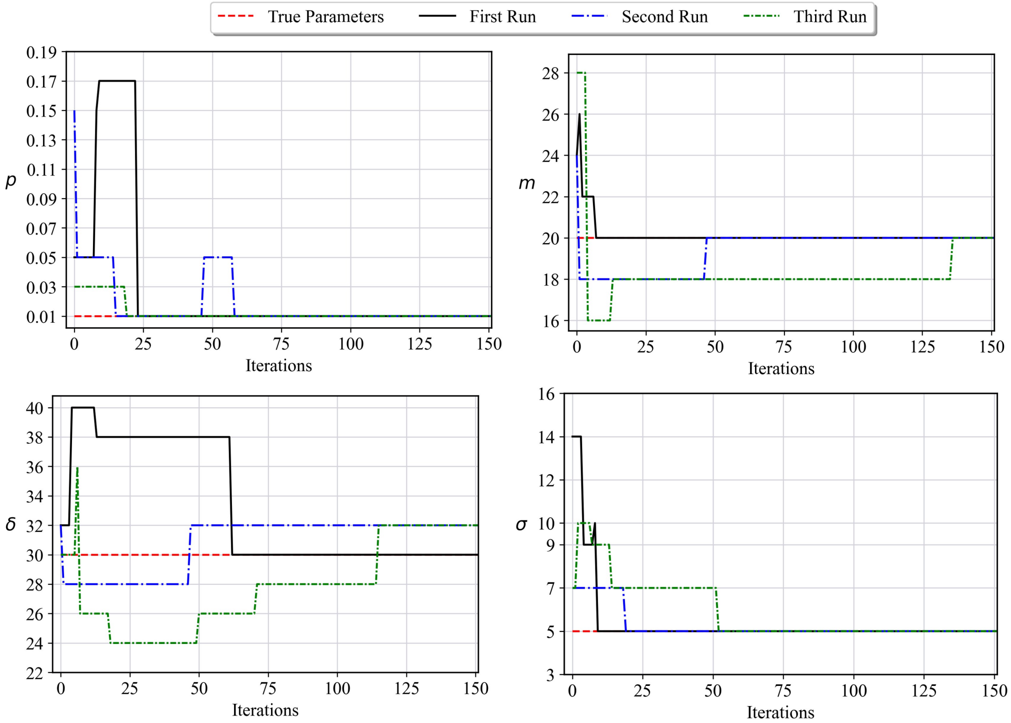

Fig. 4: Estimation of the unknown parameters (p, m, δ, σ) for p53-MDM2 network using three independent runs of the proposed method.

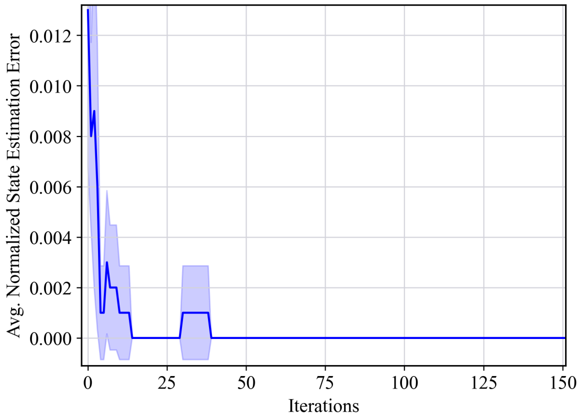

Fig. 5: The progress of average normalized state estimation error for the proposed method.

estimation error as:

$$\frac{1}{T}\sum_{k=1}^{T}||\hat{\mathbf{x}}^{\mathrm{ML}}_{k|T}-\mathbf{x}_k||_1. \tag{29}$$

Figure 5 represents the average normalized state estimation error for the results shown in Figures 2 and 3. This figure shows that the average normalized error converges to zero very quickly (in less than 50 iterations), which means that the state estimation method is able to correctly estimate the true system states efficiently.

TABLE I: RMSE Values for Different Parameters

	Case 1		Case 2	
Parameters	**True Values**	**RMSE**	**True Values**	**RMSE**
$\boldsymbol{p}$	0.1	0.02	0.01	0
$\boldsymbol{m}$	20	0.894	20	0
$\boldsymbol{\delta}$	30	1.673	30	2.683
$\boldsymbol{\sigma}$	5	0.632	5	0

Finally, to test the accuracy of our parameter estimation method, we performed experiments on two cases with 250 iterations and recorded the estimated parameter values. Table I shows the root mean square error (RMSE) values for our parameters averaged over 10 runs for two cases: the first case with the true parameters $\theta^* = (0.1, 20, 30, 5)$, and the second case with the true parameters $\theta^* = (0.01, 20, 30, 5)$. From the RMSE values shown in Table I, one can see that the proposed estimator is capable of reaching small RMSE values with a limited number of log-likelihood evaluations, thus, indicating the high accuracy of our method. Furthermore, by comparing the RMSE values of case 1 and case 2, it can be seen that all the RMSE values decrease to zero from case 1 to case 2, except the RMSE value for δ, which increases. This is because the attractor of the p53 network is "0000", meaning that all genes spend most of their time in an inactivated state [21]. Thus, for case 2, with $p = 0.01$, the available gene-expression data is a noisy realization of parameter m according to the measurement model (27). A lack of representation of state values of genes at the activated state makes inferring the differential expression parameter (δ) extremely difficult. Consequently, we can see that despite the zero error in the inference of all other parameters, a larger RMSE can be seen for the differential expression parameter.

V. Concluding Remarks

This paper focuses on estimating the state and parameters of dynamic networks with binary state variables. We employed the partially-observed Boolean dynamical systems (POBDS) model for representing these complex networked systems. Given that the parameters of POBDS are continuous, we developed a scalable and sample-efficient maximum likelihood (ML) parameter estimation built on the Bayesian optimization scheme. Then, the ML estimate of the state is achieved using the Boolean Kalman filtering theorem. A recursive and efficient matrix-based solution for the computation of the estimator is introduced, followed by performance analysis using an example of gene regulatory networks observed through gene-expression data. Our future research will study the estimation of large networks with non-continuous unknown parameters.

ACKNOWLEDGMENT

The authors acknowledge the support of the National Institute of Health award 1R21EB032480-01, National Science Foundation awards IIS-2311969 and IIS-2202395, ARMY Research Laboratory award W911NF2320179, ARMY Research Office award W911NF2110299, and Office of Naval Research award N00014-23-1-2850.

REFERENCES

[1] R. Zandi, K. Behzad, E. Motamedi, H. Salehinejad, and M. Siami, "Robofisense: Attention-based robotic arm activity recognition with wifi sensing," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–11, 2024.

[2] S. Dávila-Montero, S. Parsnejad, E. Ashoori, D. Goderis, and A. J. Mason, "Design of a multi-sensor framework for the real-time monitoring of social interactions," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 615–619, 2022.

[3] M. Imani and U. Braga-Neto, "Optimal control of gene regulatory networks with unknown cost function," in *2018 Annual American Control Conference (ACC)*, pp. 3939–3944, IEEE, 2018.

[4] M. Imani and U. Braga-Neto, "Gene regulatory network state estimation from arbitrary correlated measurements," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, pp. 1–10, 2018.

[5] M. Alali and M. Imani, "Bayesian lookahead perturbation policy for inference of regulatory networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–14, 2024.

[6] M. Alali, A. Kazeminajafabadi, and M. Imani, "Deep reinforcement learning sensor scheduling for effective monitoring of dynamical systems," *Systems Science & Control Engineering*, vol. 12, no. 1, p. 2329260, 2024.

[7] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal inference of hidden Markov models through expert-acquired data," *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2024.

[8] A. Ravari, S. F. Ghoreishi, and M. Imani, "Structure-based inverse reinforcement learning for quantification of biological knowledge," in *2023 IEEE Conference on Artificial Intelligence (CAI)*, pp. 282–284, 2023.

[9] S. H. Hosseini and M. Imani, "Learning to fight against cell stimuli: A game theoretic perspective," in *IEEE Conference on Artificial Intelligence*, 2023.

[10] A. Kazeminajafabadi and M. Imani, "Optimal joint defense and monitoring for networks security under uncertainty: A POMDP-based approach," *IET Information Security*, vol. 2024, no. 1, p. 7966713, 2024.

[11] A. Kazeminajafabadi, S. F. Ghoreishi, and M. Imani, "Optimal detection for Bayesian attack graphs under uncertainty in monitoring and reimaging," in *2024 American Control Conference (ACC)*, 2024.

[12] C. Huang, W. Wang, J. Lu, and J. Kurths, "Asymptotic stability of Boolean networks with multiple missing data," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 6093–6099, 2021.

[13] N. Asadi and F. Ghoreishi, "Dynamic sensor selection for efficient monitoring of coupled multidisciplinary systems," *Journal of Computing and Information Science in Engineering*, pp. 1–12.

[14] P. Trairatphisan, A. Mizera, J. Pang, A. A. Tantar, and T. Sauter, "optPBN: An optimisation toolbox for probabilistic Boolean networks," *PloS one*, vol. 9, no. 7, p. e98001, 2014.

[15] P. Zhu, X. Song, L. Liu, Z. Wang, and J. Han, "Stochastic analysis of multiplex Boolean networks for understanding epidemic propagation," *IEEE access*, vol. 6, pp. 35292–35304, 2018.

[16] S. H. Hosseini and M. Imani, "Modeling defensive response of cells to therapies: Equilibrium interventions for regulatory networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–13, 2024.

[17] S. H. Hosseini and M. Imani, "An optimal Bayesian intervention policy in response to unknown dynamic cell stimuli," *Information Sciences*, vol. 666, p. 120440, 2024.

[18] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[19] D. Cheng and Y. Zhao, "Identification of Boolean control networks," *Automatica*, vol. 47, no. 4, pp. 702–710, 2011.

[20] L. D. McClenny, M. Imani, and U. M. Braga-Neto, "Boolean Kalman filter with correlated observation noise," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 866–870, IEEE, 2017.

[21] M. Imani and U. M. Braga-Neto, "Maximum-likelihood adaptive filter for partially observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.

[22] M. Imani, E. R. Dougherty, and U. Braga-Neto, "Boolean Kalman filter and smoother under model uncertainty," *Automatica*, vol. 111, p. 108609, 2020.

[23] N. Kantas and et al., "On particle methods for parameter estimation in state-space models," *Statistical science*, vol. 30, no. 3, pp. 328–351, 2015.

[24] A. Wills and et al., "Identification of hammerstein–wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[25] R. G. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," *arXiv preprint arXiv:1609.09869*, 2016.

[26] T. B. Schön and et al., "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[27] M. Imani and U. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," *Automatica*, vol. 87, pp. 238–250, 2018.

[28] S. Särkkä, "Bayesian filtering and smoothing," *Cambridge university press*, no. 3, 2013.

[29] C. E. Rasmussen and et al., *Gaussian processes for machine learning*. MIT Press, 2006.

[30] M. Alali and M. Imani, "Kernel-based particle filtering for scalable inference in partially observed boolean dynamical systems," in *IFAC-PapersOnLine, 20th IFAC Symposium on System Identification (SYSID 2024)*, Elsevier, 2024.

[31] M. Alali and M. Imani, "Inference of regulatory networks through temporally sparse data," *Frontiers in Control Engineering*, vol. 3, 2022.

[32] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[33] N. Manshour, F. He, D. Wang, and D. Xu, "Integrating protein structure prediction and bayesian optimization for peptide design," in *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

[34] J. Hua, C. Sima, M. Cypert, G. C. Gooden, S. Shack, L. Alla, E. A. Smith, J. M. Trent, E. R. Dougherty, and M. L. Bittner, "Dynamical analysis of drug efficacy and mechanism of action using GFP reporters," *Journal of Biological Systems*, vol. 20, no. 04, pp. 403–422, 2012.

[35] E. Batchelor, A. Loewer, and G. Lahav, "The ups and downs of p53: understanding protein dynamics in single cells," *Nature Reviews Cancer*, vol. 9, no. 5, p. 371, 2009.

[36] M. Alali and M. Imani, "Reinforcement learning data-acquiring for causal inference of regulatory networks," in *2023 American Control Conference (ACC)*, pp. 3957–3964, 2023.

[37] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal recursive expert-enabled inference in regulatory networks," *IEEE Control Systems Letters*, vol. 7, pp. 1027–1032, 2023.

[38] C. Tsallis and D. A. Stariolo, "Generalized simulated annealing," *Physica A: Statistical Mechanics and its Applications*, vol. 233, no. 1, pp. 395–406, 1996.

[39] J. E. Handzlik and Manu, "Data-driven modeling predicts gene regulatory network dynamics during the differentiation of multipotential hematopoietic progenitors," *PLOS Computational Biology*, vol. 18, pp. 1–31, 01 2022.

[40] A. Stivala, P. Stuckey, and A. Wirth, "Fast and accurate protein substructure searching with simulated annealing and GPUs," *BMC Bioinformatics*, vol. 11, Sept. 2010.