Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 205763, 11 pages
http://dx.doi.org/10.1155/2013/205763

Research Article

Reverse Engineering Sparse Gene Regulatory Networks Using Cubature Kalman Filter and Compressed Sensing

Amina Noor,[1] Erchin Serpedin,[1] Mohamed Nounou,[2] and Hazem Nounou[3]

[1] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA*
[2] *Chemical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*
[3] *Electrical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*

Correspondence should be addressed to Amina Noor; amina@neo.tamu.edu

Received 30 November 2012; Accepted 15 April 2013

Academic Editor: Yufei Huang

This paper proposes a novel algorithm for inferring gene regulatory networks which makes use of cubature Kalman filter (CKF) and Kalman filter (KF) techniques in conjunction with compressed sensing methods. The gene network is described using a state-space model. A nonlinear model for the evolution of gene expression is considered, while the gene expression data is assumed to follow a linear Gaussian model. The hidden states are estimated using CKF. The system parameters are modeled as a Gauss-Markov process and are estimated using compressed sensing-based KF. These parameters provide insight into the regulatory relations among the genes. The Cramér-Rao lower bound of the parameter estimates is calculated for the system model and used as a benchmark to assess the estimation accuracy. The proposed algorithm is evaluated rigorously using synthetic data in different scenarios which include different number of genes and varying number of sample points. In addition, the algorithm is tested on the DREAM4 *in silico* data sets as well as the *in vivo* data sets from IRMA network. The proposed algorithm shows superior performance in terms of accuracy, robustness, and scalability.

1. Introduction

Gene regulation is one of the most intriguing processes taking place in living cells. With hundreds of thousands of genes at their disposal, cells must decide which genes are to express at a particular time. As the cell development evolves, different needs and functions entail an efficient mechanism to turn the required genes on while leaving the others off. Cells can also activate new genes to respond effectively to environmental changes and perform specific roles. The knowledge of which gene triggers a particular genetic condition can help us ward off the potential harmful effects by switching that gene off. For instance, cancer can be controlled by deactivating the genes that cause it.

Gene expression is the process of generating functional gene products, for example, mRNA and protein. The level of gene functionality can be measured using microarrays or gene chips to produce the gene expression data [1]. More accurate estimation of gene expression is now possible using the RNA-Seq method. Intelligent use of such data can help improve our understanding of how the genes are interacting in a living organism [2–4]. Gene regulation is known to exhibit several modes; a couple of important ones include transcription regulation and posttranscription regulation [5]. While the theoretical applications of gene regulation are extremely promising, it requires a thorough understanding of this complex process. Different genes may cooperate to produce a particular reaction, while a gene may repress another gene as well. The potential benefits of gene regulation can only be reaped if a complete and accurate picture of genetic interactions is available. A network specifying different interconnections of genes can go a long way in understanding the gene regulation mechanism. The control and interaction of genes can be described through a *gene regulatory network*. Such a network depicts various interdependencies among genes where nodes of the network represent the genes,

and the edges between them correspond to an interaction among them. The strength of these interactions represents the extent to which a gene is affected by other genes in the network. A key ingredient of this approach is an accurate and representative modeling of gene networks. Precise modeling of a regulatory network coupled with efficient inference and intervention algorithms can help in devising personalized medicines and cures for genetic diseases [6].

Various methods for gene network modeling have been proposed recently in the literature with varying degrees of sophistication [7–10]. These techniques can be broadly classified as static and dynamic modeling schemes. Static modeling includes the use of correlation, statistical independence for clustering [11–13], and information theoretic criteria [14–16]. On the other hand, dynamic models provide an insight into the temporal evolution of gene expressions and hence yield a more quantitative prediction on gene network behavior [17–20]. In order to incorporate the stochasticity of gene expressions, statistical techniques have been applied [13]. A rich literature is also available on the Bayesian modeling of gene networks [21–26]. Promoted in part by the Bayesian methods, the state-space approach is a popular technique to model the gene networks [27–33], whereby the hidden states can be estimated using the Kalman filter. In the case of nonlinear functions, the extended Kalman filter (EKF) and particle filter represent feasible approaches [33, 34]. However, the EKF relies on the first-order linear approximations of nonlinearities, while the particle filter may be computationally too complex. A comprehensive review of these methods can be found in [35].

In this paper, the gene network is modeled using a state-space approach, and the cubature Kalman filter (CKF) is used to estimate the hidden states of the nonlinear model [36, 37]. The gene expressions are assumed to evolve following a sigmoid squash function, whereas a linear function is considered for the expression data. The noise is assumed to be Gaussian for both the state evolution and gene expression measurements. As the gene network is assumed sparse, any simple mean square error minimization technique will not suffice for the estimation of static parameters. Therefore, a compressed sensing-based Kalman filter (CSKF) [38] is used in conjunction with CKF for reliable estimation of parameters. In case of statistical inference, it is essential to obtain some guarantees on the performance of estimators. In this regard, the Cramér-Rao lower bound (CRB) of the parameter estimates is used as a benchmarking index to assess the mean square error (MSE) performance of the proposed estimator which is evaluated here for a parameter vector. The performance of the proposed algorithm is tested on synthetically generated random Boolean networks in various scenarios. The algorithm is also tested using DREAM4 data sets and IRMA networks [39, 40].

The main contributions of this paper can be summarized as follows.

(1) CKF is proposed for the estimation of states, and a compressed sensing-based Kalman filter is used for the estimation of system parameters. The genes are known to interact with few other genes only necessitating the use of sparsity constraint for more accurate estimation. The proposed algorithm carries out online estimation of parameters and is therefore computationally efficient and is particularly suitable for large gene networks.

(2) The Cramér-Rao lower bound is calculated for the estimation of unknown parameters of the system. The performance of the proposed algorithm is compared to CRB. This comparison is significant as it shows room for improvement in the estimation of parameters.

(3) The proposed algorithm is compared with the EKF algorithm. Using the false alarm errors, true connections, and Hamming distance as fidelity criteria, rigorous simulations are carried out to assess the performance of the algorithm with the increase in the number of samples. In addition, receiver operating characteristic (ROC) curves are plotted to evaluate the algorithms for different network sizes. It is observed that the proposed algorithm outperforms EKF in terms of accuracy and precision. The proposed algorithm is then applied to the DREAM4 10-gene and 100-gene data sets to assess the algorithm accuracy. The underlying gene network for the IRMA data sets is also inferred.

The rest of this paper is organized as follows. Section 2 describes the underlying system model for the gene expressions. The proposed CKF algorithm in combination with CSKF for gene network inference is formulated in Section 3. The derivation of CRB is shown in Section 4, and the simulation results and their interpretation are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. System Model

Gene regulatory networks can be modeled as static or dynamical systems. In this work, state-space modeling is considered which is an instance of a dynamic modeling approach and can effectively cope with time variations. The states represent gene expressions, and their evolution in time, in general, can be expressed as

$$\mathbf{x}_k = g\left(\mathbf{x}_{k-1}\right) + \mathbf{w}_k \quad k = 1, \ldots, K, \tag{1}$$

where K is the total number of data points available, $\mathbf{w}_k$ is assumed to be a zero-mean Gaussian random variable with covariance $\mathbf{Q}_k = \sigma_w^2\mathbf{I}$, and the function $g(\cdot)$ represents the regulatory relationship between the genes and is generally nonlinear. The microarray data is a set of noisy observations and is commonly expressed as a linear Gaussian model [41]

$$\mathbf{y}_k = h\left(\mathbf{x}_k\right) + \mathbf{v}_k, \tag{2}$$

where $\mathbf{v}_k$ is Gaussian-distributed random variable with zero mean and covariance $\mathbf{S}_k = \sigma_v^2\mathbf{I}$ and incorporates the uncertainty in the microarray experiments. In order to capture

the gene interactions effectively, the following nonlinear state evolution model is assumed [33, 34]:

$$x_{k,n} = \sum_{m=1}^{N} b_{nm} f\left(x_{k-1,m}\right) + w_{k,n}, \quad k = 1, \ldots, K, \; n = 1, \ldots, N, \tag{3}$$

where N is the total number of genes in the network and $f(\cdot)$ is the sigmoid squash function

$$f\left(x_{k-1,m}\right) = \frac{1}{1 + e^{-x_{k-1,m}}}. \tag{4}$$

This particular choice for the nonlinear function ensures that the conditional distribution of the states remains Gaussian [41]. The multiplicative constants b_{nm} quantify the positive or negative relations between various genes in the network. A positive value of b_{nm} implies that the mth gene is activating the nth gene, whereas a negative value implies repression [34, 42]. The absolute value of these parameters indicates the strength of interaction.

The model given in (3) and (4) in the absence of any constraints may be unidentifiable and may result into overfitted solutions [43]. Assumptions on network structures are, therefore, necessary to obtain a connectivity matrix that agrees with the biological knowledge. In a gene regulatory network (GRN), the genes are known to interact with few other genes only. To this end, the coefficients b_{nm}s are estimated using sparsity constraints, as explained in the next section.

A discrete linear Gaussian model for the microarray data is considered which can be expressed at the kth time instant as [41]

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k. \tag{5}$$

Stacking the unknown parameters together, the parameter vector to be estimated is

$$\mathbf{b} \triangleq \left[\phi_1, \phi_2, \ldots, \phi_N\right], \tag{6}$$

where $\phi_n = [b_{n1}, \ldots, b_{nN}]$. Plugging the values of states from (3) into (5), it follows that

$$\mathbf{y}_k = \mathbf{R}_k \mathbf{b} + \mathbf{e}_k, \tag{7}$$

where

$$\mathbf{R}_k \triangleq \begin{bmatrix} \widetilde{\mathbf{f}}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{f}}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{f}}_k \end{bmatrix}, \tag{8}$$

$$\widetilde{\mathbf{f}}_k \triangleq \left[f\left(x_{k-1,1}\right) \cdots f\left(x_{k-1,N}\right)\right]. \tag{9}$$

Thus, the gene network inference problem boils down to the estimation of system parameters $\mathbf{b}$ using the observations $\mathbf{y}_k$, where the effective noise $\mathbf{e}_k$ is the sum of system and observation noises. The next section describes the proposed inference algorithm for sparse networks.

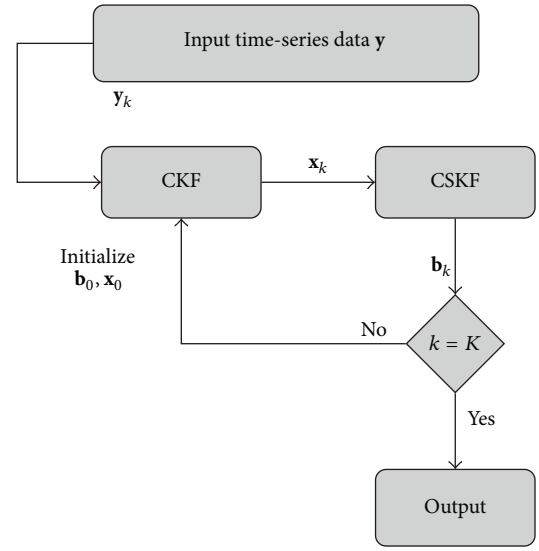

Figure 1: Block diagram of network inference methodology CKFS.

3. Method

In this section, the methodology proposed to infer the system parameters in (3) is described. The proposed cubature Kalman filter with sparsity constraints (CKFS) approach is succinctly illustrated in Figure 1. The specific details of this algorithm are as next presented.

3.1. Cubature Kalman Filter. Kalman filter is a Bayesian filter which provides the optimal solution to a general linear state space inference problem depicted by (1) and (2) and assumes a recursive *predictive-update* process. The underlying assumption of Gaussianity for the predictive and the likelihood densities simplifies the Kalman filter algorithm to a two-step process, consisting of prediction and update of the mean and covariance of the hidden states. However, the presence of nonlinear functions in the state and measurement equations requires calculation of multidimensional integrals of the form *nonlinear function* × *Gaussian density* [36], which in general is computationally prohibitive. Several solutions to this problem have been proposed including the EKF, which linearizes the nonlinear function by taking its first-order Taylor approximation, and the unscented Kalman filter (UKF), which approximates the probability density function (PDF) using a nonlinear transformation of the random variable. Recently, a new approach, CKF, has been proposed which evaluates the integrals numerically using spherical-radial cubature rules [36].

The next two subsections briefly explain the working of Bayesian filtering and the CKF solution for the nonlinear multidimensional integrals.

3.1.1. Time Update. Let the observations up to the time instant k be denoted by $\mathbf{d}_k$; that is, $\mathbf{d}_k \triangleq [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T]^T$. In the prediction phase, also called the time update of the Bayesian filter,

the mean and covariance of the Gaussian posterior density are computed as follows:

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= E\left[\mathbf{f}\left(\mathbf{x}_{k-1}\right) \mid \mathbf{d}_{k-1}\right],\\ \mathbf{P}_{xx,k|k-1} &= E\left[\mathbf{f}\left(\mathbf{x}_k\right)\mathbf{f}^T\left(x_k\right)\right] - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1},\end{aligned} \tag{10}$$

where E denotes the expectation operator and $\mathbf{x}_{k-1}$ is normally distributed with parameters $(\widehat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{xx,k-1|k-1})$. The third equality is a consequence of the zero-mean nature of Gaussian noise $\mathbf{w}$ and its independence from $\mathbf{d}_k$. The estimates $\widehat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{xx,k-1|k-1}$ are assumed to be available from the previous iteration. Here, $\mathbf{P}_{xx,k|k-1}$ is an estimate of the error covariance matrix.

3.1.2. Measurement Update. Since the measurement noise is also Gaussian, the likelihood density is given by $\mathbf{y}_{k-1} \mid \mathbf{d}_{k-1} : \mathcal{N}(\mathbf{z}_{k-1}; \widehat{\mathbf{y}}_{k|k-1}, \mathbf{P}_{xx,k|k-1})$. As the measurements become available at the kth time instant, the mean and covariance of the likelihood density are calculated as follows:

$$\begin{aligned}\widehat{\mathbf{y}}_{k|k-1} &= E\left[\mathbf{y}_k \mid \mathbf{d}_{k-1}\right],\\ \mathbf{P}_{yy,k|k-1} &= E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \widehat{\mathbf{y}}_{k|k-1}\widehat{\mathbf{y}}_{k|k-1}^T + \mathbf{S}_{k-1}.\end{aligned} \tag{11}$$

The updated posterior density, obtained from the conditional joint density of states, and the measurements can be expressed as

$$\begin{aligned}&\left(\left[\mathbf{x}_k^T\mathbf{y}_k^T\right]^T\mathbf{d}_{\mathbf{k-1}}\right)\\ &\quad\sim \mathcal{N}\left(\begin{pmatrix}\widehat{\mathbf{x}}_{k|k-1}\\ \widehat{\mathbf{y}}_{k|k-1}\end{pmatrix}, \begin{pmatrix}\mathbf{P}_{xx,k|k-1} & \mathbf{P}_{xy,k|k-1}\\ \mathbf{P}_{xy,k|k-1}^T & \mathbf{P}_{yy,k|k-1}\end{pmatrix}\right),\end{aligned} \tag{12}$$

where

$$P_{xy,k|k-1} = E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{y}}_{k|k-1}^T \tag{13}$$

is the cross-covariance matrix between the states and the measurements. Hence, the states and the corresponding error covariance matrix are updated by calculating the innovation $\mathbf{z}_k - \widehat{\mathbf{z}}_{k|k-1}$ and the Kalman gain $\mathbf{K}_{G,i}$

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{G,k}\left(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}\right),\\ P_{xx,k|k} &= P_{xx,k|k-1} - \mathbf{K}_{G,k}P_{yy,k|k-1}\mathbf{K}_{G,k}^T,\\ \mathbf{K}_{G,k} &= P_{xy,k|k-1}P_{yy,k|k-1}^{-1}.\end{aligned} \tag{14}$$

The next subsection briefly describes the computation of high-dimensional integrals present in the equations above.

3.1.3. Computation of Integrals Using Spherical-Radial Cubature Points. In order to determine the expectations in (10), using a numerical integration method, a spherical-radial cubature rule is applied. This method calculates the cubature points $\mathbf{X}_{j,k-1|k-1}$ as follows [36]:

$$\mathbf{X}_{j,k-1|k-1} = \mathbf{U}_{k-1|k-1}\zeta_j + \widehat{\mathbf{x}}_{k-1|k-1}, \tag{15}$$

where $\zeta_j = \sqrt{\ell/2}[1]_j$, $j = 1, \ldots, \ell$, $\ell = 2N$ denotes the total number of cubature points and $\mathbf{U}_{k-1|k-1}$ stands for the square root of the error covariance matrix; that is,

$$\mathbf{P}_{xx,k-1|k-1} = \mathbf{U}_{k-1|k-1}\mathbf{U}_{k-1|k-1}^T. \tag{16}$$

The cubature points are updated via the state equation

$$\mathbf{X}_{j,k|k-1}^* = g\left(\mathbf{X}_{j,k-1|k-1}\right). \tag{17}$$

The propagated cubature points yield the state and error covariance estimates

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{X}_{j,k|k-1}^*,\\ \mathbf{P}_{xx,k|k-1} &= \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{X}_{j,k|k-1}^*\mathbf{X}_{j,k|k-1}^{*T}\\ &\quad - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1}.\end{aligned} \tag{18}$$

The integrals in (11) and (14) can be evaluated in a similar manner. The next subsection explains the estimation of parameters in the system.

3.2. Estimation of Sparse Parameters Using Kalman Filter. The state estimates are obtained using the CKF as described in the previous subsection. In order to estimate the unknown parameters in the system model, one of the most commonly used methods involves stacking the parameters with the states and estimating them together. The estimation process performed in this manner is called *joint estimation*. Another method for the estimation of parameters consists of a two-step recursive process which is termed *dual estimation*. This process estimates the states in the first step, and with the assumption that states are known, parameters are estimated in the second step. These steps are repeated until the algorithm converges to the true values or until the amount of available observations is exhausted. This paper makes use of the latter technique.

The vector $\mathbf{b}$ as defined in (6) is assumed to be evolving as a Gauss-Markov model. As discussed previously, the states are assumed to be known at this step. The system evolution equations can therefore be expressed as

$$\begin{aligned}\mathbf{b}_k &= \mathbf{b}_{k-1} + \boldsymbol{\eta}_{k-1},\\ \mathbf{y}_k &= \mathbf{R}_k\mathbf{b}_k + \mathbf{e}_k,\end{aligned} \tag{19}$$

where $\boldsymbol{\eta}_k$ stands for the i.i.d Gaussian noise and $\mathbf{R}_k$ is as defined in (8). It is observed that (19) is a system of linear equations with additive Gaussian noise, and therefore, the Kalman filter is the optimal choice for the estimation of

parameter vector. The standard *predict* and *update* steps involved in Kalman filter are summarized as follows:

$$\begin{aligned}
\widehat{\mathbf{b}}_{k|k-1} &= \widehat{\mathbf{b}}_{k-1|k-1} + \boldsymbol{\eta}_k,\\
\mathbf{P}_{bb,k|k-1} &= \mathbf{P}_{bb,k-1|k-1} + \boldsymbol{\Sigma}_{\eta_k},\\
\mathbf{u}_k &= \mathbf{y}_k - \mathbf{R}_{f_k}\widehat{\mathbf{b}}_k,\\
\mathbf{K}_G &= \mathbf{P}_{bb,k|k-1}\mathbf{R}_{f_k}^T\left(\mathbf{R}_{f_k}\mathbf{P}_{bb,k|k-1}\mathbf{R}_{f_k}^T + \sigma_e^2\mathbf{I}^{-1}\right),\\
\widehat{\mathbf{b}}_{k|k} &= \widehat{\mathbf{b}}_{k|k-1} + \mathbf{K}_G\mathbf{u}_k,\\
\mathbf{P}_{bb,k|k} &= \left(\mathbf{I} - \mathbf{K}_G\mathbf{R}_{f_k}\right)\mathbf{P}_{bb,k|k-1},
\end{aligned} \tag{20}$$

where $\mathbf{K}_G$ denotes the Kalman gain and $\mathbf{P}$ represents the error covariance matrix.

The Kalman filter algorithm is based on an l_2-norm minimization criterion. As the gene networks are known to be highly sparse, the parameter vector is expected to have only a few nonzero values. A more accurate approach for estimating such a vector would be to introduce an additional constraint on its l_1-norm which is the core idea in compressed sensing [38, 44]. Such an l_1-norm constraint provides a unique solution to the underdetermined set of equations [45]. Therefore, instead of a simple l_2 norm minimization, the following constrained optimization problem is considered:

$$\min_{\widehat{\mathbf{b}}_k}\left\|\widehat{\mathbf{b}}_k - \mathbf{b}_k\right\|_2^2 \quad \text{s.t. } \left\|\widehat{\mathbf{b}}_k\right\| \le \epsilon. \tag{21}$$

The importance of this constraint can be judged by the fact that without it, the system would be rendered unidentifiable [43].

The problem (21) can be solved using a pseudomeasurement (PM) method which incorporates the inequality constraint (21) in the filtering process by assuming an artificial measurement $\|\mathbf{b}_k\|_1 - \epsilon = 0$. This is concisely expressed as

$$0 = \overline{\mathbf{R}}\widehat{\mathbf{b}}_k - \epsilon, \quad \overline{\mathbf{R}}_\tau = \left[\operatorname{sign}\left(\widehat{\mathbf{b}}_\tau(1)\right), \ldots, \operatorname{sign}\left(\widehat{\mathbf{b}}_\tau(N)\right)\right]. \tag{22}$$

The value of the covariance matrix $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2\mathbf{I}$ of the pseudonoise ϵ is selected in a similar manner as the process noise covariance in the EKF algorithm. However, it is found that large values of variances, that is, $\sigma_\epsilon^2 \ge 100$, prove sufficient in most cases [38]. Further details on selecting these parameters can be found in [38, 46]. The PM method solves (21) in a recursive manner for K_τ iterations using the following steps:

$$\begin{aligned}
\mathbf{K}_G^\tau &= \mathbf{P}_\tau\overline{\mathbf{R}}_\tau^T\left(\overline{\mathbf{R}}_\tau\mathbf{P}_\tau\overline{\mathbf{R}}_\tau^T + \boldsymbol{\Sigma}_\epsilon\right)^{-1},\\
\widehat{\mathbf{b}}_{\tau+1} &= \left(\mathbf{I} - \mathbf{K}_G^\tau\overline{\mathbf{R}}_\tau\right)\widehat{\mathbf{b}}_\tau,\\
\mathbf{P}_{\tau+1} &= \left(\mathbf{I} - \mathbf{K}_G^\tau\overline{\mathbf{R}}_\tau\right)\mathbf{P}_\tau.
\end{aligned} \tag{23}$$

At each kth time instant, $\mathbf{P}_{bb,k|k}$ and $\widehat{\mathbf{b}}_{k|k}$ obtained from (20) are considered as initial values; that is, $\widehat{\mathbf{b}}^1 = \widehat{\mathbf{b}}_{k|k}$ and $\mathbf{P}_1 = \mathbf{P}_{bb,k|k}$ which is the error covariance matrix. The value of K_τ is equal to the number of constraints, that is, the expected number of nonzero $\mathbf{b}_{mn}$s in the system. Possible ways for calculating K_τ include minimum description length (MDL) principle and Bayesian information criterion (BIC).

(1) Input time series data set $\mathbf{y}$.
(2) Initialize $I, K, \phi_0, \mathbf{x}_0$.
(3) **for** $k = 1, \ldots, K$ **do**
(4) Find the state estimates using CKF following the time and measurement update steps in Section 3.
(5) Estimate parameters $\widehat{\mathbf{b}}_k$ from $\mathbf{x}_k$ and $\mathbf{y}_k$ using (20).
(6) **for** $\tau = 1, \ldots, K_\tau$ **do**
(7) Update the parameters $\widehat{\mathbf{b}}_k$ using (23).
(8) **end for**
(9) **end for**
(10) **return**

Algorithm 1: Network inference: CKFS.

3.3. Inference Algorithm. The network inference algorithm is summarized in Algorithm 1. The algorithm consists of a recursive process which repeats itself for the number of observations present in the time-series data. For each time sample, the state estimate is obtained using the CKF, and the parameter estimate is obtained using the KF. Since the parameters are expected to be sparse, the estimates are then refined further using the CSKF algorithm. This iterative process results in a simple and accurate algorithm for gene network inference while considering a complex nonlinear model.

4. Cramér-Rao Bound

The performance of an estimator can be judged by comparing it with theoretical lower bounds proposed in parameter estimation theory. The CRB establishes a lower bound on the MSE of an unbiased estimator [47]. In particular, the CRB states that the covariance matrix of the estimator $\widehat{\mathbf{b}}$ is lower bounded by

$$\mathbb{E}\left[\left(\widehat{\mathbf{b}} - \mathbf{b}\right)\left(\widehat{\mathbf{b}} - \mathbf{b}\right)^T\right] \succeq \left[\mathbf{I}(\mathbf{b})\right]^{-1}, \tag{24}$$

where the matrix inequality $\succeq$ is to be interpreted in the semidefinite sense and $\mathbf{I}(\mathbf{b})$ is the Fisher information matrix (FIM)

$$\mathbf{I}(\mathbf{b}) = \mathbb{E}\left[\left(\frac{\partial \ln f(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)\left(\frac{\partial \ln f(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)^T\right]. \tag{25}$$

The CRB for gene network inference can be calculated as follows. By stacking all the observations for $k = 1, \ldots, K$, (7) can be written compactly in the matrix form

$$\mathbf{y} = \mathbf{R}\mathbf{b} + \mathbf{e}, \tag{26}$$

where $\mathbf{y} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_K^T]^T$, $\mathbf{R} = [\mathbf{R}_1^T, \ldots, \mathbf{R}_K^T]^T$, and $\mathbf{e} = [\mathbf{e}_1^T, \ldots, \mathbf{e}_K^T]^T$. The PDF $p(\mathbf{y} \mid \mathbf{b})$ is expressed as

$$p(\mathbf{y} \mid \mathbf{b}) = C \exp\left(-\frac{(\mathbf{y} - \mathbf{Rb})^T (\mathbf{y} - \mathbf{Rb})}{2\sigma_e^2}\right), \quad (27)$$

where C is a constant. The derivative of $\ln p(\mathbf{y} \mid \mathbf{b})$ can be expressed as

$$\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}} = -\frac{\partial}{\partial \mathbf{b}}\left[\frac{(\mathbf{y} - \mathbf{Rb})^T (\mathbf{y} - \mathbf{Rb})}{\sigma_e^2}\right] = \frac{\mathbf{R}^T\mathbf{y} - \mathbf{R}^T\mathbf{Rb}}{\sigma_e^2}. \quad (28)$$

It now follows that

$$\left(\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)\left(\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)^T = \frac{\mathbf{R}^T (\mathbf{y} - \mathbf{Rb})(\mathbf{y} - \mathbf{Rb})^T \mathbf{R}}{\sigma_e^4}. \quad (29)$$

By taking the expectation of (29), the FIM in (25) is given by

$$\mathbf{I}(\mathbf{b}) = \frac{\mathbf{R}^T\mathbf{R}}{\sigma_e^2}. \quad (30)$$

The inverse of the FIM in (30) can be used to place a lower bound on the estimation error of the parameter vector $\mathbf{b}$. Figure 2 shows the comparison of MSE of CKFS algorithm with CRB as a function of number of samples K for one representative gene from the eight-gene network considered in Section 5.1. It is observed that the MSE of the estimated parameters decreases with increasing number of samples.

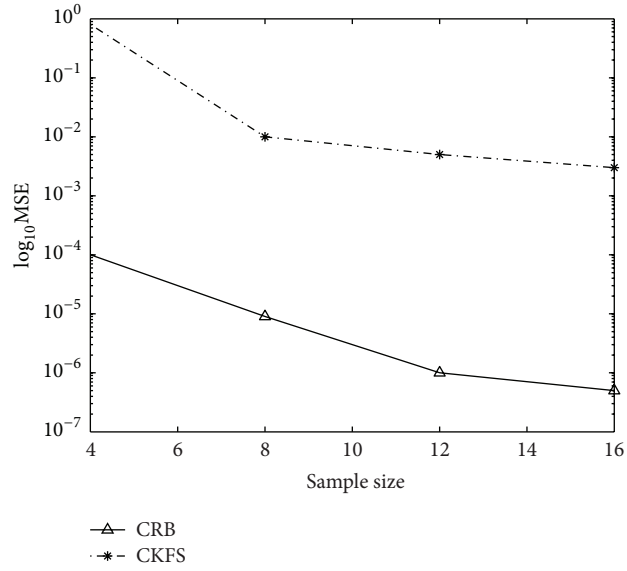

Figure 2: Cramér-Rao bound on the estimation of parameters. The MSE for one of the representative θ is shown here for a network consisting of 8 vertices.

5. Results and Discussion

The simulation results of the CKFS algorithm are discussed in this section. The performance is first tested on synthetic data obtained from randomly generated Boolean networks under various scenarios and performance metrics. The algorithm is then assessed on the DREAM4 networks and the IRMA network.

5.1. Synthetic Data. Time-series data is produced from randomly generated Boolean networks using the system model (3) and (5). Two scenarios are considered for this purpose.

First, the comparison is performed by varying the number of sample size while keeping the network size fixed. The gene network consists of 8 genes and 20 vertices. In terms of network estimation, if the algorithm predicts an edge between two nodes which may not be present in reality, an error, referred to as *false alarm error* (F), is said to have occurred. Another situation is the indication of the absence of a vertex in the graph which in fact is present in the real network. This kind of error is termed *missed detection* (M). The summation of these two errors normalized over the total number of vertices in the network yields the *Hamming distance*. It is also important to consider the probability of predicting the true connections correctly which will be assessed by the *true connections* (T) metric. An algorithm with low Hamming distance and small false alarm error is particularly desirable as predicting an edge erroneously can be troublesome for biologists. True connections indicate the reliability of the predictions. Figure 3 illustrates the performance of the CKFS algorithm and that of the EKF algorithm proposed in [34] in terms of the metrics described above. It is important to mention here that the same system model is assumed by both CKFS and EKF algorithms for the purpose of this simulation. These metrics are the same as those used in [15]. The variances of both the system and measurement noises, σ_w^2 and σ_v^2, respectively, are taken to be 10^{-5} in all the simulations and are assumed to be known. It is noticed that EKF has a slightly lower false alarm rate when the number of samples is small; however, as the number of samples increases, CKFS yields a lower false alarm error. The Hamming distance for CKFS is also smaller than EKF indicating lesser cumulative error. True connections show a consistent behavior for the two algorithms when the number of samples is increased where CKFS is able to predict connections more accurately. These experiments show the superiority of CKFS in terms of lower error rate.

To obtain a more rigorous evaluation, the performance of algorithms is then compared in a scenario which considers the sample size to be fixed and assumes networks of different sizes. The receiver operating characteristic (ROC) curves are plotted as performance measures. A higher area under the ROC curve (AUROC) shows more true positives for a given false positive, and therefore, indicates better classification. The performance of CKFS(N, E, K) and EKF(N, E, K) is shown in Figure 4, where N stands for the number of nodes,

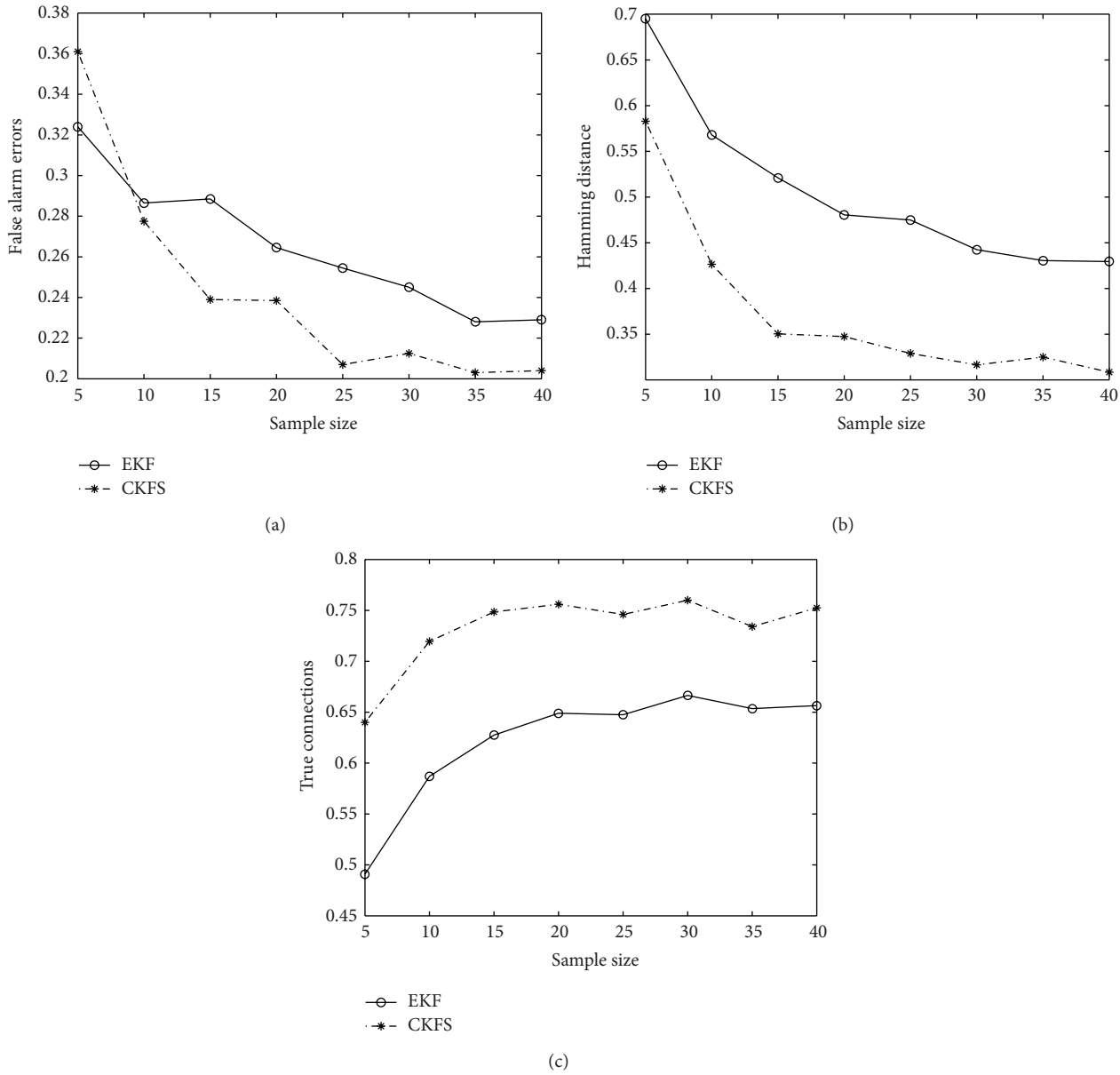

Figure 3: (a), (b), and (c) False alarm errors, Hamming distance, and true connections. The synthetic networks consist of 8 vertices and 20 edges. The metric is normalized over the number of edges. CKFS gives lower error and predicts more true connections with the increase in the sample size of data.

E represents the number of edges, and K denotes the time points. It is observed that the CKFS exhibits superior performance than the EKF for networks of different sizes.

The complexity of the two algorithms is compared for synthetically generated networks with number of genes equal to 10, 20, 30, and 40. The sample size is kept to 50 time points for each of these networks, and the run time for EKF and CKFS algorithms is calculated as shown in Table 1. It is noted that EKF is faster for smaller network sizes, but as the network size increases, the run time gets much larger than that for CKFS. The main reason for this is that EKF [34] estimates the states and parameters by stacking them together which requires large-sized matrix multiplications at each iteration. The benefit associated with performing dual estimation, as in CKFS, is that the parameters are estimated separately from the states. Since the system is linear and one-to-one for parameters, inversion of much smaller matrices can be performed reducing the computational complexity of CKFS algorithm. CKFS is therefore particularly attractive for large-sized networks.

5.2. DREAM4 Gene Networks. Several *in silico* networks have been produced in order to benchmark the performance of gene network inference algorithms. dialogue on reverse engineering assessment and methods (DREAM) *in silico* networks serve as one of the popular methods used for this

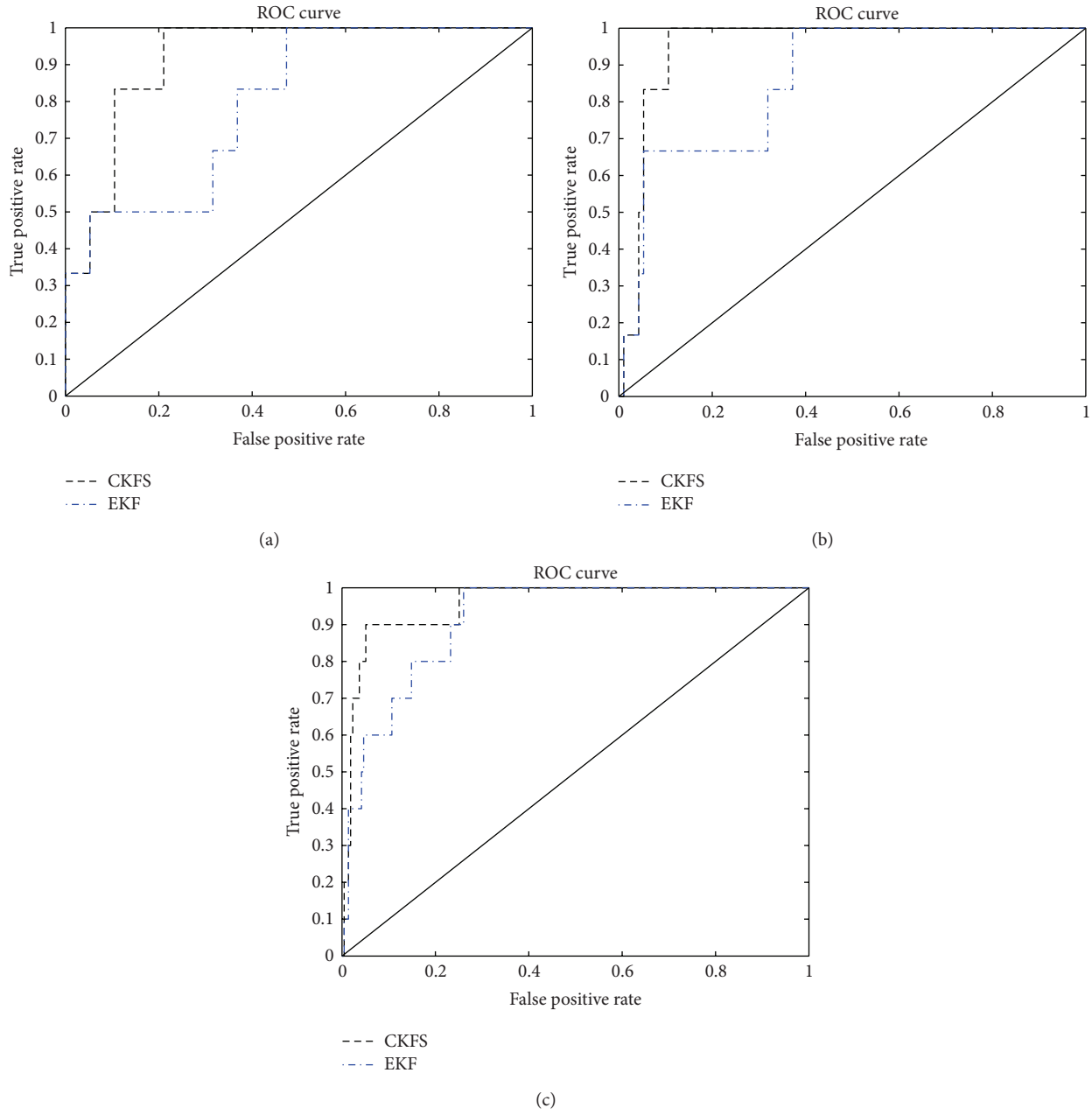

Figure 4: ROC curves for the performance of CKFS and EKF using synthetic data. (N, E, K) (a), (b), and (c) $(5, 10, 20)$, $(10, 12, 20)$, and $(15, 19, 20)$. The area under the ROC curve for CKFS is more than that for EKF for various sized networks.

Table 1: Run time in seconds for EKF and CKFS algorithms for varying network sizes for synthetically generated data. The number of sample points is fixed to 50.

Number of genes	10	20	30	40
EKF	0.16	1.9	16.5	84
CKFS	1.2	4.3	11.5	24.1

purpose [39, 48]. In this section, the performance of the CKFS algorithm is evaluated using the 10-gene and 100-gene networks released online by the DREAM4 challenge. Five networks are produced using the known GRNs of *Escherichia coli* and *Saccharomyces cerevisiae*. The data sets for each of 10-gene network consists of 21 data points for five different perturbations. The inference is performed by using all the perturbations. The 100-gene network consists of data sets for ten perturbations. AUROC and area under the precision-recall curve (AUPR) are calculated for the five networks of both the data sets and shown in Tables 2 and 3, respectively. The quantities, *precision* and *recall*, are defined as $P = T/(T + F)$ and $R = T/(T + M)$, respectively. For comparison purposes, the values of the two quantities for time-series network identification (TSNI) algorithm that

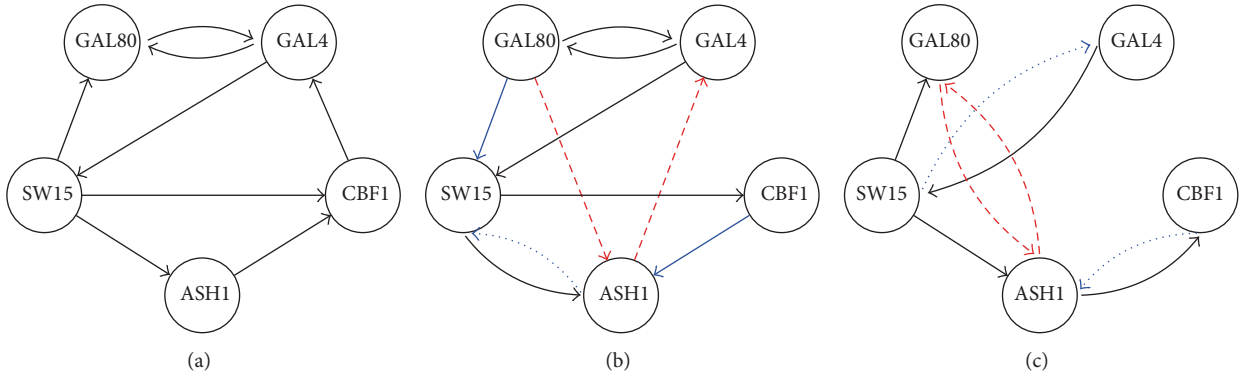

Figure 5: The inferred IRMA networks. (a), (b), and (c) Gold standard, inferred network using CKFS, and inferred network using ODE [39, 40]. Black arrows indicate true connections, blue arrows indicate the edges that are correct, but their directions are reversed, and red arrows indicate false positives.

Table 2: Area under the ROC curve (AUROC) and area under the PR curve (AUPR) for DREAM4 10-gene networks for the five different networks.

Algorithm	Network 1	Network 2	Network 3	Network 4	Network 5
ODE [39]	0.62 (0.27)	0.63 (0.32)	0.58 (0.21)	0.63 (0.23)	0.68 (0.25)
CKFS	0.63 (0.40)	0.67 (0.50)	0.72 (0.50)	0.75 (0.49)	0.81 (0.42)
Random [39]	0.55 (0.18)	0.55 (0.19)	0.55 (0.17)	0.57 (0.17)	0.56 (0.16)

Table 3: Area under the ROC curve (AUROC) and area under the PR curve (AUPR) for DREAM4 100-gene networks for the five different networks.

Algorithm	Network 1	Network 2	Network 3	Network 4	Network 5
ODE [39]	0.55 (0.02)	0.55 (0.03)	0.60 (0.03)	0.54 (0.02)	0.59 (0.03)
CKFS	0.67 (0.13)	0.57 (0.08)	0.60 (0.10)	0.62 (0.10)	0.60 (0.07)
Random [39]	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)

exploits ordinary differential equations are also given [39]. The CKFS algorithm is found to perform significantly better than the TSNI algorithm.

5.3. IRMA Gene Network. In addition to synthetic data, it is imperative to test the algorithms using real biological data. In this subsection, the performance of the CKFS algorithm is assessed using the *in vivo* reverse-engineering and modeling assessment (IRMA) network [40]. This network consists of five genes. Galactose activates the gene expression in the network, whereas glucose deactivates it. The cells are grown in the presence of galactose and then switched to glucose to obtain the switch-off data which represents the expressive samples at 21 time points. The switch-on data consists of 16 sample points and is obtained by growing the cells in a glucose medium and then changing to galactose. The system and measurement noise variances for the CKFS are assumed to be identical as in the previous simulations. Figure 5 shows the inferred network, the gold standard, and the network inferred using TSNI. It is observed that the CKFS algorithm succeeds to predict most of the interactions while giving lower false positives.

6. Conclusions

This paper presents a novel algorithm for inferring gene regulatory networks from time-series data. Gene regulation is assumed to follow a nonlinear state evolution model. The parameters of the system, which indicate the inhibitory or excitatory relationships between the genes, are estimated using compressed sensing-based Kalman filtering. The sparsity constraint on the parameters is crucial because the genes are known to interact with few other genes only. The use of CKF and the dual estimation of states and parameters renders the algorithm computationally efficient. The performance of CKFS is evaluated for synthetic data for different network sizes as well as varying sample points. ROC curves, Hamming distance, and true positives are used for comparing the accuracy of inferred network with EKF. It is observed that CKFS outperforms the EKF algorithm. In addition, CKFS

gives advantages over EKF in terms of smaller run time for large networks. The Cramér-Rao lower bound is also determined for the parameters of the model and compared with the MSE performance of the proposed algorithm. Assessment using DREAM4 10-gene and 100-gene networks and IRMA network data corroborates the superior performance of CKFS. Future research directions include incorporating the estimation of model order in the network inference algorithm.

Acknowledgments

This work was supported by US National Science Foundation (NSF) Grant 0915444 and QNRF-NPRP Grant 09-874-3-235. The material in this paper was presented in part at the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS), San Antonio, TX, USA, December 2011.

References

[1] X. Zhou, X. Wang, and E. R. Dougherty, *Genomic Networks: Statistical Inference from Microarray Data*, John Wiley & Sons, New York, NY, USA, 2006.

[2] H. Kitano, "Computational systems biology," *Nature*, vol. 420, pp. 206–210, 2002.

[3] X. Zhou and S. T. C. Wong, *Computational Systems Bioinformatics*, World Scientific, River Edge, NJ, USA, 2008.

[4] X. Cai and X. Wang, "Stochastic modeling and simulation of gene networks," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 27–36, 2007.

[5] D. Yue, J. Meng, M. Lu, C. L. P. Chen, M. Guo, and Y. Huang, "Understanding micro-RNA regulation: a computational perspective," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 77–88, 2012.

[6] R. Pal, S. Bhattacharya, and M. U. Caglar, "Robust approaches for genetic regulatory network modeling and intervention: a review of recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 66–76, 2012.

[7] H. Hache, H. Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 617281, 2009.

[8] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, no. 6, p. 9, 2007.

[9] H. D. Jong, "Modeling and simulation of genetic regulatoy systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[10] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, no. 1, pp. i248–i256, 2004.

[11] C. D. Giurcaneanu, I. Tabus, and J. Astola, "Clustering time series gene expression data based on sum-of-exponentials fitting," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 8, Article ID 358568, pp. 1159–1173, 2005.

[12] C. D. Giurcaneanu, I. Tabus, J. Astola, J. Ollila, and M. Vihinen, "Fast iterative gene clustering based on information theoretic criteria for selecting the cluster structure," *Journal of Computational Biology*, vol. 11, no. 4, pp. 660–682, 2004.

[13] X. Cai and G. B. Giannakis, "Identifying differentially expressed genes in microarray experiments with model-based variance estimation," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2418–2426, 2006.

[14] X. Zhou, X. Wang, and E. R. Dougherty, "Gene clustering based on cluster-wide mutual information," *Journal of Computational Biology*, vol. 11, no. 1, pp. 151–165, 2004.

[15] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using informationtheoretic criteria," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 2, pp. 262–274, 2008.

[16] J. Dougherty, I. Tabus, and J. Astola, "Inference of gene regulatory networks based on a universal minimum description length," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2008, Article ID 482090, 2008.

[17] L. Qian, H. Wang, and E. R. Dougherty, "Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3327–3339, 2008.

[18] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics*, vol. 22, no. 17, pp. 2129–2135, 2006.

[19] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformatics*, vol. 20, no. 17, pp. 2918–2927, 2004.

[20] J. Meng, M. Lu, Y. Chen, S.-J. Gao, and Y. Huang, "Robust inference of the context specific structure and temporal dynamics of gene regulatory network," *BMC Genomics*, vol. 11, no. 3, p. S11, 2010.

[21] Y. Zhang, Z. Deng, H. Jiang, and P. Jia, "Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural em.," in *DILS*, S. C. Boulakia and V. Tannen, Eds., vol. 4544 of *Lecture Notes in Computer Science*, pp. 204–214, Springer, New York, NY, USA, 2007.

[22] K. Murphy and S. Mian, Modeling gene expression data using dynamic Bayesian networks, University of California, Berkeley, Calif, USA, 2001.

[23] H. Liu, D. Yue, L. Zhang, Y. Chen, S. J. Gao, and Y. Huang, "A Bayesian approach for identifying miRNA targets by combining sequence prediction and gene expression profiling," *BMC Genomics*, vol. 11, no. 3, p. S12, 2010.

[24] Y. Huang, J. Wang, J. Zhang, M. Sanchez, and Y. Wang, "Bayesian inference of genetic regulatory networks from time series microarray data using dynamic Bayesian networks," *Journal of Multimedia*, vol. 2, no. 3, pp. 46–56, 2007.

[25] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. D'Alché-Buc, "Gene networks inference using dynamic Bayesian networks," *Bioinformatics*, vol. 19, no. 2, pp. ii138–ii148, 2003.

[26] C. Rangel, D. L. Wild, F. Falciani, Z. Ghahramani, and A. Gaiba, "A. modelling biological responses using gene expression profiling and linear dynamical systems," *Bioinformatics*, pp. 349–356, 2005.

[27] M. Quach, N. Brunel, and F. d'Alch Buc, "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference," *Bioinformatics*, vol. 23, no. 23, pp. 3209–3216, 2007.

[28] F.-X. Wu, W.-J. Zhang, and A. J. Kusalik, "Modeling gene expression from microarray expression data with state-space

equations," in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, Eds., pp. 581–592, World Scientific, River Edge, NJ, USA, 2004.

[29] R. Yamaguchi, S. Yoshida, S. Imoto, T. Higuchi, and S. Miyano, "Finding module-based gene networks with state-space modelsmining high-dimensional and short time-course gene expression data," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 37–46, 2007.

[30] O. Hirose, R. Yoshida, S. Imoto et al., "Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models," *Bioinformatics*, vol. 24, no. 7, pp. 932–942, 2008.

[31] J. Angus, M. Beal, J. Li, C. Rangel, and D. Wild, "Inferring transcriptional networks using prior biological knowledge and constrained state-space models," in *Learning and Inference in Computational Systems Biology*, N. Lawrence, M. Girolami, M. Rattray, and G. Sanguinetti, Eds., pp. 117–152, MIT Press, Cambridge, UK, 2010.

[32] C. Rangel, J. Angus, Z. Ghahramani et al., "Modeling T-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[33] A. Noor, E. Serpedin, M. N. Nounou, and H. N. Nounou, "Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1203–1211, 2012.

[34] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An extended kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, pp. 410–419, 2009.

[35] A. Noor, E. Serpedin, M. Nounou, H. Nounou, N. Mohamed, and L. Chouchane, "An overview of the statistical methods used for inferring gene regulatory networks and proteinprotein interaction networks," *Advances in Bioinformatics*, vol. 2013, Article ID 953814, 12 pages, 2013.

[36] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[37] A. Noor, E. Serpedin, M. N. Nounou, and H. N. Nounou, "A cubature Kalman filter approach for inferring gene regulatory networks using time series data," in *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS '11)*, pp. 25–28, 2011.

[38] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded rseudo-measurement norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, 2010.

[39] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," *Interface Focus*, pp. 857–870, 2011.

[40] I. Cantone, L. Marucci, F. Iorio et al., "A yeast synthetic network for *in vivo* assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172–181, 2009.

[41] Y. Huang, I. M. Tienda-Luna, and Y. Wang, "Reverse engineering gene regulatory networks: a survey of statistical models," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 76–97, 2009.

[42] Z. Wang, F. Yang, D. W. C. Ho, S. Swift, A. Tucker, and X. Liu, "Stochastic dynamic modeling of short gene expression time-series data," *IEEE Transactions on Nanobioscience*, vol. 7, no. 1, pp. 44–55, 2008.

[43] H. Xiong and Y. Choe, "Structural systems identification of genetic regulatory networks," *Bioinformatics*, vol. 24, no. 4, pp. 553–560, 2008.

[44] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[45] E. J. Cands and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[46] J. D. Geeter, H. V. Brussel, and J. D. Schutter, "A smoothly constrained Kalman filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1171–1177, 1997.

[47] S. M. Kay, *Fundamentals of Statistical Signal Processing. Estimation Theory*, Prentice-Hall, New York, NY, USA, 1993.

[48] http://wiki.c2b2.columbia.edu/dream/.