Online Inference for Time-varying Temporal Dependency Discovery from Time Series

Chunqiu Zeng, Qing Wang, Wentao Wang, Tao Li
School of Computing and Information Science
Florida International University
Miami, FL, USA
Email: {czeng001, qwang028, wwang041, taoli}@cs.fiu.edu

Larisa Shwartz
Operational Innovations
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
Email:lshwart@us.ibm.com

***Abstract*—Large-scale time series data are prevalent across diverse application domains including system management, biomedical informatics, social networks, finance, etc. Temporal dependency discovery performs an essential part to identify the hidden interactions among the observed time series and helps to gain more insight into the behavior of the applications. However, the time-varying sparsity of the interactions among time series often poses a big challenge to temporal dependency discovery in practice. This paper formulates the temporal dependency problem with a novel Bayesian model allowing for both the sparsity and evolution of the hidden interactions among the observed time series. Taking advantage of the Bayesian modeling, an online inference method is proposed for time-varying temporal dependency discovery. Extensive empirical studies on both the synthetic and real application time series data are conducted to demonstrate the effectiveness and the efficiency of the proposed method.**

I. Introduction

Large-scale multivariate time series data are prevalent across diverse application domains including system management, biomedical informatics, social networks, finance, etc. Temporal dependency discovery from multivariate time series has been recognized as one of the key tasks in time series analysis. Taking system management as an example, the time series data (e.g., CPU utilization, memory usage) are collected by monitoring the internal components of a large-scale distributed information system, where a great variety of involved components work together in a highly complex and coordinated manner. Temporal dependency discovered from the monitoring time series reveals important dependency relationships among components and has established its significance in system anomaly detection [1], root cause analysis for system faults [2], etc.

Mining temporal dependency structure among time series has been extensively studied in the past decades. The inference of temporal dependencies can be broadly categorized into two different frameworks: dynamic Bayesian Network [3][4] and Granger Causality [5][6][7]. An extensive comparison study between these two types of frameworks is presented in [8]. The Granger Causality framework is famous for its simplicity, robustness and extendability, and becomes increasingly popular in practice [9]. Taking these advantages into account, this paper mainly focuses on the the Granger Causality framework.

The intuitive idea of Granger Causality is that if the time series A Granger causes the time series B, the future value prediction of B can be improved by giving the value of A. The prediction is typically attained by inferring the distribution of time series. Since modeling the distribution for multivariate time series is extremely difficult while linear regression model is a simple and robust approach, regression model has evolved to be one of the principal approaches for Granger Causality. Specifically, to predict the future value of B, one regression model built only on the past values of B should be statistically significantly less accurate than the regression model inferred by giving the past values of both A and B.

Based on the regression model, two major approaches have been developed to discover the Granger Causal relationship for multivariate time series. The first approach employs the statistical significance test to identify the possible interactions among time series, where the nonzero coefficients of the regression model have been verified by hypothesis test. The second method, named Lasso-Granger, determines the Granger Causality from the time series by inferring the regression model with Lasso regularization. The main idea of Lasso-Granger is to impose a L_1 regularization penalty on the regression coefficients, so that it can effectively identify the sparse Granger Causality especially in high dimensions. It has been shown that both approaches are consistent in low dimensions, while only Lasso-Granger is consistent in high dimensions [10]. Our work is mainly based on the Lasso-Granger approach.

Most existing works related to Lasso-Granger method have been developed for Granger Causality inference by assuming that the latent causal relationships for multivariate time series are fixed yet unknown. However, this assumption rarely holds in practice, since real-world problems often involve underlying processes that are dynamically evolving over time. A scenario of system management, shown in Fig. 1, is taken as an example. In this example, multiple instances of memory intensive applications are running on a server. At the early stage, the memory of this server is sufficient for supporting running application instances. However, if the number of application instances keeps increasing and the required memory exceeds the capacity of the server, then the server has to take advantage of its virtual memory (the virtual memory is built on the

disk storage) to support the running application instances. As a result, an dynamic dependency relationship exists between the number of running application instances and the disk I/O (the number of bytes read from or written to the disk): at the beginning, no obvious relationship occurs between them, while strong relationship is indicated after the number of running application instances increases beyond a threshold related to the memory capacity. It turns out to be critical if the dynamically changing behaviors of the temporal dependency for time series can be identified instantly.

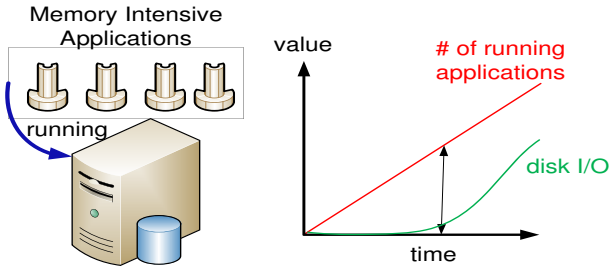


Fig. 1: The correlation between the number of memory intensive applications and the disk I/O changes dynamically over time in the system management.

In this paper, to capture the dynamical change of casual relationships among the time series, we propose a time-varying temporal dependency model based on Lasso-Granger Casuality and develop effective online inference algorithms using particle learning. The dynamical change behaviors of the temporal dependency is explicitly modeled as a set of random walk particles. The fully adaptive inference strategy of particle learning allows our model to effectively capture the varying dependency and learn the latent parameters simultaneously. We conduct empirical studies on both synthetic and real dataset. The experimental result demonstrate the effectiveness of our proposed approach.

The remainder of this paper is organized as follows. In Section II, most relevant existing works are briefly summarized. We formulate the problem for identifying time-varying temporal dependency in Section III. The solution based on particle learning for online model inference is presented in Section IV. Extensive empirical evaluation results are reported in Section V. Finally, we conclude our work and the future work in Section VI.

II. Related Work

Temporal data are a collection of data items associated with time stamps. In light of the different types of data items, temporal data are divided into two categories, i.e., time series and event data [31]. Our work focuses on time series data.

One of the major data mining tasks for time series data is to reveal the underlying temporal causal relationship among the time series. Currently, two popular approaches prevail in the literature for causal relationship inference from time series data. One is the Bayesian network inference approach [11][3][4][12], while the other approach is the Granger Causality [5][6][7]. Comparing with Bayesian network, Granger Causality is more straightforward, robust and extendable. Our proposed method is more related to the approach based on Granger Causality.

Since Granger causality is originally defined for a pair of time series, the causal relationship identification among multivariate time series can not be addressed directly until the appearance of some pioneering work on combining the notion of Granger causality with graphical model [13]. The Granger causality inference among multivariate time series is typically developed by two techniques, i.e., statistical significance test and Lasso-Granger [7]. Lasso-Granger is more preferable due to its robust performance even in high dimensions [14][30]. Our method takes the advantage of Lasso-Granger, but conducts the inference from the Bayesian perspective in a sequential online mode, borrowing the idea of Bayesian Lasso [15]. However, most of these methods assume a constant dependency structure among time series.

In order to capture the dynamic temporal dependency typically happening in real practice, a hidden Markov model regression [16] and time-varying dynamic Bayesian network [12] have been proposed. However, the number of hidden states in [16] and the decaying weights in [12] are difficult to determine without any domain knowledge. Furthermore, both methods infer the underlying dependency structure in an offline mode. In this paper, we explicitly model the dynamic changes of the underlying temporal dependencies and infer the model in an online manner.

Our proposed model makes use of sequential online inference to infer the latent state and learn unknown parameters simultaneously. Popular sequential learning methods include sequential monte carlo sampling [17], and particle learning [18].

Sequential Monte Carlo (SMC) methods consist of a set of Monte Carlo methodologies to solve the filtering problem [19]. It provides a set of simulation based methods for computing the posterior distribution. These methods allow inference of full posterior distributions in general state space models, which may be both nonlinear and non-Gaussian.

Particle learning provides state filtering, sequential parameter learning and smoothing in a general class of state space models [18]. Particle learning is for approximating the sequence of filtering and smoothing distributions in light of parameter uncertainty for a wide class of state space models. The central idea behind particle learning is the creation of a particle algorithm that directly samples from the particle approximation to the joint posterior distribution of states and conditional sufficient statistics for fixed parameters in a fully-adapted `resample-propagate` framework. We borrow the idea of particle learning for both latent state inference and parameter learning.

III. Problem Formulation

In this section, we formally define the Granger Causality problem from a Bayesian perspective first, and then model the time-varying temporal dependency problem. Some important notations mentioned in this paper are summarized in Table I.

TABLE I: Important Notations

Notation	Description
$\mathbf{Y}$	a set of time series.
K	the number of time series in $\mathbf{Y}$.
T	the length of time series.
$\mathbf{y}_i$	the i^{th} time series.
$\mathbf{y}_{j,t}$	the value of j^{th} time series at time t.
$\mathbf{y}_{\cdot,t}$	a column vector containing the values of all time series at time t.
$\mathbf{x}_t$	a column vector built from all time series with time lag L at time t.
$\mathcal{P}_{j,t}$	the set of particles for predicting $\mathbf{y}_{j,t}$ at time t and $\mathcal{P}_{j,t}^{(i)}$ is the i^{th} particle of $\mathcal{P}_{j,t}$.
$\mathbf{W}^l$	the coefficient matrix for time lag l in VAR model.
$\mathbf{w}_j$	the coefficient vector used to predict j^{th} time series value in Bayesian Lasso model.
$\mathbf{w}_{j,t}$	the coefficient vector used to predict j^{th} time series value at time t in time-varying Bayesian Lasso model.
$\mathbf{c}_{\mathbf{w}_j}$	the constant part of $\mathbf{w}_{j,t}$.
$\delta_{\mathbf{w}_{j,t}}$	the drifting part of $\mathbf{w}_{j,t}$.
$\eta_{j,t}$	the standard Gaussian random walk at time t, given $\eta_{j,t-1}$.
θ_j	the scale parameters used to compute $\delta_{\mathbf{w}_{j,t}}$.
σ_j^2	the variance of value prediction for the j^{th} time series.
α, β	the hyper parameters determine the distribution of σ_j^2.
$\mu_{\mathbf{w}}$	the hyper parameters determine the distribution of $\mathbf{w}_j$ in Bayesian Lasso model.
$\mu_{\mathbf{c}}$	the hyper parameters determine the distribution of $\mathbf{c}_{\mathbf{w}_j}$.
μ_θ	the hyper parameters determine the distribution of θ_j.
γ_p^2	the augmented random variable for $\mathbf{w}_j$, with λ.
$\gamma_{c,p}^2$	the augmented random variable for $\mathbf{c}_{\mathbf{w}_j}$, with λ_1.
$\gamma_{\theta,p}^2$	the augmented random variable for θ_j, with λ_2.
λ, λ_1, λ_2	the Lasso penalty parameters for $\mathbf{w}_j$, $\mathbf{c}_{\mathbf{w}_j}$ and θ_j, respectively.

A. Basic Concepts and Terminologies

Let $\mathbf{Y}$ be a set of time series, denoted as $\mathbf{Y} = \{\mathbf{y}_i | 1 \leq i \leq K\}$, where K is the number of time series in $\mathbf{Y}$ and $\mathbf{y}_i$ is the i^{th} time series. Assume $\mathbf{y}_{i,t} \in R$ to be the value of the i^{th} time series at time t, where $0 \leq t \leq T$. The time series $\mathbf{y}_j$ is supposed to be caused by another time series $\mathbf{y}_i$ in terms of Granger Causality, denoted as $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$, if and only if the regression for $\mathbf{y}_j$ using the past values of both $\mathbf{y}_j$ and $\mathbf{y}_i$ gains statistically significant improvement in terms of accuracy comparing with doing so with past values of $\mathbf{y}_j$ only. The Granger causal relationship among the set of time series $\mathbf{Y}$ is formulated as a directed graph G, where each vertex of G corresponds to a time series, and an edge exists directed from $\mathbf{y}_i$ to $\mathbf{y}_j$ if $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$.

In practice, the inference of Ganger causality is usually achieved by fitting the time series data $\mathbf{Y}$ with a Vector Auto-Regression (VAR) model. Let $\mathbf{y}_{\cdot,t} = (\mathbf{y}_{1,t}, ..., \mathbf{y}_{K,t})^\intercal$, a column vector containing the values of K time series at time t. Given the maximum time lag L, the VAR model is expressed as follows,

$$\mathbf{y}_{\cdot,t} = \sum_{l=1}^{L} (\mathbf{W}^l)^\intercal \mathbf{y}_{\cdot,t-l} + \epsilon, \tag{1}$$

where $\mathbf{W}^l$ is $K \times K$ coefficient matrix for time lag l, and ϵ is a $K \times 1$ vector, describing the random noise. The nonzero value of $\mathbf{W}_{ij}^l$ indicates $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. A statistics test [7] is applied to determine the nonzero values in $\mathbf{W}^l$, based on the VAR model shown in Equation 1. However, the combinational explosion for the statistics test on time series pairs brings about its inefficiency for Granger causality inference, especially analyzing time series data with high dimension.

Lasso-Granger provides a more efficient and consistent way to infer the Granger causal relation among time series, where L_1 regularization is imposed for addressing sparsity issue in high dimensional time series data [7]. Specifically, the coefficient matrix $\mathbf{W}^l$ is obtained by minimizing the following objective function,

$$\min_{\{\mathbf{W}^l\}} \sum_{t=L+1}^{T} \| \mathbf{y}_{\cdot,t} - \sum_{l=1}^{L} (\mathbf{W}^l)^\intercal \mathbf{y}_{\cdot,t-l} \|_2^2 + \lambda \sum_{l=1}^{L} \| \mathbf{W}^l \|_1, \tag{2}$$

where λ is the penalty parameter, which determines the sparsity of the coefficient matrix $\mathbf{W}^l$.

In Equation 2, Lasso-Granger provides regression for K variables, where each variable is expressed as a linear function of its own past values and past values of all other variables with L_1 regularization. To be simplified, we focus on the regression for one arbitrarily given variable $\mathbf{y}_j$, and the regression of other variables can be derived in a similar way.

Let $\mathbf{x}_t = vec([\mathbf{y}_{\cdot,t-1}, \mathbf{y}_{\cdot,t-2}, ..., \mathbf{y}_{\cdot,t-L}])$, where $vec(\cdot)$ is an operator to convert a matrix into a vector by stacking column vectors. The Lasso regression for the variable $\mathbf{y}_j$ is expressed as follows,

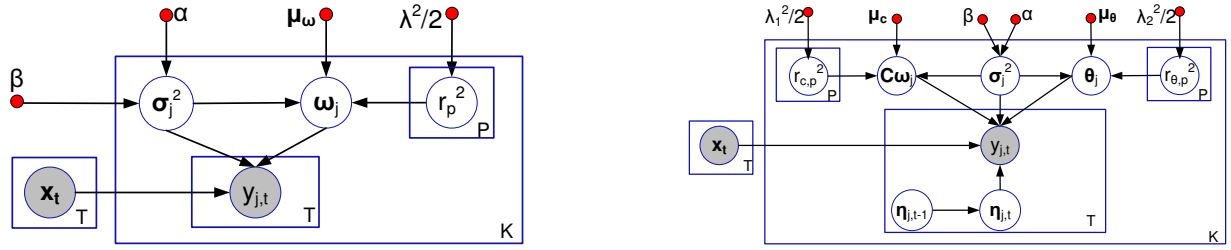
$$\min_{\mathbf{w}_j} \sum_{t=L+1}^{T} (\mathbf{y}_{j,t} - \mathbf{w}_j^\intercal \mathbf{x}_t)^2 + \lambda \| \mathbf{w}_j \|_1, \tag{3}$$

where $\mathbf{w}_j$ is the coefficient vector of the regression for the variable $\mathbf{y}_j$. Assuming $P = K * L$, both $\mathbf{x}_t$ and $\mathbf{w}_j$ are column vectors with the dimension $P \times 1$. However, Equation 3 tends to be addressed as an optimization problem, with the assumption that the coefficient vector is fixed but unknown.

B. Bayesian Modeling

In order to track the temporal dependencies among time series instantly, the problem described in Equation 3 is reformulated from a Bayesian perspective. Bayesian method provides a natural and principled way of combining prior information with data, within a solid decision theoretical framework. The past information about parameters can be incorporated and formed as prior knowledge for future analysis. When new observations become available at current time t, the previous posterior distribution of parameters at time $t-1$ can be used as a prior for current parameter inference. The parameter estimate for linear regression with Lasso penalty can be interpreted as a Bayesian posterior mode estimate when the priors on the regression parameters are independent Laplace distributions [15]. The regression for $\mathbf{y}_{j,t}$ is implemented by a linear combination of $\mathbf{x}_t$ with coefficient vector $\mathbf{w}_j$. From Bayesian perspective, given the coefficient vector (i.e., $\mathbf{w}_j$) and the variance of random observation noise (i.e., σ_j^2), it is assumed that $\mathbf{y}_{j,t}$ follows a Gaussian distribution as below,

$$\mathbf{y}_{j,t} | \mathbf{w}_j, \sigma_j^2 \sim \mathcal{N}(\mathbf{w}_j^\intercal \mathbf{x}_t, \sigma_j^2). \tag{4}$$



(a) Bayesian Lasso model.

(b) Time-varying Bayesian Lasso model.

Fig. 2: Graphical model representations for Granger Causality. Random variable is denoted as a circle. The circle with gray color filled means the corresponding random variable is observed. Red dot represents a hyper parameter.

In this setting, a graphical representation for Bayesian Lasso model is illustrated in Fig. 2a, where the predicted value $\mathbf{y}_{j,t}$ depends on random variable $\mathbf{x}_t$, $\mathbf{w}_j$ and σ_j^2. To obtain a Bayesian model equivalent to the Lasso regression in Equation 3 and simplify the computation, the conjugate prior distributions for all the coefficients in $\mathbf{w}_j$ are assumed as the independent Laplace distributions. Therefore,

$$\pi(\mathbf{w}_j|\sigma_j^2) = \prod_{p=1}^{P} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\mathbf{w}_{j,p}|/\sqrt{\sigma^2}}, \tag{5}$$

where $\pi(\cdot)$ denotes the probability density function. The distribution in Equation 5 can be equivalently expressed as a scale mixture of normals with an exponential mixing density. The augmented latent variables $\gamma_1^2, ..., \gamma_P^2$, following independent exponential distributions, are introduced to build the mixture of normals. The full Bayesian Lasso model is developed in the following hierarchical representation.

$$\begin{aligned} &\mathbf{w}_j|\sigma_j^2, \gamma_1^2, ..., \gamma_P^2 \sim \mathcal{N}(\mu_{\mathbf{w}}, \sigma_j^2 \mathbf{R}_{\mathbf{w_j}}), \\ &\sigma_j^2 \sim \mathcal{IG}(\alpha, \beta), \\ &\gamma_p^2 \sim Exp(\lambda^2/2), \quad 1 \le p \le P, \end{aligned} \tag{6}$$

where $\mathbf{R}_{\mathbf{w_j}} = \mathtt{diag}(\gamma_1^2, ..., \gamma_P^2)$. The prior of σ_j^2 follows Inverse Gamma (abbr., $\mathcal{IG}$) distribution with hyper parameters α and β. The prior of γ_p^2 is given by the exponential distribution (denoted as Exp) with the hyper parameter $\lambda^2/2$, where λ is the Lasso regularization parameter defined in Equation 3. Given σ_j^2 and $\gamma_1^2, ..., \gamma_P^2$, the prior of the coefficient vector $\mathbf{w}_j$ follows a Gaussian distribution with $\mu_{\mathbf{w}}$ and $\sigma_j^2 \mathbf{R}_{\mathbf{w_j}}$ as the mean and the variance, respectively. Typically, $\mu_{\mathbf{w}}$ is set to be $\mathbf{0}$.

The full hierarchical representation in Equation 6 can be reduced to the joint distribution of independent Laplace priors in Equation 5 after integrating out all the augmented latent variables $\gamma_1^2, ..., \gamma_P^2$. With the help of the Bayesian Lasso model, the temporal dependency in terms of Granger Causality can be determined by inferring the posterior distribution of $\mathbf{w}_j$ instantly.

C. Dynamic Causal Relationship Modeling

In real practice, the underlying causal relationship among time series tends to evolve over time. As illustrated in Fig. 3, From the time $t-1$ to t, the dynamic changes of the causal relationship among time series consist of three types: new dependency occurring, dependency fading away, and the strength of dependency varying.

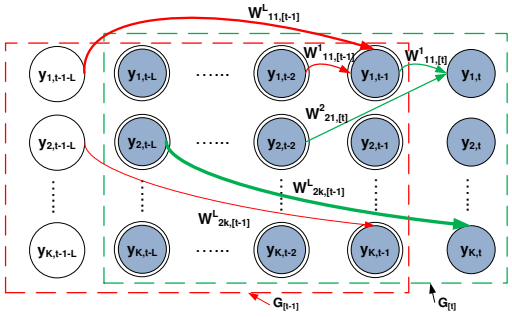


Fig. 3: L is the maximum time lag for VAR model. Temporal dependency among time series changes from $G[t-1]$ at time $t-1$ to $G[t]$ at time t. The nonzero coefficients are indicated by the directed edges. Red lines is used to denote the temporal dependencies in $G[t-1]$, while the green lines represent the temporal dependencies in $G[t]$. The thicker lines mean stronger dependencies existing.

As shown in Equation 3, the value prediction for $\mathbf{y}_j$ at time t is conducted by a linear combination of its own past values and the past values of other variables, using coefficient vector $\mathbf{w}_j$ with L_1 regularization penalty. Each element in the coefficient vector $\mathbf{w}_j$ indicates the contribution of the past value of the corresponding variable for predicting $\mathbf{y}_{j,t}$. The aforementioned model is based on the assumption that $\mathbf{w}_j$ is unknown but fixed, which does not work well with the scenario where the temporal dependency dynamically changes over time. To account for the dynamics, our goal is to come up with a model having the capability of capturing the drift of $\mathbf{w}_j$ over time so as to track the time-varying temporal dependency among the time series instantly. Let $\mathbf{w}_{j,t}$ denote the coefficient vector for predicting $\mathbf{y}_{j,t}$ at time t. Taking the drift of $\mathbf{w}_j$ into account, $\mathbf{w}_{j,t}$ is formulated as follows:

$$\mathbf{w}_{j,t} = \mathbf{c}_{\mathbf{w}_j} + \delta_{\mathbf{w}_{j,t}}, \tag{7}$$

where $\mathbf{w}_{j,t}$ is decomposed into two components including both the stationary component $\mathbf{c}_{\mathbf{w}_j}$ and the drift component $\delta_{\mathbf{w}_{j,t}}$. Both components are P-dimensional vectors. Similar to modeling $\mathbf{w}_j$ in Fig.2a, a conjugate prior distribution below is assumed to generate the stationary component $\mathbf{c}_{\mathbf{w}_j}$.

$$\mathbf{c}_{\mathbf{w}_j} \sim \mathcal{N}(\mu_c, \sigma_j^2 \mathbf{R}_{\mathbf{c_j}}), \tag{8}$$

where μ_c is the hyper parameter, and $\mathbf{R}_{\mathbf{c_j}} = \texttt{diag}(\gamma_{c,1}^2, ..., \gamma_{c,P}^2)$. The latent variables $\gamma_{c,1}^2, ..., \gamma_{c,P}^2$ follow independent exponential distributions ruled by the hyper parameter $\lambda_1^2/2$, as shown in Fig.2b.

However, it's not straightforward to model the drift component with a single function due to the diverse changing behaviors of the regression coefficients. First, some coefficients change frequently, while some coefficients keep relatively stable. Moreover, the coefficients for different variables can change with diverse scales. To simplify the inference, we assume that each element of $\delta_{\mathbf{w}_{j,t}}$ drifts independently. Due to the uncertainty of drifting, we formulate $\delta_{\mathbf{w}_{j,t}}$ by combining a standard Gaussian random walk $\eta_{j,t}$ and a scale variable θ_j using the following Equation:

$$\delta_{\mathbf{w}_{j,t}} = \theta_j \odot \eta_{j,t}, \tag{9}$$

where $\eta_{j,t} \in \mathcal{R}^P$ is the drift value at time t caused by the standard random walk and $\theta_j \in \mathcal{R}^P$ contains the changing scales for all the elements of $\delta_{\mathbf{w}_{j,t}}$. The operator $\odot$ is used to denote the element-wise product. The standard Gaussian random walk is defined with a Markov process as shown in Equation 10.

$$\eta_{j,t} = \eta_{j,t-1} + \mathbf{v}, \tag{10}$$

where $\mathbf{v}$ is a standard Gaussian random variable defined by $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathcal{I}_P)$, and $\mathcal{I}_P$ is a $P \times P$-dimensional identity matrix. It is equivalent that $\eta_{j,t}$ is sampled from the Gaussian distribution

$$\eta_{j,t} \sim \mathcal{N}(\eta_{j,t-1}, \mathcal{I}_P). \tag{11}$$

Similarly, the scale random variable θ_j is generated with a conjugate prior distribution

$$\theta_j \sim \mathcal{N}(\mu_\theta, \sigma_j^2 \mathbf{R}_{\theta_\mathbf{j}}), \tag{12}$$

where μ_θ is predefined hyper parameter, and $\mathbf{R}_{\theta_\mathbf{j}} = \texttt{diag}(\gamma_{\theta,1}^2, ..., \gamma_{\theta,P}^2)$. The latent variables $\gamma_{\theta,1}^2, ..., \gamma_{\theta,P}^2$, following the independent exponential distributions governed by the hyper parameter $\lambda_2^2/2$, are used to construct $\mathbf{R}_{\theta_\mathbf{j}}$. The random variable σ_j^2 of the time-varying Bayesian Lasso model in Fig. 2b is drawn from the Inverse Gamma distribution, which is the same as the one described in Equation 6.

Combining Equation 7 and Equation 9, we obtain:

$$\mathbf{w}_{j,t} = \mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t}, \tag{13}$$

Accordingly, the value $\mathbf{x}_j^t$ is modeled to be drawn from the following a Gaussian distribution as below,

$$\mathbf{y}_{j,t} | \mathbf{c}_{\mathbf{w}_j}, \theta_j, \eta_{j,t}, \sigma_j^2 \sim \mathcal{N}((\mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t})^\intercal \mathbf{x}_t, \sigma_j^2). \tag{14}$$

The time-varying Bayesian Lasso model is presented with a graphical model representation in Fig. 2b. Compared with the model in Fig. 2a, a standard Gaussian random walk $\eta_{j,t}$ and the corresponding scale θ_j for j^{th} time series are introduced in the new model. The new model explicitly formulates the coefficients in Lasso regression, considering the time-varying temporal dependency in real-world application. From the new model, each element value of $\mathbf{c}_{\mathbf{w}_j}$ indicates the contribution of the past values of each variable in predicting the value $\mathbf{y}_{j,t}$, while the element values of θ_j show the drift scales of their contributions to the prediction of $\mathbf{y}_{j,t}$. A large element value of θ_j signifies a great change occurring to the strength of the corresponding causal relationship over time.

Lemma 1 (Equivalent Optimization)**.** *The time-varying Bayesian Lasso model is equivalent to the optimization problem as follows:*

$$\min_{\{\mathbf{w}_{j,t}\}} \sum_{t=L+1}^{T} (\mathbf{y}_{j,t} - (\mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t})^\intercal \mathbf{x}_t)^2 + \lambda_1 \parallel \mathbf{c}_{\mathbf{w}_j} \parallel_1 + \lambda_2 \parallel \theta_j \parallel_1, \tag{15}$$

where λ_1 and λ_2 are penalty parameters, determining the sparsity of both stationary component and drift component.

Based on the idea of Bayesian Lasso, Lemma 1 is straightforward. Thus, its proof is not provided.

According to Lemma 1, λ_1 is set to determine the sparsity of stationary component and λ_2 is used for controlling the variance of drift component. It is difficult to infer the coefficient vectors $\{\mathbf{w}_{j,t}\}$ instantly directly from Equation 15, since $\eta_{j,t}$ is the latent variables. We develop our solution to infer the time-varying Bayesian Lasso model from a Bayesian perspective and the solution is presented in the following section.

IV. Methodology and Solution

In this section, we present the methodology for online inference of the time-varying Bayesian Lasso model.

The posterior distribution inference involves the latent random variables σ_j^2, $\mathbf{c}_{\mathbf{w_j}}$, θ_j, $\mathbf{R}_{\mathbf{c}_j}$, $\mathbf{R}_{\theta_j}$, and $\eta_{j,t}$. According to the graphical model in Fig. 2b, all the latent random variables are grouped into three categories: parameter random variable, augmented random variable and latent state random variable. σ_j^2, $\mathbf{c}_{\mathbf{w}_j}$, θ_j, are parameter random variables since they are assumed to be fixed and unknown, and their values do not depend on the time. $\mathbf{R}_{\mathbf{c}_j}$, $\mathbf{R}_{\theta_j}$ are regarded as augmented random variables where these variables are introduced for equivalent Lasso derivation but their specific values are not very interesting for the problem. Instead, $\eta_{j,t}$ is referred to as a latent state random variable since it is not observable and its value is time dependent according to Equation 10. On the other hand, $\mathbf{x}_t$ and $\mathbf{y}_{j,t}$ are referred to as observed random variables.

Our goal is to infer both latent parameters and latent state variables. However, since the inference partially depends on the random walk which generates the latent state variables, we use the sequential sampling based inference strategy that is widely used in sequential monte carlo sampling [20] [21], particle filtering [22], and particle learning [18] to learn the distribution of both parameters and the state random variables.

Since state $\eta_{j,t-1}$ changes over time with a standard Gaussian random walk, it follows a Gaussian distribution after accumulating $t-1$ standard Gaussian random walks. Assume $\eta_{j,t-1} \sim \mathcal{N}(\mu_{\eta_j}, \mathbf{\Sigma}_{\eta_j})$, a particle is defined as follows.

Definition 1 (Particle)**.** *A particle for predicting $\mathbf{y}_{j,t}$ is a container which maintains the current status information for*

value prediction. The status information comprises of random variables such as σ_j^2, $\mathbf{c}_{\mathbf{w}_j}$, θ_j, $\mathbf{R}_{\mathbf{c}_j}$, $\mathbf{R}_{\theta_j}$, *and* $\eta_{j,t}$, *and the hyper parameters of their corresponding distributions such as* α *and* β, $\mu_{\mathbf{c}}$, μ_θ, λ_1, λ_2, μ_{η_k} *and* $\mathbf{\Sigma}_{\eta_k}$.

A. Re-sample Particles with Weights

At time $t-1$, a fixed-size set of particles are maintained for the value prediction of the j^{th} time series, where the particle set is denoted as $\mathcal{P}_{j,t-1}$ and the number of particles in $\mathcal{P}_{j,t-1}$ is B. Let $\mathcal{P}_{j,t-1}^{(i)}$ be the i^{th} particle in the particle set $\mathcal{P}_{j,t-1}$ at time $t-1$, where $1 \leq i \leq B$. Each particle $\mathcal{P}_{j,t-1}^{(i)}$ has a weight, denoted as $\rho^{(i)}$, indicating its fitness for the new observed data at time t. Note that $\sum_{i=1}^{B} \rho^{(i)} = 1$. The fitness of each particle $\mathcal{P}_{j,t-1}^{(i)}$ is defined as the likelihood of the observed data $\mathbf{x}_t$ and $\mathbf{y}_{j,t}$. Therefore,

$$\rho^{(i)} \propto P(\mathbf{x}_t, \mathbf{y}_{j,t} | \mathcal{P}_{j,t-1}^{(i)}). \tag{16}$$

Further, according to Equation 14, the distribution of $\mathbf{y}_{j,t}$ is determined by the random variables $\mathbf{c}_{\mathbf{w}_j}$, θ_j, σ_j^2 and $\eta_{j,t}$.

Therefore, we can compute $\rho^{(i)}$ in proportional to the density value at $\mathbf{y}_{j,t}$. Thus,

$$\begin{aligned}\rho^{(i)} \propto \iint_{\eta_{j,t}, \eta_{j,t-1}} &\{\mathcal{N}(\mathbf{y}_{j,t} | (\mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t})^\intercal \mathbf{x}_t, \sigma_j^2) \\ &\mathcal{N}(\eta_{j,t} | \eta_{j,t-1}, \mathcal{I}_P) \mathcal{N}(\eta_{j,t-1} | \mu_{\eta_j}, \mathbf{\Sigma}_{\eta_j})\} \\ &d\eta_{j,t}\, d\eta_{j,t-1},\end{aligned}$$

where state variables $\eta_{j,t}$ and $\eta_{j,t-1}$ are integrated out due to their change over time, and $\mathbf{c}_{\mathbf{w}_j}$, θ_j, σ_j^2 are from $\mathcal{P}_{j,t-1}^{(i)}$. Then we obtain

$$\rho^{(i)} \propto \mathcal{N}(\mathbf{m}_j, \mathbf{Q}_j), \tag{17}$$

where

$$\begin{aligned}\mathbf{m}_j &= (\mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t})^\intercal \mathbf{x}_t \\ \mathbf{Q}_j &= \sigma_j^2 + (\mathbf{x}_t \odot \theta_j)^\intercal (\mathcal{I}_P + \mathbf{\Sigma}_{\eta_j})(\mathbf{x}_t \odot \theta_j).\end{aligned} \tag{18}$$

Before updating any parameters, a re-sampling process is conducted. We replace the particle set $\mathcal{P}_{j,t-1}$ with a new set $\mathcal{P}_{j,t}$, where $\mathcal{P}_{j,t}$ is generated from $\mathcal{P}_{j,t-1}$ using sampling with replacement based on the weights of particles. Then sequential parameter updating is based on $\mathcal{P}_{j,t}$.

B. Latent State Inference

At time $t-1$, the sufficient statistics for state $\eta_{j,t-1}$ are the mean (i.e., μ_{η_j}) and the covariance (i.e., $\mathbf{\Sigma}_{\eta_j}$). Provided with the new observation data $\mathbf{x}_t$ and $\mathbf{y}_{j,t}$ at time t, the sufficient statistics for state $\eta_{j,t}$ need to be re-computed. We apply the Kalman filtering [23] method to recursively update the sufficient statistics for $\eta_{j,t}$ based on the new observation and the sufficient statistics at time $t-1$. Let ${\mu'}_{\eta_j}$ and ${\mathbf{\Sigma}'}_{\eta_j}$ be the new sufficient statistics of state $\eta_{j,t}$ at time t. Then,

$$\begin{aligned}{\mu'}_{\eta_j} &= \mu_{\eta_j} + \underbrace{\mathbf{G}_j(\mathbf{y}_{j,t} - (\mathbf{c}_{\mathbf{w}_j} + \theta_j \odot \eta_{j,t})^\intercal \mathbf{x}_t))}_{\text{Correction by Kalman Gain}}, \\ {\mathbf{\Sigma}'}_{\eta_j} &= \mathbf{\Sigma}_{\eta_j} + \mathcal{I}_P - \underbrace{\mathbf{G}_j \mathbf{Q}_j \mathbf{G}_j^\intercal}_{\text{Correction by Kalman Gain}},\end{aligned} \tag{19}$$

where $\mathbf{Q}_j$ is defined in Equation 18 and $\mathbf{G}_j$ is Kalman Gain [23] defined as

$$\mathbf{G}_j = (\mathcal{I}_P + \mathbf{\Sigma}_{\eta_j})(\mathbf{x}_t \odot \theta_j)\mathbf{Q}_j^{-1}.$$

As shown in Equation 19, both ${\mu'}_{\eta_j}$ and ${\mathbf{\Sigma}'}_{\eta_j}$ are estimated with a correction using Kalman Gain $\mathbf{G}_j$(i.e., the last term in both two formulas). With the help of the sufficient statistics for the state random variable, $\eta_{j,t}$ can be drawn from the Gaussian distribution

$$\eta_{j,t} \sim \mathcal{N}({\mu'}_{\eta_j}, {\mathbf{\Sigma}'}_{\eta_j}). \tag{20}$$

C. Augmented Variable Inference

The augmented variables $\mathbf{R}_{\mathbf{c}_j}$ and $\mathbf{R}_{\theta_j}$ are diagonal matrices composed of independent random variables $\gamma_{c,1}^2, ..., \gamma_{c,P}^2$ and $\gamma_{\theta,1}^2, ..., \gamma_{\theta,P}^2$, respectively. The independent random variables are drawn from exponential distribution as follows,

$$\begin{aligned}\gamma_{c,p} &\sim Exp(\lambda_1^2/2), \\ \gamma_{\theta,p} &\sim Exp(\lambda_2^2/2),\end{aligned} \tag{21}$$

where $1 \leq p \leq P$. At each time stamp, those augmented random variables are sampled independently. Assume $\mathbf{R}_j = \begin{bmatrix} \mathbf{R}_{\mathbf{c_j}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\theta_\mathbf{j}} \end{bmatrix}$, where $\mathbf{R}_j$ is a $2P \times 2P$ -dimensional matrix.

D. Parameter Inference

At time $t-1$, the sufficient statistics for the parameter random variables (σ_j^2, $\mathbf{c}_{\mathbf{w_j}}$, θ_j) are (α, β, μ_c, μ_θ). Let $\mathbf{z}_t = (\mathbf{x}_t{}^\intercal, (\mathbf{x}_t \odot \boldsymbol{\eta}_{\mathbf{j,t}})^\intercal)^\intercal$, $\mu_j = (\mu_{\mathbf{c}}{}^\intercal, \mu_\theta{}^\intercal)^\intercal$, and $\nu_j = (\mathbf{c}_{\mathbf{w_j}}{}^\intercal, \theta_{\mathbf{j}}{}^\intercal)^\intercal$ where $\mathbf{z}_t$, μ_j, and ν_j are $2P$-dimensional vector.

Therefore, the inference of $\mathbf{c}_{\mathbf{w_j}}$ and θ_j is equivalent to infer ν_j with its distribution $\nu_j \sim \mathcal{N}(\mu_j, \sigma_j^2 \mathbf{R}_j^{\frac{1}{2}} \mathbf{\Sigma}_j \mathbf{R}_j^{\frac{1}{2}})$, where $\mathbf{\Sigma}_j$ is initialized with an identity matrix time 0. Assume ${\mathbf{\Sigma}'}_j$, $\mu'_\mathbf{j}$, α', and β' be the sufficient statistics at time t which are updated based on the sufficient statistics at time $t-1$ and the new observation data. The sufficient statistics for parameters are updated as follows:

$$\begin{aligned}&{\mathbf{\Sigma}'}_j = (\mathbf{\Sigma}_j^{-1} + \mathbf{R}_j^{\frac{1}{2}} \mathbf{z}_t \mathbf{z}_t^\intercal \mathbf{R}_j^{\frac{1}{2}})^{-1}, \\ &\mu'_\mathbf{j} = \mathbf{R}_j^{\frac{1}{2}} {\mathbf{\Sigma}'}_j \mathbf{R}_j^{\frac{1}{2}} \mathbf{z}_t \mathbf{y}_{j,t} + \mathbf{R}_j^{\frac{1}{2}} {\mathbf{\Sigma}'}_j \mathbf{\Sigma}_j \mathbf{R}_j^{\frac{1}{2}} \mu_j), \\ &\alpha' = \alpha + \frac{1}{2}, \\ &\beta' = \beta + \frac{1}{2}(\mu_j^\intercal \mathbf{R}_j^{-\frac{1}{2}} \mathbf{\Sigma}_j^{-1} \mathbf{R}_j^{-\frac{1}{2}} \mu_j + \mathbf{y}_{j,t}^2 - \mu_j'^\intercal \mathbf{R}_j^{-\frac{1}{2}} {\mathbf{\Sigma}'}_j^{-1} \mathbf{R}_j^{-\frac{1}{2}} \mu'_j).\end{aligned} \tag{22}$$

At time t, the sampling process for σ_j^2 and ν_j is summarized as follows:

$$\begin{aligned}\sigma_j^2 &\sim \mathcal{IG}(\alpha', \beta'), \\ \nu_j &\sim \mathcal{N}(\mu'_j, \sigma_j^2 \mathbf{R}_j^{\frac{1}{2}} {\mathbf{\Sigma}'}_j \mathbf{R}_j^{\frac{1}{2}}).\end{aligned} \tag{23}$$

E. Algorithm

Putting all the aforementioned things together, an algorithm based on the proposed time-varying Bayesian Lasso model is provided below.

Online inference for time-varying Bayesian Lasso model starts with MAIN procedure, as presented in Algorithm 1. The parameters B, L, α, β, λ_1 and λ_2 are given as the input of MAIN procedure. The initialization is executed from line 2 to line 6. As new observation $\mathbf{y}_{\cdot,t}$ arrives at time t, $\mathbf{x}_t$ is built using the time lag, then $\mathbf{w}_{j,t}$ is inferred by

calling UPDATE procedure. Especially in the UPDATE procedure, we use the $resample\text{-}propagate$ strategy in particle learning [18] rather than the $propagate\text{-}resample$ strategy in particle filtering [22]. With the $resample\text{-}propagate$ strategy, the particles are re-sampled by taking $\rho^{(i)}$ as the i^{th} particle's weight, where the $\rho^{(i)}$ indicates the occurring probability of the observation at time t given the particle at time $t-1$. The $resample\text{-}propagate$ strategy is considered as an optimal and fully adapted strategy, avoiding an importance sampling step.

Algorithm 1 The algorithm for time-varying Bayesian Lasso model

procedure MAIN(B, L, α, β, λ_1, λ_2) ▷ main entry
  Initialize $\mu_c = \mathbf{0}$, $\mu_\theta = \mathbf{0}$.
  for $j \leftarrow 1, K$ **do**
    Initialize regression for $\mathbf{y}_j$ with B particles.
    Initialize Σ_j with identity matrix.
  end for
  for $t \leftarrow 1, T$ **do**
    Get $\mathbf{x}_t$ using time lag L.
    for $j \leftarrow 1, K$ **do**
      UPDATE($\mathbf{x}_t$, $y_{j,t}$).
      Output $\mathbf{w}_{j,t}$ according to Eq. 13.
    end for
  end for
end procedure

procedure UPDATE($\mathbf{x}_t$, $y_{j,t}$) ▷ update the inference.
  for $i \leftarrow 1, B$ **do** ▷ Compute weights for each particle.
    Compute weight $\rho^{(i)}$ of particle $\mathcal{P}^{(i)}_{j,t-1}$ by Eq. 17.
  end for
  Re-sample $\mathcal{P}_{j,t}$ from $\mathcal{P}_{j,t-1}$ according to $\rho^{(i)}$s.
  for $i \leftarrow 1, B$ **do** ▷ Update statistics for each particle.
    Update the sufficient statistics for $\eta_{j,t}$ by Eq. 19.
    Sample $\eta_{j,t}$ according to Eq. 20.
    Construct augmented variables $\mathbf{R}_j$ with Eq. 21.
    Update the statistics for σ_j^2, $\mathbf{c}_{\mathbf{w}_j}$, θ_j by Eq. 22.
    Sample σ_j^2, $\mathbf{c}_{\mathbf{w}_j}$, θ_j according to Eq. 23.
  end for
end procedure

V. Empirical Study

With the purpose of demonstrating the performance of the proposed algorithm, we conduct the experiments over both synthetic and real data sets, and illustrate a real case study from the system management. Before diving into the discussion of the evaluation in detail, we first outline the general implementation of the baseline algorithms for comparison, then verify the proposed algorithm using every data set one by one. The evaluation on each data set is started with a brief description of the data and the corresponding evaluation methods, and followed by the presentation of the comparative experimental results between the proposed algorithm and the baseline algorithms.

A. Baseline Algorithms

In the empirical study, we demonstrate the performance of our method by comparing with the following baseline algorithms including:

- BLR(q_0): It infers the temporal dependencies among time series using Bayesian Linear Regression with prior distribution $\mathcal{N}(\mathbf{0}, q_0^{-1}\mathbf{I}_d)$. It has been shown that the setting of the penalty parameter λ in ridge regression can be achieved by tuning q_0 accordingly [24].
- BLasso(λ): It applies Bayesian Lasso to learn the temporal dependencies, where λ is the L_1 penalty parameter. It presents an online inference for Lasso regression from Bayesian perspective [15].
- TVLR(q): It makes use of the Time-Varying Linear Regression from Bayesian perspective, which is capable of capturing the dynamics of dependency without regularization. The parameter q specifies the prior distribution $\mathcal{N}(\mathbf{0}, q^{-1}\mathbf{I}_{2d})$ for both constant and varying components of the coefficients [29].

One the other hand, we denote our proposed method as TVLasso(λ_1,λ_2), where the Time-Varying Bayesian Lasso regression algorithm is used to infer the time-varying temporal dependency among time series. The penalty parameters λ_1 and λ_2 are presented in Equation 15, determining the sparsity of both stationary component and drift component, respectively. Note that the algorithms in [12] and [16] are not included as baseline algorithms in our experiment, since both are offline algorithms, while the work of this paper mainly focuses on online inference of time-varying temporal dependency. During our experiments, we extract small subset of data with early time stamps and employ grid search to find the optimal parameters for all the algorithm. The parameter settings are verified by cross validation in terms of the prediction errors over the extracted data subset.

B. Evaluation Measures

AUC Score: In order to further verify the efficacy of the proposed method for temporal dependency identification, AUC, the Area Under the ROC [25], is applied for performance evaluation due to its independence of priors, costs, and operating points [26]. The value of AUC is the probability that the algorithm will assign a higher value to a randomly chosen existing edge than a randomly chosen non-existing edge in the temporal dependency structure. As we have mentioned in Section III-A, nonzero value of $\mathbf{W}^l_{ij}$ indicates $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. It is reasonable to suppose that a higher absolute value of $\mathbf{W}^l_{ij}$ implies a larger likelihood of existing a temporal dependency $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. At each time t, an AUC score of the algorithm is obtained by comparing its inferred temporal dependency structure with the ground truth at t.

Prediction Error: Let $\mathbf{W}_t$ be the true coefficient matrix and $\widehat{\mathbf{W}}_t$ be the estimated coefficient matrix. We define the prediction error at time t as $\Delta = ||\mathbf{W}_t - \widehat{\mathbf{W}}_t||_F$, where $||\bullet||_F$ is the Frobenius Norm [27]. A smaller prediction error indicates a better inference of dynamic temporal structure.

In order to give a clear illustration, we segment the time line into time buckets with the same predefined size and illustrate the performance with an average value of the corresponding measure for every time bucket.

C. Synthetic Data

The main advantage of using synthetic data sets is that the detailed dependency structures are known and hence we can systematically evaluate the performance of our proposed method with different factors such as noise and sparsity levels and quantitatively compare with other alternative solutions using various performance measures.

Synthetic Data Generation: The synthetic data generation is governed by the parameters shown in Table II. The time series data are generated with the VAR model, where the coefficient value $\mathbf{W}_{ij}^{l}$ indicates the strength of dependency $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. To simulate the time-varying temporal dependencies among time series, five types of dynamics are randomly injected into the VAR model, depicting the dynamic changes of the coefficients, including:

TABLE II: Parameters for Synthetic Data Generation

Name	Description
K	The number of time series.
T	The length of time series.
L	The maximum time lag for VAR model.
I	The maximum number of intervals used to segmented the time line.
s	The sparsity of the temporal dependency, denoted as the ratio of coefficients with zero value to K.
μ	The mean of the noise introduced during regression.
σ^2	The variance of the noise introduced during regression.

(1) **Zero Value** The coefficient holds a zero value, indicating no temporal dependency existing. The number of coefficient with zero value is determined by the sparsity s.
(2) **Constant Value** The coefficient holds a constant nonzero value, which is randomly generated from the standard Gaussian distribution.
(3) **Piecewise Constant** The time line is randomly segmented into multiple intervals. The number of intervals is uniformly sampled in $(0, I]$. During each interval, the coefficient value is constant. The constant values are generated from the standard Gaussian distribution.
(4) **Periodic Change** The coefficient value varies periodically as time evolves, where the periodic change of the coefficient is simulated by sin curve whose period is uniformly sampled from the range $(0, T)$.
(5) **Random Walk** The coefficient value at time t is determined by a standard Gaussian random walk from the value at time $t-1$.

The sparsity of the temporal dependencies is regulated by s, indicating that a coefficient has the probability s to be generated by type (1). Accordingly, the other four types (2)-(5) uniformly share the probability $1-s$ for simulating the coefficient.

Dynamic Temporal Dependency Tracking: In order to show the capability in capturing the dynamic temporal dependency with a visualized straightforward example, we start with a simulation where $K = 20$, $T = 3000$, $L = 1$, $I = 10$, $s = 0.9$, $\mu = 0$ and $\sigma^2 = 1$. Both the baseline algorithms and our proposed algorithm infer the temporal dependency in an online mode. The performance of all the algorithms depends on the parameter setting. Therefore, we first conduct the performance comparison for each algorithm with diverse parameter settings. Then the one with best performance is selected for comparison study. Eight coefficients are selected and displayed in Fig. 5. It shows our proposed algorithm `TVLasso` can effectively capture the time-varying temporal dependency with different types of dynamics. The `BLasso` algorithm shows more robustness than `BL` for zero-value coefficient inference, and is more suitable for inference with high sparsity. The algorithm `TVLR` captures the dynamic change of the coefficients better than both `BLasso` and `BL`, but it is less stable when comparing with `TVLasso`.

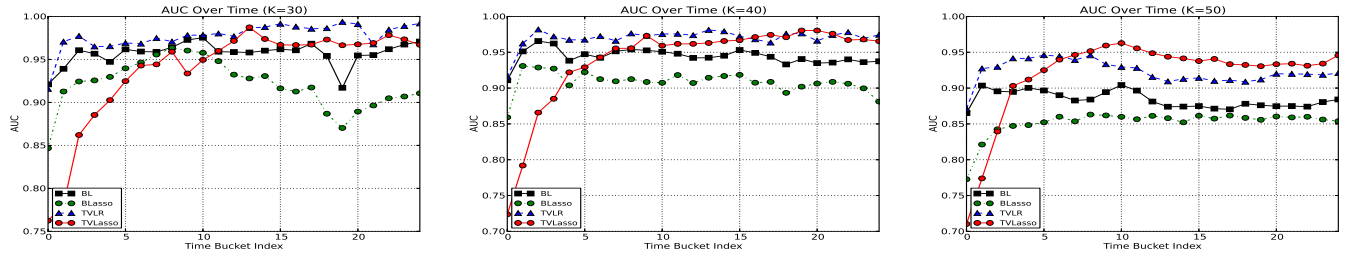
Performance Evaluation: We continue to conduct the evaluation in terms of `AUC` and prediction error over a simulation data set with higher dimension, where $K = (30, 40, 50)$, $T = 5000$, $L = 1$, $I = 10$, $s = 0.9$, $\mu = 0$ and $\sigma^2 = 1$. The evaluations with different Ks in terms of `AUC` are depicted in Fig. 4a, Fig. 4b and Fig. 4c, respectively. The performance of `TVLasso` is comparable with `TVLR` in low dimensions, while `TVLasso` quickly catches up with other baseline algorithms at the beginning and keeps outperforming them in high dimensions. Comparing with other two baseline algorithms , `TVLR` shows a relatively good performance since it models the dynamic change explicitly.

In terms of prediction error, `TVLasso` incurs lowest prediction error consistently, shown in Fig. 4d, Fig. 4e and Fig. 4f. When in high dimension, `TVLR` gets the highest prediction error even though it obtains relatively high `AUC` score, where the reason is that the `AUC` is computed based on the absolute value of the coefficient. The conclusion is that our proposed algorithm `TVLasso` is consistent in the coefficient prediction while `TVLR` may suffer coefficient prediction with opposite sign of the truth, especially in high dimensions.

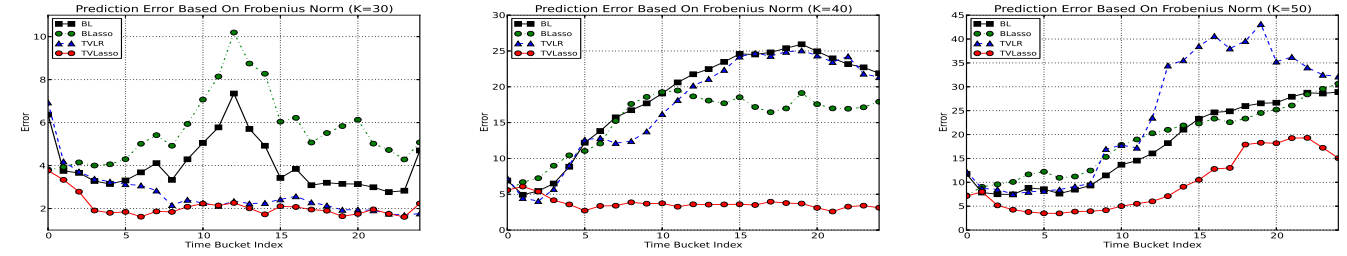
Time Cost: The time cost increases linearly as the number of particles shown in Fig. 6a.

D. A Case Study In System Management

We conduct the case study in a real system FIU-Miner [28], which is a fast, integrated and user-friendly system for data mining in distributed system. FIU-Miner composes every job as a workflow where a set of computing tasks are organized in a dependency graph. A job of FIU-Miner can be scheduled in different ways, such as one-time execution at a particular time, periodic execution every one predefined time interval. To help FIU-Miner make decisions on job scheduling, the system monitoring agents are deployed to all the computing nodes in the distributed environment, and periodically collect the information about both resource usage and running processes. The resource usage information includes CPU utilization, memory usage, disk I/O, networking I/O, etc. The running process information describes the status, the number of running instances aggregated by the program, running time, and so forth. An alert is raised if the predefined monitoring situation persists violated beyond a particular duration. We deploy



(a) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(2k,2k). (b) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(1k,2k). (c) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(1k,2k).



(d) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(2k,2k). (e) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(1k,2k). (f) BLR(1.0),BLasso(1k),TVLR(1.0),TVLasso(1k,2k).

Fig. 4: The temporal dependency identification performance is evaluated in terms of AUC and prediction error. The bucket size is 200.

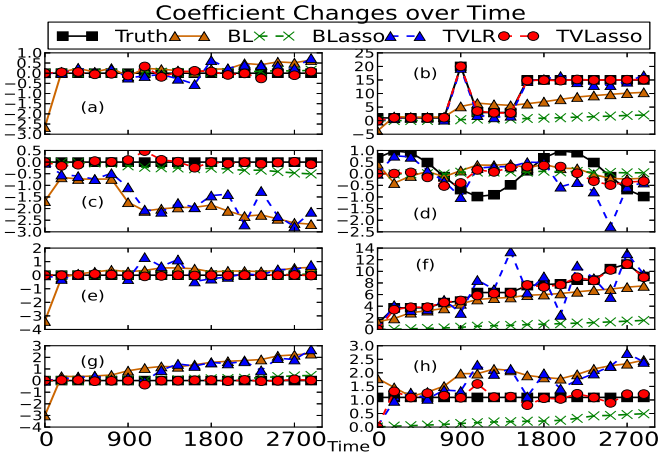
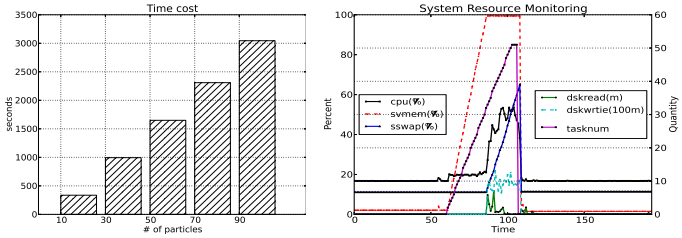


Fig. 5: The temporal dependencies among 20 time series are leant and eight coefficients among all are selected for demonstration. Coefficients with zero values are displayed in (a),(c),(e) and (g). The coefficients with piecewise constant, periodic change, random walk and constant value are shown in (b),(d),(f) and (h), respectively.



(a) (b)

Fig. 6: (a) The time cost of TVLasso with different number of particles. (b) The system resource monitoring time series collected every 5 seconds.

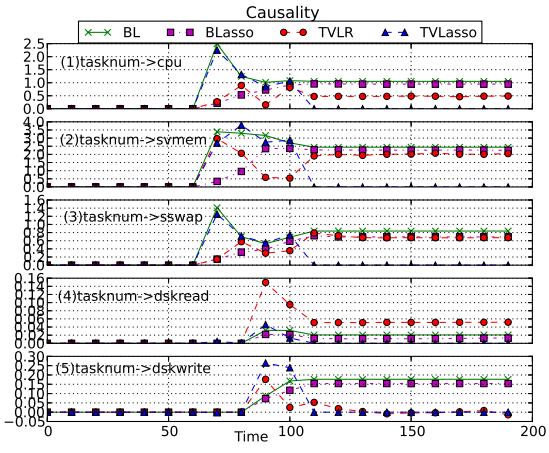


Fig. 7: Temporal dependencies among system resource monitoring time series are discovered.

our algorithm with FIU-Miner to instantly infer the causal dependency among the collected monitoring information.

To illustrate the efficacy of our method, we inspect an alert raised at the time stamp 2016-07-06 01:30:39,852, when a persistent high system load occurred. The process information is aggregated by the executable program. The number of instances for a matrix computation program is identified with strong dependencies between other system resource monitoring time series. The system monitoring time series as well as the number of instances for the identified program are displayed in Fig 6b. Here `cpu(%)`, `svmem(%)`, `sswap(%)`, `dskread(m)`, `dskwrite(100m)` and `tasknum` represent the CPU utilization, virtual memory usage, swap memory usage, the number of bytes reading from the disk, the number of bytes writing to the disk, and the number of instances

for a matrix computation program, respectively. `cpu(%)`, `svmem(%)` and `sswap(%)` share the Percent axis, and `dskread(m)`, `dskwrite(100m)` and `tasknum` share the Quantity axis. Each computing node in the distributed environment has $31G$ memory in total. The causal dependencies discovered by multiple algorithms are shown in Fig 7. The `tasknum` increases linearly to 52 at the beginning, and then decreases to 0 abruptly.

After meticulously inspecting the source code of the matrix computation program, each instance allocates $1G$ memory for holding the matrix data, but does not explicitly recycle the used memory after computation. FIU-Miner schedules the program periodically as a sub-process but does not reap the completed sub-processes until all the sub-processes have been scheduled. It ends in a number of zombie processes during scheduling and causes a resource leak.

As illustrated by the algorithm `TVLasso` in Fig. 7, at the early stage, `tasknum` strongly infers `cpu`, `svmem`, and `sswap`. After the consumed memory exceeds the total available memory of the computing node, `tasknum` has strong causal relations with `dskread` and `dskwrite`. Finally, the temporal dependencies disappear after all the sub-processes are reaped by the schedule process of FIU-Miner. However, the baseline algorithms can not effectively react with the dynamic changes of temporal dependencies.

VI. Conclusion and Future Work

In this paper, we take the dynamic change of the underlying temporal dependencies among time series into account and explicitly model the dynamic change as a random walk. We propose a method based on the particle learning to efficiently infer both parameters and latent variables simultaneously. The performance of our proposed algorithm is verified by both synthetic and real data set.

To discover the time-varying temporal dependency among time series, the choice of penalty parameters is very essential. One possible future work is to come up with online method to automatically identify the proper parameters. The time-varying temporal dependency discovery among time series unveils the dynamic change of the system structure over time. Another possible direction is to apply the discovered time-varying temporal dependency for anomaly detection.

VII. Acknowledgements

The work was supported in part by the National Science Foundation under grants CNS-1126619, IIS-1213026, and CNS-1461926, the U.S. Department of Homeland Securitys VACCINE Center under Award Number 2009-ST-061-CI0001, a gift award from Huawei Technologies Co. Ltd, and an FIU Dissertation Year Fellowship.

References

[1] Y. Jiang, C. Zeng, J. Xu, and T. Li, "Real time contextual collective anomaly detection over multiple data streams," *Proceedings of the ODD*, pp. 23–30, 2014.

[2] C. Zeng, L. Tang, T. Li, L. Shwartz, and G. Y. Grabarnik, "Mining temporal lag from fluctuating events for correlation and root cause analysis," in *CNSM 2014*, pp. 19–27.

[3] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[4] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, 2003.

[5] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[6] ——, "Testing for causality: a personal viewpoint," *Journal of Economic Dynamics and control*, vol. 2, pp. 329–352, 1980.

[7] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modeling with graphical granger methods," in *SIGKDD 2007*, pp. 66–75.

[8] C. Zou and J. Feng, "Granger causality vs. dynamic bayesian network inference: a comparative study," *BMC bioinformatics*, vol. 10, no. 1, 2009.

[9] D. Cheng, M. T. Bahadori, and Y. Liu, "Fblg: A simple and effective approach for temporal dependence discovery from time series data," in *SIGKDD 2014*, pp. 382–391.

[10] M. T. Bahadori and Y. Liu, "An examination of practical granger causality inference," in *SIAM Data Mining*, 2013.

[11] D. Heckerman, "A tutorial on learning with bayesian networks," in *Learning in graphical models*, 1998, pp. 301–354.

[12] L. Song, M. Kolar, and E. P. Xing, "Time-varying dynamic bayesian networks," in *NIPS 2009*, pp. 1732–1740.

[13] M. Eichler, "Graphical modelling of multivariate time series with latent variables," *Preprint, Universiteit Maastricht*, 2006.

[14] M. T. Bahadori and Y. Liu, "On causality inference in time series." in *AAAI Fall Symposium: Discovery Informatics*, 2012.

[15] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, **103**(482), pp. 681–686, 2008.

[16] Y. Liu, J. R. Kalagnanam, and O. Johnsen, "Learning dynamic temporal graphs for oil-production equipment monitoring system," in *SIGKDD 2009*, pp. 1225–1234.

[17] J. H. Halton, "Sequential monte carlo," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 58, Cambridge Univ Press, 1962, pp. 57–78.

[18] C. Carvalho, M. S. Johannes, H. F. Lopes, and N. Polson, "Particle learning and smoothing," *Statistical Science*, **25**(1), pp. 88–106, 2010.

[19] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[20] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[21] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

[22] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[23] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[24] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006.

[25] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[26] T. Landgrebe and R. Duin, "A simplified extension of the area under the roc to the multiclass domain," in *Seventeenth annual symposium of the pattern recognition association of South Africa*, 2006, pp. 241–245.

[27] B. Carpentieri, I. S. Duff, and L. Giraud, "Sparse pattern selection strategies for robust frobenius-norm minimization preconditioners in electromagnetism," *Numerical linear algebra with applications*, vol. 7, no. 7-8, pp. 667–685, 2000.

[28] C. Zeng, Y. Jiang, L. Zheng, J. Li, L. Li, H. Li, C. Shen, W. Zhou, T. Li, B. Duan *et al.*, "Fiu-miner: a fast, integrated, and user-friendly system for data mining in distributed environment," in *SIGKDD 2013*. pp. 1506–1509.

[29] C. Zeng, Q. Wang, S. Mokhtari, and T. Li, "Online Context-Aware Recommendation with Time Varying Multi-Armed Bandit," in *SIGKDD 2016*. pp. 2025-2034.

[30] J. Xu, P. Tan, and L. Luo, "ORION: Online Regularized multI-task regressiON and its application to ensemble forecasting" in *ICDM 2014*. pp. 1061-1066.

[31] C. Zeng, and T. Li, "Event Pattern Mining," in Event Mining: Algorithms and Applications, T. Li, Ed. Chapman and Hall/CRC, 2015, pp. 71-121.