SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2018, Vol. 61 052101:1–052101:16
doi: 10.1007/s11432-016-9070-4

Learning dynamic dependency network structure with time lag

Sizhen DU[1], Guojie SONG[1*], Haikun HONG[2] & Dong LIU[3]

[1]*Key Laboratory of Machine Perception, Ministry of Education, Peking University, Beijing* 100871*, China;*
[2]*Institute of Electronics, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*School of Computer and Information Engineering, Henan Normal University, Xinxiang* 453007*, China*

Received 18 July 2016/Revised 31 December 2016/Accepted 16 March 2017/Published online 30 August 2017

Abstract Characterizing and understanding the structure and the evolution of networks is an important problem for many different fields. While in the real-world networks, especially the spatial networks, the influence from one node to another tends to vary over both space and time due to the different space distances and propagation speeds between nodes. Thus the time lag plays an essential role in interpreting the temporal causal dependency among nodes and also brings a big challenge in network structure learning. However most of the previous researches aiming to learn the dynamic network structure only treat the time lag as a predefined constant, which may miss important information or include noisy information if the time lag is set too small or too large. In this paper, we propose a dynamic Bayesian model with adaptive lags (DBAL) which simultaneously integrates two usually separate tasks, i.e., learning the dynamic dependency network structure and estimating time lags, within one unified framework. Specifically, we propose a novel weight kernel approach for time series segmenting and sampling via leveraging samples from adjacent segments to avoid the sample scarcity. Besides, an effective Bayesian scheme cooperated with reversible jump Markov chain Monte Carlo (RJMCMC) and expectation propagation (EP) algorithm is proposed for parameter inference. Extensive empirical evaluations are conducted on both synthetic and two real-world datasets, and the results demonstrate that our proposed model is superior to the traditional methods in learning the network structure and the temporal dependency.

Keywords dependency network, time lag, dynamic network

Citation Du S Z, Song G J, Hong H K, et al. Learning dynamic dependency network structure with time lag. Sci China Inf Sci, 2018, 61(5): 052101, doi: 10.1007/s11432-016-9070-4

1 Introduction

In real-life networks, the network structure and temporal dependency are not invariant but tend to vary over both time and space. Recently, dynamic dependency network structure learning has attracted increasing attentions in various domains, such as gene regulatory network discovery [1], social network analysis [2], and climate data analysis [3]. Mining the temporal dependency is one of the fundamental tasks in network structure learning. As the key feature to indicate the exact temporal dependency relationship, time lag plays an essential role in characterizing and understanding dynamic dependency network structure and evolution [4].

Taking the spatial network as an example [5], the time lag problem is much significant and inevitable due to the different space distances and propagation speeds between nodes. For example, (1) in the

* Corresponding author (email: gjsong@pku.edu.cn)

highway traffic network, the traffic flow of one destination station in current time interval is constructed by vehicles from related upstream origin stations several time intervals ago. Thus the time lags are related with the vehicle speeds and the distances from origin stations to destination stations. (2) In the air quality monitoring network, the concentration of pollutant in one monitoring station usually has temporal dependency on the concentrations of pollutant in surrounding monitoring stations. The time lag always varies over time and is influenced by various factors, like spatial distance, weather, traffic accident, etc. Therefore, it is necessary to learn the time lag adaptively in the network structure learning process instead of treating it as a predefined constant.

In recent years, there have been a plenty of methods for dynamic network structure learning. Some researches focus on the dependency relationship of every time point in time series, such as [6–8]. While some researches regard that the dependency network structures vary by time intervals, such as [1, 9–11]. As aforementioned, most of the existing studies do not take the dynamic time lag into account and only simply treat the time lag as a predefined constant. There are some researches which focus on learning the lags from temporal data. For example, ref. [4] proposes to estimate the maximum lag existing in the temporal variables based on order statistics. A maximum likelihood (ML) estimator is developed for determining time delay in [12]. However, these researches only focus on inferring the lag from temporal data while the relationships are not considered at the same time.

Moreover, we identify two major challenges in learning dynamic network structures with simultaneous estimation of dynamic time lags: (i) How fast the dynamic structures vary by time? If we learn a structure for each single time point, the results would be redundant and the learning process would be computationally expensive. However, if the time series are first segmented and then one relationship is learned to each segment, it would suffer the scarcity of samples when the segment is short [8]. (ii) How to learn the dynamic time lags which vary with the network structures simultaneously during the structure estimation process? The uncertainty of time lags increases the difficulty of the whole learning process.

In this paper, we propose a dynamic Bayesian model with adaptive lags (DBAL) to learn the dynamic dependency network structure and the optimal time lags simultaneously in one unified framework. We extend the non-homogeneous dynamic Bayesian model where changepoints are used to segment the time series. Instead of setting the minimum length of interval, we assume the network structures vary gradually and propose a novel kernel reweighted probabilistic approach by leveraging the samples from adjacent segments to avoid sample scarcity within narrow segments. When we estimate the dependency network structures, we regard lags as variables instead of parameters set artificially. In addition, the dependency network structures are reflected by the regression coefficients which we take as the spike and slab priors. In parameter inference part, an effective Bayesian scheme incorporating the reversible jump Markov chain Monte Carlo (RJMCMC) [13] and Expectation Propagation algorithm (EP) [14] is proposed in order to infer the parameters of the model, where the dimension of the parameters is unknown and the posterior probability is intractable to sample.

In summary, the contributions of our work are described as follows:

• In this paper, a novel dynamic Bayesian model is proposed which integrates both the dynamic network structure learning and dynamic time lag estimation simultaneously in one unified framework.

• To avoid the sample scarcity, we propose a novel weight kernel approach for time series segmenting and sampling based on the assumption that the network structure of adjacent segments are similar.

• In order to infer the parameters in our model, where the dimension of parameters is unknown and some posterior probabilities are intractable to sample, we propose an effective Bayesian scheme incorporating the RJMCMC and EP algorithm.

• We evaluate our method on a synthetic dataset and two real world datasets. Experimental results show that we not only can estimate lags, but also obtain higher accuracies of learning network structures.

The rest of the paper is organized as follows: we first review the algorithms to learn dynamic network structures in Section 2; The problem definition is given in Section 3. Then we describe the details of our proposed model in Section 4 and the parameter estimation method in Section 5. The experimental results on synthetic dataset and real-world datasets are present in Section 6. Finally, we summarize the paper and conclude with future work in Section 7.

2 Related work

Serious attempts to learn dynamic networks whose topologies vary by time started in 2005 [10]. The research of dynamic relationships/networks discovery for time series can be categorized into two types: structures varying by time point and relationships varying by segment.

The research in the first category is based on that the structure of the network change at each time point. In order to make a further analysis on time series data, dynamic graph structure learning has been provided [15, 16], which reveals the dependencies between variables at the same time stamp. However, most of these algorithms only use the data from the same time stamp to learn the dependencies. Ref. [7] develops a dynamic temporal graphical model based on hidden Markov model regression and lasso-type algorithm, while the number of states has to be predefined. Ref. [8] presents a kernel reweighted $l1$-regularized auto-regressive procedure to estimate the network structure for each time stamp.

In the second category, the network structure keeps stable in a period. Ref. [6] employs a dynamic linear model with Markov switching for estimating time-dependent gene network structures. However this approach assumes that there is a fixed (user-specified) number of distinct networks or phases, and the switching between phases is modelled via a stochastic transition matrix that requires an estimation of many parameters. Some researches relax the homogeneity assumption in dynamic Bayesian networks using multiple changepoints, which segment the time series by changepoints firstly, and then fit an invariant structure to each segment [9, 11]. However, these methods is not suitable for networks whose time lag can not be ignored.

To extend conventional methods, which only aim to infer the dynamic network structure, our work builds on recent research in combining dynamic Bayesian networks with adaptive lag estimation. Different from the previous approaches [1, 17] which only learn the network structure by feature selection, we also consider the sample scarcity when we infer the network structure of each segment.

Our work is also related to some work which focus on learning the lags from temporal data. For example, the work in [4] proposes to estimate the maximum lag existing among the temporal variables for grouped graphical Granger models; a maximum likelihood estimator has been developed for determining the time delay in correlation analysis [12]; a similar work that focuses on estimating the time lag of stream correlation discovery is also proposed in [18]. However, the correlation based [12, 18] considers a different problem, since the correlation essentially differs from the dependency relations. That is, a variable A correlates to another variable B is equal to B correlates A, but in dependency modeling, this relationship is not invertible. Moreover, these studies only focus on inferring the lag from temporal data while they do not aim to infer the dependency relationship at the same time.

3 Problem definition

Supposing the network contains d observed nodes $X_1, \ldots, X_d$, we aim to discovery how the network structure varies by time. In this paper, we assume that the network structures remain unchanged in a period of time and the lags of different variables probably are distinct [17].

For example, in Figure 1, it exists three types of network structures. Here we just take X_2 as an example, in the first segment, both X_1 and X_3 are influential factors, and the lag is 1 and 2 respectively. However, in the second segment, only X_1 is the influential factor, and the lag is 2.

Given a N-by-d data matrix $\boldsymbol{D}$, where the columns correspond to the d variables and the rows correspond to N temporal observations, let $y_{i,t}$ denote the target variable selected from $\boldsymbol{X}$ associated with node i at the time point $t \in 1, \ldots, N$, and let $X_{i,t}$ denote the variable X_i at time t. In our model, the time series is segmented by k changepoints ξ. Table 1 lists the notations to be used extensively in the rest of this paper.

We introduce a lag tensor $\boldsymbol{L}$, where $\boldsymbol{L} = \boldsymbol{L}[\cdot] = (\boldsymbol{L}^1, \ldots, \boldsymbol{L}^d)$ is a sequence of $(d-1) \times (k+1)$ matrices and $L^i_{j,s}$ means the lag of X_j in the s-th segment when X_i is the target variable. Accordingly, we define the coefficient tensor $\boldsymbol{\beta}$ in $\mathbb{C}^{(d-1)\times(k+1)\times d}$ where each $(d-1) \times (k+1)$ slice $\boldsymbol{\beta}^i$ is the coefficient matrix

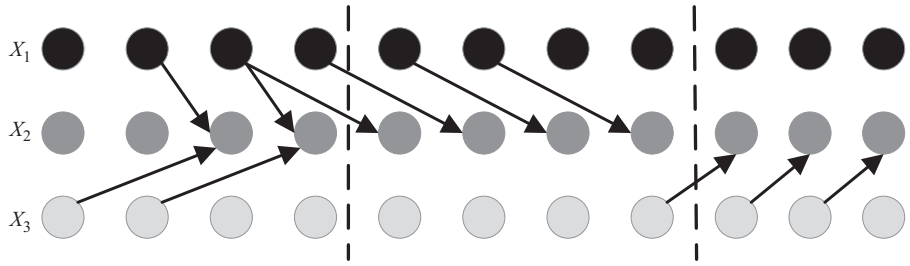

Figure 1 Illustration of the dynamic dependency network with varying lags.

Table 1 Notation explanation

Notation	Description
$\boldsymbol{X}$	The observed variables
$\boldsymbol{y}$	The target variable selected from $\boldsymbol{X}$
d	The dimension of all the observed variables
N	The length of observations
k	The number of changepoints
ξ	The vector of changepoints
$\boldsymbol{\beta}$	The tensor of regression coefficients
$\boldsymbol{L}$	The tensor of lags
s	The index of segment

of i-th target variable and $\beta^i_{j,s}$ means the coefficient of X_j in the s-th segment when X_i is the target variable.

we follow [1] and apply a linear Gaussian regression model to each target variable:

$$y_{i,t} = \sum_{j=1}^{d-1} X_{j,t-L^i_{j,s(t)}} \beta^i_{j,s(t)} + \varepsilon_{i,t}, \tag{1}$$

where $s(t)$ is the segment index which time t belongs to, and $\varepsilon_{i,t}$ is the noise, which is Gaussian distributed with zero mean and variance σ^2.

The optimal lags and structures are obtained as follows:

$$\arg\min_{\boldsymbol{L},\boldsymbol{\beta},k} \sum_{t=1}^{N} \sum_{s=1}^{k+1} \sum_{i=1}^{d} \left\| y_{i,t} - \sum_{j=1}^{d-1} X_{j,t-L^i_{j,s(t)}} \beta^i_{j,s(t)} \right\|_2^2 + \lambda \left\| \beta^i_{:,s} \right\|_1, \tag{2}$$

where $\left\|\boldsymbol{\beta}^{\boldsymbol{i}}_{:,\boldsymbol{s}}\right\|_1$ is the L1-penalty term [19].

4 Proposed model

In this paper, we propose a method which learns the dynamic network structures with lag estimation. In the proposed approach, we decompose the problem of estimating the time-varying network structures into two procedures along different axes. The first procedure is to search the optimal changepoints along the time axis; and the second one is to estimate the network structures and the adaptive lags simultaneously along the axis of variable set. The benefit of the decomposition is to reduce the estimation problem into a set of atomic optimizations.

Given the changepoints, if the number of samples in a segment is very small such as one or two, and we follow the naive assumption that each segment is a completely different network, the task of jointly estimating regression coefficients in the small segment by maximizing the log-likelihood may be statistically impossible because the estimator would suffer from high variance due to sample scarcity. Therefore, we make a statistically tractable yet realistic assumption that the network structures are vary

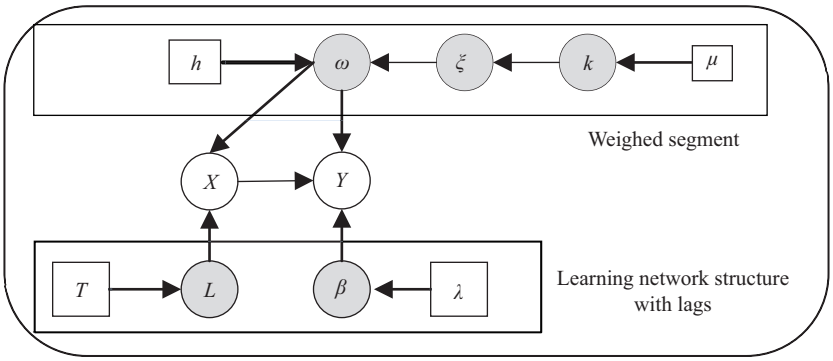

Figure 2 Plate notation for the DBAL model. Blank circles are observations, shaded circles are latent variables, and the variables in squares are model parameters.

smoothly across time; and hence temporally adjacent networks are likely to share common edges than temporally distal networks [8].

To avoid the sample scarcity, we give every segment s a weight and use all the weighted segments as the sample set to learn the network structure as well as adaptive lags for one segment. To uncover the network structures, we add a sparse prior for $\boldsymbol{\beta}$, instead of sampling network topology structure like [1], which can reduce the sampling times. Figure 2 shows the plate notation for our model.

4.1 Weighted segment

Firstly, in order to reinforce sparsity of the network and following multiple changepoint approaches, we assume that the number of changepoints k is sampled from a truncated poisson distribution with mean μ and maximum $k_{\max} = N - 1$:

$$\Pr(k\,|\mu) = \frac{\mathrm{e}^{-\mu}\mu^k}{k!}1_{\{k<k_{\max}\}}. \tag{3}$$

Here μ can be interpreted as the expected number of changepoints. Following [9], μ is drawn according to a Gamma distribution $\mu \sim Ga(a, b)$ where the shape parameter a and the scale parameter b should be chosen appropriately so that the prior probability decreases when the numbers of changepoints increase.

Given k changepoints, we assume that the changepoint positions vector $\xi = \{\xi_1, \ldots, \xi_k\}$ takes only uniformly distributed integer values:

$$\Pr(\xi\,|k) = 1 \Big/ \begin{pmatrix} N-1 \\ k \end{pmatrix}. \tag{4}$$

It is a reasonable assumption that the data in the same segment share the same network structure. Hence, we need to learn the network struture of each segment instead of each data. However, only using the data from one segment may lead to the scarcity of the training samples [8]. In this paper, we assume that the network structures of temporally adjacent segments are more similar than those of segments temporally far away [20]. According to this, we use the data from all the segments with different weights as the training set to learn the network structure and adaptive lags for one segment.

Given the target variable y_i, when we estimate the network structure of the $s1$-th segment, the weighting of all the observations in the $s2$-th segment is defined as follows:

$$w^i_{s1,s2} = \frac{K_h(tc_{s2} - tc_{s1})}{\sum_{s2=1}^{k+1} K_h(tc_{s2} - tc_{s1})}, \tag{5}$$

where $K_h(\cdot)$ is a symmetric nonnegative kernel function and h is the kernel bandwidth. tc_{s2} and tc_{s1} are the temporal center points of the $s2$-th segment and the $s1$-th segment, respectively. Here we use the distance between the temporal center points of two segments to represent the distance between these two

segments. The smaller the distance between the segment $s1$ and the target segment $s2$, the larger the weight of segment $s2$. In this paper, we use a Gaussian RBF kernel:

$$K_h(t) = \exp\left(-\frac{t^2}{h}\right). \tag{6}$$

Different time points at the same segment are equally treated by assigning them the same weight in the proposed method. Note that multiple time points at the same period are considered as i.i.d. observations and can be trivially handled by assigning them the same weight. Then when we estimate the network structure of i-th target variable y_i in the s-th segment, the likelihood becomes as follows:

$$\Pr(y_i \,|\beta^i_{:,s}\,, \boldsymbol{X}, L^i_{:,s}, \sigma_0) = \prod_{t=1}^{N} N\left(y_{i,t}\middle|\sum_{j=1}^{d-1} X_{j,t-L^i_{j,s}}\beta^i_{j,s}, \sigma_0^2\right)^{w^i_{s,s(t)}}, \tag{7}$$

where the variance of additive Gaussian noise is σ_0^2. Additionally, we define $\xi_0 = 0$ and $\xi_{k+1} = N$.

4.2 Learning network structure with lags

The proposed method not only uncovers the dynamic network structures, but also learns the corresponding lags for each segment. We assume that the maximum time delay of the effect period for all segments is T, then the value of the $L^i_{j,s}$ is within the interval $[1, T]$. Here we assume that the hyper-parameters $L^i_{j,s}$ are sampled from uniform distributions:

$$L^i_{j,s} \sim \text{uniform}(1, T). \tag{8}$$

In order to control the sparsity of regression coefficients $\boldsymbol{\beta}$, we use the spike and slab prior [21, 22] which is the prior that puts a positive probability mass on values equal to zero for the model coefficients of each variable. Because spike-and-slab priors can be expressed in terms of a set of latent binary variables that specify whether each coefficient is assigned to the spike or to the slab. The expected value of these latent variables under the posterior distribution yields an estimate of the probabilities that the corresponding model coefficients are actually different from zero. These estimates can be very useful for identifying relevant features. Besides, spike-and-slab priors have a closed-form convolution with the Gaussian distribution. This is an advantage when we use approximate inference methods based on Gaussian approximations [23].

We introduce a tensor $\boldsymbol{Z}$ where $\boldsymbol{Z}$ is a sequence of binary latent matrix $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^{(d-1)}$. Each $Z^i_{j,s}$ indicates whether $\beta^i_{j,s}$ is zero ($Z^i_{j,s}$=0) or different from zero ($Z^i_{j,s}$=1). When $\boldsymbol{Z}$ is known, the prior for $\boldsymbol{\beta}$ is defined as

$$\Pr(\boldsymbol{\beta}\,|\boldsymbol{Z}\,) = \prod_{i=1}^{d}\prod_{j=1}^{d-1}\prod_{s=1}^{k+1}\left[Z^i_{j,s}N(\beta^i_{j,s}\,|0, \nu_0\,) + (1 - Z^i_{j,s})\delta(\beta^i_{j,s})\right], \tag{9}$$

where $\delta(\cdot)$ is a Dirac delta function, also known as a point probability mass centered at 0, $N(\cdot|0, v_0)$ is a Gaussian density with zero mean and a specific variance ν_0 (the slab).

Here the value of ν_0 controls the shrinkage of the coefficients which are different from zero. If ν_0 is large, the coefficients of the groups which are different from zero are barely regularized. Conversely, if ν_0 is small, these coefficients are strongly shrunk towards zero. The parameter $\boldsymbol{Z}$ is sampled from a multivariate Bernoulli distribution:

$$\Pr(\boldsymbol{Z}) = \text{Bern}(\boldsymbol{Z}\,|\boldsymbol{p}), \tag{10}$$

where $\boldsymbol{p}$ is a sequence of d probability matrixs, and each $p^i_{j,s}$ is the prior probability that $\beta^i_{j,s}$ is different from zero.

From the hierarchical structure of the overall parameter space, we obtain the joint probability distribution over all parameters $\boldsymbol{\Theta} = (k, \xi, \boldsymbol{L}, \boldsymbol{\beta}, \boldsymbol{Z})$ as follows:

$$\Pr(\Theta|\boldsymbol{X}, \boldsymbol{y}, \mu) \propto p(k|\mu)p(\xi|k)p(\boldsymbol{\beta}|\boldsymbol{Z})p(\boldsymbol{Z})p(\boldsymbol{L})p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma, \boldsymbol{L}). \tag{11}$$

5 Parameter estimation

Since the overall parameter space of the proposed time-varying network structure model is the union of the parameter spaces of all segments delimited by k changepoints, the dimension of parameters for the model is unknown and can vary substantially. Moreover, the posterior probability is intractable to sample. Hence, in order to simultaneously consider all possible combinations of changepoints and network structures within the different segments, we propose a novel algorithm which incorporates the RJMCMC procedure [13] and the EP method [14]. RJMCMC solves the difficulty of comparing models with differing dimensionality. We can choose the "plausible" dimensionality and then get a optimal model by applying it in MCMC computation [24] and the intractable posterior probabilities can be approximated by easy ones through EP [14].

We want to infer time-varying network model, which belongs to the overall parameter space that is the union of the parameter spaces of all phases delimited by k changepoints. Adding or removing a changepoint results in a change in the dimension of the system's state-space: for each additional changepoint a new network structure has to be estimated, and for each deleted changepoint the results previously obtained for the two distinct phases have to be reconciled. Thus, the dimension of the model is unknown and can vary substantially. In order to infer the posterior distribution $\Pr(k, \xi, \boldsymbol{L}, \boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{y})$ given the observed data $\boldsymbol{X}, \boldsymbol{y}$ over all of the system's parameters, we used a Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure. The principle of RJMCMC lies in constructing a reversible Markov chain sampler that can jump between parameter subspaces of different dimensions; thus allowing the generation of an ergodic Markov chain whose equilibrium distribution is the desired posterior distribution [13, 25].

In order to traverse the parameter space of unknown dimension, we propose here four different update moves: birth of a new changepoint, death of an existing changepoint, shift of a changepoint to a different time-point, and update of the relationships within the segments. These moves occur with probabilities b_k, d_k, q_k, t_k respectively, depending only on the current number of changepoints k and satisfying $b_k + d_k + q_k + t_k = 1$.

A changepoint birth or death acceptance is performed without generating the regression model parameters for the modified segment. When we estimate the structure of i-th target variable in the s-th segment, the integration over $\beta^i_{:,s}$ yields,

$$\begin{aligned}\Pr(k, \xi, Z^i_{j,s}, \boldsymbol{X}, y_i, L^i_{j,s}) \propto & \frac{\mu^k (N-1-k)!}{(N-1)!} \prod_{j=1}^{d} \mathrm{Bern}(Z^i_{j,s}|p^i_{j,s}) \\ & \times \left[Z^i_{j,s} N(0 \,|m^i_{s,j}, v_0 + v^i_{s,j}) + (1 - Z^i_{j,s}) N(0|m^i_{s,j}, v^i_{s,j}) \right],\end{aligned} \tag{12}$$

where

$$\begin{aligned}&\boldsymbol{v}^{-1} = \tfrac{1}{\sigma_0^2}(\boldsymbol{X}')^{\mathrm{T}}\boldsymbol{X}', \quad \boldsymbol{v}^{-1} m^i_s = \tfrac{1}{\sigma_0^2}(\boldsymbol{X}')^{\mathrm{T}}\boldsymbol{y}', \quad v^i_{s,:} = \mathrm{diag}(\boldsymbol{v}), \\ &(y^i_t)' = \sqrt{w^i_{s,s(t)}}\, y^i_t, \quad X'_{:,t} = \left[X_{1,t-L^i_{1,s(t)}}, \ldots, X_{p-1,t-L^i_{p-1,s(t)}} \right] \sqrt{w^i_{s,s(t)}}.\end{aligned} \tag{13}$$

The RJMCMC acceptance probability for a changepoint change ('birth', 'death', 'shift') can be written as

$$A = \min(1, r),$$

where $r =$ (posterior distribution ratio) $\times$ (proposal ratio) $\times$ (Jacobian). The move is accepted with probability $\min(1, r)$.

We use $\Pr_{\mathrm{joint}}$ to represent the joint distribution of parameters shown in (12) and $\Pr^*_{\mathrm{joint}}$ denotes the updated joint distribution. For different moves, the probability r is given as follows respectively:

$$r_{\mathrm{birth}} = \frac{\Pr^*_{\mathrm{joint}}(N-k-1)(d-1)!}{\Pr_{\mathrm{joint}} \lambda c!(d-c)!}, \tag{14}$$

$$r_{\mathrm{death}} = \frac{\Pr^*_{\mathrm{joint}} \lambda (\mathrm{d}-1)!}{\Pr_{\mathrm{joint}}(N-k) c!(d-c)!}, \tag{15}$$

$$r_{\text{shift}} = \frac{\Pr^*_{\text{joint}}}{\Pr_{\text{joint}}}, \tag{16}$$

where c is the number of dependent variables in the new segment. We omit the tedious derivations and only present the process of the entire algorithm, as shown in Algorithm 1. The Algorithm 2 is the update function used in Algorithm 1.

Algorithm 1 Procedure for estimating dynamic dependency network structures with estimation of lags

Input: data $\boldsymbol{X}$ and y, the maximum number of changepoints $k_{\max}$
Output: $k, \xi, \boldsymbol{\beta_s}, \boldsymbol{L}, \boldsymbol{Z}$

Initialization: k, ξ
Iteration i: $i = 1$
sample $\mu_0 \sim \mu[0, 1]$
if $u_0 \leqslant b_k$ **then**
    consider changepoint birth
    Propose a new changepoint $\xi^* | \xi \sim u_{\{1,\ldots,N\}\setminus\{\xi\}}$
    Update_state (current state, Eq. (14))
else if $u_0 \leqslant b_k + d_k$ **then**
    consider changepoint death
    Choose a changepoint $\xi^* \in \xi$ to be deleted
    Update_state (current state, Eq. (15))
else if $u_0 \leqslant b_k + d_k + q_k$ **then**
    consider changepoint shift
    Randomly choose a changepoint $\xi \in \xi$ shift to a new changepoint $\xi^* | \xi \sim u_{\{1,\ldots,N\}\setminus\{\xi\}}$
    Update_state (current state, Eq. (16))
else
    Update the network structure of every segment
end if
$i \leftarrow i + 1$ and go to 4

Algorithm 2 Function update_state (current state, Eq. (e))

Input: Current state, Eq. (e)
Output: New state

Evaluate acceptance probability A according to Eq. (e)
Sample $u \sim U_{[0,1]}$
if $u \leqslant A$ **then**
    Return the updated state
else
    Return the current state
end if

Update the network structure of every segment. Computing the $\beta^i_{:,s}$ and $Z^i_{:,s}$ requires sampling them from posterior probability. Due to the intractability of posterior probability of $\beta^i_{:,s}$ and $Z^i_{:,s}$ as shown in (17) and (18):

$$\begin{aligned}\Pr(\beta^i_{:,s} | \boldsymbol{Z}, \boldsymbol{L}, \xi, k) \propto & \prod_{t=1}^{N} N\left(y_{i,t} \middle| \sum_{j=1}^{d-1} X_{j,t-L^i_{j,s}} \beta^i_{j,s}, \sigma_0^2\right)^{w^i_{s,s(t)}} \\ & \times \prod_{j=1}^{d-1} \left[Z^i_{j,s} N(\beta^i_{j,s} | 0, \nu_0) + (1 - Z^i_{j,s}) \delta(\beta^i_{j,s})\right],\end{aligned} \tag{17}$$

$$\Pr(Z^i_{:,s} | \boldsymbol{\beta}, \boldsymbol{L}, \xi, k) \propto \prod_{j=1}^{d-1} \text{Bern}(Z^i_{j,s} | p^i_{j,s}) \left[Z^i_{j,s} N(\beta^i_{j,s} | 0, \nu_0) + (1 - Z^i_{j,s}) \delta(\beta^i_{j,s})\right], \tag{18}$$

we apply the expectation propagation (EP) [14, 23] to perform the inference approximately.

The essence of the EP approach is to choose a variational distribution Q to approximate the actual joint distribution, so that the Kullback-Leibler divergence (KL-divergence) between the probability $\Pr(\boldsymbol{\beta}, \boldsymbol{Z})$

Table 2 Comparative methods

Method	Network state	Varying form	Lag
SGL	Static	–	T
TVDBN	Dynamic	Time point	1
DBN	Static	–	1
NHDBNs	Dynamic	Period	1
DBNL	Dynamic	Period	1
TLHL	Static	–	Learned

and its approximation Q is minimized. Here we approximate the joint distribution of $\beta^i_{:,s}$ and $Z^i_{:,s}$ using Q as follows:

$$Q(\beta^i_{:,s}, Z^i_{:,s}) = N(\beta^i_{:,s}|M, V) \prod_{j=1}^{d-1} \text{Bern}(Z^i_{j,s}|\sigma(Z^i_{j,s})). \tag{19}$$

The posterior probability of $\beta^i_{:,s}$ and $Z^i_{:,s}$ can be obtained approximately during the process of EP algorithm. See [23] for details and an explicit expression.

As a convergence diagnostic we monitor the potential scale reduction factor (PSRF) [26], computed from the within-chain and between-chain variances of marginal network structure posterior probabilities. Values of PSRF$\leqslant 1.1$ are usually taken as indication of sufficient convergence.

6 Experiment

Experiment setup. To evaluate the effectiveness of our model, we conduct thorough experiments on a synthetic dataset and real world datasets. Baselines used for comparison include static Granger lasso (SGL) model [27], time-varying dynamic Bayesian networks (TVDBN) [8], traditional dynamic Bayesian networks (DBN) [28,29], Non-homogeneous DBNs (NHDBN) [20], and the TLHL model [30]. To illustrate the importance of time lags in learning dependency network structure, we also compare with our simplified model which removes the time lag learning part (DBNL). Table 2 illustrates the model characteristic of comparative methods.

In our model, the range T of time lags is 10, the shape parameter $a = 1$ and the scale parameter $b = 0.5$. For TVDBN, we set the bandwidth parameter h of the Gaussian kernel according to the spacing between two adjacent segments such that $\exp(-\frac{N^2}{49h}) = \exp(-1)$. For the static granger lasso, we select the penalty parameter λ using the cross validation. For the static models SGL and TLHL, we set the maximum lag as 10. While for TVDBN and NHDBN, the time lags are set to 1 as applied in original studies [8, 20].

6.1 Synthetic data

Data generation. We generate 10 networks with 7 nodes each ($d = 7$), and the length of observation series of each node is 500 ($N = 500$). The simulation procedure involves three main steps. Firstly, to simulate changes in the network structures, we set 3 changepoints (except the beginning and end) and the localization vector is $\xi = [200, 300, 400]$ in the first data set.

Then, for each segment s, we set the network structure and then generate the corresponding regression coefficient matrix $\boldsymbol{\beta}^s$ $(d \times d)$. We choose the regression weights such that $\beta^i_{j,s} = 0$ if there is no edge from j to i in the network structure for segment s, and $\beta^s_{j,s} \sim N(0, 1)$ otherwise. Meanwhile, we randomly generate the lag matrix $\boldsymbol{L}$, where $L^s_{j,s} \sim U[1, 10]$.

Last, we generated time series of length l which is the length of segment s using a linear regression model as shown in (1).

We added Gaussian observation noise $\varepsilon_i \sim N(0, 1)$ independently for each observation of node i.

Experimental results. We first compare the learned $\boldsymbol{\beta}^*$ of network structure with the ground truth network structure parameter $\boldsymbol{\beta}$. As Figure 3 shows, Figure 3(a) contains four ground truth coefficient

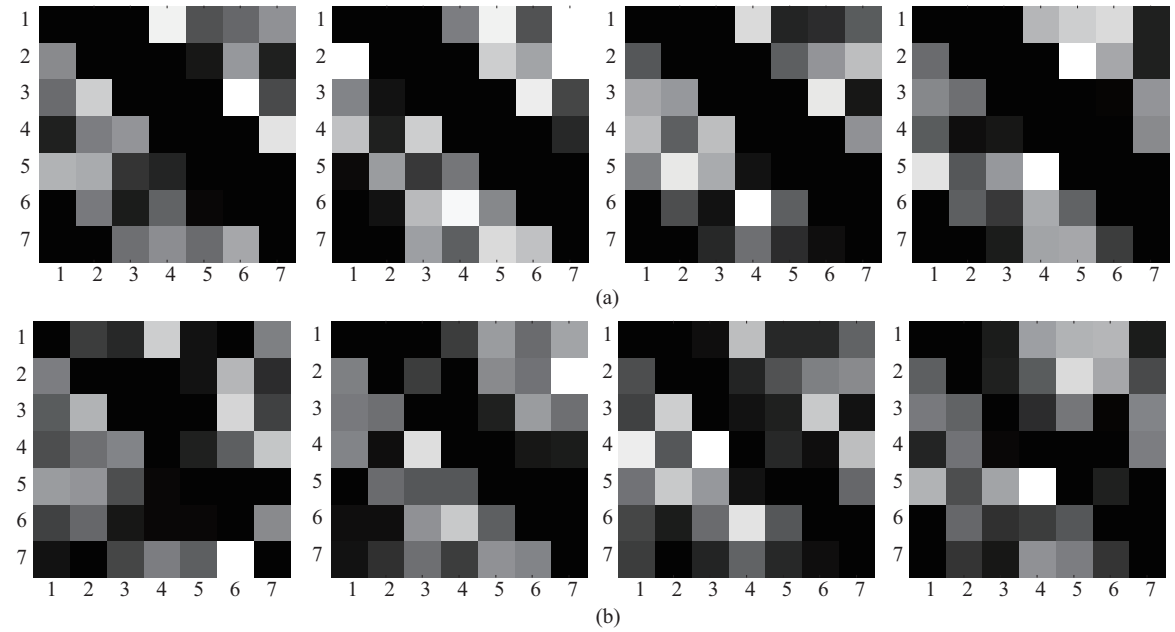

Figure 3 Comparison between the true $\boldsymbol{\beta}$ and learned $\boldsymbol{\beta}^*$. (a) The ground truth $\boldsymbol{\beta}$; (b) the learned $\boldsymbol{\beta}^*$.

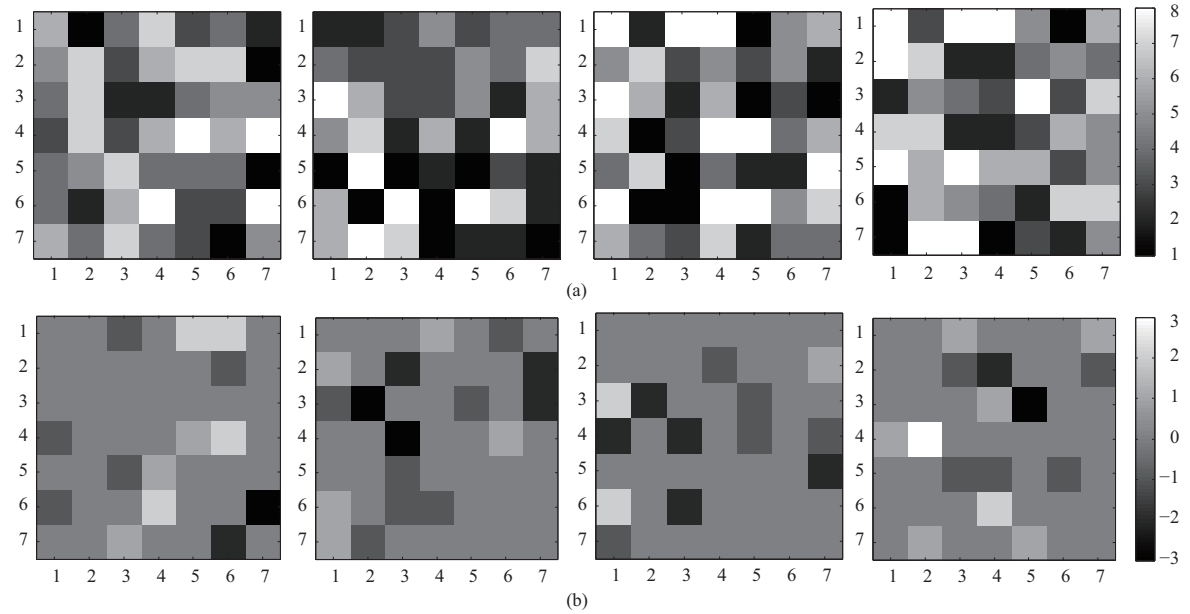

Figure 4 The true lags $\boldsymbol{L}$ and the residual matrix $\Delta\boldsymbol{L}$ between learned lags and the truth. (a) The ground truth $\boldsymbol{L}$; (b) the residual matrix $\Delta\boldsymbol{L}$ between learned lags and the truth.

matrixes for the four segments and Figure 3(b) contains four learned coefficient matrixes. Each column corresponds to one segment of the time series data. The true coefficient $\boldsymbol{\beta}$ varies over the segment. From the comparison on each column, the learned coefficient matrixes are similar with the ground truth which indicates that our model is able to learn the dynamic network structure.

Figure 4 compares the ground truth time lags with the time lags learned by our model. Figure 4(a) contains four true time lag matrixes corresponding to each segment, while Figure 4(b) contains the residual matrixes between the time lag matrixes we learned and the ground truth time lag matrixes. The time lag varies by different variable dimensions and segments. Most elements in the residual matrixes tend to be zero which means the time lags we learned are close to the ground truth.

We use the F1 score to evaluate the performance in terms of network structure learning, where the F1

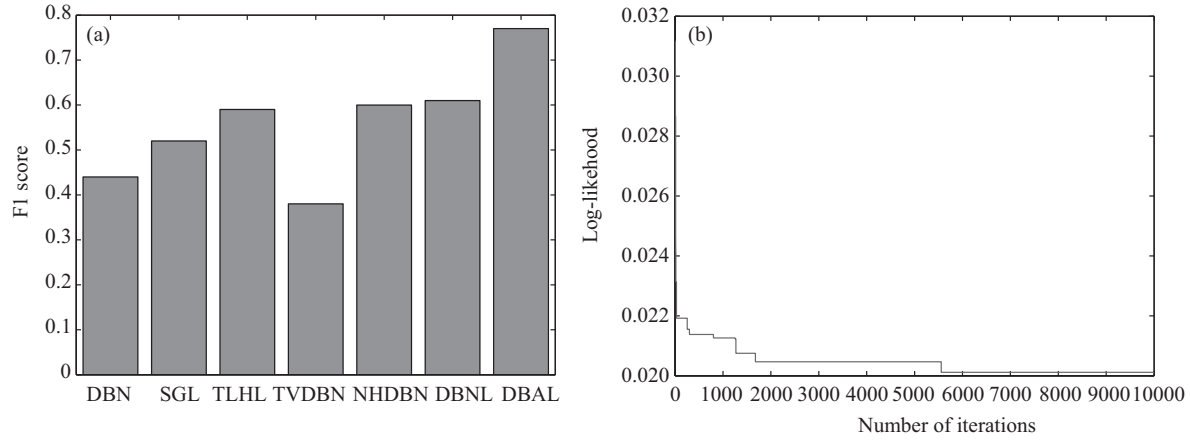

Figure 5 The F1 score of network structures learned by different methods and the log-likelihood of model. (a) The F1 score of network structures learned by different methods; (b) log-likelihood of Model.

score is computed as follows:

$$\mathrm{Pre} = \frac{\left|\left\{i,j,s \,\middle|\, \beta^i_{j,s} \neq 0,\, (\beta^i_{j,s})^* \neq 0\right\}\right|}{\left|\left\{i,j,s \,\middle|\, \beta^i_{j,s} \neq 0\right\}\right|}, \quad \mathrm{Rec} = \frac{\left|\left\{i,j,s \,\middle|\, \beta^i_{j,s} \neq 0,\, (\beta^i_{j,s})^* \neq 0\right\}\right|}{\left|\left\{i,j,s \,\middle|\, (\beta^i_{j,s})^* \neq 0\right\}\right|},$$

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Pre} \cdot \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}},$$

where $\boldsymbol{\beta}^*$ denotes the regression coefficient matrix we learned, and $\boldsymbol{\beta}$ is the true coefficient matrix we set.

From Figure 5(a), we can see that the proposed DBAL model achieves the best performance among the competing methods. Since the TLHL model learns the optimal time lag to get a smaller error, the F1 score of it is higher than that of the other methods. As for the static model, such as SGL and TLHL, they cannot capture the dynamic of networks although the F1 scores are not bad. Overall, the dynamic models outperform the static models except the TVDBN. The TVDBN method which learns network structures varying by time point with time lag set to 1 has the worst result, which is probably caused by missing important information and over-fitting. The NHDBN model learning the dynamic network structures varying by segments, which is consistent with our assumption, gains a competing result. The DBNL model without time lag learning part from the proposed model has a significant decrease in F1 score, which indicates the importance of adaptive time lag estimation in network structure learning.

Besides, Figure 5(b) shows the log-likelihood varies over the iteration in network structure learning process of our model. The log-likelihood tends to converge at about the 6000th iteration.

6.2 Air quality data

As described in Section 1, time lag is a key feature to interpret the dynamic network structure and temporal dependency, especially for spatial networks. In this section, we evaluate our DBAL model on a real-world air quality dataset collected from Beijing, China.

The air quality data is collected by 35 air quality monitoring stations, deployed in different sites in Beijing. Each air quality monitoring station measured 7 factors ($\mathrm{CO}, \mathrm{NO}_2, \mathrm{SO}_2, \mathrm{O}_3$, PM2.5, temperature and humidity) hourly during the period 2013.12.03–2014.3.31. In total, there are 2856 samples in this dataset. In this experiment, we only focus on the causal relationships of PM2.5 which attracts the most attention in recent years, and take an pollution accident as an example to illustrate the causal relationship of PM2.5.

The causal relationships of PM2.5. For the experimental setting, we select an atmospheric pollution incident in a row during February 2014 which was lasting about 15 days in Beijing. Figure 6(b) shows the estimated causal relationship graphs of PM2.5 at a station which is a place in the center of Beijing,

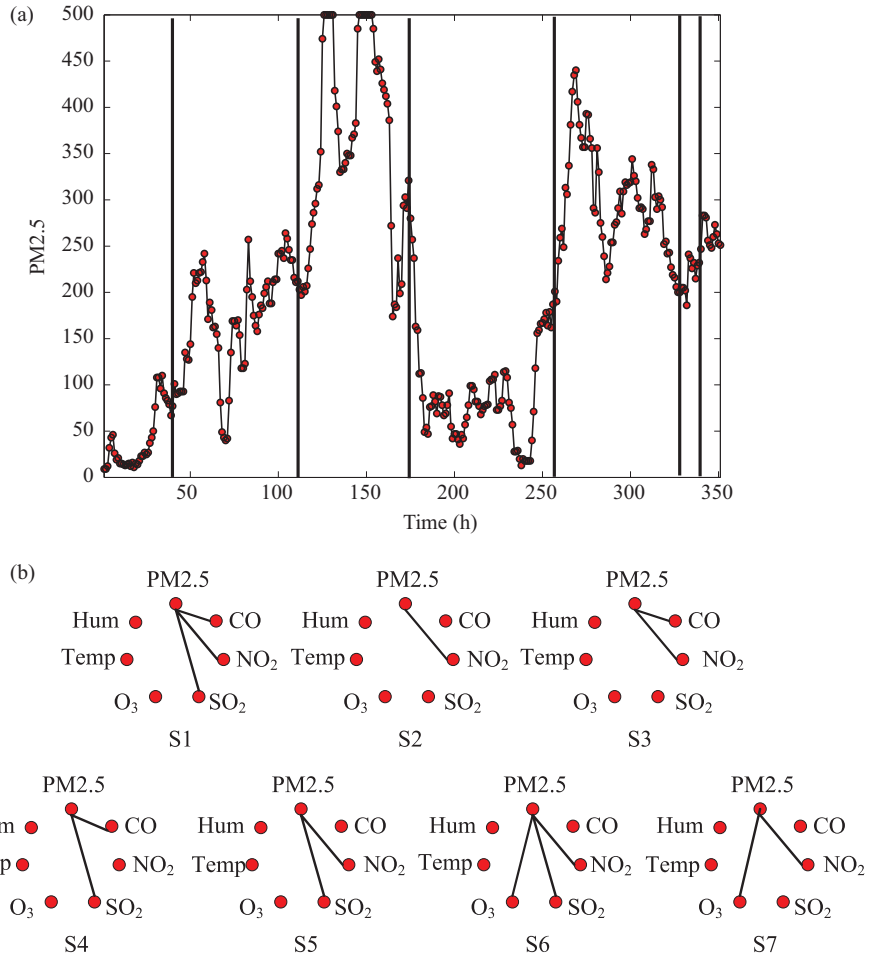

Figure 6 (Color online) The changepoints and the dynamic causal relationships of PM2.5. (a) The changepoints when we estimate the dynamic causal relationships of PM2.5; (b) the dynamic causal relationships of PM2.5.

which also reveals that the graph structures are quite similar between every two adjacent segments. It indicates that the causal relationships of air quality feature are relatively stable.

From the Figure 6(a), we can observe that concentration of PM2.5 is related to the locations of changepoints, to some extent. Based on the causal relationships graphes in Figure 6(b), it is obvious that the temperature and humidity are independent with PM2.5 all the time. However, the PM2.5 is dependent with the other factors. In other words, these factors are highly correlated and should be studied together by meteorologists. The organic carbon and elemental carbon were well correlated with PM10, CO and SO_2, which implies they have similar sources, repoted by the research [31]. In fact, there is high possibility that, due to the chemistry gas from the chemistry factories which are located in the south of Beijing. All of these facts demonstrates that our causal relationships is consistent with the truth.

The causal relationships of stations. In addition, we present some examples to illustrate the spatial casual relationships, which can help us judge the spread direction of pollution. Figure 7(a) presents the PM2.5 level of No.3 station in a curve and the changepoint using a dashed line. The changepoint learned by DBAL model partitions the PM2.5 time series data into two segments. In the left segment, the air condition is seriously polluted by PM2.5. While the right segment shows the normal air quality condition. Figure 7(b) shows the corresponding wind speed level in a curve and the changepoint using a dashed line. In both the left segment, the average wind speed level is low and the corresponding PM2.5 level is high. While in the right segment, the corresponding relationship reverse. The average wind speed level is high and the PM2.5 level is low. These segment results are quite interpretable according to domain knowledge. The wind speed level is one of the main factors for air quality pollution.

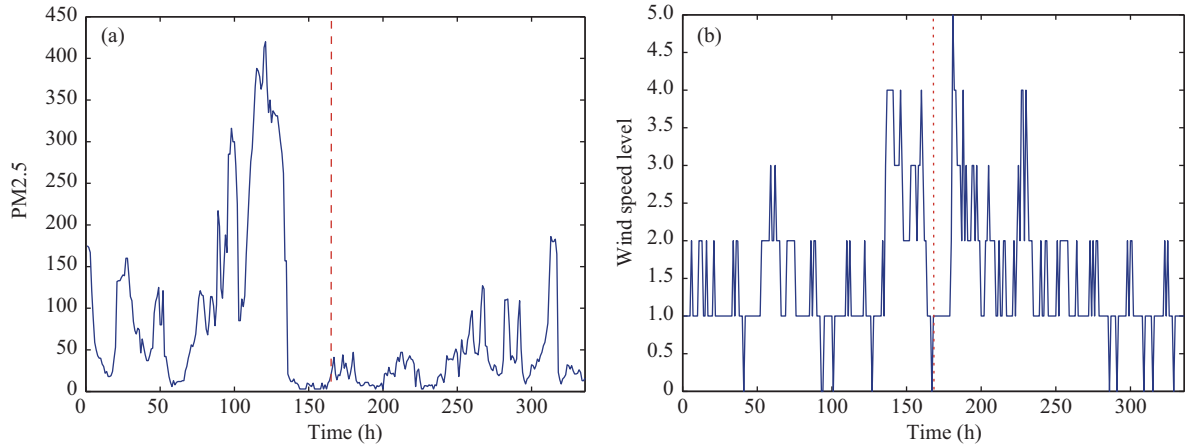

Figure 7 (Color online) The PM2.5 and and wind speed. (a) The PM2.5 and learned changepoint; (b) the wind speed level of No.3 station.

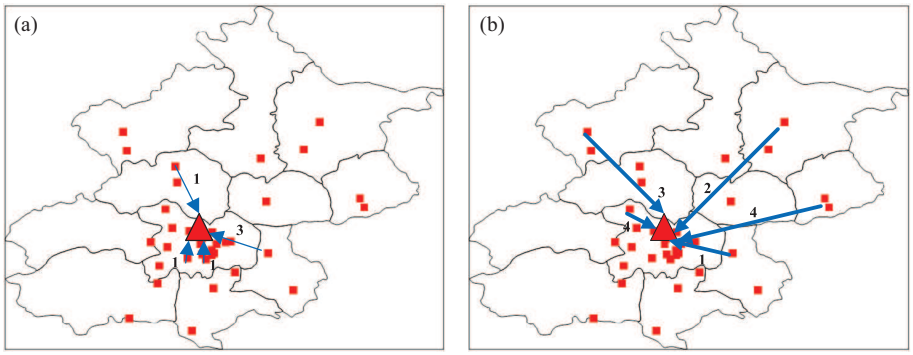

Figure 8 (Color online) The dependent stations and time lags of No.3 station. The learned dependency of No.3 station in (a) the first segment and (b) the second segment.

Figure 8(a) and (b) present the dependent stations of No.3 station with time lags for the two identified segments respectively. The triangle in the center denotes the target air quality monitoring station, i.e., the No.3 station. Each arrow origins from the dependent stations and points to the target station with a time lag number. From Figure 8(a), we can see the dependent stations are in the near neighborhood with a small learned time lag in the seriously polluted segment. While in Figure 8(b), the dependent stations are distributed in the far distance and the corresponding time lags tend to be large. These results are consistent with the fact that seriously polluted air condition tend to emerge in a windless day and good air condition is usually along with a windy day. At the same time, the influence propagation speeds from the dependent stations to the target station are small/large in a windless/windy day. Thus the time lags in the first windless segment are small. While in the second windy segment, the time lags tend to be large. These identified results are believed reasonable by domain experts, which demonstrate the proposed DBAL model is effective in learning both the dynamic network structure and the time lags.

6.3 Highway traffic data

We evaluate our method on real-life highway traffic data. The variables in this traffic dataset are observations collected from sensors located on ramps in a highway traffic network. Each observation is the vehicle count during 15 min interval. There are totally 236 traffic stations, which correspond to 236 ramps. In order to study the dynamic dependency relationships of traffic stations in the day, we use data in one day with 96 samples as an example.

The dependency structures obtained by dependency learning methods are essentially important for the analysis of the traffic systems, such as vehicle flow prediction, anomaly detection. Domain experts

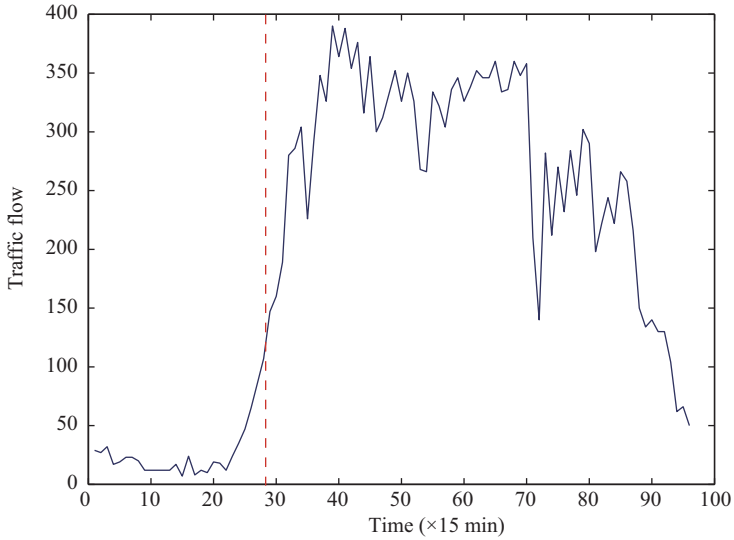

Figure 9 (Color online) The traffic flow and learned changepoint.

can obtain the accurate dependencies from the information of upstream stations, and predict the state of downstream stations. The time lags we learned companying with the network structure learning offer the time delay information between two stations.

Figure 9 presents the traffic flow in a curve and the changepoint using a dashed line. The changepoint learned by DBAL model partitions the traffic time series data into two segments. It is obvious that the traffic flow in the first segment is much smaller than that in the second segment. The location of changepoint is 28, which corresponds to the 7:00 am. From Figure 9, we can conclude that there are two types of dependency relationships of stations in one day, where one occurs from 0:00 to 7:00 and the other one is in the rest time.

Figure 10(a) and (b) present the learned dependency between traffic stations in the two segments respectively. As for the first segment, the target station which is pointed by arrows mainly depends on the stations from the east with large time lags. In the second segment, the target station depends on the stations in different directions. Comparing the time lags we learned in the two segments, the lags in the first segment are larger than those in the second segment. These results can be explained by the domain knowledge that the target station's traffic flow from the midnight to early morning mainly consists of long-distance trucks with low speeds, so the time lags tend to be much larger. While in other time the traffic flow is composed by both short-distance vehicles and long-distance vehicles from various directions. The learned results can be further applied for traffic flow prediction and traffic control. The experiment on highway traffic network indicates that the time lag problem actually exists in real-world traffic network and the proposed model is effective to learn both the dynamic network structure and time lags simultaneously.

7 Conclusion

The time lag is a crucial and inevitable parameter in dynamic network structure learning and understanding. In this paper, we propose a dynamic Bayesian model which simultaneously integrates two usually sperate tasks, i.e., learning the dynamic network structure and estimating the time lags, within one unified framework. Then a novel weight kernel approach is proposed for time series segmenting and sampling to avoid the sample scarcity based on the assumption that the network structure of adjacent segments are similar. For parameter inference, we propose an effective Bayesian scheme cooperated with RJMCMC and EP algorithms. We evaluate our model on one synthetic dataset and two real-world datasets. Experiment results demonstrate the effectiveness of the proposed model in learning the dynamic network structure with the adaptive time lag estimation. Moreover, the results on air quality network and the

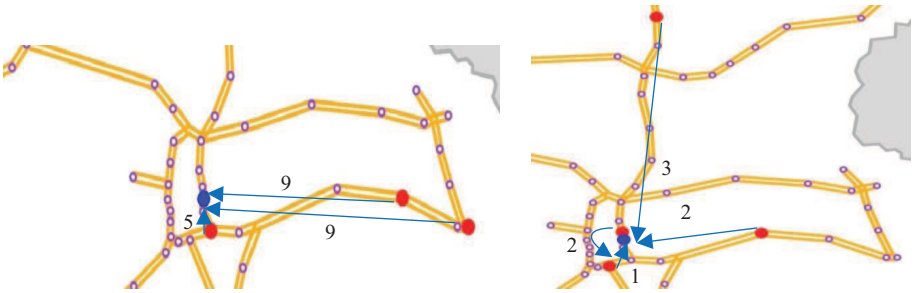

Figure 10 (Color online) The learned dependency in highway traffic network. The learned dependency between traffic ramps in (a) the first segment and (b) the second segment.

highway traffic network are believed reasonable and interpretable by domain experts. Our model is also applicable to other networks with time lag characteristic. For future work, it is interesting to apply this model in other networks.

Acknowledgements This work was supported by National Science and Technology Support Plan (Grant No. 2014BAG01B02), National Natural Science Foundation of China (Grant No. 61572041), Joint Funds of the National Natural Science Foundation of China (Grant No. U1404604), and Beijing Natural Science Foundation (Grant No. 4152023).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Lebre S, Becq J, Devaux F, et al. Statistical inference of the time-varying structure of gene-regulation networks. BMC Syst Biol, 2010, 4: 130
2 Goldenberg A, Moore A W. Bayes net graphs to understand co-authorship networks? In: Proceedings of the 3rd International Workshop on Link Discovery, New York, 2005. 1–8
3 Chen X, Liu Y, Liu H, et al. Learning spatial-temporal varying graphs with applications to climate data analysis. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, 2010
4 Dhurandhar A. Learning maximum lag for grouped graphical granger models. In: Proceedings of IEEE International Conference on Data Mining Workshops. Washington: IEEE Computer Society, 2010. 217–224
5 Barthélemy M. Spatial networks. Phys Rep, 2011, 499: 1–101
6 Grzegorczyk M. A non-homogeneous dynamic Bayesian network with a hidden Markov model dependency structure among the temporal data points. Mach Learn, 2016, 102: 155–207
7 Liu Y, Kalagnanam J R, Johnsen O. Learning dynamic temporal graphs for oil-production equipment monitoring system. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, 2009. 1225–1234
8 Song L, Kolar M, Xing E P. Time-varying dynamic Bayesian networks. In: Proceedings of Advances in Neural Information Processing Systems, Vancouver, 2009. 1732–1740
9 Dobigeon N, Tourneret J Y, Davy M. Joint segmentation of piecewise constant autoregressive processes by using a hierarchical model and a Bayesian sampling approach. IEEE Trans Signal Process, 2007, 55: 1251–1263
10 Talih M, Hengartner N. Structural learning with time-varying components: tracking the cross-section of financial time series. J Roy Stat Soc B, 2005, 67: 321–341
11 Xuan X, Murphy K. Modeling changing dependency structure in multivariate time series. In: Proceedings of the 24th International Conference on Machine Learning, Omaha, 2007. 1055–1062
12 Knapp C, Carter G. The generalized correlation method for estimation of time delay. IEEE Trans Aoust Speech, 1976, 24: 320–327
13 Green P J. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika, 1995, 82: 711–732
14 Minka T P. Expectation propagation for approximate Bayesian inference. In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, Seattle, 2001. 362–369
15 Eaton D, Murphy K. Bayesian structure learning using dynamic programming and MCMC. arXiv:1206.5247
16 Guo F, Hanneke S, Fu W, et al. Recovering temporally rewiring networks: a model-based approach. In: Proceedings of the 24th International Conference on Machine Learning, Corvallis, 2007. 321–328
17 Husmeier D, Dondelinger F, Lebre S. Inter-time segment information sharing for non-homogeneous dynamic Bayesian networks. In: Proceedings of Conference and Workshop on Neural Information Processing Systems (NIPS), Vancouver,

2010. 901–909
18 Sakurai Y, Papadimitriou S, Faloutsos C. Braid: stream mining through group lag correlations. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, Melbourne, 2005. 599–610
19 Tibshirani R. Regression shrinkage and selection via the lasso. J Roy Stat Soc B, 1996, 58: 267–288
20 Dondelinger F, Lèbre S, Husmeier D. Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. Mach Learn, 2013, 90: 191–230
21 Ishwaran H, Rao J S. Spike and slab variable selection: frequentist and Bayesian strategies. Ann Stat, 2005, 33: 730–773
22 George E I, McCulloch R E. Approaches for Bayesian variable selection. Stat Sin, 1997, 7: 339–373
23 Hernández-Lobato J M, Hernández-Lobato D, Suárez A. Expectation propagation in linear regression models with spike-and-slab priors. Mach Learn, 2015, 99: 437–487
24 Green P J, Hastie D I. Reversible jump MCMC. Genetics, 2009, 155: 1391–1403
25 Robert C P, Ryden T, Titterington D M. Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. J Roy Stat Soc B, 2000, 62: 57–75
26 Gelman A, Rubin D B. Inference from iterative simulation using multiple sequences. Stat Sci, 1992, 7: 457–472
27 Arnold A, Liu Y, Abe N. Temporal causal modeling with graphical granger methods. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, 2007. 66–75
28 Murphy K. An introduction to graphical models. Rap Tech, 2001: 1–19
29 Dagum P, Galper A, Horvitz E. Dynamic network models for forecasting. In: Proceedings of the 8th International Conference on Uncertainty in Artificial Intelligence, Stanford, 1992. 41–48
30 Zhou X, Hong H, Xing X, et al. Mining dependencies considering time lag in spatio-temporal traffic data. In: Proceedings of International Conference on Web-Age Information Management, Qingdao, 2015. 285–296
31 Zhang R J, Co J, Lee S, et al. Carbonaceous aerosols in PM10 and pollution gases in winter in Beijing. J Environ Sci, 2007, 19: 564–571