

Time Lag Concerned Dynamic Dependency Network Structure Learning

Sizhen Du
Key Laboratory of Machine
Perception, Ministry of
Education, Peking University
No.5 Yiheyuan Road
Beijing, China
sizhen@pku.edu.cn

Haikun Hong
Key Laboratory of Machine
Perception, Ministry of
Education, Peking University
No.5 Yiheyuan Road
Beijing, China
honghaikun@hotmail.com

Guojie Song^{*}
Key Laboratory of Machine
Perception, Ministry of
Education, Peking University
No.5 Yiheyuan Road
Beijing, China
gjsong@pku.edu.cn

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 time lags cost to propagate influences from one node to another tend to vary over both space and time due to the different space distances and propagation speeds between nodes. Thus 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 which simultaneously integrates two usually separate tasks, i.e. learning the dynamic dependency network structure and estimating time lags, within one unified framework. Besides, we propose a novel weight kernel approach for time series segmenting and sampling via leveraging samples from adjacent segments to avoid the sample scarcity and an effective Bayesian scheme cooperated with RJMCMC and EP algorithms for parameter inference. To our knowledge, this is the first practical work for dynamic network structure learning concerned with adaptive time lag estimation. 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

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 [18], social network analysis [10], climate data analysis [3] and so on. 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 [2], 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 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 interval 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, wind speed, wind direction, 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 and temporal causal discovery. Some researches focus on the dependency relationship of every time point in time series, such as [21] [19] [12]. While some researches regard that the dependency network structures vary by time intervals, such as [5] [23] [18] [22]. As aforementioned, most of the existing studies do not take the dynamic time lag into account and only simply treat time lag as a predefined constant.

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 [21]. (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 to learn

^{*}Corresponding author. Email: gjsong@pku.edu.cn

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 prior. In parameter inference part, an effective Bayesian scheme incorporating the RJMCMC and Expectation Propagation algorithm (EP) is proposed in order to infer the parameters of the model, where the dimension of the parameter is unknown and the posterior probability is intractable to sample.

To the best of our knowledge, this is the first practical method for dynamic structure learning with adaptive time lag estimation. 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.
- We propose a novel weight kernel approach for time series segmenting and sampling to avoid the sample scarcity 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 algorithms.
- We evaluate our method on a synthetic dataset and two real world datasets. Experimental results show the effectiveness and promise of the proposed model.

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 [24] [22]. The research of dynamic relationships/networks discovery for time series can be categorized into two types: structures varying by time points and relationships varying by segments.

The research in the first category is based on that the structures of the networks change at each time point. In order to make a further analysis on time series data, dynamic graph structure learning has been provided [7] [13], 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. [19] 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. [21] presents a kernel reweighted $l1$ -regularized auto-regressive procedure to estimate the network structure for each time stamp.

In the second category, the network structures keep stable in a period. [12] 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 research 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 [5] [23]. 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 structures, our work builds on recent research in combining dynamic Bayesian networks with adaptive lag estimation. Different from the previous approaches [18] [15] which only learn the network structure by feature selection, we also consider the sample scarcity when infer the network structure of each segment.

Our work is also related to some works which focus on learning the lags from temporal data. For example, the work [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 [17]. However, these works only focus on inferring the lag from temporal data while the relationships are not considered at the same time. In this paper, we aim to learn the dependency relationships and the lag interval simultaneously from data.

3. PROBLEM DEFINITION

Given p nodes, where the observed variables X_1, \dots, X_p , 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.

For example, in the Fig.1, the network structures are different in the first segment and the last segment. In the first segment, for X_2 , both X_1 and X_3 are influential factors, and the lag is 1 and 2 respectively. However, in the last segment, only X_3 is the influential factor, and the lag is 1.

Given a N -by- p data matrix D , where the columns correspond to the p nodes and the rows correspond to N temporal observations, let $y_{i,t}$ denotes the target variable selected from \mathbf{X} associated with node i at the time point $t \in 1, \dots, N$, and let $X_{i,t}$ denotes 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.

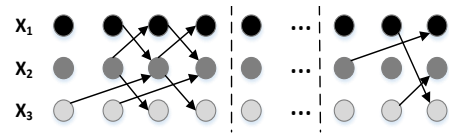


Figure 1: Illustration of the dynamic dependency network with varying lags.

Given a fixed structure coefficient matrix β , and the lag matrix $L_{s,i}$. We follow [18] and apply a linear Gaussian regression model to each target variable:

$$y_{i,t} = \sum_{j=1}^{p-1} X_{j,t-L_{s,i,j}} \beta_{s,j} + \varepsilon_i, \quad j \neq i \quad (1)$$

Table 1: Notation explanation

Notation	Description
X	The observed variables
y	The target variable selected from X
p	The dimension of all the observed variables
s	The index of segment
N	The length of observations
k	The number of segments
ξ	The vector of changepoints
β	The matrix of regression coefficients
L	The matrix of lags

where s is the segment index which time t belongs to, and ε_i is the noise, which is Gaussian distributed with zero mean and variance σ^2 .

The optimal lags and structures are obtained as follows:

$$\operatorname{argmin}_{L, \beta, \xi, k} \sum_{s=1}^{k+1} \sum_{i=1}^p \sum_{t=\xi_{s-1}+1}^{\xi_s} \left\| y_{i,t} - \sum_{j=1}^{p-1} X_{j,t-L_{s,j}} \beta_{s,j} \right\|_2^2 + \lambda \|\beta\| \quad (2)$$

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. One benefit of the decomposition is to reduce the estimation problem into a set of atomic optimizations.

Given the changepoints, to avoid the sample scarcity, we give every segment s a weight ω_s 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 β , instead of sampling network topology structure like [18], which can reduce the sampling times. Fig.2 shows the plate notation for our model.

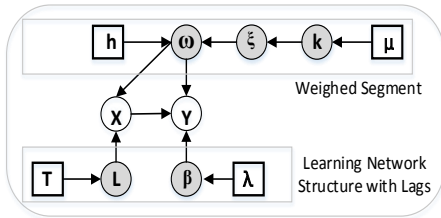


Figure 2: Plate notation for the DBAL model. Blank circles are observations, shaded circles are latent variables and the variables in squares are model parameters.

4.1 Weighted Segment

Firstly, we assume that the number of changepoints k is sampled from a truncated poisson distribution with mean μ and maximum $k_{max} = N - 1$.

$$p(k|\mu) \propto \frac{\mu^k}{k!} \{k < k_{max}\} \quad (3)$$

Here μ can be interpreted as the expected number of changepoints. Following [5], 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 $\xi = \{\xi_1, \dots, \xi_k\}$ takes only nonoverlapping and uniformly distributed integer values.

$$p(\xi|k) = 1 / \binom{N-1}{k} \quad (4)$$

It is a reasonable assumption that the data in the same segment share the same network structure. Hence, we need to learn the network structure of each segment instead of each data. However, only using the data from one segment may lead to the scarcity of the training samples. In this paper, we assume that the network structures of temporally adjacent segments are more similar than those of segments temporally far away [6]. 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.

When we estimate the network structure of the j th segment, the weights of other segments are defined as follows:

$$w_i = \frac{K_h(tc_i - tc_j)}{\sum_{i=1}^{k+1} K_h(tc_i - tc_j)} \quad (5)$$

where $K_h(\cdot)$ is a symmetric nonnegative kernel function and h is the kernel bandwidth. tc_i and tc_j are the temporal center points of the i th segment and the j 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 i and the target segment j , the larger the weight of segment i . In this paper, we use a Gaussian RBF kernel:

$$K_h(t) = \exp(-\frac{t^2}{h}) \quad (6)$$

Different time points at the same segment are equally treated by assigning them the same weight in the proposed method. Hence, the likelihood to estimate the network structure of one segment is described as follows:

$$p(y_i|\beta, X, L) = \prod_{j=1}^{k+1} \prod_{t=\xi_{j-1}+1}^{\xi_j} N(y_{i,t} | X_t \beta_{j,:}, \sigma_0^2)^{w_j} \quad (7)$$

where the variable $X_{i,t}$ is determined by the lag $L_{s,i}$, which denotes the lag of X_i in the s th segment. Given $L(s, :)$ for the s th segment, $X_t = [x_{1,t-L_1}, \dots, x_{p-1,t-L_{p-1}}]$. $\beta_{j,:}$ is the coefficient for this segment, and 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 values of the $L_{:,i}$ for each segment are both within the interval $[1, T]$. Here we assume that the hyperparameters $L_{:,i}$ are sampled from uniform distributions:

$$L_i \sim \text{uniform}(1, T) \quad (8)$$

In order to control the sparsity of regression coefficients β . We use the spike and slab prior [16] [9] which is the prior that puts a positive probability mass on values equal to zero for the model

coefficients of each variable. We introduce a matrix \mathbf{Z} where $\mathbf{Z}(i, :)$ is a $p-1$ binary latent vector $Z_{i,1}, \dots, Z_{i,p-1}$. Each $Z_{i,j}$ indicates whether $\beta_{i,j}$ is zero ($Z_{i,j}=0$) or different from zero ($Z_{i,j}=1$). When \mathbf{Z} is known, the prior for β is defined as:

$$p(\beta|\mathbf{Z}) = \prod_{s=1}^{k+1} \prod_{i=1}^{p-1} [Z_{s,i} N(\beta_{s,i} | 0, \nu_0) + (1 - Z_{s,i}) \delta(\beta_{s,i})] \quad (9)$$

where $\delta(\cdot)$ is a Dirac delta function, also known as a point probability mass centered at 0, $N(\cdot | 0, \nu_0)$ 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 \mathbf{Z} is sampled from a multivariate Bernoulli distribution:

$$p(\mathbf{Z}) = \text{Bern}(\mathbf{Z} | \mathbf{p}_0) \quad (10)$$

where $\mathbf{p}_0 = [p_{0,i,1}, \dots, p_{0,k+1,p-1}]$. $p_{0,s,i}$ is the prior probability that the coefficients of X_i in the s th segment is different from zero.

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

$$p(\Theta | \mathbf{X}, \mathbf{y}, \lambda) \propto p(k|\lambda) p(\xi|k) p(\beta|\mathbf{z}) p(\mathbf{z}) p(\mathbf{L}) p(\mathbf{X}|\mathbf{L}) p(\mathbf{y}|\mathbf{X}, \beta, \sigma, \mathbf{L}) \quad (11)$$

5. PARAMETER ESTIMATION

Since the overall parameter space of the proposed time-varying network structures 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 [11] and the Expectation Propagation (EP), where the changepoints are selected by RJMCMC procedure and the intractable posterior probabilities can be approximated by easy ones through EP.

Algorithm 1 Function Update_state(Current state, Equation e)

Input: Current state, Equation e

Output: New state

- 1: Evaluate Acceptance probability A according to Equation e
 - 2: Sample $u \sim U_{[0,1]}$
 - 3: **if** $u \leq A$ **then**
 - 4: Return the updated state
 - 5: **else**
 - 6: Return the current state
 - 7: **end if**
-

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 seg-

ment. The integration over β yields,

$$\begin{aligned} p(k, \xi, \mathbf{Z}_s, \mathbf{X}, \mathbf{y}, \mathbf{L}) \\ \propto \frac{\lambda^k (N-1-k)!}{(N-1)!} \prod_{j=1}^{p-1} \text{Bern}(z_{s,j} | p_{0,s,j}) (1/(T - L_{s,i})) \\ [Z_{s,j} N(0 | m_{s,j}, v_0 + v_{s,j}) + (1 - Z_{s,j}) N(0 | m_{s,j}, v_{s,j})] \end{aligned} \quad (12)$$

where

$$\begin{aligned} \mathbf{v}_s^{-1} &= \frac{1}{\sigma_0^2} \widetilde{\mathbf{X}}^s{}^T \widetilde{\mathbf{X}}^s & \mathbf{v}_s^{-1} \mathbf{m}_s &= \frac{1}{\sigma_0^2} \widetilde{\mathbf{X}}^s{}^T \widetilde{\mathbf{y}}^s \\ \widetilde{\mathbf{y}}_t^s &= \mathbf{y}_t * \sqrt{w_{g(t)}} & \widetilde{\mathbf{x}}_t^s &= \mathbf{x}_t * \sqrt{w_{g(t)}} \end{aligned} \quad (13)$$

and $g(t)$ denotes the segment index time t belonging to.

Algorithm 2 Procedure for Estimating Dynamic Dependency Network Structures with Estimation of Lags

Input: data \mathbf{X} and \mathbf{y} , the maximum number of changepoints k_{max}

Output: $k, \xi, \beta_s, \mathbf{L}, \mathbf{z}$

- 1: Initialization: k, ξ
 - 2: Iteration i:
 - 3: sample $\mu \sim \mu[0, 1]$
 - 4: **if** $u \leq b$ **then**
 - 5: **consider changepoint birth**
 Propose a new changepoint $\xi^* | \xi \sim u_{\{3, \dots, N\} \setminus \{\xi\}}$
 Update_state(Current state, Equation 14)
 - 6: **else if** $u \leq b + d$ **then**
 - 7: **consider changepoint death**
 Choose a changepoint $\xi^* \in \xi$ to be deleted
 Update_state(Current state, Equation 15)
 - 8: **else if** $u \leq b + d + s$ **then**
 - 9: **consider changepoint shift**
 Randomly choose a changepoint $\xi \in \xi$ shift to a new changepoint $\xi^* | \xi \sim u_{\{3, \dots, N\} \setminus \{\xi\}}$
 Update_state(Current state, Equation 16)
 - 10: **else**
 - 11: Update the network structure of every segment.
 - 12: **end if**
 - 13: $i \leftarrow i + 1$ and go to 4
-

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

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

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

We use P_{joint} to represent the joint distribution of parameters shown in the Equation 12 and P_{joint}^* denotes the updated joint distribution. For different moves, the probability r is given as follows respectively.

$$r_{birth} = \frac{P_{joint}^*(N - k - 1)(p - 1)!}{P_{joint} \lambda c! (p - c)!} \quad (14)$$

$$r_{death} = \frac{P_{joint}^* \lambda (p - 1)!}{P_{joint} (N - k) c! (p - c)!} \quad (15)$$

$$r_{shift} = \frac{P_{joint}^*}{P_{joint}} \quad (16)$$

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 2.

Update the network structure of every segment. Computing the β and \mathbf{Z} of each segment requires sampling them from posterior probability. Due to the intractability of posterior probability of

β and \mathbf{Z} as shown in the Equation 17 and Equation 18, we apply the expectation propagation(EP) [20] [14] to perform the inference approximately.

$$p(\beta | \mathbf{Z}, \mathbf{L}, \xi, k) \propto N(\mathbf{y} | \mathbf{X}\beta, \sigma^2) [\mathbf{Z}N(\beta | \mathbf{0}, \mathbf{v}_0) + (1 - \mathbf{Z})\delta(\beta)] \quad (17)$$

$$p(\beta | \mathbf{Z}, \mathbf{L}, k, \xi) \propto \text{Bern}(\mathbf{Z} | \mathbf{p}_0) [\mathbf{Z}N(\beta | \mathbf{0}, \mathbf{v}_0) + (1 - \mathbf{Z})\delta(\beta)] \quad (18)$$

The essence of the expectation propagation 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 $p(\beta, \mathbf{Z})$ and its approximation Q is minimized. Here we approximate the joint distribution of β_s and \mathbf{Z}_s using Q as follows:

$$Q(\beta_{s,:}, \mathbf{Z}_{s,:}) = N(\beta_{s,:} | \mathbf{m}_s, \mathbf{v}_s) \prod_{i=1}^{p-1} \text{Bern}(Z_{s,i} | \sigma(p_{0,s,i})) \quad (19)$$

The posterior probability of β and \mathbf{Z} can be obtained approximately during the process of EP algorithm. See [14] for details and an explicit expression.

As a convergence diagnostic we monitor the potential scale reduction factor (PSRF)[8], computed from the within-chain and between-chain variances of marginal network structure posterior probabilities. Values of $\text{PSRF} \leq 1.1$ are usually taken as indication of sufficient convergence.

6. EXPERIMENT

Experiment Setup. To evaluate the effectiveness of our model, we conduct thorough experiments on synthetic datasets and real world datasets. Baselines used for comparison include static lasso granger(SLG) model[1], Time-Varying Dynamic Bayesian Networks(TVDBN)[21], NHDBN (Non-homogeneous DBNs) [6], and the TLHL model [25]. The dynamic Bayesian model with adaptive lags we proposed is called DBAL for short. 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.

Table 2: Comparative Methods

Method	network state	varying form	lag
SLG	static	/	T
TVDBN	dynamic	time point	1
DBN	static	/	1
NHDBNs	dynamic	period	1
DBNL	dynamic	period	1
TLHL	static	/	learned

In our model, the range T of time lags is 10, the shape parameter $\alpha = 1$ and the scale parameter $\beta = 0.5$. 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 works [21][6].

6.1 Synthetic data

Data generation. We generate 10 networks with 7 nodes each ($p = 6$), 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 4 changepoints (except the beginning and end) and the localization vector is $\xi = [40, 200, 300, 400, 430]$ in the first data set.

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

Last, we generated time series of length l which is the length of segment h using a linear regression model: $\mathbf{X}(t) = \beta^h \mathbf{X}(t - \mathbf{L}) + \varepsilon$ where $\mathbf{X}(t) = [X_1(t), \dots, X_p(t)]'$ is the 6×1 vector of observations at time t , and $\mathbf{X}(t - \mathbf{L}) = [X_1(t - L_{1,1}), \dots, X_p(t - L_{i,p})]'$ where i is consistent with the subscript of $X_i(t)$. 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 β^* of network structure with the ground truth network structure parameter β . As the Fig.3 shows, Fig. 3(a) contains four ground truth coefficient matrixes for the four segments and Fig. 3(b) contains four learned coefficient matrixes. Each column corresponds to one segment of the time series data. The true coefficient β 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.

Fig.4 compares the ground truth time lags with the time lags learned by our model. Fig. 4(a) contains four true time lag matrixes corresponding to each segment, while Fig. 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.

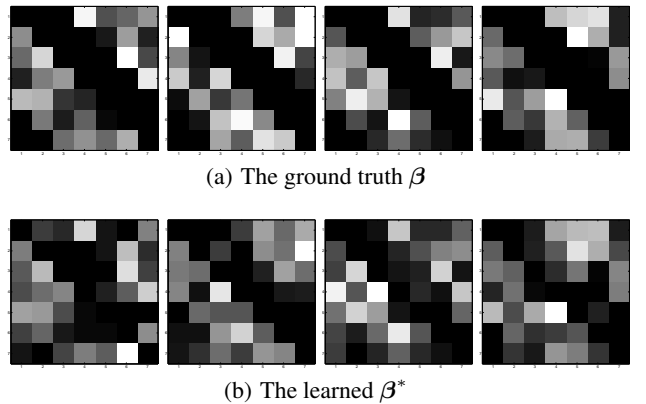


Figure 3: Comparison between the true β and learned β^* .

We use the F1-score to evaluate the performance in terms of network structure learning, where the F1-score is computed as follows:

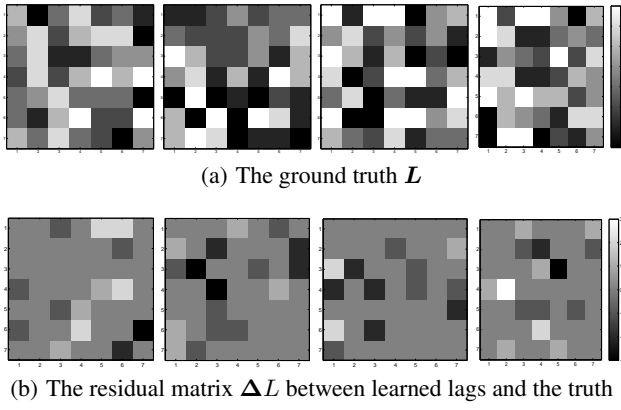


Figure 4: The true lags L and the residual matrix ΔL between learned lags and the truth.

$$Pre = \frac{|\{i, j | \beta_{i,j} \neq 0, \beta_{i,j}^* \neq 0\}|}{|\{i, j | \beta_{i,j} \neq 0\}|},$$

$$Rec = \frac{|\{i, j | \beta_{i,j} \neq 0, \beta_{i,j}^* \neq 0\}|}{|\{i, j | \beta_{i,j}^* \neq 0\}|},$$

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

where β^* denotes the regression coefficient matrix we learned, and β is the true coefficient matrix we set.

From the Fig.5, 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 the other methods. As for the static model, such as SGL and TLHL, they can not 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, Fig.6 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. The spatial distribution of stations is shown in Fig.9. Each air quality monitoring station measured the hourly air quality data such as PM2.5 and meteorological data e.g. wind, during the period 2014.12.01-2015.3.31.

Fig.7 presents the PM2.5 level of No.3 station in a blue curve and the changepoint using a red dashed line. The changepoint learned by DBAL model partitions the PM2.5 time series data into two segments.

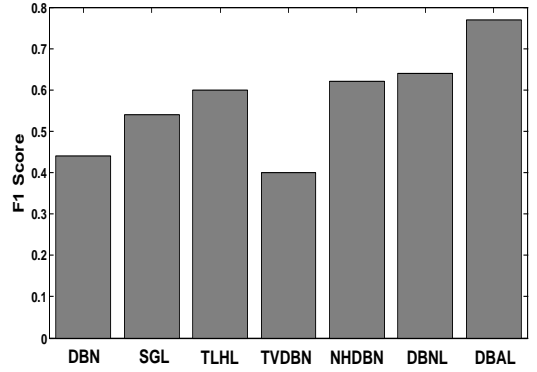


Figure 5: The F1 score of network structures learned by different methods.

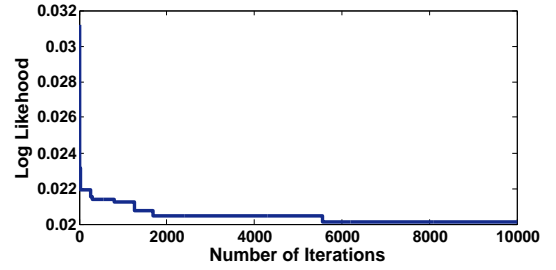


Figure 6: Log-likelihood of Model; x-axis: number of training iterations; y-axis: log-likelihood of training data

In the left segment, the air condition is seriously polluted by PM2.5. While the right segment shows the normal air quality condition. Fig.8 shows the corresponding wind speed level in a blue curve and the changepoint using a red 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.

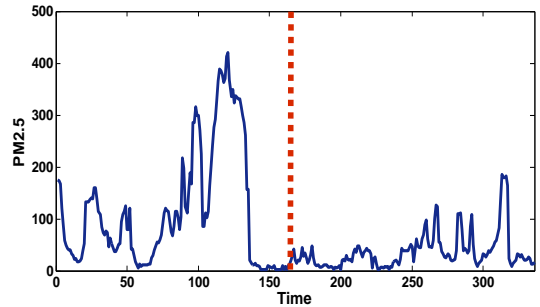


Figure 7: The PM2.5 and learned changepoint.

Fig.9(a) and Fig.9(b) present the dependent stations of No.3 station with time lags for the two identified segments respectively. The red triangle in the center denotes the target air quality monitoring station, i.e., the No.3 station. Each arrow originates from the dependent stations and points to the target station with a time lag number. From Fig.9(a), we can see the dependent stations are in

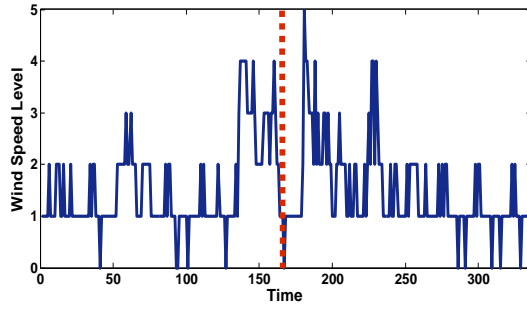
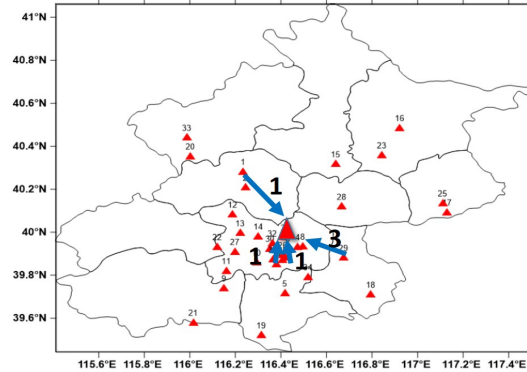
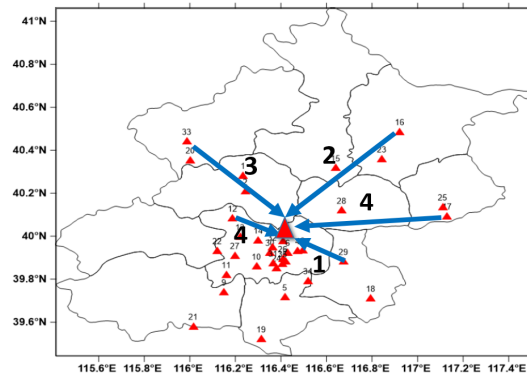


Figure 8: The wind speed level of No.3 station.

the near neighborhood with a small learned time lag in the seriously polluted segment. While in the Fig.9(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.



(a) The learned dependency of No.3 station in the first segment.



(b) The learned dependency of No.3 station in the second segment.

Figure 9: The dependent stations and time lags of No.3 station.

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 minutes interval. In order to study the dynamic dependency relationships of traffic stations in the day, we use data in one day with 96 samples. There are totally 236 traffic stations, which correspond to 236 ramps. In our experiment, we remove some stations where average traffic flows are less than 100.

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 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.

Fig.10 presents the traffic flow in a blue curve and the change-point using a red 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 flow. The location of changepoint is 28, which corresponds to the 7:00am in the day. From the Fig.10, 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.

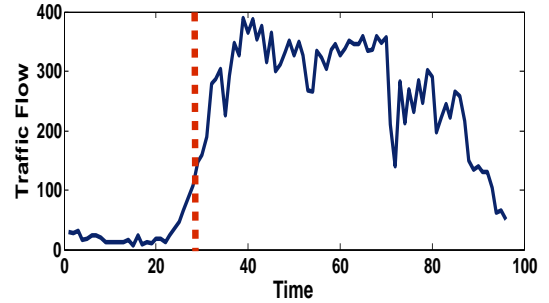
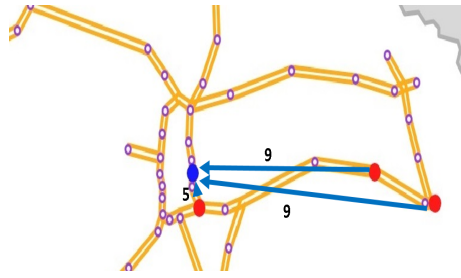


Figure 10: The traffic flow and learned changepoint.

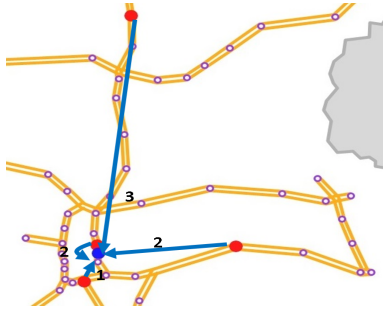
Fig.11(a) and Fig.11(b) present the learned dependency between traffic stations in the two segments respectively. As for the first segment, the target station which is marked with the blue dot mainly depends on the stations marked by red dots from the east with large time lags. In the second segment, 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 controls. 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 pro-



(a) The learned dependency between traffic ramps in the first segment.



(b) The learned dependency between traffic ramps in the second segment.

Figure 11: The learned dependency in highway traffic network.

pose a dynamic Bayesian model which simultaneously integrates two usually separate 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 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. Moreover, we will extend this model with considering context information which generate large impacts on the network structures.

8. ACKNOWLEDGEMENT

This work was supported by the National Science and Technology Support Plan(2014B AG01B02), the National Natural Science Foundation of China (61572041, 61402020), the National High Technology Research, Development Program of China (2014AA01 5103), and Beijing Natural Science Foundation (4152023) and the Ph.D. Programs Foundation of Ministry of Education of China (No. 20130 001120021).

9. REFERENCES

[1] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th*

ACM SIGKDD international conference on Knowledge discovery and data mining, pages 66–75. ACM, 2007.

[2] M. Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011.

[3] X. Chen, Y. Liu, H. Liu, and J. G. Carbonell. Learning spatial-temporal varying graphs with applications to climate data analysis. In *AAAI*, 2010.

[4] A. Dhurandhar. Learning maximum lag for grouped graphical granger models. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 217–224. IEEE, 2010.

[5] N. Dobigeon, J.-Y. Tournet, and M. Davy. Joint segmentation of piecewise constant autoregressive processes by using a hierarchical model and a bayesian sampling approach. *Signal Processing, IEEE Transactions on*, 55(4):1251–1263, 2007.

[6] F. Dondelinger, S. Lèbre, and D. Husmeier. Non-homogeneous dynamic bayesian networks with bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, 90(2):191–230, 2013.

[7] D. Eaton and K. Murphy. Bayesian structure learning using dynamic programming and mcmc. *arXiv preprint arXiv:1206.5247*, 2012.

[8] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

[9] E. I. George and R. E. McCulloch. Approaches for bayesian variable selection. *Statistica sinica*, pages 339–373, 1997.

[10] A. Goldenberg and A. W. Moore. Bayes net graphs to understand co-authorship networks? In *Proceedings of the 3rd international workshop on Link discovery*, pages 1–8. ACM, 2005.

[11] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[12] M. Grzegorzczak. A non-homogeneous dynamic bayesian network with a hidden markov model dependency structure among the temporal data points. *Machine Learning*, pages 1–53, 2015.

[13] F. Guo, S. Hanneke, W. Fu, and E. P. Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th international conference on Machine learning*, pages 321–328. ACM, 2007.

[14] J. M. Hernández-Lobato, D. Hernández-Lobato, and A. Suárez. Expectation propagation in linear regression models with spike-and-slab priors. *Machine Learning*, 99(3):437–487, 2015.

[15] D. Husmeier, F. Dondelinger, and S. Lebre. Inter-time segment information sharing for non-homogeneous dynamic bayesian networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2010.

[16] H. Ishwaran and J. S. Rao. Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773, 2005.

[17] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(4):320–327, 1976.

[18] S. Lebre, J. Becq, F. Devaux, M. P. Stumpf, and G. Lelandais. Statistical inference of the time-varying

structure of gene-regulation networks. *BMC systems biology*, 4(1):130, 2010.

- [19] Y. Liu, J. R. Kalagnanam, and O. Johnsen. 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*, pages 1225–1234. ACM, 2009.
- [20] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.
- [21] L. Song, M. Kolar, and E. P. Xing. Time-varying dynamic bayesian networks. In *Advances in Neural Information Processing Systems*, pages 1732–1740, 2009.
- [22] M. Talih and N. Hengartner. Structural learning with time-varying components: tracking the cross-section of financial time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):321–341, 2005.
- [23] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055–1062. ACM, 2007.
- [24] R. Yoshida, S. Imoto, and T. Higuchi. Estimating time-dependent gene networks from time series microarray data by dynamic linear models with markov switching. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE*, pages 289–298. IEEE, 2005.
- [25] X. Zhou, H. Hong, X. Xing, W. Huang, K. Bian, and K. Xie. Mining dependencies considering time lag in spatio-temporal traffic data. In *Web-Age Information Management*, pages 285–296. Springer, 2015.