Learning the Structure of Dynamic Bayesian Network with Hybrid Data and Domain Knowledges

Haiyang Jia[1,2], Juan Chen[1,2(✉)], and Zhiming Song[3(✉)]

[1] College of Computer Science and Technology, Jilin University, 2699 Ave. Qianjin, Changchun 130012, Jilin, People's Republic of China
{jiahy, chenjuan}@jlu.edu.cn
[2] Key Laboratory for Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, 2699 Ave. Qianjin, Changchun 130012, Jilin, People's Republic of China
[3] The Sports Medicine Department, The First Hospital of Jilin University, 2699 Ave. Qianjin, Changchun 130012, Jilin, People's Republic of China
szm3210@163.com

Abstract. Dynamic Bayesian Networks (DBNs) is a powerful graphical model for representing temporal stochastic processes. Learning the structure of DBNs is the fundamental step for parameter learning, inference, application etc. In some cases, such as computational systems biology, learning the structure of DBNs facing the two challenges (1) experimental settings only capture few time series and steady state measurements. (2) the knowledge about DBNs is uncertainty, rare and even with conflict. The paper considers the time series data, steady state and domain knowledge simultaneously, presents a novel algorithm for learning the structure of DBNs. Compare with single source learning, empirical experiment shows that learning with hybrid data and domain knowledges improved the accuracy and effectiveness of the DBNs structure learning.

Keywords: Machine learning · Dynamic system · Bayesian network
Domain knowledge

1 Introduction

Dynamic Bayesian Networks (DBNs), also known as dynamic probabilistic network or temporal Bayesian network, which generalize hidden Markov models and Kalman filters. The DBNs are widely used in many domains such as speech recognition, gene regulatory network (GRN) etc. Learning the structure of DBNs is a fundamental step for parameter learning, inference and application, but learning DBNs is a NP hard problem [1, 2]. In big data scenario, the structure learning is intractable. Despite of the computational efficacy barrier, the training set is also required to be large enough. In some domains, the training set is very noisy and rare, so learning with just one kind of training data is impractical. Domain knowledge may reduce the inherent uncertainty of

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 234–241, 2018.
https://doi.org/10.1007/978-3-319-73317-3_28

the DBNs learning. But the domain knowledge is always uncertainty, unclear and even with conflict. So, combining domain knowledge with training set is a key issue.

This paper presents an algorithm for learning the structure of DBNs with the hybrid data and domain knowledge. The paper is organized as following: Sect. 2 introduces related work and research background; Sect. 3 describes the DBNs learning algorithm; then, Sect. 4 describes the empirical experiment and last section draw the conclusion.

2 Research Background

2.1 (Dynamic) Bayesian Networks

A Bayesian networks (BNs) is a concise representation of joint probability distribution on a set of random variables [3]. A BNs is defined by a structure G and a family of parameters θ, for short $BNs = <G, \theta>$. G is a directed acyclic graph (DAG), each node is a random variable in $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, and G encodes the (condition) independencies, θ is the conditional probability distribution (CPD), encoding the conditional distributions of each node and its parent node

$$\theta = \{p(X_i \mid \pi(X_i)) \mid 1 \le i \le n\}, \ \pi(X_i) \text{ is the parent nodes of } X_i \tag{1}$$

Briefly, the joint probability distribution represented by BNs is:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \pi(X_i)) \tag{2}$$

DBNs extend the BNs by modeling the stochastic variables over time [4–6]. Let $\mathbf{X}^t = (X_1^t, \ldots, X_n^t)\ t \in [1, T]$ stand for the random variables $\mathbf{X}$ at time t. Two tiers DBNs, obey first-order Markov rules, which means $P(\mathbf{X}^t|\mathbf{X}^{t-1}, \ldots, \mathbf{X}^0) = P(\mathbf{X}^t|\mathbf{X}^{t-1})$ for all $t > 0$.

DBNs was composed of two slices: initial network BN^0 and transition network $BN^{\rightarrow}$. BN^0 encode the probability distribution of $P(\mathbf{X}^0)$, which is the initial state of the temporal process. For each time slice, $BN^{\rightarrow}$ define the probability of states translate form t−1 to t, $P(\mathbf{X}^t|\mathbf{X}^{t-1})$. With these assumptions, the joint probability distribution of a time series can be written as

$$\begin{aligned} P(\boldsymbol{X}^0, \boldsymbol{X}^1, \ldots, \boldsymbol{X}^n) &= P(\boldsymbol{X}^0) \prod_{t=1}^{T} P(\boldsymbol{X}^t|\boldsymbol{X}^{t-1}) \\ &= \prod_{i=1}^{n} P(X_i^0|\pi^0(X_i^0)) \prod_{t=1}^{T} P(\boldsymbol{X}^t|\boldsymbol{X}^{t-1}) \\ &= \prod_{i=1}^{n} P(X_i^0|\pi^0(X_i^0)) \prod_{t=1}^{T} P(\boldsymbol{X}^t|\pi^{\rightarrow}(\boldsymbol{X}^t) \\ &= \prod_{i=1}^{n} P(X_i^0|\pi^0(X_i^0)) \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_i^t|\pi^{\rightarrow}(X_i^t)) \end{aligned} \tag{3}$$

Figure 1 gives an example of DBNs. The distribution for this DBNs is

$$P(\boldsymbol{X}^0, \boldsymbol{X}^1, \ldots, \boldsymbol{X}^n) = P(X_1^0)P(X_2^0 \mid X_1^0)P(X_3^0 \mid X_4^0)P(X_4^0 \mid X_1^0)$$
$$\prod_{t=1}^{T} P(X_1^t \mid X_2^{t-1})P(X_2^t \mid X_1^{t-1}, X_4^{t-1})P(X_3^t \mid X_4^{t-1})P(X_4^t \mid X_3^{t-1}) \tag{4}$$

Fig. 1. Example of DBNs

2.2 Literature Review

Structure learning the BNs/DBNs can be considered as the general problem of selecting a probabilistic model that explains a given set of training data. a wealth of literature has been presented that seeks to understand and provide methods of learning structure from data.

Classical approaches for learning the structure can be classified to three main methods: [7] (1) A score-searching approach; (2) A constraint-based approach; (3) A dynamic programming approach;

Score-searching based approach, which define the task as an optimization problem. Based on a scoring function to evaluates different structures G related to a data set D (in the rest D was omitted for the concision). There are many score criteria such as : BD/BDe [8, 9], MDL [10] and BIC [11];

Constraint based approach define the learning task as constraint satisfaction problem. Using conditional independent test to find the independent relationships in data D, then construct a DBNs satisfied such conditional independence [12]. Each approach has its specialty: the constraint based methods are usually more efficient when the number of variables is large. However, when the data is noisy, the score-searching algorithms is more robust.

Aside from the two major techniques of structure learning that have been discussed, there is a third method that is like the score-and-search approach, but does not have the search aspect. These methods use dynamic programming to compute optimal models for a small set of variables and in some cases combine these models.

DBNs, as temporal models, are best learned from temporal data. But in some cases, such as bioinformatics and computational systems biology studies, experimental settings do not always permit collecting massive time series measurements and may only capture few time series and steady state measurements (Steady state measurements can be considered as snapshots of the long-run behavior of a system.), another challenge is that the domain knowledge is uncertain and sparse.

3 Learning Method

3.1 Formalize the Problem

Learning DBNs from steady state, temporal data and domain knowledge can be formalized as maximize the joint distribution:

$$\mathrm{P}(\mathrm{DBNs},\ \mathrm{Evidence}) \tag{5}$$

where Evidence = {Data, Prior knowledge}, Data = {DT,DS}, DT is temporal data, DS is steady state, DBNs $= \{\mathrm{G}, \theta\}$ here we only focus on transition network.

3.2 Steady State and Temporal Data

Equation (2) characterizes temporal behavior of DBNs over a given time interval. With the following Eqs. (5) and (6), the DBNs structure learning with Steady state and temporal data was formalized.

$$\begin{aligned} P(DBN|D_T, D_S) &= \frac{P(D_T, D_S|DBN)P(DBN)}{P(D_T, D_S)} \\ &\propto P(D_T, D_S|DBN)P(DBN) \end{aligned} \tag{6}$$

$$\begin{aligned} P(D|G) &= \int_\theta P(D|G,\ \theta)P\left(\theta|G\right)d\theta \\ &= \int_\theta P(D_T, D_S|G,\ \theta)P\left(\theta|G\right)d\theta \end{aligned} \tag{7}$$

Firstly, define some notations: all states for DBNs: $\mathrm{S} = \{\mathrm{S_q}|\mathrm{q} \in [1, \mathrm{N}]\}$; Size of S: N; State for $\mathrm{X_i}$: $\mathrm{S}(\mathrm{X_i}) = \{\mathrm{S_k}(\mathrm{X_i})|\mathrm{k} \in [1, \mathrm{N_i}]\}$; State for parent nodes of $\mathrm{X_i}$: $\mathrm{S}(\pi_\mathrm{i}) = \{\mathrm{S_j}(\pi_\mathrm{i})|\ \mathrm{j} \in [1, \mathrm{N}\pi_\mathrm{i}]\}$; $\boldsymbol{\theta}_{i,j,k} = \mathrm{P}(\mathrm{X}_i^t = \mathrm{S_k}(\mathrm{X_i})|\mathrm{X}_i^{t-1} = \mathrm{S_j}(\pi_i))$. For example: in Fig. 2. Assume all nodes are binary (0,1), then $S = \{(0000), (0001), \ldots\}$; $\mathrm{N} = 2^4 = 16$; $\mathrm{S}(\mathrm{X}_2) = \{0,1\}$, $\mathrm{N}_2 = 2$; $\pi_2 = \{\mathrm{X}_1, \mathrm{X}_3, \mathrm{X}_4\}$; $S(\pi_2) = \{(000), (001), (010), (011), \ldots\}$, $N_{\pi_2} = 8$; $\boldsymbol{\theta}_{2,5,1} = P(X_2^t = 1|X_1^{t-1} = 1, X_3^{t-1} = 0, X_4^{t-1} = 0)$.

Let M denote the state transition matrix, each element in M can be calculated with Eq. (8).

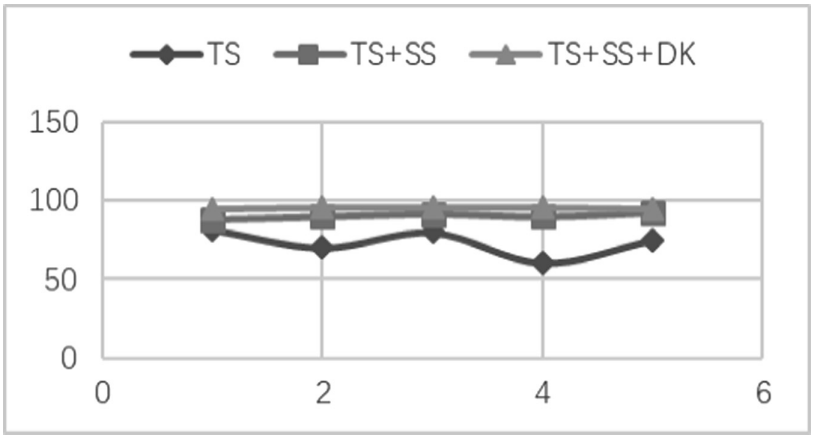

Fig. 2. Accuracy of the experiment. TS-time series; SS-steady state; DK-domain knowledge

$$M_{v,q} = P(\mathbf{X}^t = S_{\mathrm{q}}|\mathbf{X}^{t-1} = S_v) = \prod_{i=1}^{n} P(X_i^t = x_{\mathrm{i,q}}|\pi_i^{t-1} = S_{\pi_i,v}) = \prod_{i=1}^{n} \theta_{i,v_j,q_k} \tag{8}$$

The steady state S^* has the property S * M = S*, each element in S^* can be calculated with Eq. (8).

$$\lim_{t\to\infty} M_{v,q}^{(t)} = S_q^*, \text{Where } M_{v,q}^{(r)} = P(X^{t+r} = S_q|X^t = S_v) \tag{9}$$

Theorem: A finite state homogeneous Markov process corresponding to a DBNs, possess a unique stationary distribution, independent of the initial distribution if $\boldsymbol{\theta}_{i,j,k} > 0$, $\forall \mathrm{i} \in [1, \mathrm{n}]$, $\mathrm{j} \in [1, \mathrm{N}\pi_{\mathrm{i}}], \mathrm{k} \in [1, \mathrm{N_i}]$ [13].

The likelihood of a DBNs structure G give both temporal data and steady state is:

$$P(D|G) = \int_{\theta} P(D_T, D_S|G, \theta) P(\theta|G) d\theta \tag{10}$$

where $P(\theta|G) = \prod_{i=1}^{n} P(\theta_i|\pi(X_i)) = \prod_{i=1}^{n} \prod_{j=1}^{N_{\pi i}} P(\theta_{i,j}|\pi_i)$.

The prior distribution is assumed to be Dirichlet distribution (conjugate prior for multinomial), $\boldsymbol{\alpha}$: the prior for $\boldsymbol{\theta}$

$$P(\theta_{i,j}|\alpha,\pi_i) = \frac{1}{\mathrm{B}(\alpha)}\prod_{k=1}^{N_i}\theta_{i,j,k}^{\alpha_{i,j,k}-1},\ \ \mathrm{B}(\alpha) = \frac{\prod_{k=1}^{N_i}\Gamma(\alpha_{i,j,k})}{\Gamma(\sum_{k=1}^{N_i}\alpha_{i,j,k})}$$
$$\Gamma(x) = \begin{cases} (x-1)! \text{ if x is a positive integer} \\ \int_0^\infty t^{x-1}e^{-t}dt \text{ else} \end{cases} \tag{11}$$

To maxis posteriori $\tilde{\theta}$ with temporal data $\mathrm{D_T}$ is straight forward, but for steady state $\mathrm{D_S}$ we cannot compute $\tilde{\theta}$ with steady state directly, optimize both G and with $\mathrm{D_S}$ is intractable. So, we need an approximation, replace the parameter from steady state with the parameter from temporal data $\tilde{\theta}_S \approx \tilde{\theta}_T$.

3.3 Domain Knowledge

To learning the structure of DBNs means learning both G^0 and $G^{\rightarrow}$. Based on the score-searching approach, we define a score function. Suppose the number of time series is M, the NO. l sample is D_l which has T_l different time point. the score function defined as following:

$$\begin{aligned} &\log P(D|G) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{q_i}\frac{\Gamma(\alpha_{i,j})}{\Gamma(\alpha_{i,j}+N_{i,j})}\sum_{k}^{r_i}\frac{\Gamma(\alpha_{i,j,k}\ +\ N_{i,j,k})}{\Gamma(\alpha_{i,j,k})} \end{aligned} \tag{12}$$

$N^{\,0}_{i,j,k} = \sum_{l=1}^{M}\chi(X_i^0 = k, \pi^0(X_i^0) = j|D_l)$, $N^{\,\rightarrow}_{i,j,k} = \sum_{l=1}^{M}\sum_{t=1}^{T_l}\chi(X_{ii}^t = k, \pi^{\rightarrow} = j\,|D_l)$,

$N_{i,j} = \sum_{k=1}^{r_i} N_{i,j,k}$, if x is a positive integer, $\Gamma(x) = (x-1)!$

$\alpha_1, \cdots, \alpha_r$ is the hyper parameter for Dirichlet distribution.

The domain knowledge is used to calculate the prior distribution. Domain knowledge about the structure was encoded with matrix K. The initial network K^0: if there should be an edge from $\mathrm{v_i}$ to $\mathrm{v_j}$ then $\mathrm{k_{ij}}$ is 1; if there should not exist an edge $\mathrm{k_{ij}}$ is 0, otherwise $\mathrm{k_{ij}}$ is −1 for unknown.

$$K^0 = \begin{cases} k_{i,j}^0 = 1,\ \textit{if}\ v_i^0 \rightarrow v_j^0 \\ k_{i,j}^0 = 0,\ \textit{if no edge betwen}\ v_i^0\ \textit{and}\ v_j^0 \quad i \in [1,n], j \in [1,n] \\ k_{i,j}^0 = -1,\ \textit{if unknow for}\ v_i^0\ \textit{and}\ v_j^0 \end{cases} \tag{13}$$

The confidence of the knowledge defined with matrix $C^0 = c_{i,j}^0 \in [0,1]$, the distance matrix D defined as

$$D^0 = Dist(K^0, G^0) = \begin{cases} d_{i,j}^0 = 0,\ \textit{if}\ k_{i,j}^0 = -1\ \textit{or}\ k_{i,j}^0 = g_{i,j}^0 \\ d_{i,j}^0 = 1,\ \textit{if}\ k_{i,j}^0 \neq g_{i,j}^0 \end{cases} \tag{14}$$

Each structure can be weighted as following:

$$W^0 = \frac{\sum_{i=1,j=1}^{n,n} c_{i,j}^0 d_{i,j}^0}{n^2 - \sum_{i=1,j=1}^{n,n} I(k_{i,j}^0 = -1)}, \quad where\, I(x) = \begin{cases} 1, & if\ x = true; \\ 0, & if\ x = false; \end{cases} \tag{15}$$

For multiple domain knowledge, named the number of knowledge sources Q, there are a set of knowledge matrix and correspond confidence matrix. The weight of the knowledge is defined as $L_1,\ldots,L_Q$, $L_i \in [0,1]$, $\sum_{i=1}^{Q} L_i = 1$, $i \in [1,Q]$, then the W for the structure G is a weighted average: $W^0 = L_1 * W_1^0 + \ldots + L_Q * W_Q^0$.

The score function is defined as below

$$\text{Score}(G) = BI(D,G) - \beta W \tag{16}$$

The β control the ratio that data and domain knowledge effect on learning procedure.

4 Experiment

To test the behavior of the algorithm, several artificial data was generated. DBNs with 10, 30, 50, 100, 200 nodes were generated randomly. Training data were computed with such generative model. Steady state was assumed the state do not change within 30 time steps, time series data and domain knowledge were selected from give model and data respectively, the ratio for training was kept below 20%. The Fig. 2 given the accuracy with different given DBNs learning. The average accuracy was increased from 73.4% to 90.6 with steady state data added and increased another 5% when given domain knowledge.

This paper presents a novel algorithm for learning the structure of DBNs, which consider both time series data, steady state and domain knowledge simultaneously, empirical experiment shows that the proposed algorithm improved the efficiency and the accuracy of the DBNs structure learning.

Acknowledgements. This work is supported by Science and Technology Development of Jilin Province of China (20150101051JC, 20160520099JH), Special Funds of Central Colleges Basic Scientific Research Operating Expenses, Jilin University under Grant 93K172017K04.

References

1. Chickering, D.M.: Learning Bayesian networks is NP-complete. Learn. Data: Artif. Intell. Stat. **112**, 121–130 (1996)
2. Chickering, D.M., Heckerman, D., Meek, C.: Large-sample learning of Bayesian networks is NP-hard. J. Mach. Learn. Res. **5**, 1287–1330 (2004)

3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
4. Ghahramani, Z.: An introduction to hidden Markov models and Bayesian networks. In: Hidden Markov Models, pp. 9–42. World Scientific Publishing Co., Inc. (2002)
5. Murphy, K.P.: Dynamic Bayesian networks: representation, inference and learning. Ph.D. thesis, University of California, Berkeley (2002)
6. Ghahramani, Z.: Learning dynamic Bayesian networks. In: Giles, C.L., Gori, M. (eds.) NN 1997. LNCS, vol. 1387, pp. 168–197. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0053999
7. Daly, R., Shen, Q., Aitken, S.: Learning Bayesian networks: approaches and issues. Knowl. Eng. Rev. **26**, 99–157 (2011)
8. Cooper, H.: A Bayesian method for the induction of probabilistic networks from data. Mach. Learn. **9**, 309–347 (1992)
9. David, H., Dan, G., David, M.C.: Learning Bayesian networks: the combination of knowledge and statistical data. Mach. Learn. **20**, 197–243 (1995)
10. Lam, W., Bacchus, F.: Learning Bayesian belief networks: an approach based on the MDL principle. Comput. Intell. **10**, 269–293 (1994)
11. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)
12. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.R.: Learning Bayesian networks from data: an information-theory based approach. Artif. Intell. **137**, 43–90 (2002)
13. Lahdesmaki, H., Shmulevich, I.: Learning the structure of dynamic Bayesian networks from time series and steady state measurements. Mach. Learn. **71**, 185–217 (2008)