PRES LUNAM
Ecole Doctorale STIM
Sciences et Technologies de l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : LINA
Equipe : COD

Benchmarking dynamic Bayesian network structure learning algorithms

Trabelsi, Ghada
Mél : ghada.trabelsi@etu.univ-nantes.fr

Résumé : Dynamic Bayesian Networks (DBNs) are probabilistic graphical models dedicated to model multivariate time series. Two-time slice BNs (2-TBNs) are the most current type of these models. Static BN structure learning is a well-studied domain. Many approaches have been proposed and the quality of these algorithms has been studied over a range of different standard networks and methods of evaluation. To the best of our knowledge, all studies about DBN structure learning use their own benchmarks and techniques for evaluation. The problem in the dynamic case is that we don't find previous works that provide details about used networks and indicators of comparison. In addition, access to the datasets and the source code is not always possible. In this paper, we propose a novel approach to generate standard DBNs based on tiling and novel technique of evaluation, adapted from the "static" Structural Hamming Distance proposed for Bayesian networks.

Mots clés : *Dynamic Bayesian Networks, 2-TBN models, Bayesian Network Tiling, Structural Hamming Distance.*

1 Introduction

During the last two decades, there has been an increasing interest in the Bayesian Network (BN) formalism [1]. The BN are succeed as one of the most complete and consistent formalisms for the acquisition and representation of knowledge and for reasoning from incomplete and/or uncertain data.

Learning the graphical part (i.e. the structure) of these models from data is an NP-hard problem. Many studies have been conducted on this subject [2]. The most of these works and their result interpretations use standard networks and common performance indicators, such as approximation of the marginal likelihood of the obtained model or the Structural Hamming Distance (SHD) proposed by Tsamardinos [3].

Dynamic Bayesian networks (DBNs) are a general and flexible model class for representing complex stochastic temporal processes [4]. Some structure learning algorithms have been proposed, adapting principles already used in "static" BNs. Comparing these algorithms is a difficult task because the evaluation technique and/or the reference networks used change over each article. Evaluation of these algorithms is also often restricted to networks with a small number of variables, at the difference of "static" BNs where structure learning has been studied with large reference networks.

We focus on this paper to present two contributions : (1) an algorithm for generating large 2-TBN networks by using tiling approach [5] in the dynamic case ; (2) an algorithm for the evaluation of a 2-TBN structure learning algorithm by adapting the SHD measure no more correct with temporal networks.

Section 2 provides the background of our work with a brief introduction to the evaluation methods used in BN learning. In section 3, we detail our proposed approach to generate large DBN and metric to evaluate the performance of DBN structure learning algorithms. Finally in section 4, we present conclusions and future works.

2 Background

2.1 Dynamic Bayesian networks and structure learning

A dynamic Bayesian Network (DBN) is a probabilistic graphical model devoted to represent sequential systems [4]. More precisely, a DBN defines the probability distribution of a collection of random variables $\mathbf{X}[t]$ where $\mathbf{X} = \{X_1 \dots X_n\}$ is the set of variables observed along discrete time t.

In this work, we consider a special class of DBNs, namely 2-time slice Bayesian Networks (2-TBN). A 2-TBN is a DBN which satisfies the Markov property of order 1 $\mathbf{X}[t-1] \perp \mathbf{X}[t+1] \mid \mathbf{X}[t]$. As a consequence, a 2-TBN is described by a pair $(M_0, M_{\rightarrow})$.

M_0 (**initial model**) is a BN representing the initial joint distribution of the process $P(\mathbf{X}[t=0])$ and consisting of a direct acyclic graph (DAG) G_0 containing the variables $\mathbf{X}[t=0]$ and a set of conditional distributions $P(X_i[t=0] \mid pa_{G_0}(X_i))$ where $pa_{G_0}(X_i)$ are the parents of variable $X_i[t=0]$ in G_0 ;

$M_{\rightarrow}$ (**transition model**) is another BN representing the distribution $P(\mathbf{X}[t+1] \mid \mathbf{X}[t])$ and consisting of a DAG $G_{\rightarrow}$ containing the variables in $\mathbf{X}[t] \cup \mathbf{X}[t+1]$ and a set of conditional distributions $P(X_i[t+1] \mid pa_{G_{\rightarrow}}(X_i))$ where $pa_{G_{\rightarrow}}(X_i)$ are the parents of variable $X_i[t+1]$ in $G_{\rightarrow}$, parents which can belong to time t or $t+1$.

2.2 Generation of large "static" Bayesian networks

Many studies have been conducted on this subject, leading to three different families of approaches : (1) constraint-based methods, this approach estimate from the data whether certain conditional independencies between the variables hold ; (2) score-based methods [6], in this category search the space of all possible structures for the one that maximizes the score using greedy, local, or some other search algorithm ; (3) local search methods [3]. The hybrid approach or local search method that is able to overcome the limitations of two previous approachs. [5] have proposed a novel algorithm and software for the generation of arbitrarily large BN (e.g., graphical models representation and joint probability distributions) by tiling smaller real-world known networks (tiles).

2.3 Evaluation of BN structure learning algorithms

When proposing a new structure learning algorithm, Tsamardinos et al. [3] have proposed an adaptation of the usual Hamming distance between graphs taking into account the fact that some graphs with different orientations can be statistically indistinguishable. As graphical models of independence, several equivalent graphs will represent the same set of dependence/independence properties. These equivalent graphs (also named Markov or likelihood equivalent graphs) can be summarized by a partially DAG (PDAG). This new structural Hamming distance (SHD) compares the structure of the PDAG of the learned and the original networks as described in algorithm **??** in order to only compare orientations that are really statistically distinguishable.

2.4 Evaluation of a model learned from data and prior knowledge

When proposing a first algorithm to determine the PDAG of a given graph, [7] also proposed a way to take into account prior background knowledge. This solution is decomposed into three phases. The first phase consists in determining the PDAG. This step can be resolved by keeping the skeleton of the given DAG, and its V-structures, and then applying recursively a set of three rules R_1, R_2, and R_3 in order to infer all the edge orientations compatible with the initial DAG. The second phase consists in comparing this PDAG with the prior knowledge. If some information are conflicting, the algorithm returns an error. The final step consists in iteratively adding the prior knowledge (edges) not present in the PDAG and applying again the previous recursive orientation rules in order to infer all the new edge orientations induced by the addition of the prior knowledge. Meek demonstrates that another rule R_4 is needed in order to complete the three previous ones when we add the prior knowledge for the model.

3 Contributions

3.1 Generation of large 2-TBNs

In order to generate a large 2-TBN model, we propose generating two models M_0 and $M_{\rightarrow}$ from an initial static benchmark BN M.

First we use the Tsamardinos's work for generating realistic large bayesian networks by tiling. Our method consists in generating a large initial model M_0 and its conditional probability distribution by tiling n copies of the initial model M. Then we use again tiling to generate $M_{\rightarrow}$ and the transition probability distribution by tiling 2 copies of M_0 (figure 1).

Complexity of the final 2-TBN can be controlled by changing the number of tiling copies and the intra-connectivity c_i (used for generating M_0) or the temporal connectivity c_t (used for generating temporal edges).

Algorithm 1 Structural Hamming distance for 2-TBNs algorithm

Require: Learned PDAG H_0 ; Learned PDAG $H_\rightarrow$; Original PDAG G_0 ; Original PDAG $G_\rightarrow$
Ensure: SHD values for initial and transition graphs

1: $H_k = H_\rightarrow$
2: $G_k = G_\rightarrow$
% calculate SHD_0
3: $SHD_0 = SHD(H_0, G_0)$
% Temporal correction for $G_\rightarrow$
4: Select randomly a temporal edge from $G_\rightarrow$
5: Orient this temporal undirected edge in G_k
6: Recursively apply the Meek rules in G_k
7: If there exist any unprocessed temporal edge then repeat 4, 5, 6.
% Temporal correction for $H_\rightarrow$
8: Select randomly a temporal edge from $H_\rightarrow$
9: Orient this temporal undirected edge in H_k
10: Recursively apply the Meek rules in H_k
11: If there exist any unprocessed temporal edge then repeat 8, 9, 10.
% Calculate $SHD_\rightarrow$
12: $SHD_\rightarrow = SHD(H_k, G_k)$
% calculate SHD in 2-TBN
13: $SHD = (SHD_0, SHD_\rightarrow)$

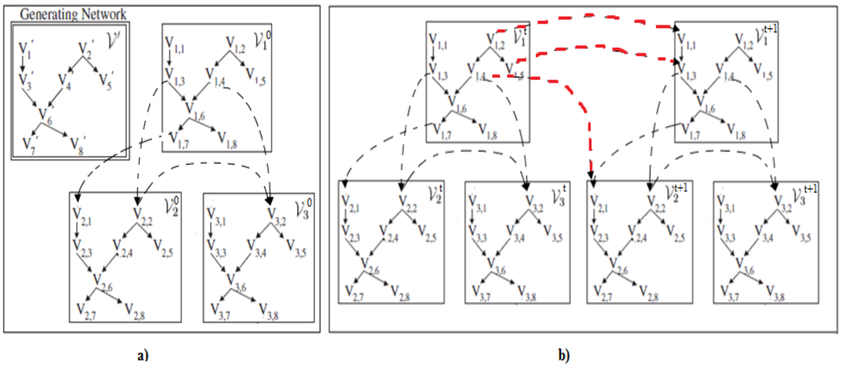

FIGURE 1 – An example output of the 2-TBN generation algorithm.

3.2 Evaluation of DBN generated by data and prior knowledge

As 2-TBNs are defined by two graphs G_0 and $G_\rightarrow$, we propose evaluating the structural difference between the theoretical 2-TBN and a learned one by the pair of the structural Hamming distance for the corresponding initial and transition graphs as described in algorithm 1.

Taking into account Markov equivalence by comparing PDAGs is important for BN structure learning evaluation, but it's not sufficient for DBNs. Some temporal information (a priori knowledge) is used for 2-TBN structure learning and can be lost by reasoning with PDAGs.

In the case of 2-TBN, two different models are learnt. The first one M_0 doesn't model temporal information, so the usual Tsamardinos' SHD can be used.

The second model named transition model $M_\rightarrow$ represents the dependency relations between nodes of the same slice t or between the nodes of slices t and $t+1$. In fact, we have here an important background (temporal) knowledge, edges between time slices are directed from t to $t+1$. We then propose to adapt the Tsamardinos' SHD in order to deal with this additional knowledge as proposed in section 2.4 for BNs. One temporal correction is applied for each PDAG in order to obtain a corrected $PDAG_k$ compatible with the prior knowledge. The structural Hamming distance is then computed between these $PDAG_k$s (figure 2).

4 Conclusion and perspectives

We focus in this paper on providing tools for benchmarking dynamic Bayesian network structure learning algorithms. Our first contribution is a 2-TBN generation algorithm inspired from the Tiling technique proposed by [5]. Our second contribution is a novel metric for evaluating performance of these structure learning algorithms, by correcting the Structural Hamming distance proposed by [3] in order

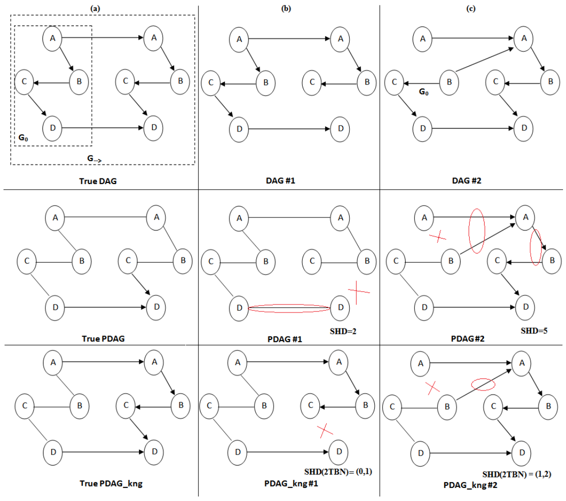

FIGURE 2 – Two examples of structural Hamming distance with or without temporal correction.

to take into account temporal background information. Our next step in this direction is proposing one website by providing some 2-TBNs benchmarks (graphs and datasets) in order to provide common evaluation tools for every researcher interested in 2-TBN structure learning.

Références

[1] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, 1988.

[2] D. Chickering. Learning bayesian networks is np-complete. In *InD. Fisher and H. Lenz (Eds.), Learning from data : Artificial intelligence and statistics V*, pages 121–130. Springer-Verlag, 1996.

[3] I. Tsamardinos, L. E. Brown, and C. F. Constantin, F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.*, 65(1) :31–78, 2006.

[4] K.P. Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning*. Phd, University of California, Berkeley, 2002.

[5] A. R. Statnikov, I. Tsamardinos, and C.F. Aliferis. An algorithm for generation of large bayesian networks. Technical Report Technical Report DSL-03-01, Department of Biomedical Informatics, Discovery Systems Laboratory, Vanderbilt University, 2003.

[6] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks : The combination of knowledge and statistical data. *Machine Learning*, 20 :197–243, 1995.

[7] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 403–410, San Francisco, CA, 1995. Morgan Kaufmann.