The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Learning Bayesian Network Structure by Self-Generating Prior Information: The Two-Step Clustering-Based Strategy

Yikun Zhang
School of Mathematics
Sun Yat-sen University
Guangzhou, China 510275
email: yikunzhang@foxmail.com

Yang Liu, Jiming Liu
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
email: {csygliu,jiming}@comp.hkbu.edu.hk

Abstract

Structure learning is a fundamental and challenging issue in dealing with Bayesian networks. In this paper we introduce a two-step clustering-based strategy, which can automatically generate prior information from data in order to further improve the accuracy and time efficiency of state-of-the-art algorithms in Bayesian network structure learning. Our clustering-based strategy is composed of two steps. In the first step, we divide the potential nodes into several groups via clustering analysis and apply Bayesian network structure learning to obtain some pre-existing arcs within each cluster. In the second step, with all the within-cluster arcs being well preserved, we learn the between-cluster structure of the given network. Experimental results on benchmark data sets show that a wide range of structure learning algorithms benefit from the proposed clustering-based strategy in terms of both accuracy and efficiency.

Introduction

Bayesian network models, first introduced into artificial intelligence by (Pearl 1982), have been applied to miscellaneous fields of science. For example, Bayesian network classifiers, carried out by (Friedman, Geiger, and Goldszmidt 1997), improved the performances of Naive Bayes classifiers under some circumstances by taking into account the correlations between variables. During applications of Bayesian networks under real-world scenarios, practitioners will inevitably face an annoying problem about how to "precisely" learn the structures of Bayesian networks from data. To address this problem, some constraint-based and score-based algorithms have been proposed in the last two decades ((Margaritis 2003)). (Chickering 1996) proved that this problem is NP-hard and the existing structure learning algorithms might produce some network structures that can hardly describe the original data sets. However, some prior information helps to ameliorate computational costs and accuracies of the existing algorithms. For instance, (Perrier, Imoto, and Miyano 2008) assumed the skeleton of a network and efficiently found an optimal Bayesian network by restricting the searching on it. Some algorithms require the prior knowledge in high quality, and they need users to specify a structure or an ordering of nodes, both of which are not easy to achieve ((Xu et al. 2015)). Moreover, in real-world applications, Bayesian network structures could be learnt without any source of expert knowledge. Hence an automatic mechanism for generating prior information should be of great merit. The previous work tends to incorporate prior knowledge to elicit informative prior probability distributions of the graph structures ((Cano, Masegosa, and Moral 2011)). However, the approach of existing work is highly constrained to the availability and correctness of prior information. This conspicuous drawback motivates us to develop a novel automatic way to generate prior information from data by the structure learning algorithm itself. The prior knowledge may come from the existence and absence of parents, and distribution knowledge including the conditional probability distribution (CPD) of edges and the probability distribution (PD) of nodes ((Xu et al. 2015)). Among various kinds of prior knowledge, nothing goes better than some pre-existing arcs, since they work as the foundation of a network structure. (Friedman, Nachman, and Peér 1999) applied clustering to figure out the candidate parents of a variable in a Bayesian network, which was considered as an innovative way to generate prior information from data itself via clustering. Moreover, (Kojima et al. 2010) divided the super-structure (a pre-assumed skeleton for the resulted network) into several clusters in order to extend the feasibility of their constrained optimal searching methods. Their ideas inspire us to resort to clustering analysis in order to automatically obtain some prior information about the existence of arcs. Our *two-step clustering-based* strategy utilizes clustering analysis in the first step. We first apply clustering to group those strongly "dependent" nodes (or variables) and learn the arcs within clusters via a conventional structure learning algorithms. In the next step, retaining all the arcs within clusters, we again implement the same traditional algorithm so as to learn the arcs between clusters. The advantages of our *two-step clustering-based* strategy fall into two aspects. First, it can tackle those real-world structure learning problems when expert knowledge is scarce or even nowhere to obtain. Second, a wide range of traditional structure learning algorithms can benefit from our strategy in terms of accuracy and time efficiency. The brief version of our method has been accepted by AAAI-18 Student Abstract and Poster Program ((Zhang, Liu, and Liu 2018)).

Background

In this section we present some related background knowledge. We begin with a description of general notations that would be used in following sections. Consider a finite set $\mathbf{U} = X_1, ..., X_N$ of discrete random variables where each variable X_i may take on values from a finite set. We use capital letters such as X, Y, Z for variable names and lowercase letters such as x_i, y_i, z_i to denote specific values taken by those variables. The notation $X \perp Y|\mathbf{S}$ indicates that X and Y are independent upon conditioning on (at least one value assignment of) the variables in the set $\mathbf{S}$.

A *Bayesian network* is an annotated directed acyclic graph that encodes a joint probability distribution over a set of random variables $\mathbf{U}$. Formally, a Bayesian network for $\mathbf{U}$ is a pair $B = \langle G, \Theta \rangle$. The first component, G, is a directed acyclic graph whose vertices correspond to the random variables $X_1, ..., X_N$, and whose edges represent direct dependencies between the variables. The second component of the pair, namely Θ, represents the set of parameters that quantifies the network. See Friedman et.al (1997) for details. Together, they sufficiently decompose the joint probability distribution over $\mathbf{U}$ into

$$Pr(X_1, X_2, ..., X_N) = \prod_{i=1}^{N} Pr(X_i|\mathbf{Pa}(X_i)),$$

where $\mathbf{Pa}(X_i)$ is the set containing the parents of X_i in the Bayesian network.

As a paramount step in learning a Bayesian network, learning its structure aims at identifying a network B that uncovers a set of conditional independence relations among the variables given the data set $\mathcal{D}$. Besides independencies, under some assumptions, the graph structure of a Bayesian network can also be used in certain domains to represent cause-effect relationships through the edges and their directions ((Margaritis 2003)). In this paper we focus primarily on two types of structure learning algorithms, namely, constraint-based and score-based methods. Later analyses and experimental results indicate that both these two types of methods benefit from our innovative strategy in terms of accuracy and time efficiency.

Constraint-based Structure Learning

The foundation of constraint-based structure learning algorithms is the profound work of (Verma and Pearl 1991), the *inductive causation (IC)* algorithm. They learn the network structure by analyzing the probabilistic relations entailed by the Markov property of Bayesian networks with conditional independence tests and then constructing a graph which satisfies the corresponding d-separation statements ((Scutari 2010)).

Classical constraint-based algorithms cannot be applied to any real-world problem due to the exponential number of possible conditional independence relationships ((Nagarajan, Scutari, and Lébre 2013)). As a result, (Margaritis 2003) proposed a novel approach, grow-shrink (GS) algorithm. The plain version of the GS algorithm utilized Markov blanket information for inducing the structure of a Bayesian network and employed dependence tests conditioned only on the minimal Markov blankets of the variables (or nodes) involved. The definition of a minimal Markov blanket is as follows,

Definition (Markov blanket): *For any variable $X \in \mathbf{U}$, the minimal Markov blanket $\mathbf{BL}(X) \subseteq \mathbf{U}$ is the minimal subset of variables such that for any $Y \in \mathbf{U} - \mathbf{BL}(X) - X$, $X \perp Y|\mathbf{BL}(X)$.*

In a Bayesian network, the (minimal) Markov blanket of a node X consists of all its parents, children, and all the other nodes sharing a child with X.

Besides the popular GS algorithm, some other constraint-based algorithms are worth being mentioned here. The *Incremental Association Markov Blanket* (IAMB) algorithm uses a two-phase selection scheme based on a forward selection followed by a backward one to detect the Markov blankets ((Tsamardinos et al. 2003)). The *Interleaved Incremental Association* (inter-IAMB) algorithm is a variant of IAMB that interleaves the grow phase with the shrink phase to reduce the size of Markov blankets in time ((Yaramakala and Margaritis 2005)). The *Max-Min Parents and Children* (MMPC) is the first local-learning algorithm for identifying the parents and children of any intended variables in a Bayesian network that faithfully represents the joint distribution of data ((Tsamardinos, Aliferis, and Statnikov 2003)).

The biggest disadvantage of constraint-based algorithms is their inherent sensitivity to the failures of conditional independence tests. Although some preceding algorithms like the GS algorithm may improve the robustness of their predecessors by incorporating random factors, their performances are still outweighed by some primitive algorithms with the assistance of prior information. Therefore, one can imagine that the robustness of statistical tests will be further improved if we also furnish them some prior information.

Score-based Structure Learning

Another category of structure learning algorithms treats the problem as an optimization problem, by assigning a statistically motivated score to each candidate Bayesian network ((Friedman, Nachman, and Peér 1999)). Given the data set $\mathcal{D}$, the score of a possible structure is

$$Score(\mathcal{G}, \mathcal{D}) = Pr(\mathcal{G}|\mathcal{D}) = \frac{Pr(\mathcal{D}|\mathcal{G})Pr(\mathcal{G})}{Pr(\mathcal{D})} \quad (1)$$

The most common score function is the *Bayesian Information Criterion* (BIC). Given the data set $\mathcal{D} = \mathbf{x}_1, ..., \mathbf{x}_N$, the BIC score function of a network B is

$$BIC(B|\mathcal{D}) = LL(B|\mathcal{D}) - \frac{1}{2}|B|log(N), \quad (2)$$

where $|B|$ is the number of parameters in the network. The first term represents the *log likelihood* of B given $\mathcal{D}$:

$$LL(B|\mathcal{D}) = \sum_{i=1}^{N} log(P_B(\mathbf{x}_i)) \quad (3)$$

The log likelihood has a statistical interpretation: the higher the log likelihood, the closer B is modeling the probability distribution in the data $\mathcal{D}$ ((Friedman, Geiger, and Goldszmidt 1997)).

Another popular score function is called *minimal description length* (MDL) score, which is simply the negate of the BIC score function.

With a preassigned score to each candidate network, score-based algorithms search over the space of all possible structures and return the optimal one. A direct searching, however, could cause an intractable problem when the number of variables is large. The reason lies in the fact that the potential space of network structures is at least exponential in the number of variables n: there are $n(n-1)$ possible directed edges and thus $2^{n(n-1)}$ possible structures for every subset of these edges. Any exhaustive searching approach for all possible structures is unwise, and instead heuristic methods are employed in practice. One obvious choice is the hill-climbing algorithm, whose idea is to generate a model in a step-by-step fashion by making the maximum possible improvement in an objective quality function at each step ((Khanteymoori, Homayounpour, and Menhaj 2009)). Other meritorious search methods like stochastic hill-climbing, *Tabu greedy search* (TABU), and genetic algorithms ((Larrañaga et al. 1997)) are also commonly used.

Score-based structure learning algorithms may get stuck in a local maximum when the initial condition of searching is not properly set. Hence if an initial structure with some pre-existing arcs is known in advance, the probability of score-based algorithms to settle down on the global maximum will increase.

Clustering Analysis

In the literature, clustering analysis aims at grouping or segmenting a collection of objects into subsets or "clusters", such that those within each cluster are more closely related to one another than objects assigned to different clusters ((Hastie, Tibshirani, and Friedman 2009)). Clustering analysis can be done horizontally or vertically on a data set of n independent measurements and N variables. More precisely, we can either segment measurements or group variables into clusters. In this paper we concentrated on partitioning the N variables into K distinct groups, where the number K is a tuning parameter.

Central to clustering analysis is the choice of a measure of the dissimilarity (or distance) between different items. A clustering method attempts to group the items based on the definition of dissimilarity supplied to it ((Hastie, Tibshirani, and Friedman 2009)). In this paper a refined version of *1-correlation* will be introduced and implemented in order to obtain some prior information about the dependencies between variables. In addition, to match up with the test statistics in constraint-based algorithms, a negative version of *Mutual Information* will also be used on data sets with purely discrete variables. In the next section, we also discuss how we are going to select a clustering method for our *two-step clustering-based* strategy.

Method

In this section we will outline the framework of our strategy and discuss some involved details of our setting. As we mention earlier, prior information helps to improve the accuracy and reduce computational costs in Bayesian network structure learning. Thus our *two-step clustering-based* (TSCB) strategy, which automatically generates some pre-existing arcs from data, can be applied to any structure learning algorithm. To obtain more accurate arcs and minimize computational costs in the first step, we group the variables with a strong "dependency" via clustering analysis. Within each cluster, a traditional structure learning algorithm is conducted to learn the arcs, which work as the prior information for the second step structure learning. To combine clusters, we implement the same traditional algorithm with all the arcs in the first step being well-preserved. See Algorithm 1 for details.

Algorithm 1 *Two-step Clustering-based* Bayesian Network Structure Learning Strategy

Input:

- Data set $\mathcal{D} = X_1, X_2, ..., X_N$ with N variables
- The number of clusters: K (Parameter)

Step 1:

1: Compute the dissimilarity matrix.
2: Carry out clustering analysis via *average linkage agglomerative clustering method* and cut the dendrogram into K groups (clusters).
3: Learn Bayesian network structures within each cluster using a traditional algorithm A[1].

Step 2:

1: Apply the algorithm A again on all variables with the retained arcs to combine clusters.

Output: Bayesian network structure learned from the data set $\mathcal{D}$.

Dissimilarity Metric

When the outline of our TSCB strategy is clear there are still some details that we need to scrutinize. The computation of the dissimilarity matrix works as the foundation of clustering analysis. More precisely, the way that we define the distance metric between variables plays a central role in the outcome of clustering. Real-world data sets may contain variables with purely discrete attributes or continuous attributes. Nevertheless, some hybrid data sets that are the mixture of continuous and discrete variables also exist in biological and medical fields. Thus we must employ different dissimilarity metrics for each type of data sets, which are suitable to measure the dependencies between variables.

(Cover and Thomas 2006) suggested that the strength of dependencies between variables can be measured using mutual information or correlation. However, the selection between these two candidate metrics is subtle. One must recall that constraint-based algorithms rely heavily on conditional independence tests. Commonly, the test statistics is the *mutual information* for categorical variables while the *linear*

[1]This could be any traditional structure learning algorithm, like the grow-shrink algorithm.

correlation for continuous variables. As a result, the principle of choosing the dissimilarity metric is to match up with the test statistics in that we can maximize the prior information obtained from data in the first step and reduce computational costs in the second step. For data sets with purely discrete variables, we use the negative *mutual information* to define the dissimilarity metric $Dis(X;Y)$ between the variable X and Y,

$$Dis(X;Y) = -\sum_{x_i,y_i} \hat{P}(x_i,y_i) log \frac{\hat{P}(x_i,y_i)}{\hat{P}(x_i)\hat{P}(y_i)} \quad (4)$$

In practice, such mutual information is computed according to the estimated entropy of the empirical probability distribution.

As for data sets with purely continuous variables, the common *1-correlation* dissimilarity is used.

$$Dis(X;Y) = 1 - \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \quad (5)$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean of X, and analogously for $\bar{y}$.

The situations become more complicated when it comes to hybrid data sets. If we still want to apply the usual Pearson correlation to define the dissimilarity metric, we have to introduce a technique to transform the variables with discrete attributes.

1. **Converting**: Label attributes of discrete (or categorical) variables by nonnegative integers
2. **Centralization**: Shift the variables such that their attributes are central at 0

Table 1 illustrates how a discrete (or categorical) variable with three levels is converted and centralization into its numeric representation.

On the other hand, continuous variables can be discretized by *quantile* or *Harteminks pairwise mutual information*. Then the preceding argument can be applied to hybrid data sets.

Clustering Method

Besides the choice of a suitable dissimilarity metric, an adoption of clustering methods may, in some extents, affect the effectiveness of clustering analysis. There are basically two categories of clustering methods, partitioning and hierarchical. The representative of partitioning methods is K-means ((MacQueen 1967)) The results of applying K-means or K-medoids clustering algorithms depend on the choice for the number of clusters to be searched and a starting configuration assignment ((Hastie, Tibshirani, and Friedman 2009)). Consequently, this type of clustering algorithms is not robust enough to be applied in our TSCB strategy, especially when the performances of constraint-based algorithms are sensitive to the results of conditional independence tests. As for hierarchical clustering, it can be further divided into two basic paradigms: *agglomerative* (bottom-up) and *divisive* (top-down). The agglomerative method requires users to specify the neighboring techniques used in clusters when comparing comparing the dissimilarity, which leads to three types of approaches, i.e., *single-linkage*, *complete-linkage*, and a compromise between these two, *average-linkage*. The single-linkage method, which uses a minimum-distance metric between clusters, often leads to long "chain" of clusters, whereas complete-linkage tends to produce many small, compact clusters ((Izenman 2008)). Therefore, we use the *average linkage agglomerative clustering method* as our default clustering method in order to distribute variables evenly between clusters.

Variable	Converting	Centralization
Attribute 1	0	-1
Attribute 2	1	0
Attribute 3	2	1

Table 1: An Example of Transforming a Discrete Variable

Accuracy Metric

To determine the accuracy of a learned network structure on a simulated data set, we use the following accuracy metric proposed by (Metz 1978).

$$Accuracy = \frac{\sum True\,positive + \sum True\,negative}{\sum Total\,population} \quad (6)$$

In practice, users should apply their own accuracy metrics to evaluate the performance of a resulted Bayesian network. For instance, to carry out a well-behaved Bayesian network classifier, the classification rates would be a more suitable accuracy metric. There is why we introduce an undefined parameter, the number of clusters K into our method, which can be tuned to the optimum in terms of users' own accuracy metric. This benign design, in some sense, extends the adaptability of our TSCB strategy in real world applications. In the upcoming experiments, the variation of the previous accuracy with respect to K will be investigated. It will show that our strategy is effective to ameliorate traditional structure learning algorithms among a wide range of K.

Experimental Methodology and Results

In this section we examine our *two-step clustering-based* (TSCB) Bayesian network structure learning strategy on some benchmark data sets. To illustrate the effectiveness of our method, two aspect of analyses will be displayed. First, we investigate how accuracies of traditional structure learning algorithms can be improved with the help of our strategy. Here we plug in six traditional structure learning algorithms to evaluate the adaptability of our method when the parameter is tuned to the optimum. Furthermore, we inspect the variation of accuracies on one of the synthetic data sets with regard to the parameter, the number of clusters K. Second, we record the running times in each step of our strategy and demonstrate the correctness of our automatic mechanism for generating prior information when it comes to the improvement of time efficiency. Meanwhile, the total elapsed times

Network Data	Number of nodes	Number of arcs	Average degrees
"asia"	8	8	2.00
"insurance"	27	52	3.85
"alarm"	37	46	2.49
"hepar2"	70	123	3.51

Table 2: The Description of Benchmark Data sets

of our algorithm with the choice of parameters corresponding to optimal states of accuracies on synthetic data sets are tested when we embed different traditional algorithms. In addition, we also analyze the variation of total running times of our algorithm with regard to the parameter K.

Experimental Methodology

Our experimental evaluations are conducted on four different sizes of commonly used Bayesian networks. Without any particular clarification, a synthetic data set with 1000 instances is randomly generated from each of Bayesian network data. These network data are "asia" ((Lauritzen and Spiegelhalter 1988)), "insurance" ((Binder et al. 1997)), "alarm" ((Beinlich et al. 1989)), and "hepar2" ((Onisko 2003)). See Table 2 for detailed descriptions of the network data.

We use the R version 3.4.2 (2017-9-28) software with the i5-4200U dual core processor to estimate the accuracy and time efficiency of our algorithm. Essentially, the implementation of our clustering-based algorithm relies on the "cluster" ((Maechler et al. 2017)), "infotheo" ((Meyer 2014)) and "bnlearn" package ((Scutari 2010)).

Accuracy Analysis

First, we are interested in how performances of traditional structure learning algorithms can be ameliorated in terms of the pre-assigned accuracy metric when we leverage clustering to construct pre-existing arcs from data. Hence six well-known algorithms are embedded into our two-step clustering-based algorithm to assess the amendments of their accuracies. Among these six traditional methods, four of them, i.e., GS, IAMB, inter-IAMB, and MMPC, belong to the constraint-based category while the other two, HC and TABU, are heuristic searching algorithms in order to maximize the pre-assigned scores. Moreover, to reduce the randomness of our experimental results, we repeat the random generating process of a synthetic data set as well as the corresponding accuracy experiment for 100 times when embedding different traditional algorithms.

Table 3 illustrates that our two-step clustering-based strategy with optimal choice of the parameter helps to improve the accuracies of traditional structure learning algorithms by automatically generating prior information from data. The improvements of accuracies seem not to be salient on the absolute values of the records. This could result from the fact that the instances simulated from data sets are not large enough to uncover sufficient pre-existing arcs in the first step. More importantly, due to the sparse configurations of Bayesian networks when the number of variables

Methods	"asia"	"insurance"	"alarm"	"hepar2"
GS	**0.9096** (0.8918)	**0.9309** (0.9263)	**0.9662** (0.9602)	**0.9763** (0.9753)
IAMB	**0.9084** (0.8896)	**0.9287** (0.9218)	**0.9715** (0.9686)	**0.9747** (0.9741)
Inter-IAMB	**0.9082** (0.8936)	**0.9281** (0.9208)	**0.9716** (0.9689)	**0.9748** (0.9742)
MMPC	**0.8557** (0.8546)	**0.9259** (**0.9259**)	**0.9649** (0.9646)	**0.9732** (0.9728)
HC	**0.9766** (**0.9766**)	**0.9328** (0.9293)	**0.9768** (0.9724)	**0.9824** (0.9822)
TABU	**0.9664** (0.9657)	**0.9422** (0.9312)	**0.9788** (0.9744)	**0.9814** (0.9810)

Table 3: Accuracy Result Comparisons Between the TSCB Strategy and the Embedded Traditional Algorithms. The records inside round brackets are the corresponding accuracies of the embedded traditional algorithms.

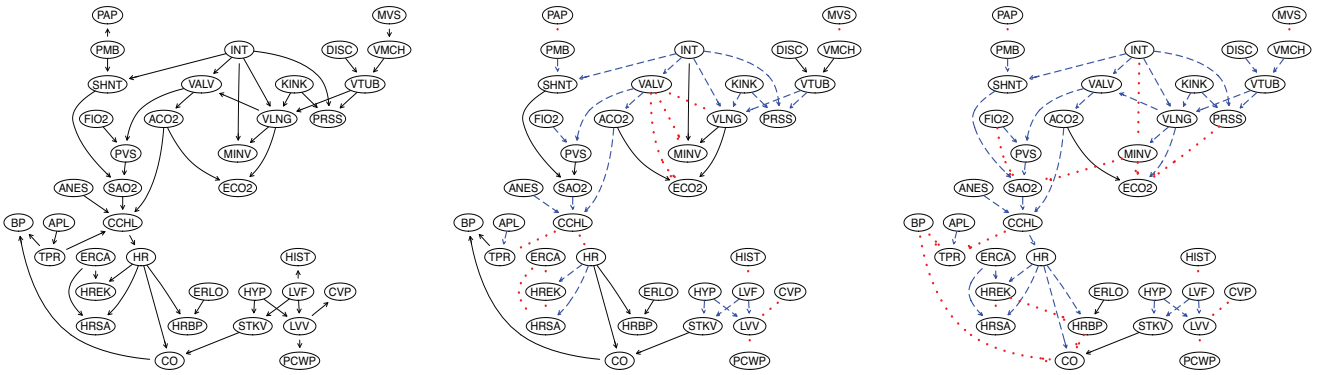
is large, any minute improvement of accuracies can indeed make a great difference to the resulted network structures. For instance, Figure 1 visualizes the actual network, the one learned with our TSCB strategy, and the one learned by the embedded traditional algorithm on the benchmark synthetic data set "alarm". From Figure 1, one can note that our TSCB strategy rectifies the error of traditional structure learning algorithms by detecting some false negative arcs. Here a benchmark "alarm" data set with 20000 instances works as the tested data set and the usual grow-shrink algorithm is plugged in.

However, skeptics may wonder how accuracies would vary with respect to different values of the critical parameter, the number of clusters K. Figure 2 displays the accuracy variations of our TSCB strategy with regards to K on the "alarm" data set when we embed the grow-shrink and hill-climbing algorithms. The accuracies of our two-step clustering-based algorithm are always identical to the embedded traditional algorithms when the number of clusters K reaches the maximum, i.e., the number of variables in the network data. The principle of this phenomenon is fairly intuitive, since there is only one variable per cluster when K is equal to the number of variables in a Bayesian network.

More importantly, Figure 2 demonstrates the robustness of our TSCB strategy when the accuracies of traditional algorithms can be improved among a wide range of effective values of K. In actual experiments, we inspect the accuracy variations on all mentioned benchmark data sets. Upon optimal stages, our TSCB strategy unanimously outweighs the performances of embedded traditional algorithms, though the effective ranges of the parameter K could vary on different data sets. For simplicity, we only present the results on the "alarm" data set.

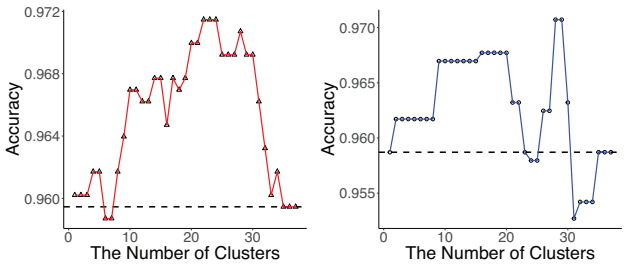
Time Analysis

Besides amendments on accuracies, our *two-step clustering-based* strategy is able to reduce computational costs of traditional structure learning algorithms simultaneously. When it is not always an easy task to measure the computational



(a) Actual Network (b) Learned by the TSCB Strategy (c) Learned by the GS algorithm

Figure 1: Network Configurations on the "alarm" Data Set. The red dotted arcs in each plotting are false positive arcs, namely, the arcs that are wrongly learned by structure learning methods. The blue dashed arcs are false negative arcs, which are not uncovered by structure learning algorithms.



(a) Embed Grow-shrink (b) Embed Hill-climbing

Figure 2: Accuracy Variation With Respect to the Number of Clusters. In each plot, there is a dash horizontal line, indicating the raw accuracy of the embedded traditional algorithm. The experiment is conducted on the "alarm" data set with 20000 instances.

Mean Elapsed Times / s	"asia"	"insurance"	"alarm"	"hepar2"
Clustering	0.00230	0.00788	0.01076	0.04432
Within clusters	0.00464	0.01670	0.05012	0.04744
Between clusters	0.00962	0.16420	0.24640	1.46168
TSCB	0.01656	**0.18878**	**0.30728**	**1.55344**
Traditional	**0.01010**	0.19362	0.35900	1.65584

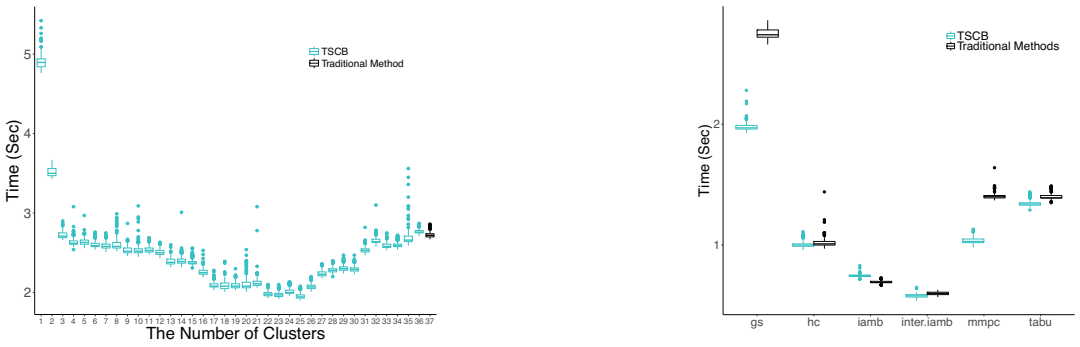
Table 4: Mean Elapsed Times Comparison.

costs of a method, recording the running times becomes an acceptable approach. The running times may vary significantly when the implementation of a method is conducted on different machines and software platforms. Hence we tend not to simply record the running times but rather make comparisons of total mean elapsed times and time distributions at different states.

To speed up the learning process of our strategy, some tradeoffs have to be made in time experiments. When computing the dissimilarity matrix of a synthetic data set we uniformly transform those discrete variables by the previously mentioned technique and apply the usual *1-correlation* metric. The reason lies in the fact that it is time-consuming to estimate the empirical mutual information for a data set with discrete variables in practice. Since the refined Pearson's correlation is able to reflect the dependencies between variables, our strategy is still well-behaved in terms of accuracies, though the improvements could be less salient.

To verify the effectiveness of pre-existing arcs when it comes to the acceleration of structure learning processes, we first segment the timing procedure on a synthetic data set into three sub-steps so as to record the elapsed times on clustering (including the computation of the dissimilarity matrix), learning arcs within clusters, and learning arcs between clusters (combining clusters), respectively. Here we embed the grow-shrink algorithm and tune the parameter to the optimum in terms of accuracy in each experiment. Again, to reduce the randomness of our experimental results, we repeat the generating process of a synthetic data set with 2000 random samples for 50 times and at the same time repeat the time recording process for 10 times. However, we exclusively generate 5000 random samples from "hepar2" network data because we want nearly all the levels in the discrete variable to appear in the simulated data set. Table 4 shows that with the optimal choice of the parameter in terms of accuracy, our TSCB can also reduce computational costs of traditional algorithms.

There are two points that are noteworthy to be scrutinized in Table 4. First, one can notice that the elapsed times for learning arcs between clusters are less than the elapsed times of the embedded traditional algorithms, especially when the size of the network is large. These running times saved from combining clusters in effect make an indispensable contribution to the reduction of the overall elapsed times of our



(a) Time Distributions

(b) Time Comparisons

Figure 3: Experimental Results of Elapsed Times on the "alarm" Data Set. Figure 3a displays time distributions of 200 repeating experiments for each possible value of the parameter when we embed the GS algorithms. The rightmost boxplot represents the time distribution of the traditional algorithm. Figure 3b presents the time comparisons between the TSCB algorithm and six traditional algorithms. For each pair of boxplots, the left one is for our TSCB method while the right one is for the embedded traditional algorithm.

TSCB strategy. More importantly, the improvement of time efficiency of learning arcs between clusters indeed verifies that such self-generating pre-existing arcs are able to accelerate the structure learning process. Second, since the clustering procedure is time-efficient, our automatic mechanism for generating prior information can be adapted to any traditional structure learning algorithm without requiring dramatic extra computational costs.

Moreover, we are going to investigate the variations of total elapsed of our TSCB strategy with respect to different values of the parameter K. Additionally, we justify that traditional structure learning algorithms from different categories benefit from our TSCB strategy in terms of time efficiency as well. Here we again conduct our experiments on the benchmark data set "alarm" with 20000 instances. For the time variation experiments, we only report the results embedding the grow-shrink algorithms. Our actual experiments on other traditional algorithms illustrate the similar tendency and thus are omitted here. See Figure 3 for details.

As shown in Figure 3a, our TSCB strategy improves time efficiency of the embedded traditional algorithm within a wide range of K. The improvement is most salient when the number of variables in most clusters is less than three. On the other hand, in Figure 3b, when embedding different traditional algorithms, our TSCB strategy also helps to reduce computational costs when the parameter is set to be optimal in terms of accuracy, especially when the traditonal algorithms come from the constraint-based category. Combined with the accuracy results, it is sufficient to demonstrate that a wide range of structure learning algorithms benefit from our TSCB strategy in terms of accuracy and time efficiency, though sometimes tuning the parameter is required.

Conclusion

In this paper we have proposed a *two-step clustering-based* strategy for Bayesian network structure learning, which can self-generate prior information from data. By dividing the original set into clusters and learning the network structure within and between clusters, the performance of a wide range of Bayesian network structure learning algorithms have been further improved. Interestingly, we observe that the optimal state of our TSCB strategy always attains when nearly all the clusters contain no more than three variables, which implies that the group of two or three nodes might be the primitive unit of the network structure. The phenomenon is consistent with the concept of "network motifs", proposed by (Milo et al. 2002). In our future work, we are particularly interested in investigating the physical meaning of each detected cluster, which will give us more insight of the performance improvement. Additionally, whether our TSCB strategy can tackle the cases with the presence of latent variables is also a possible direction for future research.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61503317, in part by the Grant from the Research Grant Council of Hong Kong SAR under Project HKBU12202417; and in part by the SZSTI Grant with the Projct Code JCYJ20170307161544087.

References

Beinlich, I.; Suermondt, H.; Chavez, R.; and Cooper, G. 1989. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, 247–256. Springer-Verlag.

Binder, J.; Koller, D.; Russell, S.; and Kanazawa, K. 1997. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning* 29(2):213–244.

Cano, A.; Masegosa, A.; and Moral, S. 2011. A Method for Integrating Expert Knowledge When Learning Bayesian Networks From Data. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the*

IEEE Systems, Man, and Cybernetics Society 41(05):1382–1394.

Chickering, D. M. 1996. *Learning Bayesian Networks is NP-Complete*. New York: Springer New York. 121–130.

Cover, T. M., and Thomas, J. A. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, Second edition.

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning* 29:131–163.

Friedman, N.; Nachman, I.; and Peér, D. 1999. Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. In *Proceedings of the 15th Conference on UAI*, 206–215. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Element of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag New York, Second edition.

Izenman, A. J. 2008. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Texts in Statistics. Springer-Verlag New York.

Khanteymoori, A. R.; Homayounpour, M. M.; and Menhaj, M. B. 2009. *A Bayesian Network Based Approach for Data Classification Using Structural Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg. 25–32.

Kojima, K.; Perrier, E.; Imoto, S.; and Miyano, S. 2010. Optimal Search on Clustered Structural Constraint for Learning Bayesian Network Structure. *Journal of Machine Learning Research* 11:285–310.

Larrañaga, P.; Sierra, B.; Gallego, M. J.; Michelena, M. J.; and Picaza, J. M. 1997. *Learning Bayesian Networks by Genetic Algorithms: A Case Study in the Prediction of Survival in Malignant Skin Melanoma*. Berlin, Heidelberg: Springer Berlin Heidelberg. 261–272.

Lauritzen, S. L., and Spiegelhalter, D. J. 1988. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 50(2):157–224.

MacQueen, J. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 281–297. Berkeley, Calif.: University of California Press.

Maechler, M.; Rousseeuw, P.; Struyf, A.; Hubert, M.; and Hornik, K. 2017. *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.6.

Margaritis, D. 2003. *Learning Bayesian Network Model Structure from Data*. Ph.D. Dissertation, Pittsburgh, USA.

Metz, C. 1978. Basic Principles of ROC Analysis. *Seminars in Nuclear Medicine* 8(4).

Meyer, P. E. 2014. *infotheo: Information-Theoretic Measures*. R package version 1.2.0.

Milo, R.; Shen-Orr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D.; and Alon, U. 2002. Network Motifs: Simple Building Blocks of Complex Networks. *Science* 298(5594):824–827.

Nagarajan, R.; Scutari, M.; and Lébre, S. 2013. *Bayesian Networks in R with Applications in Systems Biology*, volume 48 of *Use R!* Springer-Verlag New York.

Onisko, A. 2003. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*. Ph.D. Dissertation, Warsaw.

Pearl, J. 1982. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, 133–136. AAAI Press.

Perrier, E.; Imoto, S.; and Miyano, S. 2008. Finding Optimal Bayesian Network Given a Super-Structure. *Journal of Machine Learning Research* 9:2251–2286.

Scutari, M. 2010. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software, Articles* 35(3):1–22.

Tsamardinos, I.; Aliferis, C. F.; and Statnikov, A. 2003. Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, 673–678. New York, NY, USA: ACM.

Tsamardinos, I.; Aliferis, C.; Statnikov, A.; and Statnikov, E. 2003. Algorithms for Large Scale Markov Blanket Discovery. In *Proceedings of The Sixteenth International FLAIRS Conference, St*, 376–380. AAAI Press.

Verma, T., and Pearl, J. 1991. Equivalence and Synthesis of Causal Models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, 255–270. New York, NY, USA: Elsevier Science Inc.

Xu, J.-G.; Zhao, Y.; Chen, J.; and Han, C. 2015. A Structure Learning Algorithm for Bayesian Network Using Prior Knowledge. *Journal of Computer Science and Technology* 30(4):713–724.

Yaramakala, S., and Margaritis, D. 2005. Speculative Markov Blanket Discovery for Optimal Feature Selection. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, 809–812. TX, USA: IEEE Computer Society.

Zhang, Y.; Liu, J.; and Liu, Y. 2018. Bayesian Network Structure Learning: The Two-step Algorithm. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, Student Abstract and Poster Program (Accepted)*.