Bayesian Network Structure Learning: The Two-Step Clustering-Based Strategy and Algorithm Combination

Yikun Zhang

School of Mathematics
Sun Yat-sen University

A thesis submitted in fulfillment
of the requirements for the degree of
Bachelor of Science
of
Sun Yat-sen University.

May 2018

[Abstract]

Structure learning is a fundamental and challenging issue in dealing with Bayesian networks. In this thesis we introduce a two-step clustering-based strategy, which can automatically generate prior information from data in order to 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 candidate variables (or nodes) in the domain into several groups via clustering analysis and apply Bayesian network structure learning to obtain some potential 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 network datasets show that a wide range of traditional structure learning algorithms benefit from the proposed clustering-based strategy in both terms of accuracy and time efficiency. Furthermore, by combining the constraint-based version of our two-step clustering-based strategy with score-based greedy searching methods, we propose an algorithm composition technique, which is able to substantially further improve the accuracy of the resulting network structure.

[Keywords]: *Bayesian Networks, Clustering Analysis, Two-Step Structure Learning, Algorithm Combination*

【摘要】

结构学习是处理贝叶斯网络的一个基本且富有挑战性问题。本文介绍了一种基于聚类的两步策略，它可以自动从数据中生成先验信息，以提高贝叶斯网络结构学习中一些先进算法的准确性和时间效率。本文中基于聚类的策略由两个步骤组成。在第一步中，我们通过聚类分析将候选变量（或节点）分为几组，并应用贝叶斯网络结构学习来获得每个聚类中一些潜在的有向边。在第二步中，我们将聚类里的有向边保留下来，并学习网络聚类间的结构。基准网络数据集上的实验结果表明，大部分范畴下的传统结构学习算法在准确性和时间效率上均可受益于我们所提出的基于聚类的策略。此外，通过将基于约束条件的两步聚类策略与基于评分的贪心搜索方法相结合，我们提出了一种算法组合技术，该技术能够进一步大幅度地提高所得网络结构的准确性。

【关键词】：贝叶斯网络；聚类分析；两步结构学习；算法组合

Dedication

In loving memory of my grandfather (1937-2017), a respectable university president.

Contents

List of Figures

List of Tables

Abbreviations

TSCB	**T**wo-**S**tep **C**lustering-**B**ased strategy
DAG	**D**irected **A**cyclic **G**raph
CPD	**C**onditional **P**robability **D**istribution
GS	**G**row-**S**hrink algorithm
IAMB	**I**ncremental **A**ssociation **M**arkov **B**lanket algorithm
inter-IAMB	**Inter**leaved **I**ncremental **A**ssociation **M**arkov **B**lanket algorithm
MLE	**M**aximum **L**ikelihood **E**stimator
BIC	**B**ayesian **I**nformation **C**riterion
HC	**H**ill-**C**limbing algorithm
TABU	**tabu** search of algorithm
MMPC	**M**ax-**M**in **P**arents and **C**hildren algorithm
MMHC	**M**ax-**M**in **H**ill-**C**limbing algorithm

Chapter 1

Introduction and Motivation

Probabilistic graphical models based on directed acyclic graphs have a long and rich tradition, dating back to the work by a geneticist Sewall Wright in the 1920s [3]. However, such elegant and powerful models did not arouse researchers' interest in the scientific community until Judea Pearl formulated them with statistical tools and introduced the declarative representations of conditional independence relations into the field of artificial intelligence [4, 5]. These structured probabilistic models, later known as *Bayesian networks* or *belief networks*, is widely used to address uncertainty and causal relations among the variables in various scenarios. As a result, the applications of Bayesian networks range from medical research to social science. For instance, Sesen et al.(2013) applied Bayesian network approach to predict survival rates and select appropriate treatments for lung cancer patients [6]. On the other hand, the Bayesian network model can be viewed as a generalization of the powerful Naive Bayes classier and improve the classification ability of a Naive Bayes model by taking into account the correlations between variables (or features) [7].

Considering widespread applications of Bayesian networks, learning Bayesian networks from real-world datasets has been an intense research topic in the last two decades. Practitioners essentially face two main problems when learning a Bayesian network: structure learning as well as parameter learning. To address the issue of Bayesian network structure learning, some constraint-based, score-based, and hybrid algorithms have been proposed [8, 9]. As for parameter learning, common approaches are Maximum Likelihood Estimation, which chooses the parameters to maximize the likelihood

(or log-likelihood) function, and Bayesian Estimation, which incorporates the prior distribution of data into estimating procedures and carry out estimations via posterior distributions.

In this paper, we focus on Bayesian network structure learning, because [10]

1. We hope to learn models for new queries when expert knowledge is insufficient;
2. The resulting network structure can be used as a tool to reveal some important properties of the domain, especially to examine the dependencies between variables;
3. It is the prerequisite of parameter learning.

1.1 Related Work

1.1.1 Bayesian network Structure Learning

As we have mentioned earlier, existing state-of-the-art structure learning algorithms fall in three categories.

The first category utilizes *constraint-based structure learning* and views a Bayesian network as a representation of independencies. These methods attempt to test for conditional dependence and independence in the data and to find a network (or more precisely, an equivalence class of networks) that best explains these dependencies and independencies [10].

The second category is *score-based structure learning*. Score-based approaches treat structure learning as an optimization problem, where we define a hypothesis space of potential networks and assign a statistically motivated score that describes the fitness of each possible structure to the observed data [11].

The third category consists of *hybrid algorithms*, which combine constraint-based and score-based algorithms to offset their weaknesses and produce reliable network structures in a wide variety of situations [9].

The detailed discussion of representative algorithms, advantages, and disadvantages in each category will be presented in Chapter 2, where we systematically review all related background knowledge of Bayesian networks.

1.1.2 Learning Bayesian Network Structure With Prior Information

Although miscellaneous approaches have been designed to address Bayesian network structure learning, it is an NP-hard problem in general [12, 13], especially when it comes to score-based algorithms for an optimal network. Thus, practitioners need to resort to heuristic searching methods in the implementation of score-based algorithms. The constraint-based approaches, though more efficient, are highly sensitive to failures in independence tests [11]. More significantly, structure learning on large-scale datasets, like DNA expression data [14], is intractable and time-consuming. However, some prior information about network structures helps to reduce computational costs of structure learning algorithms and improve accuracies of output networks. Several studies incorporating prior knowledge into the structure learning process have been conducted recently. For example, Perrier et al.(2008) assumed the skeleton of a network and efficiently found an optimal Bayesian network by restricting the searching on its skeleton [15]. Nevertheless, some of the proposed 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 [16]. In addition, the previous work tends to introduce the expert knowledge by eliciting informative prior probability distributions of the graph structures [17]. This approach is highly constrained to the availability and correctness of prior information, which are difficult to achieve under real-world scenarios.

The preceding drawbacks in structure learning via prior information motivate us to develop a novel automatic way to generate prior information from data by the structure learning algorithm itself. The common prior information comes from the existence and absence of arcs or parent nodes, distribution knowledge including the conditional probability distribution (CPD) of edges, and the probability distribution (PD) of nodes

[16]. Among various types of prior information, nothing goes better than the existence of arcs, since they serve as the principal components of a network structure. Therefore, we seek to generate some prior knowledge of the existence of arcs directly from the data via structure learning algorithms.

1.2 Inspirations and Our Contributions

To the best of our knowledge, there are no systematic studies in investigating the existence of arcs within a Bayesian network structure based on the dataset, but several researchers has proposed algorithms to obtain other types of prior information from massive datasets. Friedman et al.(1999) applied the cluster-tree decomposition technique to figure out the candidate parents of a variable in a Bayesian network, which was regarded as an innovative way to generate prior information from data itself [11]. Additionally, Kojima et al.(2010) divided the super-structure (a pre-assumed skeleton for the resulting network) into several clusters in order to extend the feasibility of their constrained optimal searching method [18]. Both of them utilized the concept of clustering analysis, which groups variables based on their similarity, to accelerate the structure learning process and alleviate wrong arcs produced by heuristic methods. Their ideas inspire us to segment variables in the dataset into clusters and learn some strongly "prospective" arcs within each cluster as prior information for the main part of structure learning.

We thus propose a *two-step clustering-based* (TSCB) Bayesian network structure learning strategy, which can automatically retrieve some information about the existence of arcs from data. In the first step, we applying clustering analysis to group those strongly "dependent" variables (or nodes) and learn the arcs within clusters via a conventional structure learning algorithm. In the second step, retaining all the arcs within clusters, we implement the same traditional algorithm again in order to learn the arcs between clusters. The contributions of our *two-step clustering-based* strategy fall into two aspects:

1. It furnishes us an automatic mechanism to generate prior information and tackle those real-world structure learning problems when expert knowledge is scarce or even nowhere to obtain;

2. It ameliorates the performances of a wide range of traditional structure learning algorithms in terms of accuracy and time efficiency simultaneously.

In order to further improve the accuracy of a network structure learned by our TSCB strategy, we inherit the principle of ensemble learning [19] and hybrid structure learning algorithms to propose an algorithm combination technique. The combined algorithm leverages the constraint-based version of our TSCB strategy to initial a directed acyclic graph structure for the subsequent BIC scoring heuristic searching. By combining the TSCB strategy and any score-based method, the resulting network structure can better reveal the underlying distribution.

1.3 Overview and Declaration of Previous Work

The rest of the thesis document is organized as follows. In Chapter 2 we review the necessary background knowledge on Bayesian network structure learning and clustering analysis. In Chapter 3 we outline the framework of our two-step clustering-based strategy, introduce an important technique to deal with hybrid datasets that involve both discrete and continuous variables, and discuss some default settings of our strategy. In Chapter 4 the experimental setups as well as evaluation of our strategy on synthetic benchmark datasets will be presented, which demonstrate the effectiveness of our strategy on the improvement of traditional algorithms. Chapter 5 focuses on improving the accuracy of constraint-based methods via a proposed algorithm combination technique and presenting the corresponding experimental evaluation. We conclude with a discussion of our contributions and future directions in Chapter 6.

This thesis is based on the following previously accepted materials:

1. **Yikun Zhang**, Yang Liu, and Jiming Liu. **Learning Bayesian Network Structure by Self-Generating Prior Information: The Two-step Clustering-based Strategy.** *In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) Workshops Program*, New Orleans, LA, USA, 2018 (In Press).

2. **Yikun Zhang**, Jiming Liu, and Yang Liu. **Bayesian Network Structure Learning: The Two-step Clustering-based Algorithm.** *In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) Student Abstract and Poster Program*, New Orleans, LA, USA, 2018 (In Press).

Chapter 2

Background Knowledge

In this chapter we describe in detail the certain aspects of Bayesian network structure learning and clustering analysis. We begin with a description of general notations that would be used in following chapters. Consider a finite set $\mathbf{U} = \{X_1, ..., X_N\}$ of one-dimensional random variables where each variable X_i may take on values either from a finite set or a subinterval of the real line $\mathbb{R}$, all denoted by $Val(X_i)$. We use capital letters such as X, Y, Z for variable names and lower-case letters such as x, y, z to denote specific values taken by those variables. Higher-dimensional random variables are typically denoted by boldface capital letters, such as $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, and their assignments of values are denoted by boldface lowercase letters $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Finally, let P be a joint distribution over the variables in $\mathbf{U}$, and let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be some subsets of $\mathbf{U}$. The notation $\mathbf{X} \perp \mathbf{Y}$ means that $\mathbf{X}$ and $\mathbf{Y}$ are *independent*, i.e., for all $\mathbf{x} \in Val(\mathbf{X}), \mathbf{y} \in Val(\mathbf{Y})$, $P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x})P(\mathbf{y})$. Also, we says that $\mathbf{X}$ and $\mathbf{Y}$ are *conditionally independent* given $\mathbf{Z}$, denoted by $(\mathbf{X} \perp \mathbf{Y}|\mathbf{Z})$, if for all $\mathbf{x} \in Val(\mathbf{X}), \mathbf{y} \in Val(\mathbf{Y}), \mathbf{z} \in Val(\mathbf{Z})$, $P(\mathbf{x}|\mathbf{y}, \mathbf{z}) = P(\mathbf{x}|\mathbf{z})$, or equivalently, $P(\mathbf{x}, \mathbf{y}|\mathbf{z}) = P(\mathbf{x}|\mathbf{z})P(\mathbf{y}|\mathbf{z})$ whenever $P(\mathbf{y}, \mathbf{z}) > 0$.

2.1 Bayesian Network Foundation

The core of the Bayesian network representation is a directed acyclic graph (DAG) $\mathcal{G}$, whose nodes are the random variables in our domain and whose edges (or arcs) correspond, intuitively, to direct influence of one variable on another. This graph can be viewed in two different ways:

- as a data structure that provides the skeleton for representing a joint distribution compactly in a factorized way;
- as a compact representation for a set of conditional independence assumptions about a distribution.

These two views are, in a strong sense, equivalent [10]. The formal definition of the semantics of a Bayesian network requires the terminologies of conditional independence as well as factorization of the joint probability distribution.

Definition 2.1 (Bayesian Network Structure)**.** *A* Bayesian network structure $\mathcal{G}$ *is a directed acyclic graph whose nodes represent random variables* $X_1, ..., X_n$*. Let* $Pa_{X_i}^{\mathcal{G}}$ *denote the parents of* X_i *in* $\mathcal{G}$*, and* S_{X_i} *denote the variables in the graph that are not descendants of* X_i*. Then* $\mathcal{G}$ *encodes the following set of conditional independence assumptions, called the* local independencies*, and denoted by* $\mathcal{I}_\ell(\mathcal{G})$*:*

$$\textit{For each variable } X_i\textit{: } (X_i \perp S_{X_i} | Pa_{X_i}^{\mathcal{G}})\textit{.}$$

Definition 2.2 (Factorization)**.** *Let* $\mathcal{G}$ *be a Bayesian network structure over the variables* $X_1, ..., X_n$*. We say that a distribution* P *over the same variable space* factorizes *according to* $\mathcal{G}$ *if* P *can be decomposed into a product*

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | Pa_{X_i}^{\mathcal{G}}).$$

This equation is the so-called *chain rule for Bayesian networks*. The individual factors $P(X_i | Pa_{X_i}^{\mathcal{G}})$ are called *conditional probability distributions* (CPDs). We are now prepared to display the formal definition of a Bayesian network.

Definition 2.3 (Bayesian Network)**.** *A* Bayesian network *is a pair* $\mathcal{B} = (\mathcal{G}, P)$ *where a distribution* P *factorizes over* $\mathcal{G}$*, and where* P *is specified as a set of CPDs associated with* $\mathcal{G}$*'s nodes (variables).*

A simple Bayesian network of discrete variables is shown in Figure 2.1, which models the potential situation that a metastatic cancer disease patient would face. The domain is abstracted to five variables, all of which are binary. The relations between the

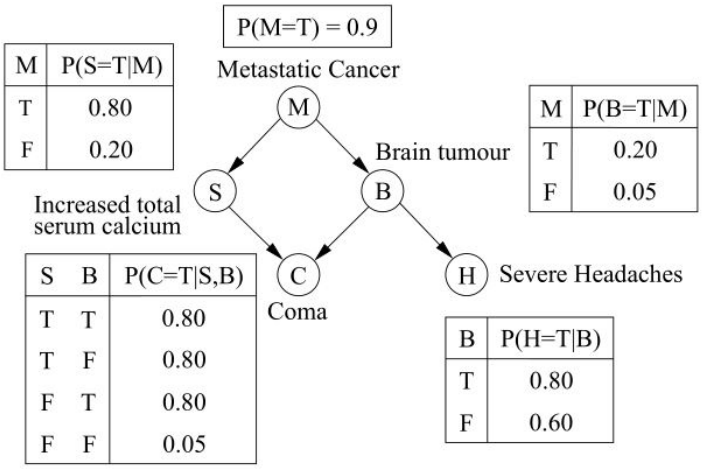


FIGURE 2.1: An Example Bayesian Network Modeling the Metastatic Cancer Problem: Structure and CPTs [1]

occurrence of the disease and some of its typical symptoms can be revealed by the network structure and CPDs annotated on the variables.

2.1.1 Independencies in Bayesian Network

We first analyze the independencies in some small components of the Bayesian network, which consist of the following five typical cases:

- Direct connection: $X \rightarrow Y$;
- Indirect causal effect: $X \rightarrow Z \rightarrow Y$;
- Indirect evidential effect: $X \leftarrow Z \leftarrow Y$;
- Common cause: $X \leftarrow Z \rightarrow Y$;
- Common effect: $X \rightarrow Z \leftarrow Y$.

The first case, when X and Y are connected, indicates that X can directly influence Y. Hence it is always possible to construct a distribution that X and Y are correlated

regardless of any evidence of the other variables in the network. In other words, X and Y are dependent[1].

We cannot obtain any independencies from the direct connection case. The other indirect connection cases, however, furnish us some intuitive conditional independencies. The indirect causal and evidential effects, which are symmetric, illuminate that X and Y can influence each other if we know nothing about Z. Nevertheless, when Z is somehow observed, X can no longer influence Y, and vice versa. Therefore, X and Y are independent given the evidence of Z, that is, $(X \perp Y|Z)$.

Similarly, in the common cause case, X can influence Y via Z if and only if Z is not observed. If the evidence of Z is given, the "flow" from X to Y is blocked and thus $(X \perp Y|Z)$.

The case becomes subtler when it comes to the common effect case, which is also known as a *v-structure*. X and Y are independent based on the assumptions of the Bayesian network, while they become correlated if we get access to some evidence in Z. A variant of this case, when we observe not Z but its descendants, also enables us to change our beliefs of Y by the evidence of X. Therefore, X and Y are independent if none of Z and its descendants are conditioned on.

Any path between two variables (nodes) in a Bayesian network is comprised of these five components. The preceding analysis induces the formal definition of *d-separation*, adapted from Pearl (1995) [20]:

Definition 2.4 (d-separation)**.** *Let $\mathcal{G}$ be a Bayesian network structure, and $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be three sets of nodes in $\mathcal{G}$. Then $\mathbf{X}$ and $\mathbf{Y}$ are* d-separated *given $\mathbf{Z}$, denoted by $(\mathbf{X} \perp_{\mathcal{G}} \mathbf{Y}|\mathbf{Z})$, if for any path between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$, there exists a node Z in $\mathbf{Z}$ satisfying one of the following two conditions:*

1. *Whenever we have a v-structure $X_i \rightarrow Z \leftarrow X_j$, then neither Z nor any of its descendants are in $\mathbf{Z}$, or*

2. *Z does not have converging arrows (v-structures) and Z is in $\mathbf{Z}$.*

[1]Here we ignore context-specific independencies. This happens when, for instance, $X = Y$ and X deterministically takes some particular value give the evidence of another variables Z, then X and Y both deterministically take that value, and are thus uncorrelated [10].

We can judge whether the d-separation assumption holds for any two nodes from the Bayesian network structure. The definition of the Bayesian network guarantees that whenever two nodes (variables) X and Y are d-separated given some other nodes $\mathbf{Z}$ they are conditionally independent given $\mathbf{Z}$. In effect, the main purpose of the Bayesian network structure is to summarize a number of conditional independence relations, graphically [8].

Definition 2.5 (I-map)**.** *A DAG $\mathcal{G}$ is called an* I-map *of a probability distribution P if every (conditional) independence displayed on* $\mathbf{G}$ *through the rules of d-separation, are valid in P.*

With the notation in Definition 2.1, we can also write the definition of an I-map as $\mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(P)$, where $\mathcal{I}(P)$ denotes the set of independence assertions that hold in P. It is interesting to know that a full-connected graph, encoding no independencies, is always an I-map of any probability distribution. Thus, for $\mathcal{G}$ to be a Bayesian network, it must be a minimal I-map of P, i.e., no edges can be removed from $\mathcal{G}$ without violating the I-map property.

2.1.2 I-equivalence

The Bayesian network structure specifies a set of conditional independence assertions. One crucial observation is that different Bayesian network structures can indeed imply the same set of conditional independence assumptions. Consider the situations where variables X, Y, Z form the indirect causal effect, evidential effect, as well as common cause structures, as mentioned in the previous subsection 2.1.1. All of them encode precisely the same independence assumptions: $(X \perp Y|Z)$. We formulate the equivalent relation of different graph structures that encodes the identical set of (conditional) independence assumptions.

Definition 2.6 (I-equivalence)**.** *Two graph structures $\mathcal{G}_1$ and $\mathcal{G}_2$ over the variable set* $\mathbf{U} = \{X_1, ..., X_n\}$ *are* I-equivalent *if $\mathcal{I}_\ell(\mathcal{G}_1) = \mathcal{I}_\ell(\mathcal{G}_2)$. The set of all graphs over* $\mathbf{U}$ *is partitioned into a set of mutually exclusive and exhaustive* I-equivalence classes*, which are the set of equivalence classes induced by the I-equivalence relation.*

Note that the v-structure network induces a very different set of d-separation assertions, and hence it does not fall into the category of indirect causal effect, evidential effect, and common cause structures [10].

The I-equivalence property of Bayesian network structure intrinsically complicates the structure learning problem, because it is unwarranted to prefer one network structure to another within the same I-equivalence based on information from data. In particular, although we can determine, for a distribution $P(X, Y)$, whether X and Y are correlated, there is nothing in the distribution that can help us determine whether the correct structure is $X \rightarrow Y$ or $Y \rightarrow X$. Therefore, the best one may hope for is a structure learning algorithm that, asymptotically, recovers the true structure $\mathcal{G}^*$'s equivalence class [10]. In practice, the Bayesian network is a competing model to address causality, and directionality of edges embraces causal significance in the domain. With the help of the domain knowledge, we distinguish a causal Bayesian network from other candidate networks within its I-equivalence class by identifying the directionality of edges. Unfortunately, learning the causal Bayesian network relies on the assumption that no *confounding factors* (or latent variables) exists in the dataset and requires high-quality prior knowledge. Although our two-step clustered strategy is able to generate a small amount of prior information and rectify the directionality of some arcs, it is still insufficient to learn causal models in many cases.

2.2 Bayesian Network Structure Learning

In this section we aim at reviewing two main categories of structure learning algorithms, constraint-based and score-based methods. As a paramount step in learning a Bayesian network, learning the structure attempts to discover a network $\mathcal{G}$ that best fits the given dataset. As we have discussed, it is almost impossible to correctly identify the directions of all the arcs in the resulting network from the dataset. The existing state-of-the-art algorithms tend to determine a particular I-equivalence class of network structures by uncovering a set of conditional independence relations among the domain variables.

Nonetheless, even on synthetic datasets, the samples are noisy and do not reconstruct the dependencies among variables correctly. Hence some compromises have to be made in our learned structure. On one hand, we may include as many edges as we can, even though some of them are spurious. On the other hand, fewer edges are retained in the learned model, and we may miss dependencies in consequence. However, we will see later that in practice, a sparser structure is beneficial to prevent overfitting and regularization methods should be applied to penalize complex structures.

2.2.1 Constraint-Based Approaches

The foundation of constraint-based structure learning algorithms is the profound work of Verma and Pearl (1991), the *inductive causation* algorithms [21]. 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 [22]. The standard framework of the conditional independence tests is to define a measure of deviance from the null hypothesis H_0, which is the assumption that $P^*(X, Y) = \hat{P}(X)\hat{P}(Y)$, where P^* is the underlying distribution and $\hat{P}$ is the empirical distribution [10]. One potential measure of this type is the χ^2 statistics:

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y} \frac{(M[x, y] - M \cdot \hat{P}(x) \cdot \hat{P}(y))^2}{M \cdot \hat{P}(x) \cdot \hat{P}(y)},$$

where M is the total number of data instances and $M[x, y]$ is the count of data instances when (X, Y) takes the value (x, y). However, the most commonly used deviance measure is the *mutual information* in the empirical distribution defined by the data set $\mathcal{D}$:

$$d_{\mathbf{I}}(\mathcal{D}) = \sum_{x,y} \hat{P}(x, y) \log \frac{\hat{P}(x, y)}{\hat{P}(x)\hat{P}(y)}. \tag{2.1}$$

Once we agree on a deviance measure d, we can devise a rule R_d to determine whether we want to accept the hypothesis

$$R_d(\mathcal{D}) = \begin{cases} \text{Accept} & d(\mathcal{D}) \leq t, \\ \text{Reject} & d(\mathcal{D}) > t, \end{cases}$$

where t is a pre-specified threshold [10].

Classical constraint-based algorithms cannot be applied to any real-world problem due to the exponential number of possible conditional independence relationships [9]. As a result, a novel approach, Grow-Shrink (GS) algorithm [23], was proposed. The plain version of the GS algorithm utilized Markov blanket information for inducing the structure of a Bayesian network and employed independence 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 2.7 (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 other words, $\mathbf{BL}(X)$ completely d-separates the variable X from any other variable outside $\mathbf{BL}(X) \cup \{X\}$. It is illuminating to mention that, in the Bayesian network framework, the Markov blanket of a node X is easily identifiable from the graph [8]: it 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 [24]. 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 [25].

When applying constraint-based algorithms one must realize that some of the independence test results could be wrong and this category of structure learning methods

are inherently sensitive to the failures of these conditional independence tests. Even a single wrong rejection of the null hypothesis could account for a totally different network structure compared to the real one. Although some preceding algorithms like the GS algorithm may improve the robustness of their original versions by incorporating random factors, practitioners still seek some prior knowledge to assure the reliability of conditional independence tests. Therefore, it is urgent and worthwhile to design an automatic mechanism to furnish constraint-based methods some prior information.

2.2.2 Score-Based Approaches

As discussed in Chapter 1, score-based algorithms treat the Bayesian network structure learning as an optimization problem, by assigning a statistically motivated score to each candidate Bayesian network. So, the choice of the scoring function becomes the most crucial factor in the whole learning process, determining the performance of score-based methods. We are supposed to scrutinize two most obvious choices of the scoring function.

2.2.2.1 Likelihood Score

Recall that a Bayesian network represents a particular form of a joint probability distribution. Given a set of data samples simulated from the probability distribution and a potential network structure, we can use the standard method to compute the (log-)likelihood function. As for structure learning task, it seems intuitive to find a model $(\mathcal{G}, \theta_{\mathcal{G}})$ that maximizes the probability of the data, or equivalently, the (log-)likelihood function. This can be achieved when we use the *Maximum Likelihood Estimators* (MLE) $\hat{\theta}_{\mathcal{G}}$ for that graph. Thus, a natural assignment of the scoring function is defined by

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}), \tag{2.2}$$

where $\ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D})$ is the logarithm of the likelihood function and $\hat{\theta}_{\mathcal{G}}$ are the maximum likelihood parameters for $\mathcal{G}$. The likelihood score can be interpreted in the view of the information theory.

Theorem 2.8. *Assuming data instances are independent and identically distributed, the likelihood score decomposes as follows:*

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(X_i; Pa_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \mathbf{H}_{\hat{P}}(X_i),$$

where $\mathbf{I}_{\hat{P}}(X, Y)$ *is the mutual information between* X *and* Y*,* $\mathbf{H}_{\hat{P}}(X)$ *is the entropy of* X*, and* M *is the total number of data instances.*

Proof. See Appendix A.

Since mutual information measures the strength of dependencies among variables, the likelihood score of a network structure indicates the strength of the dependencies between variables and their parents. In other words, the higher the likelihood score is, the more informative the parents of each variables would be. In spite of this, the likelihood score has a pronounced limitations that suppress its power to uncover the true structure. Consider the network $\mathcal{G}_\emptyset$ with two variables where X and Y are independent and the network $\mathcal{G}_{X\to Y}$ where X is the parent of Y. By Theorem 2.8, $\text{score}_L(\mathcal{G}_{X\to Y} : \mathcal{D}) - \text{score}_L(\mathcal{G}_\emptyset : \mathcal{D}) = M \cdot \mathbf{I}_{\hat{P}}(X; Y)$. However, it is easy to check that the mutual information between two variables is always nonnegative. Thus, $\text{score}_L(\mathcal{G}_{X\to Y} : \mathcal{D}) \geq \text{score}_L(\mathcal{G}_\emptyset : \mathcal{D})$ for any dataset $\mathcal{D}$. This indicates that the maximum likelihood score would consistently prefer a more complex structure unless X and Y are truly independent in the dataset. Due to statistical noise, exact independence almost never occurs in the empirical distribution, and thus, the maximum likelihood network will be a fully connected one [10]. This phenomenon is known as overfitting: the learned network will be extremely well-performed on the training data, while fails to generalize well to the test data. As a result, some techniques are required to avoid overfitting and Bayesian Information Criterion (BIC) score provides us with a feasible approach to penalize dense structures.

2.2.2.2 BIC Score

As we have seen, the maximum likelihood score tends to favor complex network structures, which are detrimental to new data cases. Hence it is necessary to penalize the

dense structure in order to obtain a generalizable network. One common approach to address this issue is to introduce a regularization term to the likelihood score, yielding the well-known Bayesian Information Criterion (BIC) score[2]:

$$\text{score}_{BIC}(\mathcal{G} : \mathcal{D}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2}\text{Dim}[\mathcal{G}] + O(1), \tag{2.3}$$

where $\text{Dim}[\mathcal{G}]$ is the model dimension, or the number of independent parameters in $\mathcal{G}$. The negate of the BIC score is also known as *minimum description length* score.

By Theorem 2.8, we can further decompose the BIC score into

$$\text{score}_{BIC}(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(X_i; Pa_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \mathbf{H}_{\hat{P}}(X_i) - \frac{\log M}{2}\text{Dim}[\mathcal{G}].$$

The BIC score exhibits a trade-off between fit to data and model complexity: the stronger the dependence of a variable with its parents, the higher the score; the more complex the network, the lower the score. However, the growth rate of the mutual information term is $O(M)$ while the regularization term grows logarithmically in M. More emphasis will be placed on the fit to data [10]. More surprisingly, it turns out that the BIC score will asymptotically favor a structure that fits the dependencies of the underlying probability distribution. This property is called the consistency of the score.

Definition 2.9 (consistent score)**.** *Assume that our data are generated by some distribution P^* for which the network $\mathcal{G}^*$ is a structure that precisely captures the independencies in P^*. We say that a scoring function is* consistent *if the following properties hold as the amount of data $M \to \infty$, with probability that approaches 1 (over possible choices of data set $\mathcal{D}$):*

- *The structure $\mathcal{G}^*$ will maximize the score.*
- *All structures $\mathcal{G}$ that are not I-equivalent to $\mathcal{G}^*$ will have strictly lower score.*

[2]In fact, the BIC score is the approximation of the $\log P(\mathcal{D}|\mathcal{G}) = \log \int_{\Theta_{\mathcal{G}}} P(\mathcal{D}|\theta_{\mathcal{G}}, \mathcal{G})P(\theta_{\mathcal{G}}|\mathcal{G})d\theta_{\mathcal{G}}$ when we use a Dirichlet prior for all parameters as $M \to \infty$, where $P(\mathcal{D}|\theta_{\mathcal{G}}, \mathcal{G})$ is the likelihood of the data given the network $(\mathcal{G}, \theta_{\mathcal{G}})$ and $P(\theta_{\mathcal{G}}|\mathcal{G})$ is our prior distribution over different parameter values for the network $\mathcal{G}$. See [10] for details.

We can prove that the BIC score satisfies the preceding definition of consistency.

Theorem 2.10. *The BIC score is consistent.*

Proof. See Appendix A.

The consistency of the BIC score guarantees that it would not be biased toward simpler but wrong structure when we have a sufficiently large amount of data. Thanks to this benign property of the BIC score, score-based algorithms will unanimously leverage the BIC score to evaluate the quality of candidate network structures in the following experiments.

After selecting a metric score for each potential network, score-based algorithms search over the space of all possible structures and return an 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 (HC) 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 [26]. More precisely, in each step, the algorithm moves from one state of the structure to another via a set of searching operators: edge addition, edge deletion, and edge reversal, while at the same time, it tests the legality of the next state network, i.e., no cycles in the network. The heuristic searching method, as is known to us, may sometimes converge to a local maximum, from which all changes are score-reducing. The other possibility is to reach a plateau: a large set of neighboring networks that have the same score. By design, the greedy Hill-Climbing procedure cannot "navigate" through a plateau, since it relies on improvement in score to guide it to better structures [10].

One plausible method to solve the plateau problem is the tabu search (TABU) of algorithm. The procedure keeps a list of recent searching operators that we have applied,

and in each step we do not consider the operators that reverse the effect of operators applied within a history window of some predetermined length L. See [10] for the detailed algorithm. In the following experiments, whenever we apply the tabu greedy searching algorithm, the length of the list is set to be 10. Other meritorious search methods like stochastic hill-climbing and genetic algorithms [27] are also commonly used.

Even though researchers strive to bypass local maxima when applying heuristic searching methods, except in rare cases, there is no guarantee that the local maximum that we found is actually the desired global one. This happens in higher chances when the initial conditions of searching is not properly set. Therefore, one can expect that if an initial structure with some potential arcs is known in advance, the possibility of score-based algorithms to converge to the global maximum will increase in consequence.

2.2.3 Hybrid Approaches

Both constraint-based and score-based algorithms embody their own pros and cons, and one feasible approach is to combine them, using constraint-based methods to initial the searching state or narrow down the searching space to a particular graph skeleton, and then applying score-based methods to refine it. For instance, the "Sparse Candidate" algorithm utilized mutual information to determine the candidate parents for each variables before applying heuristic searching methods [11]. The framework of the algorithm was later instantiated to yield a more competing hybrid method, Max-Min Hill-Climbing (MMHC) algorithm. It first reconstructs the skeleton of a Bayesian network using Max-Min Parents and Children (MMPC) algorithm, which identifies the parents and children of any intended variables in a Bayesian network that faithfully represents the joint probability distribution of data [28]. Then it performs a Bayesian-scoring greedy Hill-Climbing search to orient the edges [29].

The hybrid approach exploits the advantage of both methods. It uses the global nature of the constraint-based methods to avoid local maxima [10], and utilizes the ability

of scoring functions to prevent irreversible mistakes occurred in conditional independence tests. This principle will be leveraged in Chapter 5 so as to further improve the accuracy of our two-step clustering-based strategy when embedding constraint-based algorithms.

2.3 Clustering Analysis

Clustering analysis, a well-known form of unsupervised learning, is an effective tool to divide unstructured multivariate data into several groups so that items within the same group are more similar to each other than those in different groups. It 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. In reality, specifying an appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm [30]. The specification of dissimilarity metrics also depends on the variable type. For the quantitative (or continuous) variables, the common choices are squared-error or absolute-error loss,

$$Dis(X;Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \text{ or } \sum_{i=1}^{n}|x_i - y_i|,$$

where x_i's and y_i's are measurements (or data instances) for the variable X and Y, respectively.

Alternatively, another dissimilarity measure that is closely related to conditional independence tests can be defined to be $1-$*Pearson's correlation*,

$$Dis(X;Y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{2.4}$$

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

For discrete variables (either categorical or ordinal), we can use the *negative* mutual information to compute their pairwise dissimilarities in order to reveal their dependencies,

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

In Chapter 3, we will again discuss the choice of dissimilarity metrics for our two-step clustering-based strategy. It varies when the variable type in the input dataset is different.

Despite the importance of an appropriate dissimilarity measure, clustering algorithms are emphasized more in the clustering literature. Since we attempt not to estimate the underlying probability distribution in the clustering step, our discussion of clustering algorithms concentrates on *combinatorial algorithms*. Perhaps the most popular representative of combinatorial algorithms is K-means [31]. 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 [30]. The number of clusters K in our two-step clustering-based strategy will work as a tuning parameter, and we want our strategy to be robust and maintain decent performances within a wide range of the values of K. Thus, K-means or K-medoids would not be selected as the default clustering method. Contrary to K-means, which partitions variables into the predetermined number of groups, agglomerative methods apply a *bottom-up* paradigm. They start at the bottom and at each level recursively merge a selected pair of clusters into a single clusters [30]. Consequently, users are required to specify a measure of dissimilarity between disjoint groups of observations, which leads to three types of approaches,

i.e., *single-linkage*, *complete-linkage*, and a compromise between these two measures, *average-linkage*. The single-linkage method, which uses a minimum-distance metric between clusters, often leads to long "chain" of clusters, whereas the complete-linkage measure tends to produce many small, compact clusters [32]. Therefore, we choose the *average linkage agglomerative clustering method* as our default clustering method in order to distribute variables evenly between clusters.

After a thorough review on the essential concepts of Bayesian networks and clustering analysis, we are prepared to present the main contribution of this thesis, the two-step clustering-based strategy in Chapter 3. Experiments on synthetic benchmark datasets show that a wide range of traditional structure learning algorithms benefit from our strategy in terms of both accuracy and time efficiency.

Chapter 3

Two-Step Clustering-Based Strategy

In this chapter we outline the framework of our proposed *two-step clustering-based* (TSCB) strategy in dealing with Bayesian network structure learning. In addition, we will introduce a technique to address the computation of dissimilarity matrices in hybrid datasets and scrutinize the default settings of our strategy.

3.1 Outline of the TSCB Strategy

As we have mentioned, prior information helps to improve the accuracy and reduce the computational cost in Bayesian network structure learning. Thus our two-step clustering-based strategy, which automatically generates prior information about the existence of 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 strong "dependencies" 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.

Figure 3.1 is an illustrative example about the procedures of our TSCB strategy, where the nodes are grouped into clusters based on their similarities (in this case, shapes or colors) and then applied a traditional structure learning algorithm twice.

When the overview of our TSCB strategy is clear we are supposed to scrutinize the choice of dissimilarity metric and our data processing technique.

Algorithm 1 *Two-Step Clustering-Based* Bayesian Network Structure Learning Strategy

Input:

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

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

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.

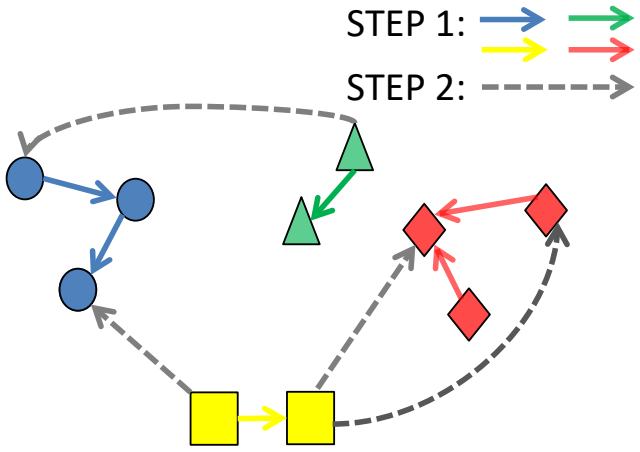


FIGURE 3.1: A Descriptive Example of the TSCB Strategy

3.2 Dissimilarity Metric and Data Processing

The choice of dissimilarity metric plays an indispensable role in the performance of clustering analysis. In Chapter 2 we have reviewed some commonly used dissimilarity metrics, which are suitable for different types of variables. Some real-world datasets, however, may contain both continuous and discrete variables. These hybrid datasets are

abound in biology and medical fields, where the Bayesian network is an effective tool to model their issues. Thus, we must develop a data processing technique to deal with the elaborate hybrid datasets, which, at the same time, should measure the dependencies between variables.

It has been suggested that the strength of dependencies between variables can be measured via mutual information or correlations [33]. Recall that to reveal pairwise dependencies, we use the negative mutual information for discrete variables but apply $1-$*(Pearson's) correlation* for continuous variables. More importantly, constraint-based algorithms learn the network structure by conditional independence tests, whose usual test statistics is the mutual information for discrete variables and linear correlation for continuous variables. Therefore, the underlying principle of choosing dissimilarity metrics is to match up with the test statistics in that we can maximize the prior information obtained from data in the first step. This, in turn, ameliorates the accuracies of conditional independence tests and reduces the computational costs in the second step.

Although the situation becomes more complicated when it comes to hybrid datasets, we still want to apply the usual Pearson's correlation to compute dissimilarity matrices because of its suitability of representing pairwise dependencies. As a result, we design a technique to transform discrete variables.

1. **Converting**: Label attributes of discrete variables by nonnegative integers

2. **Centralization**: Shift the variables such that their attributes are central at 0

Table 3.1 illustrates how a discrete (either categorical or ordinal) variable with three levels is converted and centralized into its numeric representation.

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

TABLE 3.1: An Example of Transforming a Discrete Variable

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

On the other hand, instead of transforming discrete variables, continuous variables can also be discretized by *quantile* or *Harteminks pairwise mutual information*. Then the foregoing dissimilarity metric for discrete variables, i.e., pairwise negative mutual information, can be applied to hybrid data sets. Unfortunately, to the best of my knowledge, there are no universal and widely accepted solutions to deal with hybrid datasets. Whether to transform discrete variables or to discretize continuous variables is more of an art than a science, and it often requires significant experimentations. In this thesis, we would rather transform discrete variables with our proposed technique, especially in pursuit of time efficiency.

As for the clustering algorithm, as discussed in Chapter 2, we use the *average linkage agglomerative clustering method* as our default clustering method. In the actual coding procedures, our TSCB strategy is encapsulated inside a function structure, and the clustering method is left to be one of its parameters. Therefore, in practice, users can specify the clustering method based on their actual scenarios.

3.3 Accuracy Metric

To determine the accuracy of a learned network structure on a simulated dataset, we use the following accuracy metric [34],

$$Accuracy = \frac{\sum True\ Positive + \sum True\ Negative}{\sum Total\ Population}. \tag{3.1}$$

In practice, users should apply their own accuracy metric to evaluate the performance of a resulting Bayesian network. For instance, in order to carry out a well-behaved Bayesian network classifier, classification rate would be a more suitable accuracy metric. This also explains 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 shows that our strategy is effective to ameliorate traditional structure learning algorithms among a wide range of K.

We finish the discussion of the framework of our TSCB strategy and some detailed settings. In the next chapter we are planning to display the experiments of our TSCB strategy on synthetic benchmark datasets and demonstrate its effectiveness when it comes to the improvements of traditional structure learning algorithms in terms of accuracy and time efficiency.

Chapter 4

Experimental Methodology and Results

In this chapter we examine our *two-step clustering-based* (TSCB) Bayesian network structure learning strategy on some benchmark datasets. To illustrate the effectiveness of our method, two aspects of experimental analysis will be displayed. First, we investigate how the accuracies of traditional structure learning algorithms can be improved with the assistance 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 datasets with respect to different values of 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. Furthermore, the total elapsed times 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 strategy with regard to the different values of the parameter K.

4.1 Experimental Methodology

Our experimental evaluations are conducted on four different sizes of benchmark Bayesian networks. Without any particular clarification, a synthetic dataset with 1000 instances is randomly generated from each of Bayesian network data. These network data are "asia" [35], "insurance" [36], "alarm" [37], and "hepar2" [38]. See Table 4.1 for detailed descriptions of these network data.

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 4.1: The Description of Benchmark Network Datasets

We use the R version 3.4.2 (2017-9-28) software with the i5-4200U dual core processor in an undisturbed environment to estimate the accuracies and time efficiency of different variants of our strategy. Essentially, the implementation of our TSCB strategy relies on the "cluster" [39], "infotheo" [40], and "bnlearn" [22] package.

4.2 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. Six well-known conventional algorithms are embedded into our two-step clustering-based strategy to assess the amendments of their accuracies. Among these six traditional methods, three of them, GS, IAMB, and inter-IAMB, belong to the constraint-based category; two of them, HC and TABU, are score-based algorithms; while MMPC is a local discovery algorithm that is used by a hybrid method, MMHC. To reduce the randomness of our experimental results, we repeat the random generating process of a synthetic dataset as well as the corresponding accuracy experiment for 100 times when embedding different traditional algorithms.

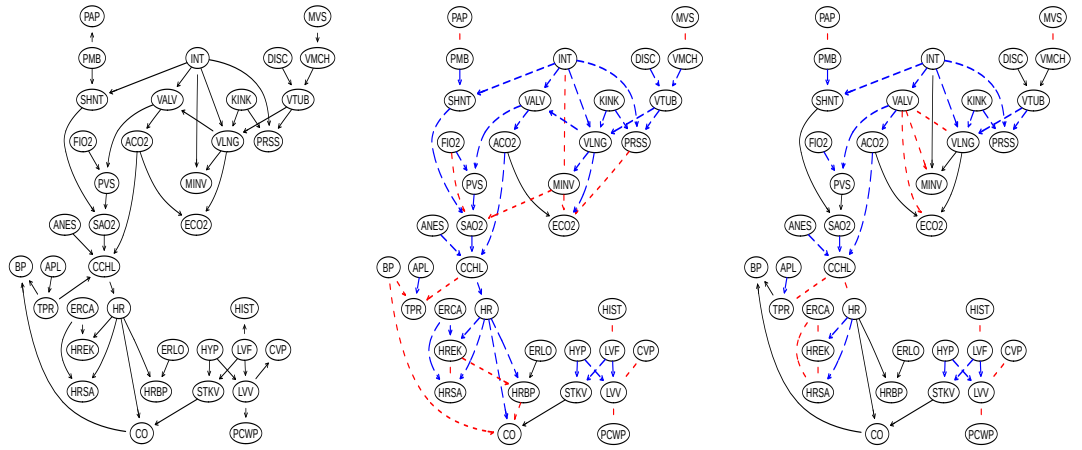
Table 4.2 illustrates that our two-step clustering-based strategy with an optimal choice of the parameter helps to improve the accuracies of traditional structure learning algorithms by automatically generating prior information from data. The improvements of accuracies seem not to be salient on the absolute values of the records. This could result from the fact that the instances simulated from datasets are not large enough to uncover sufficient candidate arcs in the first step. More importantly, due to the sparse configurations of Bayesian networks as the number of variables increases, any minute

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 4.2: **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.

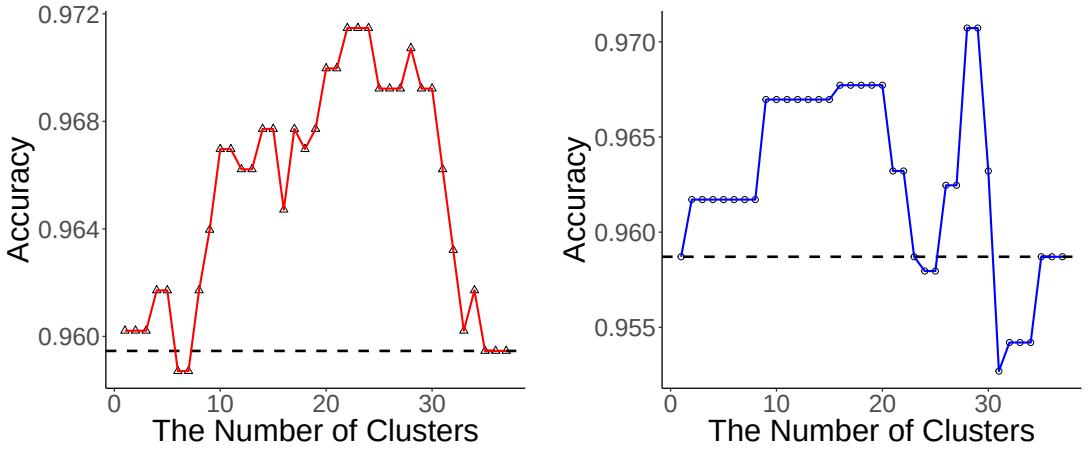
improvement of the accuracy can indeed make a great difference to the resulting network structure. For instance, Figure 4.1 visualizes the actual network, the one learned with our TSCB strategy, and the one learned by the embedded traditional algorithm on the benchmark synthetic dataset "alarm". From Figure 4.1, one can note that our TSCB strategy rectifies the error of traditional structure learning algorithms by detecting some false negative arcs and reversing some false positive arcs. Here a benchmark "alarm" data set with 20000 instances works as the test dataset 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 4.2 displays the accuracy variations of our TSCB strategy with regard to K on the "alarm" dataset when we embed the Grow-Shrink and Hill-Climbing algorithms. The accuracies of our two-step clustering-based strategy are always identical to the embedded traditional algorithms when the number of clusters K reaches the maximum, i.e., the total number of variables in the network data. The principle of this phenomenon is fairly intuitive, since



(A) Actual Network (B) Learned by the GS Algorithm (C) Learned by the TSCB Strategy

FIGURE 4.1: **Network Configurations on the "alarm" Dataset**. 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 Algorithm (B) Embed Hill-climbing Algorithm

FIGURE 4.2: **Accuracy Variation With Respect to the Number of Clusters**. In each plot, there is a horizontal dashed line, indicating the raw accuracy of the embedded traditional algorithm. The experiment is conducted on the "alarm" data set with 20000 instances.

there is only one variable per cluster when K is equal to the number of variables in the network.

More importantly, Figure 4.2 demonstrates the robustness of our TSCB strategy because the accuracies of traditional algorithms can be improved among a wide range

of the effective values of K. In actual experiments, we inspect the accuracy variations on all mentioned benchmark datasets. Upon optimal stages, our TSCB strategy unanimously outweighs the performance of any pure embedded traditional algorithms, though the effective ranges of the parameter K could vary on different datasets. For simplicity, we only present the results on the "alarm" dataset.

4.3 Time Efficiency 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 cost 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 the mean elapsed times of repeated experiments and time distributions at different states.

To speed up the learning process of our strategy, some tradeoffs have to be made during running time experiments. When computing the dissimilarity matrix of a synthetic dataset 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 more time-consuming to estimate the empirical mutual information than calculate the correlation between two variables in the R platform. Since the refined Pearson's correlation is appropriate enough to reflect the dependencies between variables, our strategy is still well-behaved in terms of accuracies, though the improvements could be less salient. In practice, computing the dissimilarity matrix via pairwise mutual information is still an optimal choice for the datasets with purely discrete variables, even when we take into account the time efficiency of our TSCB strategy.

To verify the effectiveness of the learned arcs in the first step for the acceleration of the whole structure learning process, we first segment the timing procedure on a synthetic dataset into three sub-steps so as to record the elapsed times on clustering

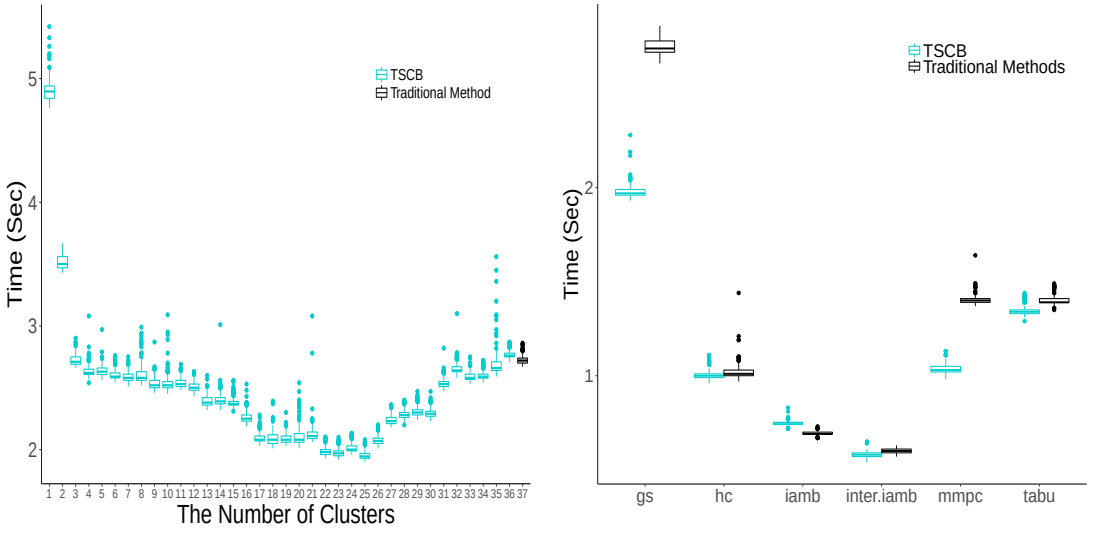
(including the computation of the dissimilarity matrix), learning arcs within clusters, and learning arcs between clusters (combining clusters), respectively. Here we embed the Grow-Shrink algorithm and tune the parameter to the optimum in terms of accuracy in each experiment. Again, to reduce the randomness 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 each time because we basically want all the levels in the variables to appear in the simulated dataset. Table 4.3 shows that with the optimal choice of the parameter in terms of accuracy, our TSCB can also reduce computational costs of traditional algorithms.

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 4.3: Mean Elapsed Time Comparisons.

There are two points that are noteworthy to be scrutinized in Table 4.3. First, one can notice that the elapsed time for learning arcs between clusters dominates the overall elapsed time for each learning process. By conducting clustering analysis in the first step, the elapsed times for combining clusters substantially decrease, 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 TSCB strategy. More importantly, the improvement of time efficiency of learning arcs between clusters indeed verifies that self-generating prospective arcs from data is 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 causing dramatic extra computational costs.

Moreover, we are going to investigate the variations of total elapsed times of our TSCB strategy with respect to different values of the parameter K. Additionally, we will 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 dataset "alarm" with 20000 instances. For the time variation experiments, we only report the results embedding the Grow-Shrink algorithm. Our actual experiments on other traditional algorithms illustrate the similar variation tendency and thus are omitted here. See Figure 4.3 for details.



(A) Time Distributions (B) Time Comparisons

FIGURE 4.3: **Experimental Results of Elapsed Times on the "alarm" Dataset**. Figure 4.3a displays time distributions of 200 repeated experiments for each possible value of the parameter when we embed the GS algorithm. The rightmost boxplot represents the time distribution of the traditional algorithm. Figure 4.3b presents the time comparisons between the TSCB strategy and six traditional algorithms. For each pair of boxplots, the left one is for our TSCB method while the right one is for the plug-in traditional algorithm.

As shown in Figure 4.3a, our TSCB strategy improves the 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 4.3b, our TSCB strategy also helps to reduce computational costs of the embedded traditional algorithms even when the parameter is set to be optimal in terms of accuracy. These amendments on computational costs are more pronounced 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.

With this chapter we conclude the discussion of our automatic mechanism for generate prior information, i.e., the two-step clustering-based strategy. In the next chapter we focus on combining constraint-based version of our TSCB strategy with score-based methods, which are committed to further improvement of our TSCB strategy in terms of accuracy.

Chapter 5

Further Improvement: Algorithm Combination

In the field of machine learning, ensemble learning methods are widely used for improving the prediction performance of a classifier by integrating multiple models [19]. One of the most popular approaches for creating correct ensembles is boosting [41, 42], which combines a sequence of weak classifiers to produce a refined prediction model. Ensemble learning methods mainly aim to tackle supervised learning problems, where we have predictors and response variables. Bayesian network structure learning, as well as clustering analysis that are leveraged in our TSCB strategy, fall into the category of unsupervised learning in most cases. Thus, ensemble methods seem to be of no use in our structure learning problem. In Chapter 2, however, we discuss with a special category of structure learning algorithms called hybrid methods, which exploit the principle of ensemble learning and synthesize both constraint-based and score-based approaches into the structure learning process. The idea of the combination is rather intuitive: it uses constraint-based methods to initialize a graph structure and then applies score-based methods to optimize it. Such an elegant combination of structure learning algorithms inspires us to combine score-based algorithms with our TSCB strategy embedded constraint-based methods so as to further ameliorate the accuracies of the resulting network structures.

5.1 Outline of Algorithm Combination

To initialize a network structure for subsequent score-based algorithms, we embed any constraint-based algorithm in our TSCB strategy and consequently obtain a preliminary network structure, which could be partially directed. Then the undirected edges in the network structure would be discarded, since score-based methods sometimes require a directed acyclic graph to start up the heuristic searching process and undirected edges do not reveal any causality between the variables. Some hybrid structure learning methods like the Max-Min Hill-Climbing algorithm require only a skeleton of a Bayesian network and orient the edges via score-based methods, which is different from our combined algorithm. Finally, given the directed acyclic graph as the initial state, any score-based method can thus be applied to refine the network structure. Recall that the BIC score is able to penalize complex structures and avoid overfitting. Thus, we again leverage the BIC scoring function in each greedy searching method. See Algorithm 2 for details.

Algorithm 2 A Combined Algorithm of the Two-Step Clustering-Based Strategy and Score-based Methods

Input:

- Dataset $\mathcal{D} = X_1, X_2, ..., X_N$ with N variables
- The number of clusters: K (Parameter)
- A constraint-based method to be optimized
- A BIC scoring heuristic searching method

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

1: Embed the constraint-based method into our TSCB strategy (Algorithm 1) to initialize a network structure
2: Retain only the directed edges of the network
3: Optimize the DAG via the BIC scoring heuristic searching method

After presenting the outline of our algorithm combination technique, we are supposed to assess the performances of the combined algorithm on benchmark network datasets.

5.2 Experimental Evaluation

As how we evaluate our TSCB strategy in the last chapter, we display the experiments of the proposed algorithm combination from two aspects, accuracy and computational cost. First we demonstrate that the accuracies of traditional constraint-based methods can be further ameliorated when they are embedded in our TSCB strategy and then combined with score-based methods. Again, we implement our TSCB strategy equipped with four conventional constraint-based algorithms, i.e., GS, IAMB, inter-IAMB, and MMPC. After learning an initial directed acyclic graph, we apply HC and TABU greedy searching methods to optimize it, respectively. All the experiments are conducted on the previous four synthetic network datasets with 1000 random data instances. (See Chapter 4 for detailed descriptions.) Meanwhile, to reduce the randomness of simulated datasets, we uniformly repeat each accuracy experiment for 50 times.

Table 5.1 illuminates that the accuracies of traditional constraint-based algorithms can be improved by our TSCB strategy and achieve more salient amendments via the combined algorithm. It is noteworthy to point out that the improvement rate of accuracies for traditional methods can even reach more than 10% when the parameter K is tuned to the optimum in our combined algorithm. This result is of great merit, showing that our algorithm combination technique can make tremendous progress from our TSCB strategy.

Likewise, we investigate whether the parameter, the number of clusters K, is robust to the choice of different values. Figure 5.1 displays the accuracy variations with respect to the parameter on the benchmark dataset "alarm" with 20000 instances. Here we use the performance of the conventional GS algorithm as the baseline of our comparison, implement our TSCB strategy to ameliorate its accuracy, and subsequently combine the Hill-Climbing greedy search to further improve the resulting network structure. As revealed by Figure 5.1, the combined algorithm outperforms the baseline GS method under a wide range of the values of the parameter. More significantly, the combined algorithm is able to upgrade the performance of our original TSCB strategy with the

Score-Based Method	Hill-Climbing											
Methods / Datasets	GS	GS (TSCB)	Combine	IAMB	IAMB (TSCB)	Combine	inter-IAMB	inter-IAMB (TSCB)	Combine	MMPC	MMPC (TSCB)	Combine
"asia"	0.8882	0.9052	**0.9805**	0.8929	0.9111	**0.9764**	0.8914	0.9104	**0.9729**	0.8571	0.8571	**0.9775**
"insurance"	0.9263	0.9306	**0.9419**	0.9221	0.9286	**0.9400**	0.9210	0.9264	**0.9389**	0.9259	0.9259	**0.9290**
"alarm"	0.9602	0.9661	**0.9775**	0.9685	0.9715	**0.9788**	0.9689	0.9712	**0.9781**	0.9644	0.9648	**0.9727**
"hepar2"	0.9750	0.9762	**0.9821**	0.9739	0.9744	**0.9818**	0.9746	0.9752	**0.9818**	0.9727	0.9730	**0.9822**
Score-Based Method	TABU Greedy Search											
Methods / Datasets	GS	GS (TSCB)	Combine	IAMB	IAMB (TSCB)	Combine	inter-IAMB	inter-IAMB (TSCB)	Combine	MMPC	MMPC (TSCB)	Combine
"asia"	0.8921	0.9100	**0.9696**	0.8893	0.9075	**0.9621**	0.8900	0.9089	**0.9675**	0.8550	0.8564	**0.9732**
"insurance"	0.9259	0.9304	**0.9488**	0.9214	0.9279	**0.9408**	0.9224	0.9301	**0.9425**	0.9259	0.9259	**0.9330**
"alarm"	0.9595	0.9657	**0.9788**	0.9689	0.9716	**0.9798**	0.9692	0.9720	**0.9811**	0.9645	0.9648	**0.9749**
"hepar2"	0.9755	0.9765	**0.9814**	0.9741	0.9747	**0.9813**	0.9741	0.9746	**0.9808**	0.9730	0.9733	**0.9812**

TABLE 5.1: Accuracy Comparisons of TSCB Strategy With and Without Algorithm Combination.

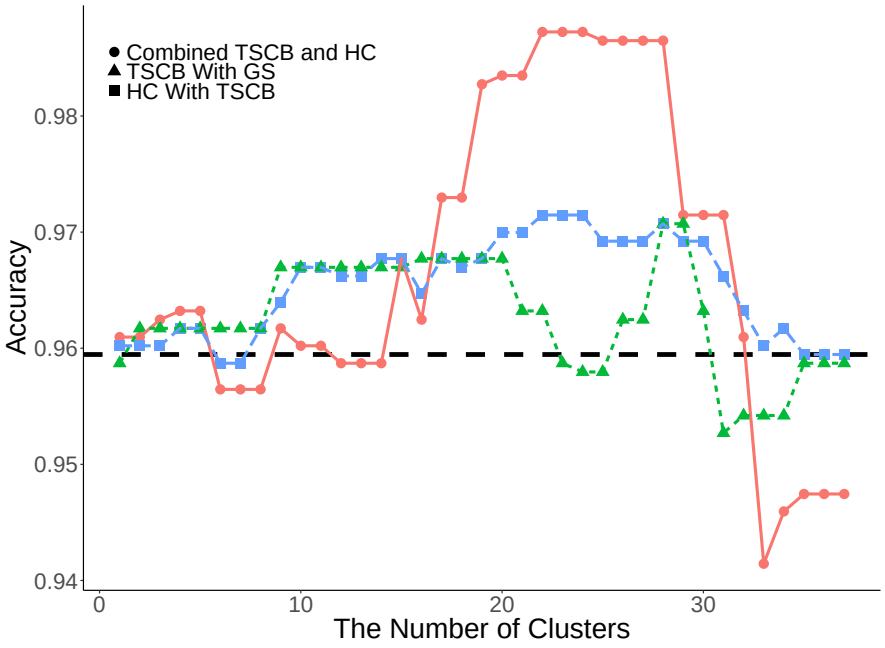


FIGURE 5.1: **Accuracy Variation of the Combined Algorithm With Respect to the Number of Clusters**. The horizontal dashed line indicates the raw accuracy of the embedded GS algorithm. The red line with solid points is the performance of our combined algorithm. The green line with triangular points represents the accuracies of the HC algorithm with our TSCB strategy, while the blue line with square points displays the accuracies of the GS algorithm with our strategy.

optimal choice of the parameter. These accuracy results justify the effectiveness and adaptability of our combined algorithm in various states.

Apart from improving the accuracies of constraint-based, can the combined algorithm exploit any advantage in terms of time efficiency? Unfortunately, there might not be the case. As one can expect, additional implementations of heuristic searching methods expend more running times and consequently increase computational costs. We conduct similar time variation experiments on the benchmark dataset "alarm". As shown in Figure 5.2, our combined algorithm is slightly more time-consuming than the embedded constraint-based method. However, the dramatic improvement on the accuracy level insinuates that it is sometimes worthwhile to sacrifice a little computational efficiency in order to pursue a well-performed and informative network structure.

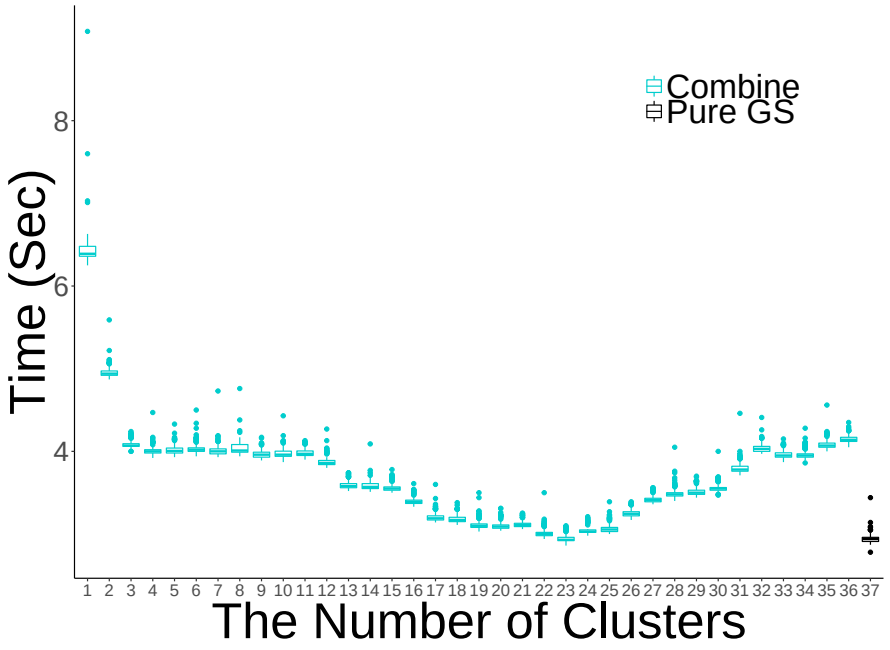


FIGURE 5.2: **Elapsed Time Distributions With Respect to the Number of Clusters**. For each possible value of the parameter, we repeat the learning process for 100 times and plot the boxplots. The rightmost boxplot represents the time distribution for the traditional constraint-based method (GS algorithm).

Chapter 6

Conclusions and Future Research

In this thesis we have proposed a *two-step clustering-based* strategy for Bayesian network structure learning, which can self-generate prior information about the existence of arcs from data. By grouping the variables in the dataset into clusters and learning the network structure within and between clusters, the performance of a wide variety of traditional structure learning algorithms on synthetic benchmark datasets have been improved in terms of accuracy and time efficiency, simultaneously. Moreover, we design a combined algorithm, integrating the constraint-based version of our proposed clustering-based strategy with BIC scoring greedy searching methods. The algorithm combination technique manages to further ameliorate the accuracy performance of constraint-based method, though some sacrifices in computational costs have to be borne.

There are several directions for future research, both theoretically and empirically. As for theoretical research, a rigorous asymptotic analysis of the running time of our TSCB strategy is of great importance, since it tells us whether our strategy would be prohibitively expensive on sufficiently large-scale datasets. In addition, there is an interesting phenomenon in our experiments. We noticed that the optimal state of our TSCB strategy always attains when nearly all the clusters contain no more than three variables, which somewhat 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 motif" [2]. See Figure 6.1 for detailed descriptions. Therefore, in our future work, we are also interested in the physical meaning of each detected cluster, which may give us more insight into the performance improvement.

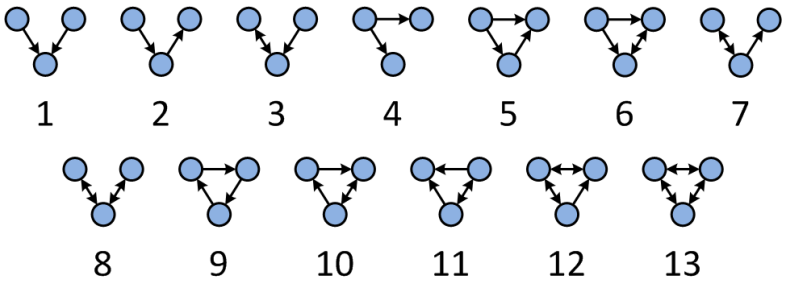


FIGURE 6.1: All 13 Types of Three-Node Connected Subgraphs [2]

Empirically, chances are high that there exist latent variables in real-world datasets. Hence the robustness of our TSCB strategy and algorithm combination technique in the presence of latent variables is also a possible direction for future research. Furthermore, our TSCB strategy and algorithm combination technique aim at learning static Bayesian network structures up to now. Thus, whether our proposed methods are capable to tackle dynamic Bayesian networks, or even more generalized template models, need investigating in the future.

Bibliography

[1] Charles R Twardy, Ann E Nicholson, Kevin B Korb, and John McNeil. Epidemiological Data Mining of Cardiovascular Bayesian Networks. *e-Journal of Health Informatics*, 1(1):e3, 2006.

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

[3] Sewall Wright. Correlation and Causation. *Journal of Agricultural Research*, 20:557–585, 1921.

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

[5] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[6] M. Berkan Sesen, Ann E. Nicholson, Rene Banares-Alcantara, Timor Kadir, and Michael Brady. Bayesian Networks for Clinical Decision Support in Lung Cancer Care. *PLoS ONE*, 8(12):e82349, 2013.

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

[8] Dimitris Margaritis. *Learning Bayesian Network Model Structure from Data*. PhD thesis, Pittsburgh, USA, 2003.

[9] Radhakrishnan Nagarajan, Marco Scutari, and Sophie Lébre. *Bayesian Networks in R with Applications in Systems Biology*, volume 48 of *Use R!* Springer-Verlag New York, 2013.

[10] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

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

[12] David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*, pages 121–130. Springer New York, New York, 1996.

[13] David Maxwell Chickering, Christopher Meek, and David Heckerman. Large-Sample Learning of Bayesian Networks is NP-Hard. 5:12871330, 10 2004.

[14] Nir Friedman, Michal Linial, and Iftach Nachman. Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, 7:601–620, 2000.

[15] Eric Perrier, Seiya Imoto, and Satoru Miyano. Finding Optimal Bayesian Network Given a Super-Structure. *Journal of Machine Learning Research*, 9:2251–2286, 10 2008.

[16] Jun-Gang Xu, Yue Zhao, Jian Chen, and Chao Han. A Structure Learning Algorithm for Bayesian Network Using Prior Knowledge. *Journal of Computer Science and Technology*, 30(4):713–724, Jul 2015.

[17] Andrés Cano, Andrés Masegosa, and Serafín Moral. 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, 06 2011.

[18] Kaname Kojima, Eric Perrier, Seiya Imoto, and Satoru Miyano. Optimal Search on Clustered Structural Constraint for Learning Bayesian Network Structure. *Journal of Machine Learning Research*, 11:285–310, 2010.

[19] Lior Rokach. Ensemble-based Classifiers. *Artificial Intelligence Review*, 33:1–39, 2009.

[20] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[21] Thomas Verma and Judea Pearl. Equivalence and Synthesis of Causal Models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, pages 255–270, New York, NY, USA, 1991. Elsevier Science Inc.

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

[23] Dimitris Margaritis and Sebastian Thrun. Bayesian Network Induction via Local Neighborhoods. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 505–511, 2000.

[24] Ioannis Tsamardinos, Constantin Aliferis, Alexander Statnikov, and Er Statnikov. Algorithms for Large Scale Markov Blanket Discovery. In *Proceedings of The Sixteenth International FLAIRS Conference, St*, pages 376–380. AAAI Press, 2003.

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

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

[27] Pedro Larrañaga, Basilio Sierra, Miren J. Gallego, Maria J. Michelena, and Juan M. Picaza. *Learning Bayesian Networks by Genetic Algorithms: A Case*

Study in the Prediction of Survival in Malignant Skin Melanoma, pages 261–272. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[28] Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander Statnikov. 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, pages 673–678, New York, NY, USA, 2003. ACM.

[29] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The Max-min Hill-climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, Oct 2006.

[30] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Element of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag New York, Second edition, 2009.

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

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

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

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

[35] S. L. Lauritzen and D. J. Spiegelhalter. 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, 1988.

[36] John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning*, 29(2):213–244, Nov 1997.

[37] I. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. 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*, pages 247–256. Springer-Verlag, 1989.

[38] A. Onisko. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*. PhD thesis, Warsaw, 2003.

[39] Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2017. R package version 2.0.6.

[40] Patrick E. Meyer. *infotheo: Information-Theoretic Measures*, 2014. R package version 1.2.0.

[41] Robert E Schapire. The Strength of Weak Learnability. *Machine Learning*, 5:197–227, 06 1990.

[42] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[43] Marco Scutari. bnlearn - an r package for bayesian network learning and inference. `http://www.bnlearn.com/`.

Acknowledgements

Foremost, I would like to express my deep and sincere gratitude to my undergraduate advisor Professor Lixin Yan. It has been an honor to be his student and teaching assistant. Since I attended his lectures, I have benefited incredibly from his broad and profound view in Mathematics, his generosity in nurturing undergraduates, and his invaluable support and encouragement in my graduate school application. I am also thankful to him for hiring me as his teaching assistant. Conducting the introductory lecture of Fourier Analysis and discussing math with younger undergraduates are among the most memorable experiences in my undergraduate study at Sun Yat-sen University.

I am also grateful to Professor Emeritus Xizhi Wu from Renmin University of China, who had kindly guided me through the adventure in the fields of Computer Science and Statistics. I deeply appreciate his warm-heartedness and dedication to pushing me to higher academic stages.

I pay sincere thanks to Professor Jiming Liu and his team from Hong Kong Baptist University, including his research assistant professor Yang Liu and Ph.D. student Qi Tan. Part of the thesis grew out when I participated a summer Ph.D. research program at the Department of Computer Science, Hong Kong Baptist University. Their tremendous supports enabled me to touch the frontier of artificial intelligence. I also acknowledge Professor David Poole from University of British Columbia, for raising a useful question during my talk and pointing out an error in my slides at the W3PHIAI-2018 workshop , and Professor Tuomas Sandholm from Carnegie Mellon University, for correcting my misunderstanding of NP-Hardness in Bayesian network structure learning during the AAAI-18 Fellow Lunch.

Among all my undergraduate friends, I would like to pay special thanks to Chang Su, who bore my low emotional intelligence, guided me out of my frustrating life, and encouraged me to keep making progress in my academic research. In addition, I thank all the people at Sun Yat-sen University and UC Berkeley who listen to my ideas, rectify my misunderstanding, and help me acquire new knowledge.

Lastly and most importantly, I would like to thank my family for all their selfless love, unswerving encouragement, and meticulous care. I am greatly indebted to my grandparents for opening the door of the realm of Mathematics as well as nurturing my passion and determination toward science. I feel beholden to my aunt for pointing out my shortages and giving me helpful suggestions when I came across tough decisions. Finally, I owe a great debt of gratitude to my parents, who predominantly advise and support me in my pursuits. They have created a liberal environment for me to suck up knowledge and encouraged me when I was in failure. Without their sacrifice and deterministic support, I would never make all the achievements in my life. I would like to take up this opportunity to say that I love you all.

Appendix A

Proofs of Theorems

A.1 Decomposition of Likelihood Score

Theorem A.1. *The likelihood score decomposes as follows:*

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(X_i; Pa^{\mathcal{G}}_{X_i}) - M \sum_{i=1}^{n} \mathbf{H}_{\hat{P}}(X_i),$$

where $\mathbf{I}_{\hat{P}}(X, Y)$ *is the mutual information between* X *and* Y*,* $\mathbf{H}_{\hat{P}}(X)$ *is the entropy of* X*, and* M *is the total number of data instances.*

Proof. Let $\mathbf{U_i} = Pa_{X_i}$ and $\mathbf{u_i}$ be its instantiation. We can rewrite the likelihood function by combining all the occurrences of each parameter $\theta_{x_i|\mathbf{u}}$ as

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = \sum_{i=1}^{n} \left[\sum_{\mathbf{u_i} \in Val(Pa^{\mathcal{G}}_{X_i})} \sum_{x_i} M[x_i, \mathbf{u_i}] \log \hat{\theta}_{x_i|\mathbf{u_i}} \right].$$

Note that $M[x_i, \mathbf{u_i}] = M \cdot \hat{P}(x_i, \mathbf{u_i})$ and the MLE $\hat{\theta}_{x_i|\mathbf{u_i}} = \hat{P}(x_i|\mathbf{u_i})$. Then

$$
\begin{aligned}
\text{score}_L(\mathcal{G}:\mathcal{D}) &= \sum_{i=1}^{n} M\left[\frac{1}{M}\sum_{\mathbf{u_i}}\sum_{x_i} M[x_i,\mathbf{u_i}]\log\hat{\theta}_{x_i|\mathbf{u_i}}\right] \\
&= \sum_{i=1}^{n} M\left[\sum_{\mathbf{u_i}}\sum_{x_i}\hat{P}(x_i,\mathbf{u_i})\log\hat{P}(x_i|\mathbf{u_i})\right] \\
&= \sum_{i=1}^{n} M\left[\sum_{\mathbf{u_i}}\sum_{x_i}\hat{P}(x_i,\mathbf{u_i})\log\left(\frac{\hat{P}(x_i,\mathbf{u_i})}{\hat{P}(\mathbf{u_i})}\cdot\frac{\hat{P}(x_i)}{\hat{P}(x_i)}\right)\right] \\
&= \sum_{i=1}^{n} M\left[\sum_{\mathbf{u_i}}\sum_{x_i}\hat{P}(x_i,\mathbf{u_i})\log\left(\frac{\hat{P}(x_i,\mathbf{u_i})}{\hat{P}(\mathbf{u_i})\hat{P}(x_i)}\right)+\sum_{x_i}\left(\sum_{\mathbf{u_i}}\hat{P}(x_i,\mathbf{u_i})\right)\log\hat{P}(x_i)\right] \\
&= M\sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i;\mathbf{U_i}) - M\sum_{i=1}^{n}\sum_{x_i}\hat{P}(x_i)\log\frac{1}{\hat{P}(x_i)} \\
&= M\sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i;\mathbf{U_i}) - M\sum_{i=1}^{n}\mathbf{H}_{\hat{P}}(X_i),
\end{aligned}
$$

where (as implied by Formula 2.1) the mutual information $\mathbf{I}_{\hat{P}}(X_i;Pa^{\mathcal{G}}_{X_i})$ is 0 if $Pa^{\mathcal{G}}_{X_i}=\emptyset$. □

A.2 Consistency of BIC Score

Theorem A.2. *The BIC score is consistent [10].*

Proof. It suffices to prove that for sufficiently large sample size M, if the graph that maximizes the BIC score is $\mathcal{G}$, then $\mathcal{G}$ is I-equivalent to $\mathcal{G}^*$.

First, consider some graph $\mathcal{G}$ that implies an independence assumption that $\mathcal{G}^*$ does not support. Then $\mathcal{G}$ cannot be an I-map of the true underlying distribution P. Hence, $\mathcal{G}$ cannot be a maximum likelihood model with respect to the true distribution P^*, so that we must have:

$$\sum_i \mathbf{I}_{P^*}(X_i;Pa^{\mathcal{G}^*}_{X_i}) > \sum_i \mathbf{I}_{P^*}(X_i;Pa^{\mathcal{G}}_{X_i}).$$

As $M\to\infty$, our empirical distribution $\hat{P}$ will converge to P^* with probability 1. Therefore, for large M,

$$\text{score}_L(\mathcal{G}^*:\mathcal{D}) - \text{score}_L(\mathcal{G}:\mathcal{D}) \approx \Delta\cdot M,$$

where $\Delta = \sum_i \mathbf{I}_{P^*}(X_i; Pa_{X_i}^{\mathcal{G}^*}) > \sum_i \mathbf{I}_{P^*}(X_i; Pa_{X_i}^{\mathcal{G}})$. Therefore, asymtotically we have that

$$\text{score}_{BIC}(\mathcal{G}^* : \mathcal{D}) - \text{score}_{BIC}(\mathcal{G} : \mathcal{D}) \approx \Delta \cdot M + \frac{1}{2}(\text{Dim}[\mathcal{G}] - \text{Dim}[\mathcal{G}^*]) \log M.$$

The first term grows much faster than the second as $M \to \infty$, so that eventually its effect will dominate, and the BIC score of $\mathcal{G}^*$ will be better.

Second, assume that $\mathcal{G}$ implies all the independence assumptions in $\mathcal{G}^*$, but that $\mathcal{G}^*$ implies an independence assumption that $\mathcal{G}$ does not. (In other words, $\mathcal{G}$ is a superset of $\mathcal{G}^*$.) In this case, $\mathcal{G}$ can represent any distribution that $\mathcal{G}^*$ can. In particular, it can represent P^*. As $\hat{P}$ converges to P^*, we will find that:

$$\text{score}_L(\mathcal{G}^* : \mathcal{D}) - \text{score}_L(\mathcal{G} : \mathcal{D}) \to 0.$$

Thus, asymptotically we obtain that

$$\text{score}_{BIC}(\mathcal{G}^* : \mathcal{D}) - \text{score}_{BIC}(\mathcal{G} : \mathcal{D}) \approx \frac{1}{2}(\text{Dim}[\mathcal{G}] - \text{Dim}[\mathcal{G}^*]) \log M.$$

Now, since $\mathcal{G}$ makes fewer independence assumptions than $\mathcal{G}^*$, it must be parameterized by a larger set of parameters, that is, $\text{Dim}[\mathcal{G}] > \text{Dim}[\mathcal{G}^*]$, so that the BIC score metric prefers $\mathcal{G}^*$ to $\mathcal{G}$. □

Appendix B

Supplementary Materials

The code for our *two-step clustering-based strategy* as well as experimental evaluations is available at `https://github.com/zhangyk8/TSCB-strategy`. The related network structure can be downloaded from `http://www.bnlearn.com/bnrepository/` [43].

The published paper, poster, and talk slides associated with this thesis can be found on my personal website (`https://zhangyk8.github.io/`).