Exact Learning of Augmented Naive Bayes Classifiers

Shouta Sugahara

A dissertation submitted in partial satisfaction of the requirements
for the degree of Doctor of Philosophy in Engineering

GRADUATE SCHOOL OF INFORMATICS AND ENGINEERING

THE UNIVERSITY OF ELECTRO-COMMUNICATIONS

MARCH 2023

Exact Learning of Augmented Naive Bayes Classifiers

APPROVED BY SUPERVISORY COMMITTEE:

CHAIRPERSON:	Professor Maomi Ueno
MEMBER:	Professor Yasutada Ohama
MEMBER:	Professor Yoshio Okamoto
MEMBER:	Professor Masakazu Muramatsu
MEMBER:	Associate Professor Hideki Yagi

論文の和文概要

論文題目: Exact Learning of Augmented Naive Bayes Classifiers

氏名: 菅原　聖太

本論では, 目的変数が親変数を持たないAugmented Naive Bayes(ANB)構造を制約とした，生成モデルとしてのBNCの厳密学習手法を提案する．また，全説明変数が分類に影響を及ぼし，全説明変数が目的変数と隣接しているという仮定のもとで，厳密学習したANBは漸近的に真の構造と全く同じ分類確率を表現することを証明する．さらに，本論では制約ベースアプローチの一つであるRecursive Autonomy Identification(RAI)アルゴリズムを用い，CIテストとしてBayes factorを組み込むことで大規模なANBを学習できる手法を提案する．提案手法学習されたANBがパラメータ数を最小にして真の同時確率分布を推定できることを示す．実験により，提案手法の優位性を示す．

Abstract

Earlier studies have shown that classification accuracies of Bayesian networks (BNs) obtained by maximizing the conditional log likelihood (CLL) of a class variable, given the feature variables, were higher than those obtained by maximizing the marginal likelihood (ML). However, differences between the performances of the two scores in the earlier studies may be attributed to the fact that they used approximate learning algorithms, not exact ones. This paper compares the classification accuracies of BNs with approximate learning using CLL to those with exact learning using ML. The results demonstrate that the classification accuracies of BNs obtained by maximizing the ML are higher than those obtained by maximizing the CLL for large data. However, the results also demonstrate that the classification accuracies of exact learning of BNs using the ML are much worse than those of other methods when the sample size is small and the class variable has numerous parents. To resolve the problem, we propose an exact learning of an augmented naive Bayes classifier (ANB), which ensures a class variable with no parents. The proposed method is guaranteed to asymptotically estimate the identical class posterior to that of the exactly learned BN. Comparison experiments demonstrated the superior performance of the proposed method. Nevertheless, exact learning of large ANBs is difficult because it entails an associated NP-hard problem that becomes more difficult as the number of variables increases. Recent reports have described that constraint-based learning methods with Bayes factor achieve larger network structures than the structure achieved using traditional methods. This study proposes an efficient learning algorithm of ANBs using recursive autonomy identification (RAI) with Bayes factor. A

unique benefit of the proposed method is that is guaranteed to accelerate execution of the RAI algorithm when the data follow an ANB model. Numerical experiments were conducted to demonstrate the effectiveness of the proposed method.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Classification contributes to solving real-world problems. The naive Bayes classifier, in which the feature variables are conditionally independent given a class variable, is a popular classifier (Minsky 1961). Initially, the naive Bayes was not expected to provide highly accurate classification because actual data were generated from more complex systems. Therefore, the general Bayesian network (GBN) with learning by marginal likelihood (ML) as a generative model was expected to outperform the naive Bayes, because the GBN is more expressive than the naive Bayes. However, Friedman et al. (1997) demonstrated that the naive Bayes sometimes outperformed the GBN using a greedy search to find the smallest minimum description length (MDL) score, which was originally intended to approximate ML. They explained the inferior performance of the MDL by decomposing the MDL into the log likelihood (LL) term, which reflects the model fitting to training data, and the penalty term, which reflects the model complexity. Moreover, they decomposed the LL term into a conditional log likelihood (CLL) of the class variable given the feature variables, which is directly related to the classification, and a joint LL of the feature variables, which is not directly related to the classification. Furthermore, they proposed conditional MDL (CMDL), a modified MDL replacing the LL with the CLL.

Consequently, Grossman and Domingos (2004) claimed that the Bayesian network (BN) minimizing CMDL as a discriminative model shows better accuracy than that maximizing ML. Unfortunately, the CLL has no closed-form equation for estimating the optimal parameters. This implies that optimizing CLL requires greedy search algorithms for structure learning such as gradient descent algorithms (e.g., extended logistic regression algorithm (Greiner and Zhou 2002)). Nevertheless, the optimization algorithm involves the reiteration of each structure candidate, which renders the method computationally expensive. To avoid searching a structure which minimizes CMDL, Friedman et al. (1997) proposed an augmented naive Bayes classifier (ANB) in which the class variable directly links to all feature variables, and links among feature variables are allowed. ANB ensures that all feature variables can contribute to classification. Later, various types of restricted ANBs were proposed, such as tree-augmented naive Bayes classifiers (TANs) (Friedman et al. 1997) and forest-augmented naive Bayes classifiers (FANs) (Lucas 2004).

Because maximization of CLL entails heavy computation, various approximation methods have been proposed to maximize it. Carvalho et al. (2013) proposed *approximated CLL* (aCLL), which is decomposable and computationally efficient. Grossman and Domingos (2004) proposed BNC2P, which is a greedy learning method with at most two parents per variable using the hill-climbing search by maximizing CLL while estimating parameters by maximizing LL. Mihaljević et al. (2018) proposed MC-DAGGES, which reduces the space for the greedy search of BN Classifiers (BNCs) using the CLL score. These reports described that the BNC maximizing the approximated CLL performed better than that maximizing the approximated ML. Nevertheless, they did not explain why CLL outperformed ML. For large data, the classification accuracies presented by maximizing ML are expected to be comparable to those presented by maximizing CLL because ML has asymptotic consistency. Differences between the performances of the two scores in these studies might depend on their respective learning algorithms; they were approximate learning algorithms, not exact ones.

Recent studies have explored efficient algorithms for the exact learning of GBNs to maximize ML (Koivisto and Sood 2004, Singth and Moore 2005, Silander and

Myllymäki 2006, De Campos and Ji 2011, Malone et al. 2011, Yuan and Malone 2013, Cussens 2012, Barlett and Cussens 2013, Suzuki 2017).

This study compares the classification performances of the BNC with exact learning using ML as a generative model and those with approximate learning using CLL as a discriminative model. The results show that maximizing ML shows better classification accuracy when compared with maximizing CLL for large data. However, the results also show that classification accuracies obtained by exact learning of BNCs using ML are much worse than those obtained by other methods when the sample size is small, and the class variable has numerous parents in the exactly learned networks. When a class variable has numerous parents, estimation of the conditional probability parameters of the class variable becomes unstable because the number of parent configurations becomes large and the sample size for learning the parameters becomes small.

To improve the classification accuracies of BNCs learned by ML, this study proposes an exact learning of ANBs which maximizes ML and ensures that the class variable has no parents. In earlier studies, the ANB constraint was used to learn the BNC as a discriminative model. In contrast, we use the ANB constraint to learn the BNC as a generative model. The proposed method asymptotically learns the optimal ANB, which asymptotically represents the true probability distribution with the fewest parameters among all possible ANB structures. Moreover, the proposed ANB is guaranteed to asymptotically estimate the identical conditional probability of the class variable to that of the exactly learned GBN. Furthermore, learning ANBs has lower computational costs than learning GBNs. Although the main theorem assumes that all feature variables are included in the Markov blanket of the class variable, this assumption does not necessarily hold. To address this problem, we propose a feature selection method using Bayes factor for exact learning of the ANB so as to avoid increasing the computational costs. Comparison experiments show that our method outperforms the other methods.

However, the exact learning of ANBs cannot be applied to network structures with more than 30 variables. In the field of causal models, a more computationally efficient structure learning method has been proposed, although it has no asymptotic matching of the true structure. This method, called the constraint-based approach, learns structure by orienting edges using orientation rules (Pearl 2000) on an undirected graph that is learned by application of the Conditional Independence test (CI test) between two variables to a fully undirected graph. In the study of constraint-based approaches, the PC algorithm (Spirtes et al. 2000), the TPDA algorithm (Cheng et al. 2002), the MMHC algorithm (Tsamardinos et al. 2006), and the RAI algorithm (Yehezkel and Lerner 2009) have been reported. The RAI algorithm is known as an extremely efficient method with this approach. The salient benefit of the RAI algorithm is that it decreases the number of conditional variables of CI tests in the constraint-based approach because it decomposes the entire structure into partial structures based on observed convergence connections. Steck and Jaakkola (2002b) proposed a conditional independence test with an asymptotic consistency, a Bayes factor with BDeu. Abellán et al. (2006) proposed a learning method by application of the CI test with the BDeu score to the PC algorithm. Furthermore, Natori et al. (2017) reported that the RAI algorithm based on the Bayes factor yielded the largest and the most accurate learning results. More recently, researchers challenged to employ constraint-based learning methods with Bayes factor to increase the available learning Bayesian networks size (e.g. Rohekar et al. (2018), Mokhtarian et al. (2021)).

We propose a constraint-based Learning of ANBs using RAI with Bayes factor to learn large ANBs. The proposed method is expected to improve efficiency of the original RAI algorithm without the ANB constraint because the proposed method is guaranteed to accelerate the structure decompositions that occur during the RAI algorithm execution when the data follow an ANB model.

Numerical experiments using benchmark datasets show that the proposed algorithm can learn larger networks than the exact solution search approach can.

Chapter 2

Background

In this chapter, we introduce the notation and background material required for our discussion.

2.1 Bayesian Network

A BN is a graphical model that represents conditional independence among random variables as a directed acyclic graph (DAG). For the discussions presented herein, we call a DAG of BN a *structure* throughout. The BN provides a good approximation of the joint probability distribution because it decomposes the distribution exactly into a product of the conditional probabilities for each variable.

Let $\mathbf{V} = \{X_0, X_1, \ldots, X_n\}$ be a set of discrete variables, where $X_i, i = 0, \ldots, n$, can take values in the set of states $\{1, \ldots, r_i\}$. One can say $X_i = k$ when X_i takes the state k. According to a structure G, the joint probability distribution is represented as

$$P(X_0, X_1, \ldots, X_n \mid G) = \prod_{i=0}^{n} P(X_i \mid \mathbf{Pa}_{X_i}^{G}, G),$$

where $\mathbf{Pa}_{X_i}^{G}$ is a set of parent variables of X_i in G. When the structure G is obvious from the context, we use $\mathbf{Pa}_{X_i}$ to denote the parents. In addition, $q^{\mathbf{Pa}_{X_i}}$ denotes the number of possible patterns of states of variables in $\mathbf{Pa}_{X_i}$, i.e., $q^{\mathbf{Pa}_{X_i}} = \prod_{v:X_v \in \mathbf{Pa}_{X_i}} r_v$. We assign numbers $1, \ldots, q^{\mathbf{Pa}_{X_i}}$ to the respective patterns of states of variables in

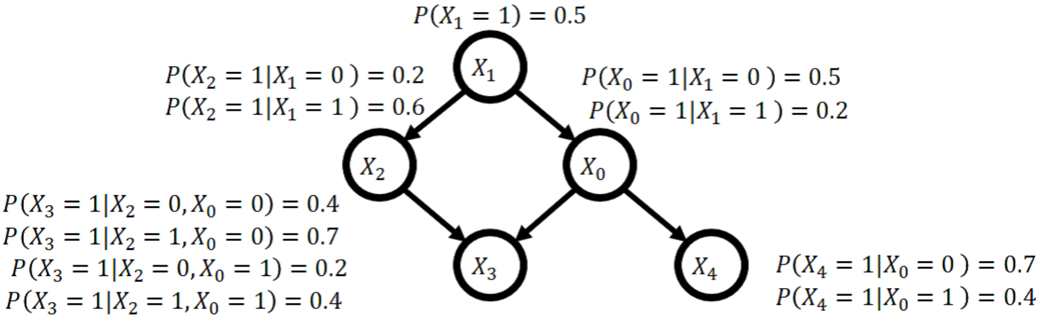
$P(X_1 = 1) = 0.5$
$P(X_2 = 1|X_1 = 0) = 0.2$
$P(X_2 = 1|X_1 = 1) = 0.6$
$P(X_0 = 1|X_1 = 0) = 0.5$
$P(X_0 = 1|X_1 = 1) = 0.2$
X_1
X_2
X_0
X_3
X_4
$P(X_3 = 1|X_2 = 0, X_0 = 0) = 0.4$
$P(X_3 = 1|X_2 = 1, X_0 = 0) = 0.7$
$P(X_3 = 1|X_2 = 0, X_0 = 1) = 0.2$
$P(X_3 = 1|X_2 = 1, X_0 = 1) = 0.4$
$P(X_4 = 1|X_0 = 0) = 0.7$
$P(X_4 = 1|X_0 = 1) = 0.4$

Figure 2.1: Example of a Bayesian network.

$\mathbf{Pa}_{X_i}$. When variables in $\mathbf{Pa}_{X_i}$ take the pattern j, we write $\mathbf{Pa}_{X_i}$ takes the state j. Let θ_{ijk} be a conditional probability parameter of $X_i = k$ when $\mathbf{Pa}_{X_i}$ takes the state j. Then, we define $\Theta_{ij} = \bigcup_{k=1}^{r_i}\{\theta_{ijk}\}, \Theta = \bigcup_{i=0}^{n}\bigcup_{j=1}^{q^{\mathbf{Pa}_{X_i}}}\{\Theta_{ij}\}$. A BN is a pair $B = (G, \Theta)$. Figure 2.1 depicts an example of a Bayesian network.

A structure of BN represents conditional independence assertions in the probability distribution by *d-separation*. First, we define *collider*, for which we need to define the d-separation. We designate a sequence of distinct variables, each one adjacent to the next, a *path*. Then the collider is defined as shown below.

Definition 1. For any structure G consisting of a variable set $\mathbf{V}$ and for any path ρ in G, a variable $Z \in \mathbf{V}$ on ρ is a collider if and only if Z has two parent variables which are adjacent to Z on ρ.

We then define “d-separated” as explained below.

Definition 2. For any structure G consisting of a variable set $\mathbf{V}$ and for any $X, Y \in \mathbf{V}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, the two variables X and Y are d-separated, given $\mathbf{Z}$ in G, if and only if every path ρ between X and Y satisfies either of the following two conditions.

- $\mathbf{Z}$ includes a non-collider on ρ.
- There is a collider Z on ρ; $\mathbf{Z}$ does not include Z and its descendants.

We write $Dsep_G(X, Y \mid \mathbf{Z})$ to denote that X and Y are d-separated given $\mathbf{Z}$ in G (We designate $Dsep_G(X, Y \mid \mathbf{Z})$ d-separation between X and Y given $\mathbf{Z}$ in G). We write $\neg Dsep_G(X, Y \mid \mathbf{Z})$ to denote that X and Y are d-connected given $\mathbf{Z}$ in G (We designate $\neg Dsep_G(X, Y \mid \mathbf{Z})$ d-connection).

If we have $X, Y, Z \in \mathbf{V}$ and X and Y are not adjacent, then the following three possible types of connections characterize the d-separations: serial connections such as $X \to Z \to Y$, divergence connections such as $X \leftarrow Z \to Y$, and convergence connections such as $X \to Z \leftarrow Y$. The following theorem of d-separations for these connections holds.

Theorem 1. (Koller and Friedman (2009))
First, assume a structure $G = (\mathbf{V}, \mathbf{E})$, $X, Y, Z \in \mathbf{V}$. If G has a convergence connection $X \to Z \leftarrow Y$, then the following two propositions hold:

- $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z\}, \neg Dsep_G(X, Y \mid \mathbf{Z}, Z)$,
- $\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z\}, Dsep_G(X, Y \mid \mathbf{Z})$.

If G has a serial connection $X \to Z \to Y$ or divergence connection $X \leftarrow Z \to Y$, then negations of the above two propositions hold.

Two DAGs are *Markov equivalent* when they have the same d-separations.

Definition 3. Let G_1 and G_2 be two DAGs consisting of a variable set $\mathbf{V}$; then G_1 and G_2 are called Markov equivalent if the following holds:

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}, \quad Dsep_{G_1}(X, Y \mid \mathbf{Z}) \Leftrightarrow Dsep_{G_2}(X, Y \mid \mathbf{Z}). \tag{2.1}$$

Verma and Pearl (1990) described the following theorem to identify Markov equivalence.

Theorem 2. (Verma and Pearl (1990))
Two DAGs are Markov equivalent if and only if they have identical links (edges without direction) and identical convergence connections.

Let $I_{P^*}(X, Y \mid \mathbf{Z})$ denote that X and Y are conditionally independent given $\mathbf{Z}$ in the true joint probability distribution (the underlying distribution) P^*. A structure G is an *independence map (I-map)* if all the d-separations in G are entailed by conditional independences in P^*:

Definition 4. For any structure G consisting of a variable set $\mathbf{V}$, G is an I-map if the following proposition holds:

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}, Dsep_G(X, Y \mid \mathbf{Z}) \Rightarrow I_{P^*}(X, Y \mid \mathbf{Z}).$$

We introduce the following notations required for our discussion on learning BNs. Let $D = \{\mathbf{x}^1, \dots, \mathbf{x}^d, \dots, \mathbf{x}^N\}$ be a complete dataset consisting of N i.i.d. instances, where each instance $\mathbf{x}^d$ is a data-vector $(x_0^d, x_1^d, \dots, x_n^d)$. For a variable set $\mathbf{Z} \subseteq \mathbf{V}$, we define $N_j^{\mathbf{Z}}$ as the number of samples when $\mathbf{Z}$ takes a state j in the dataset D, and define $N_{ijk}^{\mathbf{Z}}$ as the number of samples of $X_i = k$ when $\mathbf{Z}$ takes a state j in D. In addition, we define a joint frequency table $JFT_D(\mathbf{Z})$ as a list of $N_j^{\mathbf{Z}}$ for $j = 1, \dots, q^{\mathbf{Z}}$. For a variable $X \in \mathbf{V}$, we define a conditional frequency table $CFT_D(X, \mathbf{Z})$. For example, $CFT_D(X_i, \mathbf{Z})$ is a list of $N_{ijk}^{\mathbf{Z}}$ for $j = 1, \dots, q^{\mathbf{Z}}$, and $k = 1, \dots, r_i$.

The most popular parameter estimator of BNs is the *expected a posteriori* (EAP) of Equation (2.2), which is the expectation of θ_{ijk} with respect to the density $p(\Theta_{ij} \mid D, G)$ of Equation (2.3), assuming Dirichlet prior density $p(\Theta_{ij} \mid G)$ of Equation (2.4).

$$\hat{\theta}_{ijk} = E(\theta_{ijk} \mid D, G) = \int \theta_{ijk} \cdot p(\Theta_{ij} \mid D, G) d\Theta_{ij} = \frac{N'_{ijk} + N_{ijk}^{\mathbf{Pa}_{X_i}}}{N'_{ij} + N_j^{\mathbf{Pa}_{X_i}}}. \tag{2.2}$$

$$p(\Theta_{ij} \mid D, G) = \frac{\Gamma(\sum_{k=1}^{r_i}(N'_{ijk} + N_{ijk}^{\mathbf{Pa}_{X_i}}))}{\prod_{k=1}^{r_i} \Gamma(N'_{ijk} + N_{ijk}^{\mathbf{Pa}_{X_i}})} \prod_{k=1}^{r_i} \theta_{ijk}^{N'_{ijk} + N_{ijk}^{\mathbf{Pa}_{X_i}} - 1}. \tag{2.3}$$

$$p(\Theta_{ij} \mid G) = \frac{\Gamma(\sum_{k=1}^{r_i} N'_{ijk})}{\prod_{k=1}^{r_i} \Gamma(N'_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{N'_{ijk} - 1}. \tag{2.4}$$

In Equations (2.2) through (2.4), N'_{ijk} denotes the hyperparameters of the Dirichlet prior distributions, with $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$. In addition, for every positive real number x, $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

The structure must be estimated from observed data because it is generally unknown. This problem is called "structure learning." A goal of the structure learning is to obtain the I-map with the fewest parameters. The number of parameters of a structure G consisting of a variable set $\mathbf{V}$ is represented as $\sum_{i=0}^{n} q^{\mathbf{Pa}_{X_i}}(r_i - 1)$. The most common learning approach is a score-based approach, which seeks the best structure maximizing a score function $Score(G, D)$. Seeking the best structure among all the possible structures consisting of $\mathbf{V}$ is designated as "exact learning." To learn the I-map with the fewest parameters, we maximize the score with an *asymptotic consistency* defined as shown below.

Definition 5. (Chickering (2002))
Let G_1 and G_2 be two structures consisting of a variable set $\mathbf{V}$. A score function *Score* has an *asymptotic consistency* if the following two properties almost surely hold when the sample size of D is sufficiently large.

- If G_1 is an I-map and G_2 is not an I-map, then $Score(G_1, D) > Score(G_2, D)$.
- If G_1 and G_2 both are I-maps, and if G_1 has fewer parameters than G_2, then $Score(G_1, D) > Score(G_2, D)$.

The marginal likelihood (ML), $P(D \mid G)$, is known to have asymptotic consistency (Chickering 2002). Moreover, the ML score has the following *asymptotic local consistency* (Chickering 2002).

Definition 6. (Chickering (2002))
Let G_1 be any structure consisting of a variable set $\mathbf{V}$, and let G_2 be the structure that results from adding the edge $Y \rightarrow X$ to G_1. A score function *Score* has an asymptotic local consistency if the following two properties almost surely hold when the sample size is sufficiently large.

- $I_{P^*}(X, Y \mid \mathbf{Pa}_X^{G_1}) \Rightarrow Score(G_1) > Score(G_2)$.
- $\neg I_{P^*}(X, Y \mid \mathbf{Pa}_X^{G_1}) \Rightarrow Score(G_1) < Score(G_2)$.

When we assume the Dirichlet prior density of Equation (2.4), ML is represented as

$$P(D \mid G) = \prod_{i=0}^{n} \prod_{j=1}^{q^{\mathbf{Pa}_{X_i}}} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_j^{\mathbf{Pa}_{X_i}})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk}^{\mathbf{Pa}_{X_i}})}{\Gamma(N'_{ijk})}.$$

In particular, Heckerman et al. (1995) presented the following constraint related to hyperparameters N'_{ijk} for ML satisfying the *score-equivalence assumption*, where the ML takes the same value for the Markov equivalent structures:

$$N'_{ijk} = N'P(X_i = k, \mathbf{Pa}_{X_i} = j \mid G^h),$$

where N' is a equivalent sample size (ESS) determined by users, and G^h is a hypothetical structure that reflects the prior knowledge of users. ML with the above constraint of N'_{ijk} is designated as the *Bayesian Dirichlet equivalent* (BDe) score. As Buntine (1991) described, $N'_{ijk} = N'/(r_i q^{\mathbf{Pa}_{X_i}})$ is regarded as a special case of the BDe score. Heckerman et al. (1995) called this special case the *Bayesian Dirichlet equivalent uniform* (BDeu), defined as

$$P(D \mid G) = \prod_{i=0}^{n} \prod_{j=1}^{q^{\mathbf{Pa}_{X_i}}} \frac{\Gamma(N'/q^{\mathbf{Pa}_{X_i}})}{\Gamma(N'/q^{\mathbf{Pa}_{X_i}} + N_j^{\mathbf{Pa}_{X_i}})} \prod_{k=1}^{r_i} \frac{\Gamma(N'/(r_i q^{\mathbf{Pa}_{X_i}}) + N_{ijk}^{\mathbf{Pa}_{X_i}})}{\Gamma(N'/(r_i q^{\mathbf{Pa}_{X_i}}))}.$$

In addition, the *minimum description length* (MDL) score presented in (5), which approximates the negative logarithm of ML, is often used for learning BNs.

$$MDL(B \mid D) = \frac{\log N}{2} \sum_{i=0}^{n} q^{\mathbf{Pa}_{X_i}}(r_i - 1) - \sum_{d=1}^{N} \log P(x_0^d, x_1^d, \ldots, x_n^d \mid B). \qquad (2.5)$$

The first term of Equation (2.5) is the penalty term, which signifies the model complexity. The second term, LL, is the fitting term that reflects the degree of model fitting to the training data.

Both BDeu and MDL are *decomposable*, i.e., the scores can be expressed as a sum of *local scores* depending only on the conditional frequency table for one variable and its parents as follows.

$$Score(G, D) = \sum_{i=0}^{n} LocalScore(CFT_D(X_i, \mathbf{Pa}_{X_i})),$$

For example, the local score of log BDeu for $CFT_D(X_i, \mathbf{Pa}_{X_i})$ is

$$
\begin{aligned}
&\mathit{LocalScore}(CFT_D(X_i, \mathbf{Pa}_{X_i})) \\
&\quad = \sum_{j=1}^{q^{\mathbf{Pa}_{X_i}}} \left(\log \frac{\Gamma(N'/q^{\mathbf{Pa}_{X_i}})}{\Gamma(N'/q^{\mathbf{Pa}_{X_i}} + N_j^{\mathbf{Pa}_{X_i}})} \sum_{k=1}^{r_i} \log \frac{\Gamma(N'/(r_i q^{\mathbf{Pa}_{X_i}}) + N_{ijk}^{\mathbf{Pa}_{X_i}})}{\Gamma(N'/(r_i q^{\mathbf{Pa}_{X_i}}))} \right).
\end{aligned}
\tag{2.6}
$$

The decomposable score enables an extremely efficient search for structures (Silander and Myllymäki 2006, Barlett and Cussens 2013).

2.2 Bayesian Network Classifiers

A Bayesian network classifier (BNC) can be interpreted as a BN for which X_0 is the class variable and $X_1, \ldots, X_n$ are feature variables. Given an instance $\mathbf{x} = (x_1, \ldots, x_n)$ for feature variables $X_1, \ldots, X_n$, the BNC B infers class c by maximizing the posterior probability of X_0 as

$$
\begin{aligned}
\hat{c} &\in \operatorname*{argmax}_{c \in \{1,\ldots,r_0\}} P(c \mid x_1, \ldots, x_n, B) \qquad (2.7) \\
&= \operatorname*{argmax}_{c \in \{1,\ldots,r_0\}} \prod_{i=0}^{n} \prod_{j=1}^{q^{\mathbf{Pa}_i}} \prod_{k=1}^{r_i} (\theta_{ijk})^{1_{ijk}} \\
&= \operatorname*{argmax}_{c \in \{1,\ldots,r_0\}} \prod_{j=1}^{q^{\mathbf{Pa}_{X_0}}} \prod_{k=1}^{r_0} (\theta_{0jk})^{1_{0jk}} \times \prod_{i:X_i \in \mathbf{C}} \prod_{j=1}^{q^{\mathbf{Pa}_{X_0}}} \prod_{k=1}^{r_i} (\theta_{ijk})^{1_{ijk}},
\end{aligned}
$$

where $1_{ijk} = 1$ if $X_i = k$ and $\mathbf{Pa}_{X_i}$ takes a state j in $\mathbf{x}$, and $1_{ijk} = 0$ otherwise. Furthermore, $\mathbf{C}$ is the set of children of the class variable X_0. From Equation (2.7), we can infer class c given only the values of the parents of X_0, the children of X_0, and the parents of the children of X_0. A set of these feature variables is called a *Markov blanket* of X_0.

However, Friedman et al. (1997) reported that BNC minimizing MDL cannot optimize classification performance. They proposed the sole use of the following CLL of the class variable given feature variables, instead of the LL for learning BNC structures.

$$
\begin{aligned}
CLL(B \mid D) &= \sum_{d=1}^{N} \log P(x_0^d \mid x_1^d, \ldots, x_n^d, B) \\
&= \sum_{d=1}^{N} \log P(x_0^d, x_1^d, \ldots, x_n^d \mid B) - \sum_{d=1}^{N} \log \sum_{c=1}^{r_0} P(c, x_1^d, \ldots, x_n^d \mid B). \quad (2.8)
\end{aligned}
$$

Furthermore, they proposed conditional MDL (CMDL), which is a modified MDL replacing LL with CLL, as shown below.

$$CMDL(B \mid D) = \frac{\log N}{2} \sum_{i=0}^{n} q^{\mathbf{Pa}_{X_i}} (r_i - 1) - CLL(B \mid D).$$

Consequently, they claimed that the BN minimizing CMDL as a discriminative model showed better accuracy than that maximizing ML as a generative model.

Unfortunately, CLL is not decomposable because we cannot describe the second term of Equation (2.8) as a sum of the log parameters in Θ. This finding implies that no closed-form equation exists for the maximum CLL estimator for Θ. Therefore, learning the network structure that minimizes the CMDL requires a search method such as gradient descent over the space of parameters for each structure candidate. Therefore, exact learning of structures by minimizing CMDL is computationally infeasible.

As a simple way of resolving that difficulty, Friedman et al. (1997) proposed an augmented naive Bayes classifier (ANB) that ensures an edge from the class variable to each feature variable and allows edges among feature variables. Furthermore, they proposed a tree-augmented naive Bayes classifier (TAN) in which the class variable has no parents and each feature variable has a class variable and at most one other feature variable as parent variables.

Various approximate methods to maximize CLL have been proposed. Carvalho et al. (2013) proposed an aCLL score, which is decomposable and computationally efficient. Let G_{ANB} be an ANB structure. In addition, let N_{ijck} be the number of samples of $X_i = k$ when $X_0 = c$ and $\mathbf{Pa}_{X_i} \setminus \{X_0\}$ takes the state j, $(i = 1, \ldots, n; j = 1, \ldots, q^{\mathbf{Pa}_{X_i} \setminus \{X_0\}}; c = 1, \ldots, r_0; k = 1, \ldots, r_i)$. In addition, let $N'' > 0$ represent hyperparameters. Under several assumptions, aCLL can be represented as

$$aCLL(G_{ANB} \mid D) \propto \sum_{i=1}^{n} \sum_{j=1}^{q^{\mathbf{Pa}_{X_i} \setminus \{X_0\}}} \sum_{k=1}^{r_i} \sum_{c=1}^{r_0} \left(N_{ijck} + \beta \sum_{c'=1}^{r_0} N_{ijc'k} \right) \log \frac{N_{ij+ck}}{N_{ij+c}},$$

where

$$N_{ij+ck} = \begin{cases} N_{ijck} + \beta \sum_{c'=1}^{r_0} N_{ijc'k} & \mathit{if}\ N_{ijck} + \beta \sum_{c'=1}^{r_0} N_{ijc'k} \geq N'' \\ N'' & \mathit{otherwise}, \end{cases}$$

$$N_{ij+c} = \sum_{k=1}^{r_i} N_{ij+ck}.$$

The value of β is found by using the Monte Carlo method to approximate CLL. There exists a value of β such that aCLL becomes a minimum-variance unbiased approximation of the CLL.

Moreover, Grossman and Domingos (2004) proposed a learning structure method using a greedy hill-climbing algorithm (Heckerman et al. 1995) by maximizing the CLL while estimating the parameters by maximizing the LL. Recently, Mihaljević et al. (2018) identified the smallest subspace of DAGs that covered all possible class-posterior distributions when the data were complete. All the DAGs in this space, which they call *minimal class-focused* DAGs (MC-DAGs), are such that every edge is directed toward a child of the class variable. In addition, they proposed a greedy search algorithm in the space of Markov equivalent classes of MC-DAGs using the CLL score. These reports described that the BNC maximizing the approximated CLL provides better performance than that maximizing the approximated ML. However, they did not explain why CLL outperformed ML. For large data, the classification accuracies obtained by maximizing ML are expected to be comparable to those obtained by maximizing CLL because ML has asymptotic consistency. Differences between the performances of the two scores in these earlier studies might depend on their learning algorithms to maximize ML; they were approximate learning algorithms, not exact ones.

Chapter 3

Classification Accuracies of Exact Learning of GBNs

This chapter presents experiments comparing the classification accuracies of the exactly learned GBN by maximizing BDeu as a generative model with those of the approximately learned BNC by maximizing CLL as a discriminative model. Although determining the hyperparameter N' of BDeu is difficult (Silander et al. 2007, Steck 2008, Ueno 2008, Suzuki 2017), we use $N' = 1.0$ that allows the data to reflect the estimated parameters to the greatest degree possible (Ueno 2010, 2011).

The experiment compares the respective classification accuracies of seven methods in Table 3.1. All the methods are implemented in Java. The source code is available at `http://www.ai.lab.uec.ac.jp/software/`. Throughout this paper, our experiments are conducted on a computational environment in Table 3.2. This experiment uses 43 classification benchmark datasets from the *UCI repository* (Lichman 2013). Continuous variables are discretized into two bins using the median value as the cut-off, as in (De Campos et al. 2014). In addition, data with missing values are removed from the datasets. We use EAP estimators as conditional probability parameters of the respective classifiers. Hyperparameters N'_{ijk} of EAP are found to be $1/(r_i q^{\mathbf{Pa}_{X_i}})$. Through our experiments, we define "small datasets" as the datasets

Table 3.1: Seven methods compared in the experiments.

Abbreviation	Methods
Naive Bayes	Navie Bayes classifier.
GBN-BDeu	Exact learning of GBNs by maximizing BDeu.
GBN-CMDL (Grossman and Domingos 2004)	Greedy learning GBN method using the hill-climbing search by minimizing CMDL while estimating parameters by maximizing LL.
BNC2P (Grossman and Domingos 2004)	Greedy learning method with at most two parents per variable using the hill-climbing search by maximizing CLL while estimating parameters by maximizing LL.
TAN-aCLL (Carvalho et al. 2013)	Exact learning of TANs by maximizing aCLL.
gGBN-BDeu	Greedy learning GBN method using hill-climbing by maximizing BDeu.
MC-DAGGES (Mihaljević et al. 2018)	Greedy learning method in the space of the Markov equivalent classes of MC-DAGs using the greedy equivalence search (Chickering 2002) by maximizing CLL while estimating parameters by maximizing LL.

Table 3.2: Computational environment.

CPU	2.2 GHz XEON 10-core processor
System Memory	128 GB
OS	Windows 10
Software	Java

with less than 200 samples, and define “large datasets” as the datasets with 10,000 or more samples.

Table 3.3 presents the classification accuracies of the respective classifiers. We will discuss the results of *ANB-BDeu* and *fsANB-BDeu* in a later chapter. The values shown in bold in Table 3.3 represent the best classification accuracies for each dataset. Here, the classification accuracies represent the average percentage of correct classifications from a ten-fold cross-validation. Moreover, to investigate the relation between the classification accuracies and *GBN-BDeu*, Table 3.4 presents the details of the achieved structures using *GBN-BDeu*. “Parents” in Table 3.4 represents the average number of parents of the class variable in the structures learned by *GBN-BDeu*. “Children” denotes the average number of children of the class variable in the structures learned by *GBN-BDeu*. “Sparse data” denotes the average number of value patterns j of the parents of X_0 with null data, $N_j^{\mathbf{Pa}_{X_0}} = 0$ $(j = 1, \ldots, q^{\mathbf{Pa}_{X_0}})$ in the structures learned by *GBN-BDeu*.

From Table 3.3, *GBN-BDeu* shows the best classification accuracies among the methods for large data, such as dataset Nos 22, 29, and 33. From the asymptotic consistency of BDeu, *GBN-BDeu* almost surely converges to an I-map with the fewest parameters. The joint probability distribution represented by an I-map approaches the true distribution as the sample size increases. However, it is worth noting that *GBN-BDeu* provides much worse accuracy than the other methods in datasets No. 3 and No. 9. In these datasets, the learned class variables by *GBN-BDeu* have no children. Numerous parents are shown in “Parents” and “Children” in Table 3.4. When a class variable has numerous parents, the estimation of the

Table 3.3: Classification accuracies of *GBN-BDeu*, *ANB-BDeu*, *fsANB-BDeu*, and traditional methods (bold text signifies the highest accuracy).

No.	Dataset	Variables	Classes	Sample size	Naive-Bayes	GBN-CMDL	BNC2P	TAN-aCLL	gGBN-BDeu	MC-DAG GES	GBN-BDeu	ANB-BDeu	fsANB-BDeu
1	Balance Scale	5	3	625	**0.9152**	0.3333	0.8560	0.8656	**0.9152**	0.7432	**0.9152**	**0.9152**	**0.9152**
2	banknote authentication	5	2	1372	0.8433	**0.8819**	0.8797	0.8761	**0.8819**	0.8768	0.8812	0.8812	0.8812
3	Hayes–Roth	5	3	132	0.8182	0.6136	0.6894	0.6742	0.7525	0.6970	0.6136	0.8182	**0.8333**
4	iris	5	3	150	0.7133	0.7800	0.8200	0.8200	0.8133	0.7800	**0.8267**	0.8200	0.8200
5	lenses	5	3	24	0.7500	0.8333	0.6667	0.7083	0.8333	0.8333	0.8333	0.7500	**0.8750**
6	Car Evaluation	7	4	1728	0.8571	**0.9497**	0.9416	0.9433	0.9416	0.9126	0.9416	0.9427	0.9416
7	liver	7	2	345	0.6319	0.6145	0.6290	**0.6609**	0.6029	0.6435	0.6087	0.6348	0.6377
8	MONK's Problems	7	2	432	0.7500	**1.0000**	**1.0000**	**1.0000**	0.8449	**1.0000**	**1.0000**	**1.0000**	**1.0000**
9	mux6	7	2	64	0.5469	0.3750	0.5625	0.4688	0.4063	**0.7656**	0.4531	0.5469	0.5547
10	LED7	8	10	3200	0.7294	0.7366	**0.7375**	0.7350	0.7297	0.7331	0.7294	0.7294	0.7294
11	HTRU2	9	2	17898	0.7031	0.7096	0.7070	0.7018	0.7188	0.7214	**0.7305**	0.7188	0.7161
12	Nursery	9	5	12960	0.6782	**0.7126**	0.6092	0.5862	**0.7126**	0.6322	**0.7126**	0.6782	**0.7126**
13	pima	9	2	768	0.8966	0.9086	0.9118	0.9130	0.9092	0.9093	0.9112	**0.9141**	0.**9141**
14	post	9	3	87	0.9033	0.5823	**0.9442**	0.9177	0.9291	0.9046	0.9340	0.9181	0.9177
15	Breast Cancer	10	2	277	**0.9751**	0.8917	0.9473	0.9488	0.7058	0.6354	**0.9751**	**0.9751**	**0.9751**
16	Breast Cancer Wisconsin	10	2	683	0.7401	0.6209	0.6823	0.7184	0.7094	**0.9780**	0.7184	0.7040	0.7473
17	Contraceptive Method Choice	10	3	1473	0.4671	0.4501	**0.4745**	0.4705	0.4440	0.4576	0.4542	0.4650	0.4725
18	glass	10	6	214	0.5561	0.5654	0.5794	0.6308	0.4626	0.5888	0.5701	**0.6449**	0.5888
19	shuttle-small	10	6	5800	0.9384	0.9660	0.9703	0.9583	0.9683	0.9586	0.9693	**0.9716**	0.9695
20	threeOf9	10	2	512	0.8164	**0.9434**	0.8691	0.8828	0.8652	0.8750	0.8887	0.8730	0.8633
21	Tic-Tac-Toe	10	2	958	0.6921	**0.8841**	0.7338	0.7203	0.6754	0.7557	0.8340	0.8497	0.8570
22	MAGIC Gamma Telescope	11	2	19020	0.7482	0.7849	0.7806	0.7631	0.7844	0.7781	0.7873	**0.7874**	0.7865
23	Solar Flare	11	9	1389	0.7811	0.8265	0.8315	0.8229	**0.8431**	0.8013	**0.8431**	0.8229	0.8373
24	heart	14	2	270	0.8259	0.8185	0.8037	0.8148	0.8222	**0.8333**	0.8259	0.8185	0.8296
25	wine	14	3	178	0.9270	**0.9438**	0.9157	0.9326	0.9045	**0.9438**	0.9270	0.9270	0.9270
26	cleve	14	2	296	0.8412	0.8209	0.8007	**0.8378**	0.7973	0.8041	0.7973	0.8277	0.8243
27	Australian	15	2	690	0.8290	0.8312	0.8348	0.8464	0.8420	0.8406	**0.8536**	0.8246	0.8522
28	crx	15	2	653	0.8377	0.8346	0.8208	0.8560	**0.8622**	0.8576	0.8591	0.8515	0.8591
29	EEG	15	2	14980	0.5778	0.6787	0.6374	0.6125	0.6732	0.6182	0.6814	**0.6864**	**0.6864**
30	Congressional Voting Records	17	2	232	0.9095	0.9698	0.9612	0.9181	**0.9741**	0.9009	0.9655	0.9483	0.9397
31	zoo	17	5	101	0.9802	0.9109	0.9505	1.0000	0.9505	0.9802	0.9307	0.9505	0.9604
32	pendigits	17	10	10992	0.8032	0.9062	0.8719	0.8700	0.9253	0.8359	**0.9290**	0.9279	0.9279
33	letter	17	26	20000	0.4466	0.5796	0.5132	0.5093	0.5761	0.4664	0.5761	**0.5935**	0.5881
34	ClimateModel	19	2	540	0.9222	**0.9407**	0.9241	0.9333	0.9370	0.9296	0.9000	0.8426	0.9278
35	Image Segmentation	19	7	2310	0.7290	0.7918	0.7991	0.7407	0.8026	0.7476	0.8156	**0.8225**	**0.8225**
36	lymphography	19	4	148	**0.8446**	0.7939	0.7973	0.8311	0.7905	0.8041	0.7500	0.7770	0.7838
37	vehicle	19	4	846	0.4350	0.5910	0.5910	0.5816	0.5461	0.5414	0.5768	**0.6253**	0.6217
38	hepatitis	20	2	80	0.8500	0.7375	**0.8875**	0.8750	0.8500	**0.8875**	0.5875	0.6250	0.8375
39	German	21	2	1000	0.7430	0.6110	0.7340	**0.7470**	0.7140	0.7180	0.7210	0.7380	0.7410
40	bank	21	2	30488	0.8544	0.8618	0.8928	0.8618	0.8952	0.8708	**0.8956**	0.8950	0.8953
41	waveform-21	22	3	5000	0.7886	0.7862	0.7754	0.7896	0.7698	0.7926	0.7846	0.7966	**0.7972**
42	Mushroom	22	2	5644	0.9957	**1.0000**	**1.0000**	0.9995	**1.0000**	0.9986	0.9949	**1.0000**	**1.0000**
43	spect	23	2	263	0.7940	0.7940	0.7903	0.8090	0.7603	0.8052	0.7378	**0.8240**	**0.8240**
	Arithmetic average				0.7764	0.7721	0.7936	0.7943	0.7867	0.7944	0.7963	0.8061	**0.8184**
	p-value (*ANB-BDeu* vs. the other methods)				0.00308	0.04136	0.00672	0.05614	0.06876	0.06010	0.22628	-	-
	p-value (*fsANB-BDeu* vs. the other methods)				0.00001	0.00014	0.00013	0.00280	0.00015	0.00212	0.00064	0.01101	-

Table 3.4: Statistical summary of *GBN-BDeu* and *fsANB-BDeu*.

			Sample			Sparse		Max	Removed
No.	Variables	Classes	size	Parents	Children	data	MB size	parents	variables
1	5	3	625	0.4	3.6	0.0	4.0	1.0	0.0
2	5	2	1372	0.0	2.0	0.0	4.0	4.0	0.0
3	5	3	132	3.0	0.0	17.2	3.0	1.0	1.0
4	5	3	150	1.8	1.2	0.0	3.0	2.0	0.0
5	5	3	24	1.1	1.0	0.0	2.1	1.1	2.0
6	7	4	1728	2.0	3.0	0.0	5.0	2.0	1.0
7	7	2	345	0.0	1.9	0.0	3.4	2.0	0.1
8	7	2	432	3.0	0.0	0.0	3.0	3.0	0.0
9	7	2	64	5.8	0.0	5.2	5.8	1.0	2.1
10	8	10	3200	0.9	6.1	0.0	7.0	1.0	0.0
11	9	2	17898	1.8	4.2	0.0	4.2	2.0	0.9
12	9	5	12960	4.0	3.0	0.0	0.0	0.0	8.0
13	9	2	768	1.4	1.7	0.0	7.0	4.0	0.0
14	9	3	87	0.0	0.0	0.0	7.0	3.0	0.1
15	10	2	277	0.9	8.0	0.0	1.0	1.0	0.0
16	10	2	683	0.7	0.3	0.0	8.9	2.0	5.0
17	10	3	1473	0.7	0.8	0.0	1.7	2.5	0.6
18	10	6	214	0.6	3.1	0.0	4.3	2.7	2.0
19	10	6	5800	2.0	4.0	0.0	7.0	5.0	1.9
20	10	2	512	5.0	2.1	0.0	7.6	2.7	0.2
21	10	2	958	1.2	2.2	0.0	5.3	3.0	0.3
22	11	2	19020	0.0	6.1	0.0	8.0	4.0	1.7
23	11	9	1389	0.8	0.2	0.0	1.0	2.0	5.3
24	14	2	270	1.8	4.2	0.0	6.3	2.0	1.8
25	14	3	178	1.7	5.3	0.0	8.1	2.1	0.0
26	14	2	296	1.8	4.5	0.0	6.6	2.0	3.1
27	15	2	690	1.4	2.8	0.0	4.5	2.8	3.3
28	15	2	653	1.3	2.8	0.0	4.2	2.2	2.7
29	15	2	14980	0.4	8.2	0.0	12.8	5.0	0.0
30	17	2	232	1.3	2.6	0.1	6.2	3.8	1.8
31	17	5	101	4.3	1.6	20.3	7.4	5.1	1.2
32	17	10	10992	2.6	13.4	0.1	16.0	5.6	0.0
33	17	26	20000	2.9	9.1	0.0	13.0	5.0	2.0
34	19	2	540	1.8	4.4	0.0	16.6	1.0	12.9
35	19	7	2310	0.7	10.4	0.0	13.2	4.0	0.0
36	19	4	148	1.6	5.9	0.2	13.1	2.2	5.3
37	19	4	846	1.1	5.1	0.1	10.1	4.1	0.5
38	20	2	80	1.3	6.1	0.4	16.0	6.9	5.4
39	21	2	1000	1.1	2.8	0.0	4.1	2.1	7.4
40	21	2	30488	4.1	2.0	32.5	9.9	6.0	4.0
41	22	3	5000	3.8	10.1	0.0	14.5	4.0	2.0
42	22	2	5644	1.3	3.3	9.0	6.4	6.4	0.0
43	23	2	263	2.0	3.4	0.0	7.7	3.0	0.0

conditional probability parameters of the class variable becomes unstable because the configurations of parents of the class variable become numerous. Then, the sample size for learning the parameters becomes small, as presented in “Sparse data” in Table 3.4. Therefore, numerous parents of the class variable might be unable to reflect the feature data for classification when the sample is not sufficiently large.

Chapter 4

Exact Learning of ANBs

The preceding chapter suggested that exact learning of GBNs by maximizing BDeu to have no parents of the class variable might improve the accuracy of *GBN-BDeu*. In this chapter, we propose an exact learning of ANBs, which maximizes BDeu and ensures that the class variable has no parents. In earlier reports, the ANB constraint was used to learn the BNC as a discriminative model. In contrast, we use the ANB constraint to learn the BNC as a generative model. The space of all possible ANB structures includes at least one I-map because it includes a complete graph, which is an I-map. From the asymptotic consistency of BDeu (Definition 5), the proposed method is guaranteed to achieve the I-map with the fewest parameters among all possible ANB structures when the sample size becomes sufficiently large. Our empirical analysis in Chapter 3 suggests that the proposed method can improve the classification accuracy for small data. We employ the dynamic programming (DP) algorithm learning GBN (Silander and Myllymäki 2006) for the exact learning of ANBs. The DP algorithm for the exact learning of ANBs is almost twice as fast as that for the exact learning of GBNs. We prove that the proposed ANB asymptotically estimates the identical conditional probability of the class variable to that of the exactly learned GBN.

4.1 Learning Procedure

Our method is intended to seek the optimal structure that maximizes the BDeu score among all possible ANB structures. Our algorithm employs dynamic programming (DP) based on the decomposability of BDeu. The local score of the class variable in ANB structures is constant because the class variable has no parents in the ANB structure. Therefore, we can ascertain the optimal ANB structure by maximizing $Score_{ANB}(G, D) = Score(G, D) - LocalScore(CFT_D(X_0, \emptyset))$.

Before we describe the procedure of our method, we introduce the following notations. Let $G^*(\mathbf{Z})$ denote the optimal ANB structure composed of a variable set $\mathbf{Z}, (X_0 \in \mathbf{Z})$. When a variable has no child in a structure, we say it is a *sink* in the structure. We use $X_s^*(\mathbf{Z})$ to denote a sink in $G^*(\mathbf{Z})$. Additionally, letting $\Pi(\mathbf{Z})$ denote the set of all subsets of $\mathbf{Z}$ that include X_0, we define the *best parents* of X_i in a candidate set $\Pi(\mathbf{Z})$ as the parent set that maximizes the local score in $\Pi(\mathbf{Z})$:

$$g_i^*(\Pi(\mathbf{Z})) = \operatorname*{argmax}_{\mathbf{W} \in \Pi(\mathbf{Z})} LocalScore(CFT_D(X_i, \mathbf{W})).$$

Our algorithm has four logical steps. The following process improves the DP algorithm proposed by (Silander and Myllymäki 2006) to learn the optimal ANB structure.

1. For all possible pairs of a variable $X_i \in \mathbf{V} \setminus \{X_0\}$ and a variable set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X_i\}, (X_0 \in \mathbf{Z})$, calculate the local score $LocalScore(CFT_D(X_i, \mathbf{Z}))$ (Equation (2.6)).

2. For all possible pairs of a variable $X_i \in \mathbf{V} \setminus \{X_0\}$ and a variable set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X_i\}, (X_0 \in \mathbf{Z})$, calculate the best parents $g^*(\Pi(\mathbf{Z}))$.

3. For all $\mathbf{Z} \subseteq \mathbf{V}, (X_0 \in \mathbf{Z})$, calculate the sink $X_s^*(\mathbf{Z})$.

4. Calculate $G^*(\mathbf{V})$ using Steps 3 and 4.

Steps 3 and 4 of the algorithm are based on the observation that the best network $G^*(\mathbf{Z})$ necessarily has a sink $X_s^*(\mathbf{Z})$ with incoming edges from its best parents

$g_s^*(\Pi(\mathbf{Z} \setminus \{X_s^*(\mathbf{Z})\}))$. The remaining variables and edges in $G^*(\mathbf{Z})$ necessarily construct the best network $G^*(\mathbf{Z} \setminus \{X_s^*(\mathbf{Z})\})$. More formally,

$$X_s^*(\mathbf{Z}) = \underset{X_i \in \mathbf{Z} \setminus \{X_0\}}{\operatorname{argmax}} \left\{ LocalScore(CFT(X_i, g_i^*(\Pi(\mathbf{Z} \setminus \{X_i\})))) + Score_{ANB}(G^*(\mathbf{Z} \setminus \{X_i\}), D) \right\}. \tag{4.1}$$

From Equation (4.1), we can decompose $G^*(\mathbf{Z})$ into $G^*(\mathbf{Z} \setminus \{X_s^*(\mathbf{Z})\})$ and $X_s^*(\mathbf{Z})$ with incoming edges from $g_s^*(\Pi(\mathbf{Z} \setminus \{X_s^*(\mathbf{Z})\})$. Moreover, this decomposition can be done recursively. At the end of the recursive decomposition, we obtain n pairs of the sink and its best parents, which comprise $G^*(\mathbf{V})$.

The number of iterations to calculate all the local scores, best parents, and best sinks for our algorithm are $(n-1)2^{n-2}$, $(n-1)2^{n-2}$, and 2^{n-1}, respectively, and those for GBN are $n2^{n-1}$, $n2^{n-1}$, and 2^n, respectively. Therefore, the DP algorithm for ANB is expected to be almost twice as fast as that for GBN.

4.2 Asymptotic Properties of the Proposed Method

Under some assumptions, the proposed ANB is proven to asymptotically estimate the identical conditional probability of the class variable, given the feature variables of the exactly learned GBN. When the sample size becomes sufficiently large, the structure learned by the proposed method and the exactly learned GBN are *classification-equivalent* defined as follows:

Definition 7. (Acid et al. (2005))
Let $\mathcal{G}$ be the set of all structures consisting of a variable set $\mathbf{V}$. Also, let D be any finite dataset. For all $G_1, G_2 \in \mathcal{G}$, we say that G_1 and G_2 are classification-equivalent if $P(X_0 \mid \mathbf{x}, G_1, D) = P(X_0 \mid \mathbf{x}, G_2, D)$ for any value $\mathbf{x}$ of the feature variables.

To derive the main theorem, we introduce five lemmas as below.

Lemma 1. (Mihaljević et al. (2018))
For any structure G consisting of a variable set $\mathbf{V}$, G is classification-equivalent to G', which is a modified G by the following operations.

1. For all $X, Y \in \mathbf{Pa}_{X_0}^{G}$, add an edge between X and Y in G.

2. For all $X \in \mathbf{Pa}_{X_0}^{G}$, reverse an edge from X to X_0 in G.

Next, we use the following lemma from Chickering (2002) to derive the main theorem:

Lemma 2. (Chickering (2002))
Let $\mathcal{G}^{Imap}$ be the set of all I-maps consisting of a variable set $\mathbf{V}$. When the sample size becomes sufficiently large, then the following proposition holds.

$$\begin{aligned}
&\forall G_1, G_2 \in \mathcal{G}^{Imap}, \qquad (4.2)\\
&((\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}, Dsep_{G_1}(X, Y \mid \mathbf{Z}) \Rightarrow Dsep_{G_2}(X, Y \mid \mathbf{Z}))\\
&\ \Rightarrow Score(G_1, D) \leq Score(G_2, D)).
\end{aligned}$$

Moreover, we provide Lemma 3 under the following assumption.

Assumption 1. There exists a structure G^* consisting of a variable set $\mathbf{V}$ which satisfies the following property:

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}, Dsep_{G^*}(X, Y \mid \mathbf{Z}) \Leftrightarrow I_{P^*}(X, Y \mid \mathbf{Z}).$$

For the discussion presented herein, we call G^* a true structure throughout.

Lemma 3. Let $\mathcal{G}_{ANB}^{Imap}$ be the set of all I-map ANBs consisting of a variable set $\mathbf{V}$. For all $G_{ANB}^{Imap} \in \mathcal{G}_{ANB}^{Imap}$, and all $X, Y \in \mathbf{V}$, if G^* has a convergence connection $X \rightarrow X_0 \leftarrow Y$, then G_{ANB}^{Imap} has an edge between X and Y.

Proof. We prove Lemma 3 by contradiction. We assume that G_{ANB}^{Imap} has no edge between X and Y. Because G_{ANB}^{Imap} has a divergence connection $X \leftarrow X_0 \rightarrow Y$, we obtain

$$\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}, Dsep_{G_{ANB}^{Imap}}(X, Y \mid X_0, \mathbf{Z}). \qquad (4.3)$$

Because G^* has a convergence connection $X \rightarrow X_0 \leftarrow Y$, the following proposition holds from Theorem 1:

$$\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}, \neg Dsep_{G_{ANB}^{Imap}}(X, Y \mid X_0, \mathbf{Z}). \qquad (4.4)$$

This result contradicts (4.3). Consequently, G_{ANB}^{Imap} has an edge between X and Y. □

Furthermore, under Assumption 1 and the following assumptions, we derive Lemma 4.

Assumption 2. All feature variables are included in the Markov blanket M of the class variable in the true structure G^*.

Assumption 3. For all $X \in M$, X and X_0 are adjacent in G^*.

Lemma 4. We assume Assumptions 1 through 3. Let G_1^* be the modified G^* by the operation 1 in Lemma 1. In addition, let G_{12}^* be the modified G_1^* by the operation 2 in Lemma 1. G_1^* is Markov equivalent to G_{12}^*.

Proof. From Theorem 2, we prove Lemma 4 by showing the following two propositions: (I) G_1^* and G_{12}^* have the same links (edges without direction) and (II) they have the same set of convergence connections. Proposition (I) can be proved immediately because the difference between G_1^* and G_{12}^* is only the direction of the edges between X_0 and the variables in $\mathbf{Pa}_{X_0}^{G^*}$. For the same reason, G_1^* and G_{12}^* have the same set of convergence connections as colliders in $\mathbf{V} \setminus (\mathbf{Pa}_{X_0}^{G^*} \cup \{X_0\})$. Moreover, there are convergence connections with colliders in $\mathbf{Pa}_{X_0}^{G^*} \cup \{X_0\}$ in neither G_1^* nor G_{12}^* because all the variables in $\mathbf{Pa}_{X_0}^{G^*} \cup \{X_0\}$ are adjacent in the two structures. Consequently, they have the same set of convergence connections; i.e., Proposition (II) holds. This completes the proof. □

Finally, under Assumptions 1 through 3, we derive the following lemma.

Lemma 5. We assume Assumptions 1 through 3. Let G_1^* be the modified G^* by the operation 1 in Lemma 1. In addition, let G_{12}^* be the modified G_1^* by the operation 2 in Lemma 1. G_{12}^* is an I-map.

Proof. From Assumption 1 and Definition 4, G^* is an I-map. The DAG G_1^* results from adding the edges between the variables in $\mathbf{Pa}_{X_0}^{G^*}$ to G^*. Because adding edges does not create a new d-separation, G_1^* remains an I-map. Lemma 5 holds because G_1^* is a Markov equivalent to G_{12}^* from Lemma 4. □

Under Assumptions 1 through 3, we prove the following main theorem using Lemmas 1 through 5.

Theorem 3. Under Assumptions 1 through 3, when the sample becomes sufficiently large, the proposal (learning ANB using BDeu) achieves the classification-equivalent structure to G^*.

Proof. Let G^*_{12} be the modified G^* by the operations 1 and 2 in Lemma 1. Because G^*_{12} is classification-equivalent to G^* from Lemma 1, we prove Theorem 3 by showing that the proposed method asymptotically learns a Markov-equivalent structure to G^*_{12}. That is, we show that G^*_{12} asymptotically has the maximum BDeu score among all the ANB structures:

$$\forall G_{ANB} \in \mathcal{G}_{ANB},\ Score(G_{ANB}, D) \leq Score(G^*_{12}, D). \tag{4.5}$$

From Definition 5, the BDeu scores of the I-maps are higher than those of any non-I-maps when the sample size becomes sufficiently large. Therefore, it is sufficient to show that the following proposition holds asymptotically to prove that Proposition (4.5) asymptotically holds:

$$\forall G^{Imap}_{ANB} \in \mathcal{G}^{Imap}_{ANB},\ Score(G^{Imap}_{ANB}, D) \leq Score(G^*_{12}, D). \tag{4.6}$$

From Lemma 5, G^*_{12} is an I-map. Therefore, from Lemma 2, a sufficient condition of (4.6) is as follows:

$$\begin{aligned}&\forall G^{Imap}_{ANB} \in \mathcal{G}^{Imap}_{ANB}, \forall X, Y \in M \cup \{X_0\},\\ &\forall \mathbf{Z} \subseteq M \cup \{X_0\} \setminus \{X, Y\}, Dsep_{G^{Imap}_{ANB}}(X, Y \mid \mathbf{Z}) \Rightarrow Dsep_{G^*_{12}}(X, Y \mid \mathbf{Z}).\end{aligned} \tag{4.7}$$

We prove (4.7) by dividing it into two cases: $X \in \mathbf{Pa}^{G^*}_{X_0} \wedge Y \in \mathbf{Pa}^{G^*}_{X_0}$ and $X \notin \mathbf{Pa}^{G^*}_{X_0} \vee Y \notin \mathbf{Pa}^{G^*}_{X_0}$.

Case I: $X \in \mathbf{Pa}^{G^*}_{X_0} \wedge Y \in \mathbf{Pa}^{G^*}_{X_0}$

From Lemma 3, all variables in $\mathbf{Pa}^{G^*}_{X_0}$ are adjacent in G^{Imap}_{ANB}. Moreover, all variables in $\mathbf{Pa}^{G^*}_{X_0}$ are adjacent in G^*_{12}. From Definition 2, we obtain

$$\begin{aligned}\forall \mathbf{Z} \subseteq &M \cup \{X_0\} \setminus \{X, Y\},\\ &\neg Dsep_{G^{Imap}_{ANB}}(X, Y \mid \mathbf{Z}) \wedge \neg Dsep_{G^*_{12}}(X, Y \mid \mathbf{Z}).\end{aligned} \tag{4.8}$$

For two Boolean propositions p and q, the following holds:

$$(\neg p \wedge \neg q) \Rightarrow (p \Rightarrow q). \tag{4.9}$$

From (4.8) and (4.9), we obtain

$$\begin{aligned}\forall \mathbf{Z} \subseteq & M \cup \{X_0\} \setminus \{X, Y\}, \\ & Dsep_{G_{ANB}^{Imap}}(X, Y \mid \mathbf{Z}) \Rightarrow Dsep_{G_{12}^*}(X, Y \mid \mathbf{Z}).\end{aligned}$$

This completes the proof of (4.7) in **Case I**.

Case II: $X \notin \mathbf{Pa}_{X_0}^{G^*} \vee Y \notin \mathbf{Pa}_{X_0}^{G^*}$

From Definition 4 and Assumption 1, we obtain

$$\begin{aligned}\forall \mathbf{Z} \subseteq & M \cup \{X_0\} \setminus \{X, Y\}, \\ & Dsep_{G_{ANB}^{Imap}}(X, Y \mid \mathbf{Z}) \Rightarrow Dsep_{G^*}(X, Y \mid \mathbf{Z}).\end{aligned}$$

Thus, we can prove (4.7) by showing that the following proposition holds:

$$\begin{aligned}\forall \mathbf{Z} \subseteq & M \cup \{X_0\} \setminus \{X, Y\}, \\ & Dsep_{G^*}(X, Y \mid \mathbf{Z}) \Leftrightarrow Dsep_{G_{12}^*}(X, Y \mid \mathbf{Z}).\end{aligned} \tag{4.10}$$

For the remainder of the proof, we prove (4.10) by dividing it into two cases: $X_0 \in \mathbf{Z}$ and $X_0 \notin \mathbf{Z}$.

Case i: $X_0 \in \mathbf{Z}$

All pairs of variables in $\mathbf{Pa}_{X_0}^{G^*}$ in G^* comprise a convergence connection with collider X_0. From Theorem 1, these pairs are necessarily d-connected, given X_0 in G^*. Therefore, G^* and G_1^* represent identical d-separations given $\mathbf{Z}$ because $X_0 \in \mathbf{Z}$. Because G_1^* is Markov equivalent to G_{12}^* from Lemma 4, G^* and G_{12}^* represent identical d-separations given $\mathbf{Z}$; i.e., Proposition (4.10) holds.

Case ii: $X_0 \notin \mathbf{Z}$

We divide (4.10) into two cases: $X = X_0 \vee Y = X_0$ and $X \neq X_0 \wedge Y \neq X_0$.

Case 1: $X = X_0 \lor Y = X_0$

Because all the variables in the X_0's Markov blanket M are adjacent to X_0 in both G^*_{12} and G^* from Assumption 2, we obtain $\neg Dsep_{G^*_{12}}(X, Y \mid \mathbf{Z}) \land \neg Dsep_{G^*}(X, Y \mid \mathbf{Z})$. From (4.9), Proposition (4.10) holds.

Case 2: $X \neq X_0 \land Y \neq X_0$

If both G^*_{12} and G^* have no edge between X and Y, they have a serial or divergence connection: $X \to X_0 \to Y$ or $X \leftarrow X_0 \to Y$. When $X_0 \notin \mathbf{Z}$, the serial and divergence connections represent d-connections between X and Y given $\mathbf{Z}$ from Theorem 1. Therefore, we obtain $\neg Dsep_{G^*_{12}}(X, Y \mid \mathbf{Z}) \land \neg Dsep_{G^*}(X, Y \mid \mathbf{Z})$. From (4.9), Proposition (4.10) holds.

Thus, we complete the proof of (4.7) in **Case II**.

Consequently, Proposition (4.7) is true, which completes the proof of Theorem 3. □

We proved that the proposed ANB asymptotically estimates the identical conditional probability of the class variable to that of the exactly learned GBN.

4.3 Numerical Examples

This section presents numerical experiments conducted to demonstrate the asymptotic properties of the proposed method. To demonstrate that the proposed method asymptotically achieves the I-map with the fewest parameters among all the possible ANB structures, we evaluate the structural Hamming distance (SHD) (Tsamardinos et al. 2006), which measures the distance between the structure learned by the proposed method and the I-map with the fewest parameters among all the possible ANB structures. The SHD is the total number of three types of errors: *extra edges*, which exist in the learned structure but which do not exist in the true structure; *missing edges*, which exist in the true structure but which do not exist in the learned structure; and *incorrect edges*, which are edges oriented incorrectly in the learned

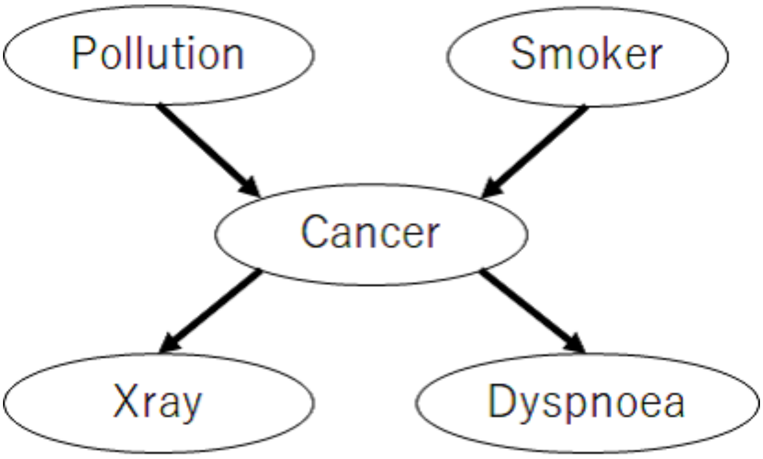

Figure 4.1: A network which satisfies Assumptions 2 and 3 (CANCER network (Scutari 2010)).

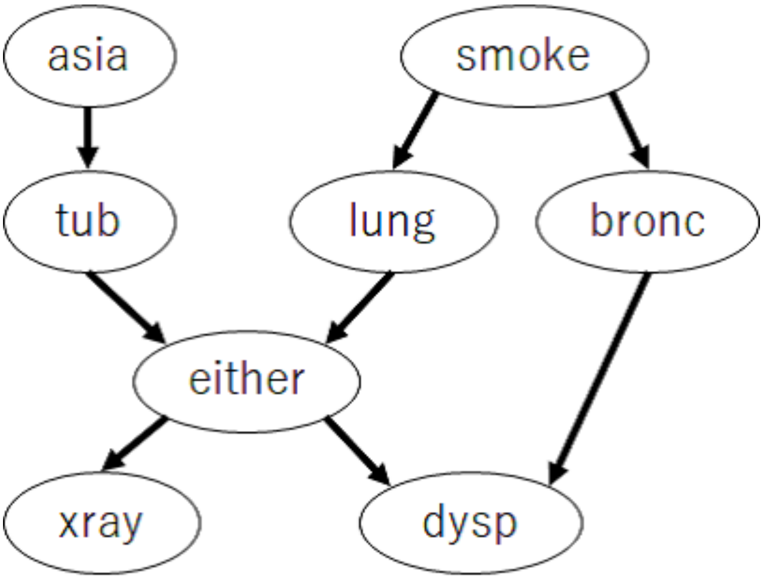

Figure 4.2: A network which violates Assumptions 2 and 3 (ASIA network (Scutari 2010)).

structure. To demonstrate Theorem 3, we evaluate the Kullback-Leibler divergence (KLD) between the learned class variable posterior using the proposed method and that by the true structure. For discrete probability distributions P and Q for the same sample space Ω, the KLD between P and Q is defined as

$$KLD(P \mid\mid Q) = \sum_{x \in \Omega} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

This experiment uses two benchmark datasets from *bnlearn* (Scutari 2010): CANCER and ASIA, as depicted in Figures 4.1 and 4.2. We use the variables "Cancer" and "either" as the class variables in CANCER and ASIA, respectively. In that case, CANCER satisfies Assumptions 2 and 3, but ASIA satisfies neither Assumption 2 nor Assumption 3.

From the two networks, we randomly generate sample data for each sample size $N = 100,\ 500,\ 1,000,\ 5,000,\ 10,000,\ 50,000,$ and $100,000$. Based on the generated data, we learn BNC structures using the proposed method and then evaluate the SHDs and KLDs.

Table 4.1 presents results. The results show that the SHD converges to 0 when the sample size increases in both CANCER and ASIA. Thus, the proposed method asymptotically learns the I-map with the fewest parameters among all possible ANB structures. Furthermore, in CANCER, the KLD between the learned class variable posterior by the proposed method and that by the true structure becomes 0 when $N \geq 1,000$. The results demonstrate that the proposed method learns a classification-equivalent structure of the true one when the sample size becomes sufficiently large, as described in Theorem 3. In ASIA however, the KLD between the learned class variable posterior by the proposed method and that by the true structure does not reach 0 even when the sample size becomes large because ASIA does not satisfy Assumptions 2 and 3.

Table 4.1: The SHD between the structure learned by the proposed method and the I-map with the fewest parameters among all the ANB structures, the KLD between the learned class variable posterior by the proposed method and learned one using the true structure.

Network	Variables	Sample size	SHD-(Proposal, I-map ANB)	KLD-(Proposal, True structure)
ASIA	8	100	3	2.31×10^{-2}
		500	2	1.24×10^{-1}
		1000	2	7.63×10^{-2}
		5000	1	3.67×10^{-3}
		10000	0	9.26×10^{-4}
		50000	0	6.28×10^{-4}
		100000	0	3.59×10^{-5}
CANCER	5	100	1	8.79×10^{-2}
		500	1	2.43×10^{-3}
		1000	0	0.00
		5000	0	0.00
		10000	0	0.00
		50000	0	0.00
		100000	0	0.00

4.4 Learning Markov Blanket

Theorem 3 assumes all feature variables are included in the Markov blanket of the class variable. However, this assumption does not necessarily hold. To solve this problem, we must know the Markov blanket of the class variable before learning the ANB. Under Assumption 3, the Markov blanket of the class variable is equivalent to a set of parents and children of the class variable (PC set). It is known that the exact learning of a PC set of a variable is computationally infeasible when the number of variables increases (Tsamardinos et al. 2006). To reduce the computational cost of learning a PC set, Niinimäki and Parviainen (2012) proposed a score-based local learning algorithm (SLL), which has two learning steps. In the step 1, the algorithm sequentially learns the PC set by repeatedly using the exact learning structure algorithm on a set of variables containing the class variable, the current PC set, and one new query variable. In the step 2, SLL enforces the symmetry constraint: if X_i is a child of X_j, then X_j is a parent of X_i. This allows us to try removing extra variables from the PC set, proving that the SLL algorithm always finds the correct PC of the class variable when the sample size is sufficiently large. Moreover, Gao and Ji (2017) proposed the S^2TMB algorithm, which improved the efficiency over the SLL by removing the symmetric constraints in PC search steps. However, S^2TMB is computationally infeasible when the size of the PC set exceeds 30.

As an alternative approach for learning large PC sets, previous studies proposed constraint-based PC search algorithms, such as MMPC (Tsamardinos et al. 2006), HITON-PC (Aliferis et al. 2003), and PCMB (Peña et al. 2007). These methods produce an undirected graph structure by cutting edges using conditional independence (CI) tests such as statistical hypothesis tests or information theory tests. As statistical hypothesis tests, the G^2 and χ^2-tests were used for these constraint-based methods. In these tests, the independence of two variables was set as a null hypothesis. A p-value signifies the probability that the null hypothesis is correct at a user-determined significance level. If the p-value exceeds the significance level, the null hypothesis is accepted. However, Sullivan and Feinn (2012) reported that statistical hypothesis tests have a significant shortcoming: the p-value sometimes becomes much smaller than the significance level as the sample size increases. Therefore, statistical hypothesis tests suffer from Type I errors (detecting dependence for an independent conditional relation in the true DAG). Conditional mutual information (CMI) is often used as a CI test (Cover and Thomas 1991). The CMI strongly depends on a hand-tuned threshold value. Therefore, it is not guaranteed to estimate the true CI structure. Consequently, CI tests have no asymptotic consistency.

For a CI test with asymptotic consistency, Steck and Jaakkola (2002a) proposed a Bayes factor with BDeu (the "BF method," below), where the Bayes factor is the ratio of marginal likelihoods between two hypotheses (Kass and Raftery 1995). For two variables $X, Y \in \mathbf{V}$ and a set of conditional variables $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, the BF method $\log BF_D(X, Y \mid \mathbf{Z})$ is defined as

$$\log BF_D(X, Y \mid \mathbf{Z}) = \mathit{LocalScore}(CFT_D(X, \mathbf{Z})) - \mathit{LocalScore}(CFT_D(X, \mathbf{Z} \cup \{Y\})),$$

where $\mathit{LocalScore}(CFT_D(X, \mathbf{Z}))$ and $\mathit{LocalScore}(CFT_D(X, \mathbf{Z} \cup \{Y\}))$ can be obtained using Equation (2.6). The BF method detects $I_{P^*}(X, Y \mid \mathbf{Z})$ if $BF_D(X, Y \mid \mathbf{Z})$ is larger than the threshold δ, and detects $\neg I_{P^*}(X, Y \mid \mathbf{Z})$ otherwise. From an asymptotic local consistency of BDeu (Definition 6) (Chickering 2002), we can prove easily that the BF method also has the following asymptotic consistency.

Theorem 4. The following two properties almost surely hold when the sample size is sufficiently large.

1. If X and Y is conditionally independent given $\mathbf{Z}$, then $\log BF_D(X, Y \mid \mathbf{Z}) > 0$.

2. If X and Y is not conditionally independent given $\mathbf{Z}$, then $\log BF_D(X, Y \mid \mathbf{Z}) < 0$.

Proof. 1. From the asymptotic local consistency of BDeu, $\mathit{LocalScore}(CFT_D(X, \mathbf{Z})) > \mathit{LocalScore}(CFT_D(X, \mathbf{Z} \cup \{Y\}))$ holds almost surely. Therefore, $\log BF_D(X, Y \mid \mathbf{Z}) > 0$.

2. From the asymptotic local consistency of BDeu, $\mathit{LocalScore}(CFT_D(X, \mathbf{Z})) < \mathit{LocalScore}(CFT_D(X, \mathbf{Z} \cup \{Y\}))$ holds almost surely. Therefore, $\log BF_D(X, Y \mid \mathbf{Z}) < 0$.

□

Natori et al. (2015) and Natori et al. (2017) applied the BF method to a constraint-based approach, and showed that their method is more accurate than the other methods with traditional CI tests.

We propose the constraint-based PC search algorithm using a BF method. The proposed PC search algorithm finds the PC set of the class variable using a BF method between the class variable and all feature variables because the Bayes factor has an asymptotic consistency for the CI tests (Natori et al. 2017). However, missing a variable that significantly affects the classification is known to degrade the classification accuracy (Friedman et al. 1997). Therefore, we redundantly learn the PC set of the class variable with no missing variables as follows.

- The proposed PC search algorithm only conducts the CI tests at the zero order (given no conditional variables) which is more reliable than those at the higher order.
- We use a positive value as the threshold δ for the Bayes factor.

Furthermore, we compare the accuracy of the proposed PC search method with those of MMPC, HITON-PC, PCMB, and S^2TMB. Learning Bayesian networks is known to be highly sensitive to the chosen an equivalent sample size (ESS) (Ueno 2010, 2011, Silander et al. 2007). Therefore, we determine the ESS $N' \in \{1.0, 2.0, 5.0\}$ and the threshold $\delta \in \{3, 20, 150\}$ in the Bayes factor using 2-fold cross validation to obtain the highest classification accuracy. The three ESS-values of N' are determined according to Ueno (2010, 2011). The three values of δ are determined according to Heckerman et al. (1995). All the compared methods are implemented in Java.[1] This experiment uses six benchmark datasets from *bnlearn*:

[1] Source code is available at `http://www.ai.lab.uec.ac.jp/software/`

ASIA, SACHS, CHILD, WATER, ALARM, and BARLEY. From each benchmark network, we randomly generate sample data for sample size $N = 10,000$. Based on the generated data, we learn the PC sets of all variables using each method. Table 4.2 shows the average runtime of each method. We calculate missing variables, representing the number of removed variables existing in the true PC set, and extra variables, which indicate the number of remaining variables that do not exist in the true PC set. Table 4.2 also shows the average missing and extra variables from the learned PC sets of all the variables. We compare the classification accuracies of the exact learning of ANBs with BDeu score (designated as *ANB-BDeu*) using each PC search method as a feature selection method. Table 4.3 shows the average accuracies of each method from the 43 UCI repository datasets listed in Table 3.3.

Table 4.2 shows that the runtimes of the proposed method are shorter than those of the other methods. Moreover, the results show that the missing variables of the proposed method are fewer than those of the other methods. On the other hand, Table 4.2 also shows that the extra variables of the proposal are more than those of the other methods in all datasets. From Table 4.3, the results show that the *ANB-BDeu* using the proposed method provides a much higher average accuracy than the other methods. This is because missing variables degrade classification accuracy more significantly than extra variables (Friedman et al. 1997).

4.5 Experiments to Evaluate Exact Learning of ANBs

This section presents numerical experiments conducted to evaluate the effectiveness of the exact learning of ANBs. First, we compare the classification accuracies of *ANB-BDeu* with those of the other methods in Chapter 3. We use the same experimental setup and evaluation method described in Chapter 3. The classification accuracies of *ANB-BDeu* are presented in Table 3.3. To confirm the significant dif-

Table 4.2: Missing variables and extra variables, and runtimes (ms) of each method.

		MMPC			HITON-PC			PCMB			S^2TMB			Proposal		
Network	Variables	Missing	Extra	Runtime	Missing	Extra	Runtime	Missing	Extra	Runtime	Missing	Extra	Runtime	Missing	Extra	Runtime
ASIA	8	1.25	0.00	251	1.75	0.63	117	1.75	0.63	163	0.25	0.50	888	0.00	3.50	13
SACHS	11	1.91	0.00	1062	2.64	0.36	248	2.00	0.00	610	0.00	0.00	4842	0.00	2.55	12
CHILD	20	1.75	0.05	6756	2.35	0.95	380	2.00	0.25	1191	0.05	0.05	6669	0.00	11.80	16
WATER	32	3.59	0.00	407	4.00	0.19	140	3.78	0.31	260	2.03	1.47	29527	0.25	13.44	25
ALARM	37	1.81	0.14	3832	2.38	0.57	281	2.19	0.19	1025	0.14	0.11	11272	0.05	10.92	39
BARLEY	48	2.85	1.23	4928	3.46	0.42	269	3.19	0.42	830	1.15	0.46	99290	0.38	9.75	49
Arithmetic average		2.19	0.24	2872	2.76	0.52	239	2.48	0.30	680	0.60	0.43	25415	0.11	8.66	26
Geometric average				1550			221			545			9911			22

Table 4.3: Average classification accuracy of each method.

	MMPC	HITON-PC	PCMB	S^2TMB	Proposal
Average	0.6185	0.6219	0.6302	0.7980	0.8164

ferences of *ANB-BDeu* from the other methods, we apply Hommel's tests (Hommel 1988), which are used as a standard in machine learning studies (Demšar 2006). The p-values are presented at the bottom of Table 3.3. In addition, "MB size" in Table 3.4 denotes the average of the Markov blanket size of the class variable in the structures learned by *GBN-BDeu*.

The results show that *ANB-BDeu* outperforms *Naive Bayes*, *GBN-CMDL*, *BNC2P*, *TAN-aCLL*, *gGBN-BDeu*, and *MC-DAGGES* at the $p < 0.1$ significance level. Moreover, the results show that *ANB-BDeu* improves the accuracy of *GBN-BDeu* when the class variable has numerous parents such as No. 3, No. 9, and No. 31 datasets, as shown in Table 3.4. Furthermore, *ANB-BDeu* provides higher accuracies than *GBN-BDeu*, even for large data such as datasets 13, 22, 29, and 33 although the difference between *ANB-BDeu* and *GBN-BDeu* is not statistically significant. These actual datasets do not necessarily satisfy Assumptions 1 through 3 in Theorem 3. These results imply that the accuracies of *ANB-BDeu* without satisfying Assumptions 1 through 3 might be comparable to those of *GBN-BDeu* for large data. It is worth noting that the accuracies of *ANB-BDeu* are much worse than those provided by *GBN-BDeu* for datasets No. 5 and No. 12. "MB size" in these datasets are much smaller than the number of all feature variables, as shown in Table 3.4. The results show that feature selection by the Markov blanket is expected to improve the classification accuracies of the exact learning of ANBs, as described in Chapter 4.4.

We compare the classification accuracies of *ANB-BDeu* using the PC search method proposed in Chapter 4.4 (referred to as "*fsANB-BDeu*") with the other methods in Table 3.3. Table 3.3 shows the classification accuracies of *fsANB-BDeu* and the p-values of Hommel's tests for differences in *fsANB-BDeu* from the other methods. The results show that *fsANB-BDeu* outperforms all the compared methods at the $p < 0.05$ significance level.

"Max parents" in Table 3.4 presents the average maximum number of parents learned by *fsANB-BDeu*. The value of "Max parents" represents the complexity of the structure learned by *fsANB-BDeu*. The results show that the accuracies of *Naive Bayes* are better than those of *fsANB-BDeu* when the sample size is small, such as No. 36 and No. 38 datasets. In these datasets, the values of "Max parents" are large. The estimation of the variable parameters tends to become unstable when a variable has numerous parents, as described in Chapter 3. *Naive Bayes* can avoid this phenomenon because the maximum number of parents in *Naive Bayes* is one. However, *Naive Bayes* cannot learn relationships between the feature variables. Therefore, for large samples such as No. 8 and No. 29 datasets, *Naive Bayes* shows much worse accuracy than those provided by other methods.

Similar to *Naive Bayes*, *BNC2P* and *TAN-aCLL* show better accuracies than *fsANB-BDeu* for small samples such as No. 38 dataset because the upper bound of the maximum number of parents is two in the two methods. However, the small upper bound of the maximum number of parents tends to lead to a poor representational power of the structure (Ling and Zhang 2003). As a result, the accuracies of both methods tend to be worse than those of *fsANB-BDeu* of which the value of "Max parents" is greater than two, such as No. 29 dataset.

For large samples such as dataset Nos. 29 and 33, *GBN-CMDL*, *gGBN-BDeu*, and *MC-DAGGES* show worse accuracies than *fsANB-BDeu* because the exact learning methods estimate the network structure more precisely than the greedy learned structure.

We compare *fsANB-BDeu* and *ANB-BDeu*. The difference between the two methods is whether the proposed PC search method is used. "Removed variables" in Table 3.4 represents the average number of variables removed from the Markov blanket of the class variable by our proposed PC search method. The results demonstrate that the accuracies of *fsANB-BDeu* tend to be much higher than those of *ANB-BDeu* when the value of "Removed variables" is large, such as Nos. 5, 12, 16, 34, and 38. Consequently, discarding numerous irrelevant variables in the features improves the classification accuracy.

Finally, we compare the runtimes of *fsANB-BDeu* and *GBN-BDeu* to demonstrate the efficiency of the ANB constraint. Table 4.4 presents the runtimes of *GBN-BDeu*, *fsANB-BDeu*, and the proposed PC search method. The results show that the runtimes of *fsANB-BDeu* are shorter than those of *GBN-BDeu* in all the datasets because the execution of the exact learning of ANBs is almost twice as fast as that of the exact learning of GBNs, as described in Chapter 4.1. Moreover, the runtimes of *fsANB-BDeu* are much shorter than those of *GBN-BDeu* when our PC search method removes many variables, such as No. 34 and No. 39 datasets. This is because the runtimes of *GBN-BDeu* decrease exponentially with the removal of variables, whereas our PC search method itself has a negligibly small runtime compared to those of the exact learning as shown in Table 4.4. As a result, the proposed method *fsANB-BDeu* provides the best classification performances in all the methods with a lower computational cost than that of the *GBN-BDeu*.

Table 4.4: Runtimes (ms) of GBN-BDeu, fsANB-BDeu, and the proposed PC search method.

No.	Variables	Sample size	Classes	GBN-BDeu	fsANB-BDeu	The proposed PC search method
1	5	625	3	169.4	23.0	6.3
2	5	1372	2	19.3	10.3	2.0
3	5	132	3	15.6	3.0	0.2
4	5	150	3	16.7	5.0	0.2
5	5	24	3	15.3	1.0	0.1
6	7	1728	4	90.8	22.9	1.7
7	7	345	2	21.1	15.6	0.3
8	7	432	2	31.0	20.7	0.5
9	7	64	2	18.9	9.1	0.1
10	8	3200	10	114.6	55.1	3.1
11	9	17898	2	300.5	251.3	10.2
12	9	12960	3	707.4	525.8	5.8
13	9	768	9	66.8	27.6	0.6
14	9	87	5	39.6	0.3	0.1
15	10	277	2	162.6	6.9	0.3
16	10	683	2	453.1	258.9	0.4
17	10	1473	3	161.1	121.4	0.8
18	10	214	6	63.0	22.3	0.2
19	10	5800	6	159.6	67.2	2.8
20	10	512	2	102.7	58.2	0.4
21	10	958	2	212.2	193.0	0.5
22	11	19020	2	979.8	277.2	5.3
23	11	1389	9	379.4	17.2	0.9
24	14	270	2	1988.6	299.8	0.1
25	14	178	3	1233.7	585.0	0.1
26	14	296	2	2034.5	115.2	0.2
27	15	690	2	10700.3	927.6	0.3
28	15	653	2	23069.5	2774.3	0.2
29	15	14980	2	12407.6	8248.8	4.1
30	17	232	2	11682.6	1623.6	0.2
31	17	101	5	7326.5	1985.1	0.1
32	17	10992	10	84967.1	48636.9	3.4
33	17	20000	26	339910.2	30224.8	6.3
34	19	540	2	217457.0	12.0	0.3
35	19	2310	7	190895.9	103447.5	1.0
36	19	148	4	107641.8	1171.4	0.2
37	19	846	4	144669.5	62663.0	0.4
38	20	80	2	98841.9	821.6	0.1
39	21	1000	2	2706616.6	8885.1	0.5
40	21	30488	2	15626734.5	130491.6	11.8
41	22	5000	3	10022030.7	757611.7	2.1
42	22	5644	2	4640293.5	2382657.7	2.3
43	23	263	2	2553290.4	1386088.2	0.2
Geometric average				2361.0	362.4	0.6

Chapter 5

Learning ANB for Large Networks

5.1 Constraint-Based Learning Bayesian Networks using a Bayes factor

The most popular structure learning approach is score-based learning, which seeks a best structure with the score function. However, score-based learning is an NP-hard problem (Chickering 1996), entailing heavy computational costs as the number of variables increases. As exact learning methods, dynamic programming Silander and Myllymäki (2006), A^*search(Yuan et al. 2011), branch and bound search (Malone et al. 2011), and integer programming (Cussens 2012) have been proposed. However, no state-of-the-art exact learning method can learn structures with more than 60 variables (Cussens 2012).

Alternatively, a constraint-based approach relaxes computational costs and learns huge networks. Methods using such an approach learn structures by conditional independence (CI) tests and by direction using orientation rules. Among these approaches, the Peter and Clark (PC) algorithm (Spirtes et al. 2000), max-min hill climb (MMHC) algorithm (Tsamardinos et al. 2006), and recursive autonomy identification (RAI) algorithm (Yehezkel and Lerner 2009) are well known. Of those, the RAI algorithm is the state-of-the-art algorithm. The salient benefit of the RAI algorithm is that it decreases the number of conditional variables of CI tests in the constraint-based approach because it decomposes the entire structure into partial structures based on observed convergence connections. However, this approach relies on CI tests conducted between each pair of variables using statistical tests or information theory tests. The statistical test necessarily has type I error (detecting incorrect dependences) even for large data. The information theory test also depends on the user-determined threshold. Therefore, earlier methods using this approach have no asymptotic consistency.

However, Steck and Jaakkola (2002b) proposed a conditional independence test with an asymptotic consistency: a Bayes factor with BDeu. Moreover, Abellán et al. (2006) and Natori et al. (2017) proposed constraint-based learning methods using the RAI with a Bayes factor, which can learn large networks. We will apply the constraint-based learning methods using a Bayes factor to our proposed method to accommodate much greater numbers of variables in our method.

5.2 Learning ANB using the RAI Algorithm with the Bayes Factor

This section presents the algorithm of the constraint-based learning method of ANB with RAI algorithm. Let $\mathbf{NDA}_X^G$ be a set of variables that are adjacent to X via an undirected edge in G. Our algorithm has six logical steps as follows.

(1) Input data D, initial order of CI tests $n_z = 1$, and initial graph G_s and G_{all}, which are complete undirected graphs consisting of all the feature variables. Let $\mathbf{V}_s$ be a set of variables in G_s and let $\mathbf{E}_s$ be a set of edges in G_s.

(2) For all $X \in \mathbf{V}_s, Y \in \mathbf{Pa}_X^{G_{all}} \cup \mathbf{NDA}_X^{G_{all}}, \mathbf{Z} \subseteq \mathbf{Pa}_X^{G_{all}} \cup \mathbf{NDA}_X^{G_{all}}, (|\mathbf{Z}| = n_z)$, when X and Y given $\mathbf{Z} \cup \{X_0\}$ are determined to be conditionally independent by CI tests using Bayes factor, the edges between X and Y in G_s and G_{all} are removed.

(3) Apply the orientation rule to the graph obtained in (2).

(4) If there exists a variable set $\mathbf{A}$ such that any pair of variables in $\mathbf{A}$ can reach each other in G_{all} and $\forall X \in \mathbf{V} \setminus (\mathbf{A} \cup \{X_0\}), (\mathbf{Pa}_X^{G_{all}} \cup \mathbf{NDA}_X^{G_{all}}) \cap \mathbf{A} = \emptyset$, then decompose G_s into a subgraph $G_{\mathbf{A}}$ consisting of $\mathbf{A}$ and a subgraph $G_{n\mathbf{A}}$ consisting of $\mathbf{V} \setminus (\mathbf{A} \cup \{X_0\})$.

(5) $n_z = n_z + 1$. For each subgraph G, recursively invoke RAI with $G_s = G$.

(6) Add X_0 and the edges from X_0 to all the feature variables to G_{all}.

In Step (1), the initial graph G_{ucf} does not include the class variable and the edges from the class variable to all the feature variables. The proposed method starting without X_0 is more efficient than that with X_0 because the former has smaller number of edges than the latter does although they achieve the same results.

The proposed method is expected to improve the efficiency of the original RAI algorithm without the ANB constraint for the following reasons. First, the proposed method performs CI tests only among feature variables whereas the original RAI performs CI tests among all variables. Second, the proposed method is guaranteed to accelerate decomposition of the structure in the RAI algorithm when the true Bayesian network has an ANB structure. The CI tests given the class variable in Step (2) earlier detect the conditional independence than those without the class variable do. As the number of removed edges is larger, the number of the decomposition in the RAI algorithm increases. Consequently, it is expected to decrease the number of conditional variables of CI tests in the RAI algorithm.

If we assume ANB, then the number of parameters necessarily increases compared to GBN because it forces addition of edges from class variables to feature variables. In this case, almost sure convergence to the true value of the joint probability distribution represented by the estimation structure is expected theoretically to be slower than that of GBN. However, as described in Chapter 3, because the number of the prior distribution parameter of the class variable increases exponentially, GBNs are known to have unstable estimation accuracy when the number of parent variables of a class variable is large (Sugahara et al. 2018, Sugahara and Ueno

2021). Although the number of parameters is greater with the ANB structure, no parent of class variables is expected to improve the classification accuracy.

5.3 Asymptotic Consistency to an I-map ANB with the Fewest Parameters

This section presents the theorem which says that the proposed RAI algorithm asymptotically learns an I-map ANB with the fewest parameters. Let G_{ANB} be an ANB structure learned by the proposed RAI algorithm. Then, we provide the following theorem.

Theorem 5. When the sample size N becomes sufficiently large, G_{ANB} converges almost surely to an I-map ANB with the fewest parameters.

Proof. Step (2) of the proposed RAI algorithm does not remove edges which exist in G^* and removes extra edges for an I-map ANB because CI tests using Bayes factor asymptotically detect the true conditional independences from Theorem 4. Moreover, the orientation rule in Step (3) makes only the true conditional independences. Therefore, G_{ANB} almost surely converges to an I-map ANB with the fewest parameters. □

It is notifiable that the proposed algorithm might not necessarily orient all the edges. If orienting the undirected edges causes a new convergence connection, G_{ANB} does not necessarily become an I-map. In this case, we should not orient the undirected edges but directly calculate the joint distribution over the variables with the undirected edges.

5.4 Experiments to Evaluate the Proposed RAI Algorithm

This section presents evaluation experiments conducted to underscore the effectiveness of the proposed RAI algorithm. First, we use the following nine methods to compare classification accuracies for small networks.

- Naive Bayes
- TAN: Learn a TAN that optimizes the log likelihood (Friedman et al. 1997).
- GBN-CMDL: Greedy learning GBN method using the hill-climbing search by minimizing CMDL while estimating parameters by maximizing LL (Grossman and Domingos 2004).
- BNC2P: Greedy learning method with at most two parents per variable using the hill-climbing search by maximizing CLL while estimating parameters by maximizing LL (Grossman and Domingos 2004).
- TAN-aCLL: Exact learning of TANs by maximizing aCLL (Carvalho et al. 2013).
- GBN-BDeu: Exact learning of GBNs with BDeu score (Silander and Myllymäki 2006).
- ANB-BDeu: Exact learning of ANBs with BDeu score.
- RAI-GBN: Constraint-based learning GBN using Bayes factor.
- RAI-ANB: Learning ANB using proposed method.

The value of the pseudo-sample (hyperparameter) for the BDeu score and Bayes factor was set as 1.0 to maximize the posterior variance, as suggested by Ueno (2010). For all methods, the conditional probability parameters of the BNCs after structure learning were estimated using expected a posteriori (EAP).

This experiment used 43 classification benchmark datasets with 5–23 variables from the UCI repository (Lichman 2013). The continuous quantities in each dataset were discretized into binary values around a median. For each method and dataset, we obtain the average classification accuracy using ten-fold cross validation. To demonstrate the importance of the proposed method, the p-value is obtained using multiple comparison using the Hommel method (Hommel 1988), which is used as a standard in machine learning studies (Demšar 2006). In "Classification accuracy" shown at the bottom of Table 5.1, "Arithmetic average" denotes the average classification accuracy of each method for all datasets. Also, "p-value" denotes the p-value obtained by multiple comparison. For "Runtime", "Arithmetic average" and "Geometric average" respectively denote the arithmetic average runtime and the geometric average runtime for structure learning of each method for all datasets. Table 5.2 presents the average maximum number of parents (MNP) for each method and the average number of edges in the Markov blanket (MNB) of the class variable for each method.

Table 5.1 shows that the proposed method outperforms Naive Bayes, TAN, GBN-CMDL, BNC2P, TAN-aCLL, and RAI-GBN at the $p < 0.1$ significance level. Because Naive Bayes, TAN, GBN-CMDL, BNC2P, and TAN-aCLL limit the number of parent variables of feature variables, Max Parents are fixed at 1 and 2, as shown in Table 5.2. However, the small upper bound of the maximum number of parents tends to lead to poor representational power of the structure (Ling and Zhang 2003). As a result, the accuracies of Naive Bayes and TAN tend to be worse than those obtained using the proposed method, such as No. 8 and No. 11 datasets. For large samples such as datasets Nos 11 and 19, RAI-ANB provides higher accuracies than GBN-CMDL does, because RAI-ANB guarantees to asymptotically estimate the true conditional probability of the class variable although GBN-CMDL does not. Because Naive Bayes requires no structural learning, the computation time is 0.00. In addition, because TAN can be learned in polynomial time (Friedman et al. 1997, Madden 2009). Its computation time is shorter than that of RAI-ANB.

Table 5.1: Accuracies of the respective classifiers for small networks.

	dataset	variable	number of data	classes	Naive Bayes	TAN	GBN-CMDL	BNC2P	TAN-aCLL	GBN-BDeu	ANB-BDeu	RAI-GBN	RAI-ANB
1	magic	11	19020	2	0.7447	0.7767	0.7849	0.7806	0.7631	**0.7865**	0.7863	0.7793	0.7790
2	Flare	11	1389	9	0.7804	0.7976	0.8265	0.8315	0.8229	**0.8430**	0.8265	0.8423	0.8178
3	heart	14	270	2	0.8296	0.8407	0.8185	0.8037	0.8148	**0.8444**	0.8148	0.7666	0.8333
4	wine	14	178	3	0.9205	0.9212	0.9438	0.9157	0.9326	0.9424	**0.9490**	0.9212	0.9150
5	Cleve	14	296	2	0.8309	0.8175	0.8209	0.8007	**0.8378**	0.8144	0.8309	0.7771	0.8212
6	Australian	15	690	2	0.8362	0.8304	0.8312	0.8348	0.8464	**0.8492**	0.8449	0.8405	0.8463
7	crx	15	653	2	0.8391	0.8483	0.8346	0.8208	**0.8560**	0.8481	0.8482	0.8544	0.8436
8	EEG	15	14980	2	0.5774	0.6298	0.6787	0.6374	0.6125	0.6843	**0.6937**	0.6421	0.6709
9	Congressional	17	232	2	0.9137	0.9398	0.9698	0.9612	0.9181	**0.9699**	**0.9699**	0.9655	0.9438
10	zoo	17	101	5	0.9709	0.9427	0.9109	0.9505	**1.0000**	0.9900	0.9700	0.8809	0.9418
11	pendigits	17	10992	10	0.7998	0.8477	0.9062	0.8719	0.8700	**0.9329**	0.9326	0.8757	0.9254
12	letter	17	20000	26	0.4456	0.4866	0.5796	0.5132	0.5093	0.5777	0.5950	0.5560	**0.6145**
13	ClimateModel	19	540	2	0.9203	0.9314	**0.9407**	0.9241	0.9333	0.9259	0.9055	0.9074	0.9203
14	ImageSegmentation	19	2310	7	0.7324	0.7510	0.7918	0.7991	0.7407	0.8233	**0.8290**	0.7839	0.8121
15	lymphography	19	148	4	0.8523	0.8109	0.7939	0.7973	0.8311	**0.8647**	0.7909	0.6842	0.8514
16	vehicle	19	846	4	0.4266	0.5472	0.5910	0.5910	0.5816	0.5910	**0.6417**	0.4893	0.6028
17	hepatitis	20	80	2	0.8750	0.8750	0.7375	0.8875	0.8750	**0.9250**	0.9000	0.8125	0.8875
18	German	21	5000	2	0.7440	0.7340	0.6110	0.7340	0.7470	0.7320	0.7420	0.7000	**0.7540**
19	bank	21	30488	2	0.8542	0.8774	0.8618	0.8928	0.8618	0.8954	0.8956	**0.8959**	0.8926
20	waveform-21	22	5000	3	0.7894	0.7914	0.7862	0.7754	0.7896	0.7938	**0.8048**	0.7328	0.7870
21	Mushroom	22	5644	2	0.9962	**1.0000**	**1.0000**	**1.0000**	0.9995	0.9946	**1.0000**	**1.0000**	**1.0000**
22	spect	23	263	2	0.7868	0.8101	0.7940	0.7903	0.8090	0.7759	**0.8207**	0.7937	0.8096
	Classification accuracy	Arithmetic average			0.7939	0.8094	0.8097	0.8143	0.8160	**0.8366**	0.8360	0.7955	0.8304
		p-value			0.0024	0.0117	0.0324	0.0099	0.0574	> 0.1	> 0.1	0.0013	-
	Runtime (s)	Arithmetic average			0.00	2.58	30.53	21.11	10.05	1790.93	500.76	26.06	3.14
		Geometric average			0.00	0.798	9.50	6.87	3.27	201.76	110.69	7.90	1.26

Table 5.2: Number of Max parents and edges in the Markov blanket of the class variable for small networks.

	dataset	Naive Bayes		TAN		GBN-BDeu		ANB-BDeu		RAI-GBN		RAI-ANB	
		NMP	NMB	NMP	NMB	NMP	NMB	NMP	NMB	NMP	NMB	NMP	NMB
1	magic	1	10	2	19	4	20.4	4	30	4	10.7	5	29
2	Flare	1	10	2	19	2	1	3	18.9	1.9	1.3	2.7	17.6
3	heart	1	13	2	25	2	6.6	2	18.4	2	2	2	16.4
4	wine	1	13	2	25	2.2	9.5	2.1	19	3.2	3.2	2.1	16.6
5	Cleve	1	13	2	25	2	7.5	2	18.3	2	2	2	16.6
6	Australian	1	14	2	27	2.4	6.2	2.9	24.1	2	2	2.3	20.3
7	crx	1	14	2	29	3	5.3	2.2	23.9	2	1.9	2	21.1
8	EEG	1	14	2	27	5	34.2	5	57.5	5	9.1	5.3	51.9
9	Congressional	1	16	2	31	3.5	7.1	4	37.1	2.5	1.8	3	29.2
10	zoo	1	16	2	31	4.9	9.4	4.9	36.9	3.9	3.9	3	27.6
11	pendigits	1	16	2	31	5.5	63.4	5.6	66.5	9.1	9.1	6	61.2
12	letter	1	16	2	31	6	41.4	5	57.9	7.8	9.5	5.3	50.7
13	ClimateModel	1	18	2	35	14	32.1	14.1	69.7	3.1	3.1	1	18
14	ImageSegmentation	1	18	2	35	4.1	31.5	4	48	6	6	5.3	45
15	lymphography	1	18	2	35	8.7	16.6	9.9	36.7	2	1.5	2.3	23.9
16	vehicle	1	18	2	35	4.2	14.3	4.1	50.8	4	3.7	3.2	40.9
17	hepatitis	1	19	2	37	10.4	31.6	11.4	78.1	2.1	1.1	2.9	29.5
18	German	1	20	2	39	2	4.1	3	33.3	2	1	3	29.6
19	bank	1	20	2	39	5	13.1	6	63.9	5	5.1	5.8	52
20	waveform-21	1	21	2	41	4	39.8	4	60.3	4.8	4.7	3.5	43.5
21	Mushroom	1	21	2	41	2.4	6.7	7.6	83	5.2	14.9	5.2	74.6
22	spect	1	22	2	43	2.7	9.3	3	49.2	2.6	2.2	3.1	46.2

Table 5.1 also shows that the proposed method much improves the classification accuracy of RAI-GBN, although RAI-GBN has the lowest classification accuracy among the compared methods. The reason might be that RAI-GBN tends to learn structures with small Markov blankets of class variables. In fact, Table 5.2 shows that the edges in the Markov blanket of the class variable are fewer than those of the other methods. In contrast, because the proposed method has all the feature variables as children of the class variable, the Markov blanket size is always the same as the number of feature variables. Moreover, because the proposed method performs CI tests among feature variables only, it requires less computational time than RAI-GBN, which performs CI tests among all variables.

The average classification accuracy of RAI-ANB is slightly worse than that of either GBN-BDeu or ANB-BDeu. The exact learning methods are known to estimate network structures more accurately than constraint-based approaches do when the sample size is large (Scutari et al. 2019). However, the runtime of RAI-ANB is much shorter than that of either GBN-BDeu or ANB-BDeu.

Next, we compare the classification accuracies of intractable large networks for the exact learning methods. This experiment used 16 datasets with 37–1301 variables. Table 5.3 shows the average accuracies and p-values of Hommel's tests. Table 5.4 presents the average number of edges in the Markov blanket of the class variable for each method.

From Table 5.3, the average classification accuracy of the proposed method is the highest among all the methods. The proposed method outperforms Naive Bayes, TAN, and RAI-GBN at the $p < 0.05$ significance level. Similarly to the results for small networks, the average runtime of the proposed method is shorter than that of RAI-GBN by the reason described earlier.

The classification accuracies of Naive Bayes and TAN are lower than those of the proposed method for all datasets except for No. 3 and No. 5. Table 5.4 shows that the edges in the Markov blanket of the class variable in RAI-ANB for No. 3 and 5 are few. Therefore, the true structure of these datasets might resemble the structure of Naive Bayes.

Table 5.3: Accuracies of the respective classifiers for large networks.

	dataset	variables	num of data	classes	Naive Bayes	TAN	RAI-GBN	RAI-ANB
1	kr-vs-kp	37	3196	2	0.8773	0.9239	0.9405	**0.9518**
2	Connect-4	43	67557	3	0.7212	0.7643	0.7467	**0.7973**
3	Flowmeters D	44	180	4	**0.8388**	**0.8388**	0.8055	0.8277
4	movement libras	91	360	15	0.5027	0.5388	0.1611	**0.5666**
5	dota2	117	102944	2	**0.5980**	0.5810	0.5435	0.5957
6	Musk1	167	478	2	0.6538	0.7565	0.6658	**0.8219**
7	Musk2	167	6598	2	0.7443	0.8408	0.8808	**0.9639**
8	Epileptic Seizure	179	11500	5	0.2344	0.3650	0.1886	**0.3820**
9	mfeat-fac	219	2000	10	0.3520	0.4590	0.3030	**0.4730**
10	semeion	257	1600	10	0.8556	0.8719	0.4106	**0.8794**
11	madelon	501	2000	2	0.5905	0.5270	**0.6280**	0.5830
12	pd speech features	755	756	2	0.7182	0.7897	0.7657	**0.8228**
13	pure-spectra-matrix	1301	571	20	0.9088	0.8984	0.4833	**0.9159**
	Classification accuracy	Arithmetic average			0.6612	0.7042	0.5787	**0.7370**
		p-value			0.0044	0.0012	0.0015	-
	Runtime (s)	Arithmetic average			0.0	545.7	2002.1	1665.9
		Geometric average			0.0	52.6	375.3	227.4

Table 5.4: Number of edges in the Markov blanket of the class variable.

	dataset	Naive Bayes	TAN	RAI-GBN	RAI-ANB
1	kr-vs-kp	36	71	5.1	136.5
2	Connect-4	42	83	31.6	157
3	Flowmeters D	43	85	4.0	91.9
4	movement libras	90	179	2.1	210.2
5	dota2	116	231	2.9	215.8
6	Musk1	166	331	2.0	553
7	Musk2	166	331	6.1	1115.8
8	Epileptic Seizure	178	355	0	367
9	mfeat-fac	216	431	3.7	600.4
10	semeion	256	511	3.8	771.4
11	madelon	500	999	2.7	537.7
12	pd speech features	754	1507	2.1	2095.1
13	pure-spectra-matrix	1300	2599	6.6	2399.9

The classification accuracies of the proposed method are higher than those of RAI-GBN for all datasets except for No. 11, perhaps because RAI-GBN tends to learn structures with small Markov blankets of class variables similarly to results of small networks. Table 5.4 shows that the edges in the Markov blanket of the class variable are fewer than those of the other methods. However, because the proposed method assumes ANB structure, all the feature variables are used for class variable estimation, which improves the classification accuracy.

Finally, we demonstrate that the proposed method accelerates the structure decompositions that occur during the RAI algorithm execution when the class variable is the root in the true Bayesian network. Table 5.5 presents the numbers of edges (NE), the numbers of decomposed structures (NDS) to subgraphs in the RAI algorithm, and the runtimes for RAI-GBN and RAI-ANB.

Table 5.5: Numbers of edges, decomposed structures, and runtime for RAI-GBN and RAI-ANB.

	dataset	NE		NDS		Runtime	
		RAI-GBN	RAI-ANB	RAI-GBN	RAI-ANB	RAI-GBN	RAI-ANB
1	kr-vs-kp	121	139	6	4	27	19.5
2	connect-4	155	158	13	14	1103.6	398.7
3	Flowmeters D	70	87	4	5	3.9	2.6
4	movement libras	125	202	3	8	9.7	21.4
5	dota2	188	227	4	6	320.5	218.3
6	Musk1	479	563	5	5	170.9	109.4
7	Musk2	1047	1152	10	9	12669.5	14624.2
8	Epileptic Seizure	357	379	3	13	2568.1	398.4
9	mfeat-fac	717	610	6	4	1010.5	304.4
10	semeion	880	781	5	4	419.9	134.9
11	madelon	234	529	134	2	267.3	306.3
12	pd speech features	1606	2072	15	14	2720.6	1656
13	pure spectra matrix	3101	2313	9	116	4735.5	3462.7

The numbers of edges (NEs) learned by RAI-ANB and RAI-GBN from the same data theoretically become identical when the true Bayesian network has an ANB structure. When the class variable is not the root in the true Bayesian network, the NE of RAI-ANB becomes larger than that of RAI-GBN. From Table 5.5, the NE of RAI-ANB for No. 13, which provides the largest difference of the accuracies between RAI-ANB and RAI-GBN, is less than that of RAI-GBN. This result suggests that No. 13 approximately follows an ANB. Therefore, the NDS of RAI-ANB for No. 13 is much larger than that of RAI-GBN. This result means that the proposed method accelerates the structure decompositions that occur during the RAI algorithm execution. As a result, it reduces the runtime of the proposed method.

In contrast, the NE of RAI-ANB for No. 11, for which RAI-GBN provides better accuracy than RAI-ANB does, is much larger than that of RAI-GBN. Therefore, the NDS of RAI-ANB for No. 11 is much less than that of RAI-GBN because the dense structure of RAI-GBN interrupts the structure decompositions in the RAI algorithm execution. As a result, it increases the runtime of the proposed method. Thus, it is important for the proposed method to select the class variable so as to be the root variable.

Chapter 6

Conclusions

First, this study compares the classification performances of the BNs exactly learned by BDeu as a generative model and those learned approximately by CLL as a discriminative model. Surprisingly, the results demonstrate that the performance of BNs achieved by maximizing ML was better than that of BNs achieved by maximizing CLL for large data. However, the results also show that the classification accuracies of the BNs that are learned exactly by BDeu are much worse than those that are learned by the other methods when the class variable had numerous parents. To solve this problem, this study proposes an exact learning ANB by maximizing BDeu as a generative model. The proposed method asymptotically learns the optimal ANB, which is an I-map with the fewest parameters among all possible ANB structures. In addition, the proposed ANB is guaranteed to asymptotically estimate the identical conditional probability of the class variable to that of the exactly learned GBN. Based on these properties, the proposed method is effective for not only classification but also decision making, which requires a highly accurate probability estimate of the class variable. Furthermore, learning ANBs has lower computational costs than learning BNs does. The experimental results demonstrate that the proposed method significantly outperforms the approximately learned structure by maximizing CLL. Moreover, we proposed an extension of constraint-based learning method using Bayes factor applied to the learning ANB. Comparison experiments showed that our method outperforms the other methods. Isozaki et al. (2008, 2009) proposed an effective learning Bayesian network method by adjusting the hyperparameter for small data. As future work, we will employ their method instead of the BDeu to improve the classification accuracy for small data. Sugahara et al. (2020, 2022) also reported a Bayesian network model averaging classifier to improve the classification accuracies. We expect to extend our proposed method to the model averaging classifier using those methods described above.

Related Journal Papers

- Shouta Sugahara, Maomi Ueno, "Exact Learning Augmented Naive Bayes Classifier," Entropy, Vol.23, No.12, 2021 (corresponding to Chapters 3 and 4).
- Shouta Sugahara, Maomi Ueno, "Exact Learning Bayesian Network Classifier with Augmented Naive Bayes Structure Constraint," The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J103, No.4, pages 301-313, 2020 (in Japanese) (corresponding to Chapters 3 and 4).

Related Conference Papers

- Shouta Sugahara, Wakaba Kishida, Koya Kato, Maomi Ueno, "Recursive Autonomy Identification-Based Learning of Augmented Naive Bayes Classifiers," In Proceedings of Machine Learning Research, PGM2022, Vol.186, pages 265–276, 2022 (corresponding to Chapter 5).
- Shouta Sugahara, Masaki Uto, Maomi Ueno, "Exact Learning Augmented Naive Bayes Classifier," In Proceedings of Machine Learning Research, PGM2018, Vol.72, pages 439–450, 2018 (corresponding to Chapters 3 and 4).

Other Papers

- Kento Aoki, Shouta Sugahara, Maomi Ueno, "Constraint-Based Learning Bayesian Network IRT using Bayes Factor," The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J106, No.2, pages 84-95, 2023 (in Japanese).
- Shouta Sugahara, Maomi Ueno, "Learning Bayesian Network Classifiers to Minimize the Class Variable Parameters," The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J105, No.11, pages 679-690, 2022 (in Japanese).

- Shouta Sugahara, Itsuki Aomi, Maomi Ueno, “Bayesian Network Model Averaging Classifiers by Subbagging,” Entropy, Vol.24, No.5, 2022.

- Naruchika Kikuya, Shouta Sugahara, Kazuki Natori, Maomi Ueno, “Learning Huge Bayesian Network Classifier with Augmented Naive Bayes,” The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J104, No.1, pages 65-81, 2021 (in Japanese).

- Shouta Sugahara, Itsuki Aomi, Maomi Ueno, “Bayesian Network Model Averaging Classifiers by Subbagging,” In Proceedings of Machine Learning Research, PGM2020, Vol.138, pages 461–472, 2020.

- Itsuki Aomi, Shouta Sugahara, Maomi Ueno, “Model Averaging Bayesian Network Classifier by Ensemble Learning,” The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J103, No.3, pages 183-193, 2020 (in Japanese).

- Kazunori Honda, Kazuki Natori, Shouta Sugahara, Takashi Isozaki, Maomi Ueno, “Learning Huge Bayesian Network Structures Using the Transitivity,” The Institute of Electronics, Information and Communication Engineers (IEICE), Vol.J102, No.12, pages 796-811, 2019 (in Japanese).

Acknowledgements

I have not been able to complete the doctoral thesis without the support of many people. First of all, I would like to express my sincere gratitude to my supervisor, Professor Maomi Ueno, for his valuable comments, suggestions, and encouragement throughout the research period. I would like to thank Professors Masakazu Muramatsu, Yoshio Okamoto, Yasutada Oohama, and Hideki Yagi of the University of Electro-Communications, who are my committee members, for proofreading the draft of this thesis and providing useful comments. This research was supported by JSPS KAKENHI Grant Numbers JP19H05663, JP22K19825, and JP22J15279. Finally, I would like to express my gratitude to my parents for their tremendous encouragement.

Shouta Sugahara

Bibliography

J. Abellán, M. Gómez-Olmedo, and S. Moral. Some variations on the pc algorithm. In *Proceedings of the International Conference on Probabilistic Graphical Models*, pages 1–8, 2006.

S. Acid, L. M. De Campos, and J. G. Castellano. Learning Bayesian Network Classifiers: Searching in a Space of Partially Directed Acyclic Graphs. *Machine Learning*, 59:213–235, 2005.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: A Novel Markov Blanket Algorithm for Optimal Variable Selection. *AMIA*, pages 21–25, 2003.

M. Barlett and J. Cussens. Advances in Bayesian Network Learning Using Integer Programming. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 182–191, 2013.

W. Buntine. Theory Refinement on Bayesian Networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 52–60, 1991.

A. M. Carvalho, P. Adão, and P. Mateus. Efficient Approximation of the Conditional Relative Entropy with Applications to Discriminative Learning of Bayesian Network Classifiers. *Entropy*, 15:2716–2735, 2013.

J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian Networks from Data: An Information-Theory Based Approach. *Artificial Intelligence*, 137: 43–90, 2002.

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

D. M. Chickering. Optimal Structure Identification With Greedy Search. *JMLR*, 3: 507–554, 2002.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

J. Cussens. Bayesian network learning with cutting planes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 153–160, 2012.

C. P. De Campos and Q. Ji. Efficient Structure Learning of Bayesian Networks Using Constraints. *JMLR*, 12:663–689, 2011.

C. P. De Campos, M. Cuccu, G. Corani, and M. Zaffalon. *Extended Tree Augmented Naive Classifier*, pages 176–189. 2014.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7: 1–30, 2006.

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

T. Gao and Q. Ji. Efficient score-based Markov Blanket discovery. *IJAR*, 80:277–293, 2017.

R. Greiner and W. Zhou. Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, pages 167–173, 2002.

D. Grossman and P. Domingos. Learning Bayesian Network classifiers by maximizing conditional likelihood. In *Proceedings of the International Conference on Machine Learning*, pages 361–368, 2004.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20:197–243, 1995.

G. Hommel. A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, pages 383–386, 1988.

T. Isozaki, N. Kato, and M. Ueno. Minimum Free Energies with "Data Temperature" for Parameter Learning of Bayesian Networks. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 371–378, 2008.

T. Isozaki, N. Kato, and M. Ueno. "Data temperature" in Minimum Free energies for Parameter Learning of Bayesian Networks. *IJAIT*, 18:653–671, 2009.

R. E. Kass and A. E. Raftery. Bayes Factors. *JASA*, 90:773–795, 1995.

M. Koivisto and K. Sood. Exact Bayesian Structure Discovery in Bayesian Networks. *JMLR*, 5:549–573, 2004.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

M. Lichman. UCI Machine Learning Repository, 2013. URL `http://archive.ics.uci.edu/ml`.

C. X. Ling and H. Zhang. The Representational Power of Discrete Bayesian Networks. *JMLR*, 3:709–721, 2003.

P. J. F. Lucas. Restricted bayesian network structure learning. 146:217–234, 2004.

M. G. Madden. On the classification performance of TAN and general Bayesian networks. *Knowledge-Based Systems*, 22:489–495, 2009.

B. Malone, C. Yuan, E. A. Hansen, and S. Bridges. Improving the Scalability of Optimal Bayesian Network Learning with External-Memory Frontier Breadth-First Branch and Bound Search. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 479–488, 2011.

B. Mihaljević, C. Bielza, and P. Larrañaga. Learning Bayesian network classifiers with completed partially directed acyclic graphs. In *Proceedings of the International Conference on Probabilistic Graphical Models*, pages 272–283, 2018.

M. Minsky. Steps toward Artificial Intelligence. In *Proceedings of the IRE*, pages 8–30, 1961.

E. Mokhtarian, S. Akbari, F. Jamshidi, J. Etesami, and N. Kiyavash. Learning Bayesian networks in the presence of structural side information, 2021.

K. Natori, M. Uto, Y. Nishiyama, S. Kawano, and M. Ueno. Constraint-based learning bayesian networks using bayes factor. In *Proceedings of Machine Learning Research (AMBN2015)*, pages 15–31, 2015.

K. Natori, M. Uto, and M. Ueno. Consistent Learning Bayesian Networks with Thousands of Variables. In *Proceedings of Machine Learning Research (AMBN2017)*, volume 73, pages 57–68, 2017.

T. Niinimäki and P. Parviainen. Local Structure Discovery in Bayesian Networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 634–643, 2012.

J. Pearl. *Models, Reasoning, and Inference.* Cambridge University Press, 2000.

J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *IJAR*, 45:211–232, 2007.

R. Y. Rohekar, Y. Gurwicz, S. Nisimov, G. Koren, and G. Novik. Bayesian structure learning by recursive bootstrap. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 10546–10556, 2018.

M. Scutari. Learning Bayesian Networks with the bnlearn R Package. *JSSA*, 35: 1–22, 2010.

M. Scutari, C. E. Graafland, and J. M. Gutiérrez. Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. *IJAR*, 115:235–253, 2019.

T. Silander and P. Myllymäki. A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 445–452, 2006.

T. Silander, P. Kontkanen, and P. Myllymäki. On Sensitivity of the MAP Bayesian Network Structure to the Equivalent Sample Size Parameter. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 360–367, 2007.

A. Singth and A. Moore. Finding optimal Bayesian networks by dynamic programming. Technical report, Technical Report, Carnegie Mellon University, 2005.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, 2000.

H. Steck. Learning the Bayesian Network Structure: Dirichlet Prior vs. Data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 511–518, 2008.

H. Steck and T. S. Jaakkola. On the Dirichlet Prior and Bayesian Regularization. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 713–720, 2002a.

H. Steck and T.S. Jaakkola. *On the Dirichlet prior and Bayesian regularization.*, pages 697–704. 2002b.

S. Sugahara and M. Ueno. Exact learning augmented naive bayes classifier. *Entropy*, 23, 2021.

S. Sugahara, M. Uto, and M. Ueno. Exact learning augmented naive Bayes classifier. In *Proceedings of the International Conference on Probabilistic Graphical Models*, pages 439–450, 2018.

S. Sugahara, I. Aomi, and M. Ueno. Bayesian network model averaging classifiers by subbagging. In *Proceedings of the International Conference on Probabilistic Graphical Models*, pages 461–472, 2020.

S. Sugahara, I. Aomi, and M. Ueno. Bayesian network model averaging classifiers by subbagging. *Entropy*, 24, 2022.

G. M. Sullivan and R. Feinn. Using Effect Size – or Why the P Value Is Not Enough. *JGME*, 4:279–282, 2012.

J. Suzuki. A theoretical analysis of the BDeu scores in Bayesian network structure learning. *Behaviormetrika*, 44:97–116, 2017.

I. Tsamardinos, L.E. Brown, and C.F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.

M. Ueno. Learning likelihood-equivalence Bayesian networks using an empirical Bayesian approach. *Behaviormetrika*, 35:115–135, 2008.

M. Ueno. Learning Networks Determined by the Ratio of Prior and Data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 598–605, 2010.

M. Ueno. Robust learning Bayesian networks for prior belief. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 689–707, 2011.

T. Verma and J. Pearl. Equivalence and Synthesis of Causal Models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 255–270, 1990.

R. Yehezkel and B. Lerner. Bayesian network structure learning by recursive autonomy identification. *JMLR*, 10:1527–1570, 2009.

C. Yuan and B. Malone. Learning Optimal Bayesian Networks: A Shortest Path Perspective. *JAIR*, 48:23–65, 2013.

C. Yuan, H. Lim, and T. Lu. Most Relevant Explanation in Bayesian Networks. *JAIR*, 42:309–352, 2011.