修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院　情報理工学研究科　情報・ネットワーク工学専攻　博士前期課程		
氏　　名	菊谷　成慎	学籍番号	１９３１０４３
論 文 題 目	Constraint-Based Learning Bayesian Networks Classifier With Augmented Naïve Bayes (Augmented Naïve Bayes によるベイジアンネットワーク分類器の制約ベース学習)		

要　旨

ベイジアンネットワーク分類器 (Bayesian Network Classifier: BNC) は同時確率分布をモデル化した生成モデルの分類器である．BNC の構造学習はこれまで候補構造から近似的に識別モデルの学習スコアが最大となる構造を探索する手法が用いられてきた．近年，菅原ら (2020) はベイジアンネットワークの学習スコアにより厳密学習した BNC は分類精度が低いとは限らないと報告している．さらに彼らは，構造に Augmented Naive Bayes (ANB) を仮定し生成モデルとして BNC を厳密学習する手法を提案した．これによりデータが少ない場合も分類精度の高い BNC を学習できることを示した．しかし，厳密学習は変数の増加に対して計算量が指数的に増加するため，菅原ら (2020) の手法は数十変数の ANB 学習が限界である．そこで，本論文では大規模変数を持つ BNC を学習できる手法を提案する．因果モデルの研究分野では，条件付き独立性検定 (CI テスト) とエッジの方向付けによる計算効率の高い構造学習法が提案されており，制約ベースアプローチと呼ぶ．名取らは，CI テストに Bayes factor を用いることで真の構造への漸近一致性を有しつつ 1000 変数以上の構造学習を実現し，本田らはその手法に推移性を組み込むことで 3500 変数の構造学習を実現した．そこで本論文では，本田らの手法を用いて従来より大規模な ANB を学習できる手法を提案し，提案手法が ANB 構造について漸近的にパラメータ数を最小にして真の同時確率分布に収束することを示す．実験により，大規模構造学習において提案手法が有用であることを示す．

令和 2 年（2020 年度）　修士論文

Constraint-Based Learning Bayesian Networks Classifier With Augmented Naive Bayes

電気通信大学　情報理工研究科
情報・ネットワーク工学専攻
植野真臣研究室

学籍番号 1931043
氏名　菊谷成慎

指導教員　植野真臣 教授
副指導教員　川野秀一 准教授

提出日 2021 年 3 月 25 日

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A Bayesian network is a probabilistic graphical model in which discrete random variables are represented by nodes and dependencies between nodes are represented by Directed Acyclic Graph (DAG). It decomposes a joint probability distribution into a product of conditional probabilities by assuming a DAG in the probability structure. The structure of a Bayesian network generally needs to be estimated from the data. A most common score for learning Bayesian network structures is the marginal likelihood of the structure. The structure which maximizes marginal likelihood is called a generative model, which represents the joint probability distribution of all variables. The marginal likelihood is known to have asymptotic consistency, which guarantees that the structure which maximizes the marginal likelihood converges to the true structure when the sample size is sufficiently large (Heckerman, Geiger, & Chickering, 1995). A Bayesian network classifier (BNC), in which one node is a class variable and the other nodes are feature variables in a Bayesian network, is known as a classifier for discrete variables (Friedman, Geiger, & Goldszmidt, 1997). It is known that discriminative models expressing the conditional probability of the class variables given the feature variables have higher classification accuracy than generative models expressing the joint probability of all variables (Carvalho, Roos, Oliveira, & Myllymäki, 2011; Carvalho, Adão, & Mateus, 2013; Grossman & Domingos, 2004). Recently, however, Sugahara et al. (2018) showed that when the data is large enough, the generative model has higher classification accuracy. On the other hand, they pointed out that when the data is small and the structure of the class variable has many parent variables, the data used for parameter learning becomes sparse, resulting in a significant decrease in accuracy. To solve this problem, they proposed a method for exact learning of Augmented Naive Bayes (ANB) (Friedman et al., 1997), which assumes a structure in which class variables have no parents and

all feature variables as children. They showed that this method can achieve high classification accuracy even when the data is small. Furthermore, Sugahara et al. (2020) proposed an efficient algorithm for exact learning of ANB structures. However, they used dynamic programming, which cannot be applied to large-scale variables because it is limited to learning structures of several dozen variables.

On the other hand, as an alternative approach for learning large network structures, a constraint-based approach has been known. This method learns structure by orienting edges using orientation rules (Pearl, 2000) on an undirected graph that is learned by applying 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, Glymour, & Scheines, 2000), the TPDA algorithm (Cheng, Greiner, Kelly, Bell, & Liu, 2002), the MMHC algorithm (Tsamardinos, Brown, & Aliferis, 2006), and the RAI algorithm (Yehezkel & Lerner, 2009) have been proposed. As CI tests, the G^2 test, χ^2 test, and conditional mutual information have been used for these constraint-based methods. However, these tests have no asymptotic consistency. To solve the problem, Natori et al. (2015, 2017) use the Bayes factor, which has asymptotic consistency with respect to independence for CI testing of RAI algorithms, and improve the learning accuracy. Furthermore, Honda et al. (2019) reduce the number of CI tests and computation time by incorporating Bayesian network transitivity into the RAI algorithm to achieve large-scale structure learning with 3500 variables.

In this paper, we propose an extension of Honda's method (Honda, Natori, Sugahara, Isozak, & Ueno, 2019) to the learning of BNCs under the assumption of ANB structure, which enables us to learn larger scale BNCs than before. We also prove that the proposed method can learn the ANB structure representing the true joint probability with a minimal number of parameters when the sample size is sufficiently large. Furthermore, simulation experiments on random networks and numerical experiments on benchmark datasets show that the proposed method can learn larger networks than the exact solution search approach and has higher classification accuracy than the conventional methods on large networks.

Chapter 2

Bayesian Network

1. Learning Bayesian Networks

Let $\mathbf{V} = \{X_0, X_1, \ldots, X_n\}$ be a set of N discrete variables. Each can take values in the set of states $\{1, \ldots, r_i\}$. We write $X_i = k$ when we observe that variable X_i is state k. According to the Bayesian network structure $G = (\mathbf{V}, \mathbf{E})$, where $\mathbf{E}$ is a set of edges, the joint probability distribution is given as

$$P(X_0, X_1, \ldots, X_n) = \prod_{i=0}^{n} P(X_i \mid \Pi_i, G). \tag{2.1}$$

where Π_i is the parent variable set of X_i.

Let a path between node X and Y in G be a sequence of distinct nodes $\{X_0, X_1, \ldots, X_n\}$. On the path, the three variables A, B, and C are $A \to C \to B$, $A \leftarrow C \to B$ and $A \to C \leftarrow B$, which are called head-to-tail, tail-to-tail, and head-to-head with the variable C. The Bayesian network structure expresses conditional independence by blocking the path connecting the two variables. The blocks are defined as follows.

Theorem 2.1 *When a path p in two variables X and Y satisfies one of the following conditions, the path p is blocked by a variable set $\mathbf{Z}$.*

1. *The path p is head-to-tail or tail-to-tail with the variable $Z \in \boldsymbol{Z}$.*
2. *The path p is head-to-head with the variable $Z \notin \boldsymbol{Z}$, and no descendant of Z belongs to $\mathbf{Z}$.*

When all paths connecting two variables X and Y are blocked by the set of variables $\mathbf{Z}$, we represent the conditional independence between A and B in the structure G as $I_G(X, Y|\mathbf{Z})$. Also, when X and Y are conditionally independent given $\mathbf{Z}$ in the true

joint probability distribution, we denote $I_P(X, Y|\mathbf{Z})$. Conversely, when X and Y are dependent given $\mathbf{Z}$ in the true joint probability distribution, we denote $D_P(X, Y|\mathbf{Z})$.

Bayesian networks represent the joint probability distribution of all variables with a DAG structure G and a parameter set Θ. However, DAGs are not capable of representing all conditional independence in the joint probability distribution. Therefore, Bayesian networks handle only the conditional independence that can be represented by the DAG, and the following I-map is defined.

Theorem 2.2 *Let G^* be true structure. If any independence implied by a structure G is also implied by the structure G^*, G is an independence map (I-map) of G^*.*

The joint probability distribution represented by the I-map asymptotically converges to the true distribution. In addition, the following relationship holds between the true structure and the I-map.

Theorem 2.3 *Let $G^* = (\boldsymbol{V}, \boldsymbol{E}^*)$ be true structure, and let E_{XY} be the edge between X and Y. If any structure $G = (\boldsymbol{V}, \boldsymbol{E})$ satisfies the following three conditions, then G is an I-map.*

1. *For $\forall X, Y \in \mathbf{V}$, if $E_{XY} \in \boldsymbol{E}^*$, then $E_{XY} \in \boldsymbol{E}$.*
2. *For $\forall X, Y, Z \in \mathbf{V}$, if G^* has head-to-head $X \rightarrow Z \leftarrow Y$, then $E_{XY} \in \boldsymbol{E}$ or G has head-to-head $X \rightarrow Z \leftarrow Y$.*
3. *For $\forall X, Y, Z \in \boldsymbol{V}$, if G^*has head-to-tail $X \rightarrow Z \rightarrow Y$ or tail-to-tail $X \leftarrow Z \rightarrow Y$, then $E_{XY} \in \boldsymbol{E}$ or G has head-to-tail $X \rightarrow Z \rightarrow Y$ or tail-to-tail $X \leftarrow Z \rightarrow Y$.*

The proof of Theorem 2.3 is shown in (Chickering, 2002).

1.1 Parameter estimation

Assume a Dirichlet distribution $P(\Theta)$ as the prior distribution of the parameters. Let $\mathbf{D} = \{D_1, \ldots, D_N\}$ denote the data for the set of variables $\mathbf{X}$. The Dirichlet distribution $P(\Theta)$ and the posterior distribution $P(\Theta \mid D, G)$ are represented as

$$P(\Theta) = \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1}, \tag{2.2}$$

$$P(\Theta \mid D, G) = \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma\{\sum_{k=1}^{r_i} (\alpha_{ijk} + N_{ijk})\}}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}+N_{ijk}-1}, \tag{2.3}$$

where N_{ijk} represents the number of frequencies of $X_i = k$ when the parent variable set $\Pi_i = j$, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Also, α_{ijk} denotes the hyperparameters of the Dirichlet prior distributions (α_{ijk} is a pseudo-sample corresponding to N_{ijk}); $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.

In Bayesian networks, the most commonly used parameter estimate is Expected a Posteriori (EAP) estimate. It is obtained by taking the expected value in equation 2.3.

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}. \tag{2.4}$$

1.2 Learning structure

In order to estimate the parameters of a Bayesian network, it is necessary to estimate the optimal structure from the data. A popular structure learning approach is the score-based approach, which uses score functions to seek the best structure. The learning score is generally based on the marginal likelihood $P(\mathbf{D} \mid G)$. The marginal likelihood can be expressed in closed form by marginalizing the parameter estimates from the posterior distribution of the equation 2.3.

$$P(\mathbf{D} \mid G) = \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}. \tag{2.5}$$

The results for $\alpha_{ijk} = 1$ (uniform prior distribution) are known to asymptotically converge to the results of Bayesian Information Criterion (BIC) (Schwarz, 1978) and Minimum Description Length (MDL) (Rissanen, 1978). Recently, Bayesian Dirichlet equivalent uniform (BDeu) ($\alpha_{ijk} = \alpha/(r_i q_i)$) is the most used score (Heckerman et al., 1995; Buntine, 1991), where α is a pseudo-sample that represents the weight of prior knowledge, called Equivalent Sample Size (ESS). Ueno et al. (2008, 2010, 2011, 2012) reported that BDeu using a non-informative prior distribution is the most useful. The score-based approach by BDeu has the following asymptotic consistency.

Definition 2.1 *When the number of data is sufficiently large, if the structure estimated by a structure learning approach is almost sure convergence to I-map with the minimum number of parameters, then this approach has asymptotic consistency for the Bayesian network structure.*

However, this approach has NP-hard problem [14], entailing heavy computational costs as the number of variables increases. In order to efficiently search for the best structure, several methods have been proposed, such as dynamic programming (Cowell, 2009; Koivisto & Sood, 2004; Singth & Moore, 2005; Silander & Myllymäki, 2006;

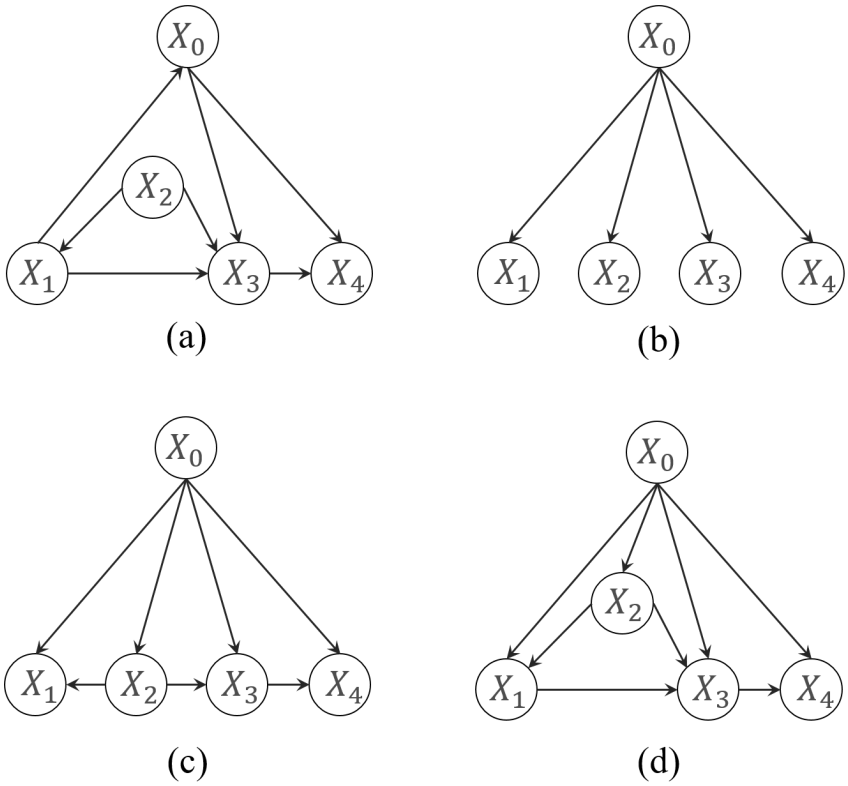

Figure 2.1: (a) Example of GBN; (b) Example of Naive Bayes; (c) Example of TAN; (d) Example of ANB;

Malone, Yuan, Hansen, & Bridges, 2011), A^* search (Yuan, Lim, & Lu, 2011), and integer programming (Barlett & Cussens, 2013). However, even the most advanced method, (Barlett & Cussens, 2013), is limited to learning structures of about 60 variables.

2. Bayesian Network Classifier

2.1 Learning Bayesian network classifier

A Bayesian network classifier (BNC) can be interpreted as a Bayesian network for which X_0 is the class variable and for which $X_0, \ldots, X_n$ are feature variables. Given an instance $x = \langle x_0, \ldots, x_n \rangle$ for feature variables $X_0, \ldots, X_n$, the BNC predicts the

class c by maximizing the posterior probability as

$$\begin{aligned}\hat{c} &= \arg\max_{c\in\{1,\ldots,r_0\}} P(c \mid x_1,\ldots,x_n,G,\Theta) \\ &= \arg\max_{c\in\{1,\ldots,r_0\}} \prod_{i=0}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}(\theta_{ijk})^{1_{ijk}} \\ &= \arg\max_{c\in\{1,\ldots,r_0\}} \prod_{j=1}^{q_0}\prod_{k=1}^{r_0}(\theta_{0jk})^{1_{0jk}} \times \prod_{i:X_i\in\mathbf{Ch}}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}(\theta_{ijk})^{1_{ijk}},\end{aligned} \tag{2.6}$$

where 1_{ijk} if $X_i = k$ and $\Pi_i = j$ in case $\langle x_0,\ldots,x_n\rangle$ and $1_{ijk} = 0$ otherwise. Furthermore, **Ch** is a set of children of the class variable X_0. From Equation 2.6, we can infer class c given only the values of the X_0 ' s parents, the X_0 ' s children, and the parents of the X_0 ' s children, which are called a Markov blanket of X_0.

A BNC that use a general Bayesian network is called a General Bayesian Network (GBN) (Figure 2.1 (a)). A GBN exactly learned by BDeu may learn a structure that has a small number of child class variables and a large number of parent variables. Therefore, the conditional probability parameter estimation of the class variable becomes unstable because the number of patterns of the parents becomes large. As a result, the classification accuracy tends to decrease significantly (Jensen & Nielsen, 2007; Mitchell, 1997). As a means of resolving this difficulty, Minsky (1961) proposed Naive Bayes classifier (Figure 2.1 (b)), for which the class variable has no parents and for which each feature variable is independent, and Friedman et al. (1997) proposed the tree-augmented naive Bayes (TAN) classifier (Figure 2.1 (c)), for which the class variable has no parents and for which each feature variable has a class variable and at most one other feature variable as a parent variable. It is known that TAN with likelihood as a score can be learned in polynomial time (Friedman et al., 1997; Madden, 2009). In adiition, as a more expressive model that generalizes Naive Bayes and TAN, they proposed the augmented naive Bayes (ANB) classifier, for which the class variable has no parent and in which all feature variables have the class variable as a parent. Similar to the GBN, the ANB structure can be used to interpret causal relationships between the feature variables.

Although the GBN that maximizes the marginal likelihood is a generative model that represents the joint probability of all variables, discriminative models that represent the conditional probability of the class variables given the feature variables have been reported to have higher asymptotic classification accuracy (Carvalho et al., 2013; Grossman & Domingos, 2004). However, previous studies have not theoretically shown the basis for GBN as a discriminative model to have higher classification accuracy than GBN as a generative model with asymptotic consistency for the Bayesian

network structure. And, in their experiments, they used approximate learning approaches even though the BNC as a generative model can be learned using exact ones. Sugahara et al. (2018, 2020) showed that GBN as a generative model have higher classification accuracy than ones with as a discriminative model when the data is sufficient. However, they showed in their experiments that the number of parent variables of the class variable in GBN as a generative model may become excessive when the data is small, and the classification accuracy is significantly reduced in this case. To solve this problem, they proposed an exact learning approach with ANB as a generative model. The results show that their approach can obtain stable classification accuracy even when the data is small, and the classification accuracy is significantly higher BNC as a discriminative model. However, the exact learning approach is limited to learning dozens of variables and cannot be applied to the cases with large variables. Therefore, in this paper, we propose an approach that can learn a BNC with large variables.

3. Constraint-based learning Bayesian Networks

As an alternative approach for learning large network structures, a constraint-based approach has been known. This section presents an overview of the constraint-based approach and a RAI algorithm using a Bayes factor.

3.1 RAI algorithm using Bayes factor

Now assume that a true Bayesian network is as follows.

Assumption 2.1 *Let $G^* = (\boldsymbol{V}, \boldsymbol{E}^*)$ be true Bayesian network structure. In this case, G^* satisfies the following.*

$$\forall X, \forall Y \in \boldsymbol{V}, \forall \boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}, I_P(X, Y | \boldsymbol{Z}) \Leftrightarrow E_{XY} \notin \boldsymbol{E}^*.$$

Next, we define Markov equivalent.

Definition 2.2 *Let G_1 and G_2 be two DAGs consisting of a set of variables $\boldsymbol{V}$. If the following holds for $\forall X, \forall Y \in \boldsymbol{V}, \forall \boldsymbol{Z} \subseteq \boldsymbol{V}$, G_1 and G_2 are Markov equivalent,*

$$I_{G_1}(X, Y | \boldsymbol{Z}) \Leftrightarrow I_{G_2}(X, Y | \boldsymbol{Z}).$$

A graph representing a Markov equivalent class are commonly represented using undirected edges. A DAG containing such a subgraph is called a Partially DAG (PDAG). In the constraint-based approach, if there exists the true Bayesian network structure

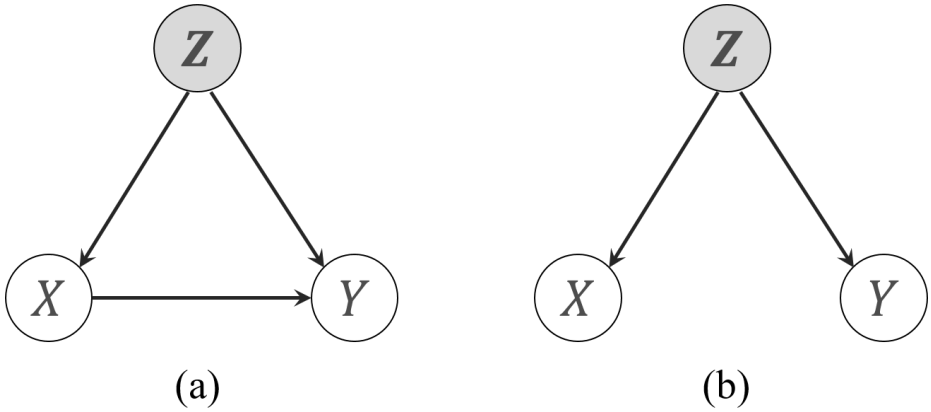

Figure 2.2: (a) Dependent model; (b) independent model

satisfying the assumption 2.1, the goal is to estimate a class of PDAG that are Markov equivalent to that structure.

The basic algorithm of this approach is as follows.

(1) Generates a complete undirected graph.

(2) Remove edges from the complete undirected graph generated in (1) by a Conditional Independence test (CI test).

(3) Orient the undirected graph obtained in (2) using a orientation rule (Pearl, 2000).

In general, the learning accuracy of a constraint-based approach depends on the accuracy of CI tests, and the learning speed depends on the number of CI test to be executed. As constraint-based approaches, PC algorithm (Spirtes et al., 2000), TPDA algorithm (Cheng et al., 2002), MMHC algorithm (Tsamardinos et al., 2006), and RAI algorithm (Yehezkel & Lerner, 2009) have been proposed. However, these algorithms use the χ^2 test, the G^2 test, and a conditional mutual information measure as CI tests, and do not have asymptotic consistency as defined below.

Definition 2.3 *When the number of data is $N \to \infty$, if a CI test determines true conditional independence with probability 1.0 for any variable, then the CI test has asymptotic consistency for independence.*

On the other hand, Steck et al. (2002) proposed a CI test using the Bayes factor, which defines the ratio of BDeu scores of the independent model and the dependent model between two variables. As an example, in a Bayesian network of two variables

X and Y, where $\mathbf{Z}$ is a set of common parent variables of X and Y, a dependent model is G_1 and a independent model is G_2, and they are shown in (a) and (b) of Figure 2.2, respectively. When the Bayes factor is $\mathrm{BF}(X, Y \mid \mathbf{Z})$, the Log Bayes factor can be expressed as

$$\log \mathrm{BF}(X, Y \mid \mathbf{Z}) = \log \frac{P(\mathbf{D} \mid G_1, \boldsymbol{\alpha})}{P(\mathbf{D} \mid G_2, \boldsymbol{\alpha})}, \tag{2.7}$$

where, $P(\mathbf{D} \mid G_1, \boldsymbol{\alpha})$ and $P(\mathbf{D} \mid G_2, \boldsymbol{\alpha})$ use BDeu from Equation (5).

In the CI test using Bayes factor, the choice between (a) and (b) in Figure 2.2 is determined by whether a log Bayes factor is greater than 0. However, Steck et al. (2002) used it only for theoretical analysis and not for the Bayesian network learning. On the other hand, Natori et al. (2015, 2017) showed that the CI test using Bayes factor have asymptotic consistency for independence.

Theorem 2.4 *When the number of data is $N \to \infty$,*

(1) If X and Y given $\mathbf{Z}$ are not the conditionally independent in the true structure, then $\log BF(X, Y \mid \boldsymbol{Z}) > 0$ with probability 1.0.

(2) If X and Y given $\mathbf{Z}$ are the conditionally independent in the true structure, $\log BF(X, Y \mid \boldsymbol{Z}) < 0$ with probability 1.0.

The proof of Theorem 2.4 is shown in (Natori, Uto, & Ueno, 2017).

3.2 RAI algorithm with transitivity

As mentioned above, the speed of constraint-based learning depends on the number of CI tests. In general, the number of CI tests can be reduced by removing edges early in the learning process. For this purpose, the following transitivity method has been proposed.

Theorem 2.5 *Let G be DAG, and for $X, Y \in \boldsymbol{V}$, Y is a non-descendant of X. If $A \in \boldsymbol{V} \setminus (\{X, Y\} \cup \boldsymbol{Pa}(X, G) \cup \boldsymbol{W})$ then,*

$$I_G(X, Y \mid \boldsymbol{Pa}(X, G)) \to I_G(X, A \mid \boldsymbol{Pa}(X, G)) \text{ or } I_G(A, Y \mid \boldsymbol{Pa}(X, G)), \tag{2.8}$$

where, $\boldsymbol{W}$ is a variable set consisting of a set of common child variables of X and Y and their descendants.

The proof of Theorem 2.5 is shown in (Honda et al., 2019). From Theorem 2.5, the conditional independence of any two variables guarantees at least one of their conditional independence from the other. Therefore, we can enumerate the edges that are guaranteed to have at least one conditional independence from the conditional independence between the two variables. By prioritizing CI test of enumerated edges, the edges can be removed earlier. Honda et al. (2019) reduced the number of CI tests by incorporating the transitive edge removal method into the RAI algorithm using Bayes factor, and achieved large structure learning with 3500 variables. In this paper, we use Honda's method to achieve structural learning of BNC on a larger than before.

Chapter 3

Proposed Method

An ANB was treated as a discriminative model, but Sugahara et al. (2018) and Sugahara & Ueno (2020) proposed a method for exact learning of the ANB as a generative model. Furthermore, they showed that high classification accuracy can be achieved using the learned ANB structure. In the constraint-based approach, it is also possible to learn the ANB structure by setting constraints on the initial structure and the scope of CI test execution. In this section, we first describe an algorithm for learning ANB structures based on the Honda's method. Furthermore, we show that the proposed method has asymptotic consistency as defined below.

Definition 3.1 *When the number of data is $N \to \infty$, if the structure estimated by a structure learning method is a almost sure convergence to the I-map with the smallest number of parameters in the ANB (called I-map ANB), then the method has asymptotic consistency for the ANB structure.*

1. Learning ANB using RAI algorithm

First, we introduce the basic operation of Honda's method (Honda et al., 2019). Let graph be $G = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of variables in G and $\mathbf{E}$ is a set of edges in G. And, G has both directed and undirected edges. Also, let $G_{ex} = (\mathbf{V}_{ex}, \mathbf{E}_{ex})$ be a subgraph partitioned by the RAI algorithm.

(1) Input a complete undirected graph G_{uc} and the data $\mathbf{D}$.

(2) When X and Y are determined to be the conditionally independent by the CI test of each order, the edges between X and Y are removed. And, immediately after the deletion of edges by the CI test after the first order, we remove the edges based on transitivity.

Algorithm 1 CI test for ANB

1: **function** CI-TEST($n_z, G_s, \mathbf{G}_{ex}, X_0, \mathbf{D}$)
   n_z: order of CI tests
   $G_s = (\mathbf{V}_s, \mathbf{E}_s)$: input graph
   $\mathbf{G}_{ex}$: set of subgraphs
   $G_{all} = (\mathbf{V}, \mathbf{E})$: output graph obtained by CI tests and direction
   X_0: class variable
   // remove edges by CI tests
2: **for** $G_{ex} = (\mathbf{V}_{ex}, \mathbf{E}_{ex}) \in \mathbf{G}_{ex}$ **do**
3: **for** $X \in \mathbf{V}_s$, $Y \in \mathbf{V}_{ex}$ **do**
4: **for** $\mathbf{Z} \subseteq \mathbf{Pa}_p(X, G_s) \cup \mathbf{Pa}(X, \mathbf{G}_{ex}) \setminus \{Y\}$ **do**
5: **if** $|\mathbf{Z}| = n_z - 1$ and $\log \mathrm{BF}(X, Y \mid \mathbf{Z} \cup \{X_0\}) < 0$ **then**
6: $\mathbf{E}_{all} \leftarrow \mathbf{E}_{all} \setminus \{E_{XY}\}$ ▷ E_{XY}: edge between X and Y
   //remove edges by transitivity
7: TRANSITIVE_CUT($G_s, G_{all}, X, Y, \mathbf{Z}, X_0, \mathbf{D}$)
8: **end if**
9: **end for**
10: **end for**
11: **end for**
12: **for** $X \in \mathbf{V}_s$, $Y \in \mathbf{V}_s$ **do**
13: **for** $\mathbf{Z} \subseteq \mathbf{Pa}_p(X, G_s) \cup \mathbf{Pa}(X, \mathbf{G}_{ex}) \setminus \{Y\}$ **do**
14: **if** $|\mathbf{Z}| = n_z - 1$ and $\log \mathrm{BF}(X, Y \mid \mathbf{Z} \cup \{X_0\}) < 0$ **then**
15: $\mathbf{E}_{all} \leftarrow \mathbf{E}_{all} \setminus \{E_{XY}\}$, $\mathbf{E}_s \leftarrow \mathbf{E}_s \setminus \{E_{XY}\}$
   //remove edges by transitivity
16: TRANSITIVE_CUT($G_s, G_all, X, Y, \mathbf{Z}, X_0, \mathbf{D}$)
17: **end if**
18: **end for**
19: **end for**
20: **return** (G_s, G_{all})
21: **end function**

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

(4) Partition the graph into the subgraphs G_{ex} based on the direction.

(5) Recursively invokes the RAI on each subgraph.

The CI test using Bayes factor asymptotically determines true conditional independence, and Honda's method asymptotically converges to the true structure (Honda et al., 2019). We now explain how to learn the ANB using Honda's method described above. In the above algorithm, we implement the ANB learning using the RAI algorithm by modifying the initial graph in step (1) and the CI test in step (2). Algorithm1 shows the CI test using Bayes factor in the RAI algorithm.

First, in step (1), the initial graph to be input is not the complete undirected graph of all variable sets, but the complete undirected graph of all feature variable sets. Next, in step (2), since the feature variables always have the class variable as a parent in the ANB, we add a constraint to perform CI tests between two variables given X_0. Let input graph be $G = (\mathbf{V}_s, \mathbf{E}_s)$. In addition, let $\mathbf{Adj}(X, G)$ be a set of adjacent variables of X in G, and $\mathbf{Ch}(X, G)$ is a set of child variables of X in G. Then, $\mathbf{Pa}_p(X, G) = \mathbf{Adj}(X, G) \setminus \mathbf{Ch}(X, G)$, and $\mathbf{Pa}(X, G)$ is the set of parent variables of the variable X in G. Also, $\mathbf{Pa}(X, \mathbf{G}) = \cup_{\mathbf{G}\in\mathbf{G}}\mathbf{Pa}(X, G)$, where $\mathbf{G}$ is a set of graph. The function CI_test removes the edge E_{XY} between X and Y for the set of variables $\mathbf{V}_s$ in the input graph when X and Y are determined to be conditionally independent in the CI test of each order. In the ANB, the class variables and the feature variables are always connected, so the class variable X_0 is always included in the conditional part when conducting the CI test. In other words, perform the CI test between X and Y given $\mathbf{Z} \cup \{X_0\}$ (lines 5 and 14). Also, call the function TRANSITIVE_CUT immediately after the edge are removed according to the CI test (lines 7 and 16). This function removes the edges by transitivity. The details are shown in Algorithm 2. In the function TRANSITIVE_CUT, the CI test is performed on a set of variables $\mathbf{A}$ detected by transitivity from the conditional independence estimated by the CI test, and as before, the class variable X_0 is always included in the conditional part. By making these changes, a PDAG consisting of a set of feature variables is output. Finally, a PDAG with the ANB structure is output by connecting edges from the class variables to all feature variables.

2. I-map ANB with minimum number of parameters

In this section, we show that the proposed method has asymptotic consistency for the ANB structure. First, we prove the following three Lemmas. Let $\mathbf{V} = \{X_0, X_1, ..., X_n\}$ be a set of variables, and X_0 is a class variable, and $X_1, ..., X_n$ is feature variables. Let $G^* = (\mathbf{V}, \mathbf{E}^*)$ be a true Bayesian network structure and $G_{ANB} = (\mathbf{V}, \mathbf{E}_{ANB})$ is the ANB structure learned by the proposed algorithm.

Algorithm 2 Edge cutting with transitivity for ANB

function TRANSITIVE_CUT($G_s, G, X, Y, \mathbf{Z}, X_0, \mathbf{D}$)
$G_s = (\mathbf{V}_s, \mathbf{E}_s)$:autonomous sub-structure
$G = (\mathbf{V}, \mathbf{E})$: entire structure
$X, Y, \mathbf{Z}$: two variables X, Y and a set of variables $\mathbf{Z}$ such that $I_P(X \perp Y | \mathbf{Z})$
X_0: class variable
  $\mathbf{A} \leftarrow \mathbf{Adj}(X, G) \cap \mathbf{Adj}(Y, G) \setminus (\mathbf{Ch}(X, G) \cap \mathbf{Ch}(Y, G) \cup \mathbf{Z} \cup \{X_0\})$
  for $A \in \mathbf{A}$ **do**
    if $\log \mathrm{BF}(X, A \mid \mathbf{Z} \cup \{X_0\}) < 0$ **then**
      $\mathbf{E}_s \leftarrow \mathbf{E}_s \setminus \{E_{AY}\}, \mathbf{E} \leftarrow \mathbf{E} \setminus \{E_{AY}\}$
    else
      $\mathbf{E} \leftarrow \mathbf{E} \setminus \{E_{XA}\}$ ▷ E_{XA}: edge between X and A
    end if
    if $\log \mathrm{BF}(A, Y \mid \mathbf{Z} \cup \{X_0\}) < 0$ **then**
      if $A \in \mathbf{V}_s$ and $Y \in \mathbf{V}_s$ **then**
        $\mathbf{E}_s \leftarrow \mathbf{E}_s \setminus \{E_{AY}\}, \mathbf{E} \leftarrow \mathbf{E} \setminus \{E_{AY}\}$
      else
        $\mathbf{E} \leftarrow \mathbf{E} \setminus \{E_{AY}\}$
      end if
    end if
  end for
  return (G_s, G)
end function

Lemma 3.1 *When the number of data $N \to \infty$, for $\forall X, Y \in \boldsymbol{V}$, if $E_{XY} \in \boldsymbol{E}^*$ then $E_{XY} \in \boldsymbol{E}_{ANB}$.*

Proof: If $E_{XY} \in \mathbf{E}^*$, then $\forall \mathbf{Z} \subseteq \mathbf{V} \backslash \{X, Y\}$, $D_P(X, Y | \mathbf{Z})$. Thus, $\forall \mathbf{Z} \subseteq \mathbf{V} \backslash \{X, Y, X_0\}$, $D_P(X, Y | \mathbf{Z} \cup \{X_0\})$. From Theorem 2.4, the CI test performed by the proposed algorithm does not detect the independence between X and Y, so the edge E_{XY} is not removed. Therefore, $E_{XY} \in \mathbf{E}_{ANB}$. □

Lemma 3.2 *When the number of data $N \to \infty$, for $\forall X, Y, Z \in \boldsymbol{V}$, if G^* has head-to-head $X \to Z \leftarrow Y$, then $E_{XY} \in \boldsymbol{E}_{ANB}$ or G_{ANB} has head-to-head $X \to Z \leftarrow Y$.*

Proof: It is divided into the case where Z is the class variable and the other cases.

1. the case $Z = X_0$:
   $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}$, $D_P(X, Y | \mathbf{Z} \cup \{X_0\})$, because G^* has head-to-head $X \to$

$X_0 \leftarrow Y$. From Theorem 2.4, the CI test performed by the proposed algorithm does not detect independence between X and Y, so the edge E_{XY} is not removed. Therefore, $E_{XY} \in \mathbf{E}_{ANB}$.

2. the case $Z \neq X_0$: It is divided into the case where X or Y is the class variable and the other cases.

   (a) the case $X = X_0$ or $Y = X_0$ **:**
   From the ANB assumption, the class variable is the parent of all feature variable, so $E_{XY} \in \mathbf{E}_{ANB}$.

   (b) the case $X \neq X_0$ and $Y \neq X_0$:
   The cases are divided into the following contrary events.

      i. the case $\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $I_P(X, Y|\mathbf{Z} \cup \{X_0\})$:
      From Theorem 2.4, the proposed algorithm detects independence and removes the edge E_{XY} by the CI test. X and Y given Z are dependent because G^* has head-to-head $X \rightarrow Z \leftarrow Y$. Therefore, by the orientation rule, G_{ANB} has head-to-head $X \rightarrow Z \leftarrow Y$.

      ii. the case $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $D_P(X, Y|\mathbf{Z} \cup \{X_0\})$ (3.1) :
      X and Y given Z are dependent because G^* has head-to-head $X \rightarrow Z \leftarrow Y$. Thus, $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $D_P(X, Y|\mathbf{Z} \cup \{Z, X_0\})$. From this and equation (3.1), $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}$, $D_P(X, Y|\mathbf{Z} \cup \{X_0\})$. From Theorem 2.4, the CI test performed by the proposed algorithm does not detect independence between X and Y, so the edge E_{XY} is not removed. Therefore, $E_{XY} \in \mathbf{E}_{ANB}$.

From (1) and (2), for $\forall X, Y, Z \in \mathbf{V}$, if G^* has head-to-head $X \rightarrow Z \leftarrow Y$, then $E_{XY} \in \mathbf{E}_{ANB}$ or G_{ANB} has head-to-head $X \rightarrow Z \leftarrow Y$. □

Lemma 3.3 *When the number of data $N \rightarrow \infty$, for $\forall X, Y, Z \in \boldsymbol{V}$, if G^* has head-to-tail $X \rightarrow Z \rightarrow Y$ or tail-to-tail $X \leftarrow Z \rightarrow Y$, then $E_{XY} \in \boldsymbol{E}_{ANB}$ or G_{ANB} has head-to-tail $X \rightarrow Z \rightarrow Y$ or tail-to-tail $X \leftarrow Z \rightarrow Y$.*

Proof: It is divided into the case where Z is the class variable and the other cases.

1. the case $Z = X_0$: $\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}$, $I_P(X, Y|\mathbf{Z} \cup \{X_0\})$ because G^* has head-to-tail $X \rightarrow X_0 \rightarrow Y$ or tail-to-tail $X \leftarrow X_0 \rightarrow Y$. From Theorem 2.4, the proposed algorithm detects independence and removes the edge E_{XY} with probability 1.0 by the CI test when $N \rightarrow \infty$. From the ANB assumption, the class variable is the parent of all feature variable. therefore, G_{ANB} has tail-to-tail $X \leftarrow X_0 \rightarrow Y$.

2. the case $Z \neq X_0$:
   It is divided into the case where X or Y is the class variable and the other cases.

   (a) the case $X = X_0$ or $Y = X_0$:
   From the ANB assumption, the class variable is the parent of all feature variable, so $E_{XY} \in \mathbf{E}_{ANB}$.

   (b) the case $X \neq X_0$ and $Y \neq X_0$:
   The cases are divided into the following contrary events.

      i. the case $\exists \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $I_P(X, Y|\mathbf{Z} \cup \{Z, X_0\})$:
      From Theorem 2.4, the proposed algorithm detects independence and removes the edge E_{XY} with probability 1.0 by the CI test when $N \to \infty$. X and Y not given Z are dependent because G^* has head-to-tail $X \to Z \to Y$ or tail-to-tail $X \leftarrow Z \to Y$. Therefore, by the orientation rule, G_{ANB} has head-to-tail $X \to Z \to Y$ or tail-to-tail $X \leftarrow Z \to Y$.

      ii. the case $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $D_P(X, Y|\mathbf{Z} \cup \{Z, X_0\})$ $(3, 2)$:
      X and Y not given Z are dependent because G^* has head-to-tail $X \to Z \to Y$ or tail-to-tail $X \leftarrow Z \to Y$. Thus, $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, Z, X_0\}$, $D_P(X, Y|\mathbf{Z}, \cup\{X_0\})$. From this and equation (3,2), $\forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y, X_0\}$, $D_P(X, Y|\mathbf{Z} \cup \{X_0\})$. From Theorem 2.4, the CI test performed by the proposed algorithm does not detect independence between X and Y, so the edge E_{XY} is not removed. Therefore, $E_{XY} \in \mathbf{E}_{ANB}$.

From (1) and (2), for $\forall X, Y, Z \in \mathbf{V}$, $E_{XY} \in \mathbf{E}_{ANB}$ or G_{ANB} has head-to-tail $X \to Z \to Y$ or tail-to-tail $X \leftarrow Z \to Y$. □

From the above three Lemmas, we prove the following.

Theorem 3.1 *When the number of data $N \to \infty$, The structure G_{ANB} learned by the proposed algorithm is a almost sure convergence for the I-map ANB.*

Proof: For Lemma 3.1, 3.2, 3.3, G_{ANB} satisfies each of the three conditions in Theorem 2.3 with probability 1.0 when $N \to \infty$. Therefore, the structure G_{ANB} learned by the proposed algorithm is the almost sure convergence to the I-map ANB. □

Furthermore, as the following theorem shows, G_{ANB} is almost sure convergence to the I-map ANB with the minimum number of parameters.

Theorem 3.2 *The proposed algorithm has asymptotic consistency for the ANB structure.*

Proof: From Theorem 3.1, when $N \to \infty$, G_{ANB} is the almost sure convergence to I-map ANB. From Theorem 2.4, when $N \to \infty$, the CI test of the proposed algorithm detects true conditional independence with probability 1.0, so all edges between variables that are truly conditionally independent given X_0 are removed. Thus, G_{ANB} has the smallest number of edges in the I-map ANB when $N \to \infty$. Therefore, G_{ANB} is the almost sure convergence to the I-map ANB with the minimum number of parameters. □

Therefore, we can expect that the proposed algorithm can achieve the same level of accuracy as the exact learning ANB in a shorter computation time.

On the other hand, if we assume the ANB, the number of parameters increases compared to the GBN because it forces the subtraction of edges from the class variables to the feature variables. In this case, the convergence to the true value of the joint probability distribution represented by the estimation structure should theoretically be slower than that of the GBN. However, the GBN is known to have unstable estimation accuracy when the number of parent variables of the class variable is large, as the prior distribution parameter of the class variable increases exponentially (Sugahara, Uto, & Ueno, 2018; Sugahara & Ueno, 2020). By assuming the ANB, the number of parameters becomes redundant, but the number of the parent variables of the class variables is suppressed and the prior distribution can be estimated robustly, which is expected to improve the classification accuracy.

Chapter 4

Experiments

1. Comparsion with BNC

In this section, we conduct the following evaluation experiments to show the effectiveness of the proposed method. First, we show that the proposed method can learn a larger network than the exact learning approach by simulation experiment. Next, we compare the classification accuracy of the proposed method and other BNC learning methods using real data.

In this section, we compare the following six methods.

- Naive Bayes
- TAN: learning TAN method by maximizing log likelihood.
- exact-GBN: Exact learning GBN method by maximizing BDeu.
- exact-ANB: Exact learning ANB method by maximizing BDeu.
- RAI-GBN: Learning GBN using the Honda's method
- RAI-ANB: Learning GBN using the proposed method

In this paper, the proposed method is RAI-ANB. the TAN was learned using the Friedman's method (Friedman et al., 1997). The exact-GBN and the exact-ANB were learned using the exact learning method (Silander & Myllymäki, 2006) with BDeu. The value of the pseudo-sample (hyperparameter) for the BDeu score and the Bayes factor was set to 1.0 to maximize the posterior variance as suggested by Ueno (2010, 2011). In all methods, all parameters of the BNC after the structure learning were estimated by EAP. The computational environment for each method is shown in Table 1.

Table 4.1: Computational environment

Naive Bayes	
CPU	2.10GHz 8-Cores Intel XEON
System Memory	128GB
OS	OS ubuntu 16.04.4 lts
Software	Python
TAN, exact-GBN, exact-ANB	
CPU	2.10GHz 8-Cores Intel XEON
System Memory	128GB
OS	OS ubuntu 16.04.4 lts
Software	JAVA
RAI-GBN, RAI-ANB	
CPU	2.10GHz 8-Cores Intel XEON
System Memory	128GB
OS	OS ubuntu 16.04.4 lts
Software	MATLAB

1.1 Experiments using random networks

In this section, we perform simulation experiment using random networks to show that the proposed method can learn larger networks than the exact learning approach. Random networks are generated using the BNGenerator, which uses a Markov chain Monte Carlo method to randomly generate networks from a uniform distribution(Ide & Cozman, 2002; Ide, Cozman, & Ramos, 2004). In this section, the number of variables is set to {5, 10, 20, 50, 100, 200, 500, 1000} and the maximum order of each network is set to 5. In addition, we generate 10000 data from the generated networks, and calculate the computation time for each method by learning the structure of each network. However, we set a time limit of 6 hours and terminated the learning process if the time exceeded the limit.

The results are shown in Figure 4.1. The horizontal axis represents the number of variables and the vertical axis represents the computation time (in seconds). From Figure 4.1, we can see that the computation time increases as the number of variables increases, except for Naive Bayes. Next to Naive Bayes, the computation time for TAN is short. This is because TAN can be computed in polynomial time (Friedman et al., 1997; Madden, 2009). Next, in both exact-GBN and exact-ANB, the networks with more than 50 variables did not finish training in time. On the other hand, the

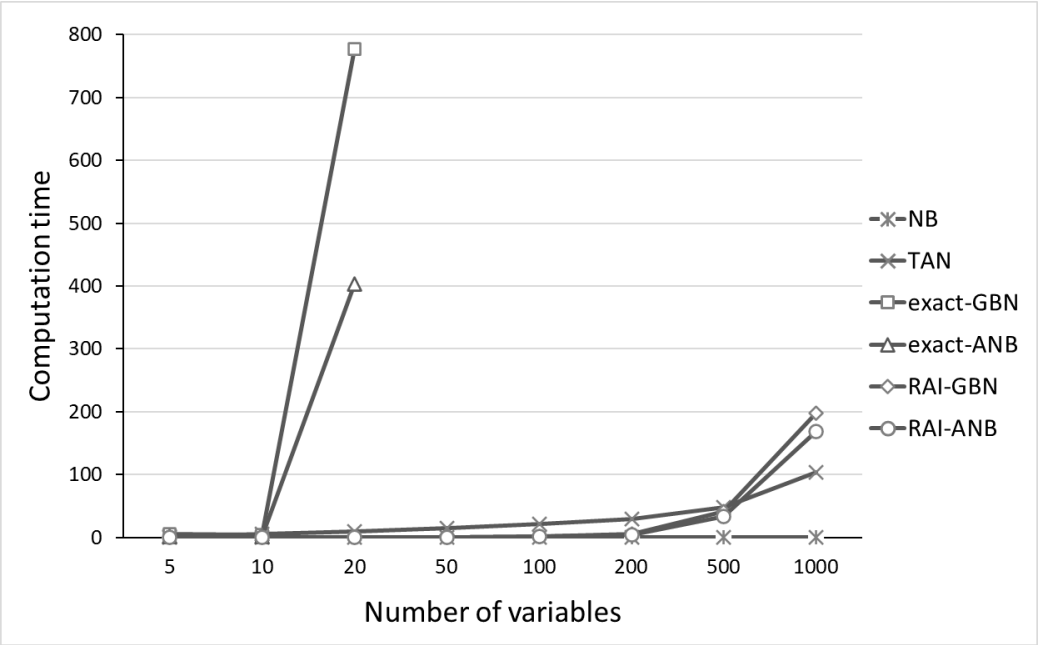

Figure 4.1: Relationship between the number of variables and computation time

constraint-based methods, RAI-GBN and RAI-ANB, were able to learn the structure of networks with 1000 variables. Besides, the proposed method, RAI-ANB, completes the structure learning in a shorter time than RAI-GBN. This is because RAI-GBN judges the independence among all variables, while RAI-ANB only needs to judge the independence among feature variables. These results show that the proposed method can learn larger networks than the exact learning approach. In addition, we showed that the proposed method has a shorter computation time than RAI-GBN.

1.2 Experiments using real data

In this section, we compare the classification accuracy and computation time using real data to demonstrate the significance of the proposed method. First, we compare the classification accuracy of small networks to compare the proposed method with the exact learning approach. Next, we compare the classification accuracies of the large networks that cannot be exact learning. In this experiment, we used the datasets registered in 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 10-fold cross validation. In order to show the significance of the proposed method, the p-value is obtained

Table 4.2: Accuracies of respective classifiers for small networks

	dataset	variables	number of data	classes	Naive Bayes	TAN	exact-GBN	exact-ANB	RAI-GBN	RAI-ANB
1	Balance	5	625	3	**0.9168**	0.8640	**0.9168**	**0.9168**	0.6352	**0.9168**
2	banknote	5	1372	2	0.8433	**0.8819**	0.8812	0.8812	0.8776	0.8812
3	Hayes-Roth	5	132	3	**0.8484**	0.6808	0.5610	**0.8484**	0.5132	**0.8484**
4	iris	5	150	3	0.7133	**0.8267**	**0.8267**	0.8200	0.8133	**0.8267**
5	lenses	5	24	3	0.6833	0.6833	**0.8500**	0.6833	**0.8500**	0.6833
6	Car	7	1728	4	0.8583	0.9381	0.9415	**0.9421**	0.8137	0.9410
7	liver	7	345	2	0.6461	0.6402	0.6200	0.6402	0.5830	**0.6634**
8	monk1	7	432	2	0.7500	**1.0000**	**1.0000**	**1.0000**	0.7500	**1.000**
9	mux6	7	64	2	0.5762	**0.6357**	0.5500	0.5762	0.3262	0.5333
10	led7	8	3200	10	0.7294	**0.7306**	0.7294	0.7294	0.7056	0.7294
11	HTRU2	9	17898	2	0.8966	**0.9141**	0.9112	**0.9141**	0.9027	0.9084
12	Nursery	9	12960	3	0.9033	0.9250	0.9340	0.9181	0.8921	**0.9356**
13	pima	9	768	9	0.7031	0.7175	**0.7279**	0.7175	0.7122	0.7032
14	post	9	87	5	0.6764	0.5986	**0.7125**	0.6764	**0.7125**	0.6764
15	Breast Cancer	10	277	2	**0.7443**	0.7187	0.7295	0.7119	0.6972	0.7192
16	Breast Cancer Wisconsin	10	683	2	**0.9752**	0.9649	**0.9752**	**0.9752**	0.9254	**0.9752**
17	cmc	10	1473	3	0.4657	0.4704	0.4548	0.4677	0.4358	**0.4752**
18	glass	10	214	6	0.5524	0.5431	0.5656	**0.6361**	0.5890	0.6087
19	shuttle-small	10	5800	6	0.9384	0.9566	0.9693	**0.9716**	0.9659	0.9707
20	threeOf9	10	512	2	0.8144	0.8477	**0.8865**	0.8673	0.7071	0.8399
21	TicTac	10	958	2	0.6919	0.7567	0.8319	**0.8549**	0.6992	0.7546
22	magic	11	19020	2	0.7482	0.7768	0.7873	**0.7874**	0.7801	0.7700
23	Flare	11	1389	9	0.7804	0.7948	**0.8431**	0.8229	**0.8431**	0.8236
24	heart	14	270	2	**0.8259**	**0.8259**	**0.8259**	0.8185	0.7815	**0.8370**
25	wine	14	178	3	**0.9330**	0.9275	0.9327	0.9216	0.8938	**0.9330**
26	cleve	14	296	2	**0.8410**	0.8338	0.7900	0.8344	0.7798	0.8308
27	australian	15	690	2	0.8290	0.8348	0.8536	0.8246	**0.8551**	0.8377
28	crx	15	653	2	0.8393	0.8531	0.8592	0.8531	**0.8639**	0.8531
29	EEG	15	14980	2	0.5778	0.6305	0.6814	**0.6864**	0.6411	0.6697
30	Congressional	17	232	2	0.9092	0.9478	**0.9652**	0.9478	**0.9652**	0.9522
31	zoo	17	101	5	**0.9800**	0.9600	0.9400	0.9600	0.9000	**0.9800**
32	pendigits	17	10992	10	0.8032	0.8504	**0.9289**	0.9278	0.8790	0.9085
33	letter	17	20000	26	0.4466	0.4868	0.5761	**0.5935**	0.5448	0.5609
34	ClimateModel	19	540	2	0.9222	**0.9315**	0.9000	0.8426	0.8963	0.9222
35	ImageSegmentation	19	2310	7	0.7290	0.7515	0.8156	**0.8225**	0.7758	0.8039
36	lymphography	19	148	4	**0.8386**	0.7648	0.7586	0.8186	0.7033	**0.8386**
37	vehicle	19	846	4	0.4339	0.5722	0.5732	**0.6241**	0.5543	0.5793
38	hepatitis	20	80	2	**0.8625**	0.8375	0.5875	0.6250	0.7375	**0.8625**
39	german	21	1000	2	**0.7430**	0.7310	0.7210	0.7380	0.6830	0.7390
40	bank	21	30488	2	0.8544	0.8774	**0.8956**	0.8949	0.8939	0.8907
41	waveform-21	22	5000	3	0.7886	0.7896	0.7846	**0.7966**	0.7336	0.7826
42	Mushroom	22	5644	2	0.9957	**1.0000**	0.9949	**1.0000**	**1.0000**	**1.0000**
43	spect	23	263	2	0.8013	0.8128	0.7450	**0.8164**	0.7946	0.8051
	classification accuracy	average			0.7770	0.7927	0.7985	0.8071	0.7583	**0.8086**
		p-value			0.00004	0.00126	0.46812	0.46812	0.00004	-
	calculation time (s)	average			0.00	2.58	1790.93	500.76	11.94	1.57
		standard error			0.00	0.16	895.76	252.69	9.55	0.6.2

by the multiple comparison using a Hommel's test (Hommel, 1988), and the average computation time for structure learning is obtained for each method and each dataset.

Table 4.3: The number of Max parents of respective classifiers for small networks

	dataset	Naive Bayes	TAN	exact-GBN	exact-ANB	RAI-GBN	RAI-ANB
1	Balance	1	2	1	1	1	1
2	banknote	1	2	4	4	3.9	4
3	Hayes-Roth	1	2	3	1	1.8	1
4	iris	1	2	2	2	2.7	2
5	lenses	1	2	1.3	1.1	2	1.1
6	Car	1	2	2	2	3	2
7	liver	1	2	2	2	1.5	2
8	monk1	1	2	3	3	1	2
9	mux6	1	2	5.8	1	0.7	1
10	led7	1	2	1	1	1.8	1
11	HTRU2	1	2	3	4	5	4
12	Nursery	1	2	4	3	3	3
13	pima	1	2	2	2	2.1	2.3
14	post	1	2	0.2	1	0.3	1
15	Breast Cancer	1	2	1	2	1	2
16	Breast Cancer Wisconsin	1	2	1	1	1.1	1
17	cmc	1	2	2	2.5	2	2.1
18	glass	1	2	2.9	3	2	2.3
19	shuttle-small	1	2	5	5	5	3.7
20	threeOf9	1	2	5	2.7	4.4	2
21	TicTac	1	2	3	3	1.7	2
22	magic	1	2	4	4	4	5
23	Flare	1	2	2	3	1.6	3
24	heart	1	2	2	2	2	2
25	wine	1	2	2.2	2.1	3.2	2.1
26	cleve	1	2	2	2	2	2
27	australian	1	2	2.4	2.9	2	2.3
28	crx	1	2	3	2.2	1.6	2
29	EEG	1	2	5	5	4.9	5.3
30	Congressional	1	2	3.5	4	2.3	3
31	zoo	1	2	4.9	4.9	3.7	3
32	pendigits	1	2	5.5	5.6	8	5.9
33	letter	1	2	6	5	7.6	5.3
34	ClimateModel	1	2	14	14.1	3.1	1
35	ImageSegmentation	1	2	4.1	4	6	5
36	lymphography	1	2	8.7	9.9	2.1	2.3
37	vehicle	1	2	4.2	4.1	3.5	3.6
38	hepatitis	1	2	10.4	11.4	2.4	2.9
39	german	1	2	2	3	2	3
40	bank	1	2	5	6	5.4	5.4
41	waveform-21	1	2	4	4	5	3.7
42	Mushroom	1	2	2.4	7.6	4.8	4.8
43	spect	1	2	2.7	3	2.6	3.2

1.2.1 Classification accuracy in small networks

In this section, we compare the classification accuracy of BNCs learned by each method using datasets to which the exact learning approach can be applied. For this experiments, we used 43 datasets with 5 to 23 variables. Table 4.2 shows the classification accuracy of each method for each dataset. In "classification accuracy" shown at the bottom of Table 4.2, "average" is the average classification accuracy of each method for all datasets, and "p-value" is the p-value obtained by multiple comparison. In "computation time" is the average computation time for structure learning of each method for all datasets. "standard error" is the standard error of

Table 4.4: The number of edges used to estimate the class variable of respective classifiers for small networks

	dataset	Naive Bayes	TAN	exact-GBN	exact-ANB	RAI-GBN	RAI-ANB
1	Balance	4	7	4	4	1	4
2	banknote	4	7	7	10	6.8	9.8
3	Hayes-Roth	4	7	3	4	1.7	4
4	iris	4	7	4.1	7	3.1	5.2
5	lenses	4	7	2.1	4.1	2	4
6	Car	6	11	7	9	3	6
7	liver	6	11	3.8	10.6	1.1	9
8	monk1	6	11	3	8	1	6
9	mux6	6	11	5.8	6	0.4	6
10	led7	7	13	7	7	5.1	7
11	HTRU2	8	15	12.5	20	5	18
12	Nursery	8	15	8	13	3	8
13	pima	8	15	4.2	15	2.2	12.4
14	post	8	15	0	8	0	8
15	Breast Cancer	9	17	1	13	0.1	11.1
16	Breast Cancer Wisconsin	9	17	8.9	9	1	9
17	cmc	9	17	1.7	16.1	1	13.2
18	glass	9	17	4.3	15.5	3.1	13
19	shuttle-small	9	17	15	23.8	4.4	22
20	threeOf9	9	17	9.7	13.4	4.5	9
21	TicTac	9	17	7.4	18.9	1	12.3
22	magic	10	19	20.4	30	13	26.8
23	Flare	10	19	1	18.9	0.9	17.4
24	heart	13	25	6.6	18.4	2	17.1
25	wine	13	25	9.5	19	3.2	16
26	cleve	13	25	7.5	18.3	2	16.3
27	australian	14	27	6.2	24.1	3.6	20.2
28	crx	14	29	5.3	23.9	2.5	21.2
29	EEG	14	27	34.2	57.5	11.8	48.6
30	Congressional	16	31	7.1	37.1	3.4	27.7
31	zoo	16	31	9.4	36.9	3.9	24.9
32	pendigits	16	31	63.4	66.5	9.2	60.5
33	letter	16	31	41.4	57.9	17.7	51.1
34	ClimateModel	18	35	32.1	69.7	3	18
35	ImageSegmentation	18	35	31.5	48	5.8	38.5
36	lymphography	18	35	16.6	36.7	1.8	23.6
37	vehicle	18	35	14.3	50.8	6.3	41.1
38	hepatitis	19	37	31.6	78.1	1.4	28.2
39	german	20	39	4.1	33.3	1	28.8
40	bank	20	39	13.1	63.9	5.8	47.6
41	waveform-21	21	41	39.8	60.3	5.9	42.9
42	Mushroom	21	41	6.7	83	17.1	61.3
43	spect	22	43	9.3	49.2	2.1	44.8

the computation time of each method. Next, we introduce four metrics to compare each method. Table 4.3 shows the average of a Max parents of each method for each dataset. Max parents is the average of the maximum number of parent variables that a variable in the structure learned by each method in the 10-fold cross-validation method, and the higher the value of Max parents, the more complex the structure (Ling & Zhang, 2003). Table 4.4 shows the average number of edges involved in the estimation of the class variable for each method for each dataset. The higher the

value, the more complex the structure of the Markov blanket of the class variable has been learned.

First, Naive Bayes and TAN limit the number of parent variables taken by feature variables, so Max Parents are fixed at 1 and 2. Naive Bayes does not require structural learning, so the computation time is 0. In addition, TAN can be computed in polynomial time, so its computation time is shorter than other methods (Friedman et al., 1997; Madden, 2009).

Next, we compared the proposed method with exact-ANB and found that the classification accuracy of the proposed method was comparable to that of exact-learned ANB, although we could not show any significant difference. However, exact-ANB requires more computation time than the proposed method. Also, from Table 4.4, the number of edges involved in the estimation of the class variable of the proposed method is smaller than that of exact-ANB for all datasets, which means that the proposed method tends to learn a sparser structure than exact-ANB. On datasets No. 31, 36, and 38, the classification accuracy of the proposed method is higher than that of exact-ANB. Table 4.2 shows that the number of these data is small. In addition, Table 4.3 and Table 4.4 show that the number of edges involved in the estimation of Max parents and the class variable in the exact-ANB is smaller than the proposed method. Because of the small number of data and the complex structure learned, the exact learning approach is considered to have low classification accuracy for the class variables. On the other hand, the proposed method tends to learn a sparser structure than the exact laerning approach, so it is thought that the classification accuracy did not decrease much even though the number of data was small.

Comparing RAI-GBN and RAI-ANB, assuming ANB in the structure improves the classification accuracy as well as the exact learning approach; RAI-GBN has the lowest classification accuracy among the compared methods. This is due to the accuracy of the CI test; the CI test using Bayes factor may remove wrong edges when the number of data is small. This may result in learning networks with small Markov blankets of class variables. In fact, Table 4.4 shows that the number of edges involved in the estimation of the class variables is smaller than the other methods, and that the proposed method learns the structure where the class variable and feature variables are almost independent in No. 9, 14, and 15. In contrast, the proposed method has all feature variables as children, so the number of Markov blankets is always the same as the number of the feature variables, and the classification accuracy is improved by inferring using all feature variables. In addition, since the proposed method only performs CI test among feature variables, it takes less computational time than RAI-GBN, which performs CI test among all variables.

Table 4.5: Accuracies of respective classifiers for huge networks

	dataset	variables	num of data	classes	Naive Bayes	TAN	RAI-GBN	RAI-ANB
1	kr-vs-kp	37	3196	2	0.8774	0.9240	0.9402	**0.9524**
2	Connect-4	43	67557	3	0.7213	0.7643	0.7467	**0.7938**
3	Flowmeters D	44	180	4	0.8389	0.8389	0.8389	**0.8500**
4	movement libras	91	360	15	0.5028	**0.5389**	0.2278	0.5333
5	dota2	117	102944	2	**0.5981**	0.5810	0.5564	0.5957
6	Musk1	167	478	2	0.6517	0.7566	0.5756	**0.7986**
7	Musk2	167	6598	2	0.7445	0.8406	0.9047	**0.9615**
8	Epileptic Seizure	179	11500	5	0.2344	0.3650	0.1187	**0.3808**
9	mfeat-fac	219	2000	10	0.3520	0.4590	0.3310	**0.4650**
10	semeion	257	1600	10	0.8550	0.8719	0.3521	**0.8776**
11	madelon	501	2000	2	**0.5905**	0.5270	0.5740	**0.5905**
12	HART	563	10929	12	0.7967	0.8685	0.8456	**0.8805**
13	HAR	563	10929	6	0.7633	0.8797	0.8657	**0.8987**
14	Parkinson's Disease	755	756	2	0.7182	0.7898	0.7419	**0.7964**
15	MNIST	785	70000	10	0.8258	0.8911	0.9482	**0.9493**
16	MicroMass	1301	360	10	0.9472	0.9472	0.8756	**0.9556**
	classification accuracy	average			0.6886	0.7402	0.6527	**0.7675**
		p-value			0.0080	0.0060	0.0022	-
	calculation time (s)	average			0.0	545.7	3647.2	565.7
		standard error			0.0	434.6	1966.1	220.9

1.2.2 Classification accuracy in large networks

In this section, we describe an evaluation experiment using a large dataset that cannot be handled by the exact learning approach. Using 15 datasets with 36-1300 variables, we compare the classification accuracies of BNCs learned by the traditional method with those learned by the proposed method. The exact learning approach is excluded from the comparison because it is not applicable. As in previous section, Table 4.5 shows the average and p-value of the classification accuracy, and the average and the standard error of the computation time for each method on each dataset.

From Table 4.5, the classification accuracy of the proposed method was the highest. The classification accuracy of the proposed method is significantly higher than Naive Bayes, TAN, and RAI-GNB at the 5% level of significance. Furthermore, the proposed method was able to learn the BNC with large networks that cannot be learned by dynamic programming. Similar to the results of computation time for small networks, the computation time of the proposed method is shorter than that of RAI-GBN and longer than that of Naive Bayes and TAN. The reason for this is as described in the previous section.

Table 4.6: The number of Max parents of respective classifiers for huge networks

	datset	Naive Bayes	TAN	RAI-GBN	RAI-ANB
1	kr-vs-kp	1	2	6	6.2
2	Connect-4	1	2	5.1	5.5
3	Flowmeters D	1	2	3.6	4
4	movement libras	1	2	2.5	3.1
5	dota2	1	2	3.4	4
6	Musk1	1	2	4.7	5
7	Musk2	1	2	9.3	10.3
8	Epileptic Seizure	1	2	3	3
9	mfeat-fac	1	2	5.1	5.1
10	semeion	1	2	5	4
11	madelon	1	2	3.5	4.4
12	HAPT	1	2	5.8	5
13	HAR	1	2	6.5	6.8
14	Parkinson's Disease	1	2	5	6
15	mnist	1	2	8.6	8.5
16	MicroMass	1	2	6.8	4.4

The classification accuracy of Naive Bayes and TAN is lower than that of the proposed method for most of the datasets. This is due to the fact that Naive Bayes and TAN have limited Max parents as can be seen in Table 4.6. As the number of variables increases, the number of variables that may influence each other increases, and thus a small Max parents may lead to a decrease in classification accuracy. However, dataset No. 4 has the highest classification accuracy in Naive Bayes, and dataset No. 5 has the highest classification accuracy in TAN. This can be attributed to the small correlation between the variables. Looking at Table 4.7, we can see that the number of edges involved in the estimation of class variables for RAI-GBN datasets 4 and 5 is 2.4 and 14.3, which is very small. This indicates that the Markov blanket of the class variable is small, and most of the feature variables are not involved in the estimation of the class variable. Looking at the number of edges involved in the estimation of the class variable in RAI-ANB, the value of No. 4 is close to that of TAN, and the value of No. 5 is smaller than that of TAN and close to that of Naiva Bayes. This means that RAI-ANB estimates a sparse structure similar to Naive Bayes and TAN. Therefore, the classification accuracy of these datasets was high because the true structure was close to Naive Bayes and TAN in the ANB candidate space.

Comparing RAI-GBN and RAI-ANB, the classification accuracy of the proposed method is higher on all datasets. Table 4.7 shows that for datasets 4, 6, 8, 11, and

Table 4.7: The number of edges used to estimate the class variable of respective classifiers for huge networks

	dataset	Naive Bayes	TAN	RAI-GBN	RAI-ANB
1	kr-vs-kp	36	71	46.1	116.2
2	Connect-4	42	83	43.1	115
3	Flowmeters D	43	85	13.1	83.2
4	movement libras	90	179	2.4	187.4
5	dota2	116	231	14.3	209.9
6	Musk1	166	331	1.1	513.1
7	Musk2	166	331	61.6	1026.9
8	Epileptic Seizure	178	355	0	387
9	mfeat-fac	216	431	19.5	561.3
10	semeion	256	511	15.3	753.3
11	madelon	500	999	3	533.5
12	HAPT	561	1121	164.7	1524.6
13	HAR	561	1121	204.9	1625.8
14	Parkinson's Disease	753	1505	2.5	1973.2
15	mnist	784	1567	2218.0	3253.2
16	MicroMass	1300	2599	169.8	1940.4

14, the number of edges involved in the estimation of class variables for RAI-GBN is very small, which may indicate undertraining due to CI testing. On the other hand, since the proposed method assumes ANB structure, all feature variables are used for class variable estimation, which may improve the classification accuracy.

2. Comparison with Other Classifiers

In the previous section, we showed that the proposed method has higher classification accuracy than other BNC in large networks. In this section, we analyze the proposed method by comparing its accuracy with classifiers other than BNC. The dataset used for the experiments is the same as the one used in the previous section.

In this section, the following three methods are used as general classifiers.

- RF: random forest
- SVM: support vector machine
- MLP: deep neural network (Multilayer perceptron)

Both of these methods are trained using scikit-learn (https://scikit-learn.org/stable/). MLP is tuned using grid search with $\{10, 100, 1000\}$ as the candidate number of hidden layers and $\{0.0001, 0.001, 0.01\}$ as the candidate learning rate. The proposed method, RAI-ANB, is tuned using grid search with $\{1, 5, 10, 20\}$ as the candidate ESS for structure learning and classification accuracy estimation. For each method and each dataset, we obtain the average classification accuracy using 10-fold cross validation. The proposed method, RAI-ANB, is tuned using grid search with $\{1, 5, 10, 20\}$ as the candidate ESS for structure learning and classification accuracy estimation. For each method and each dataset, we obtain the average classification accuracy using 10-fold cross validation. However, since the comparison method cannot deal with input data containing missing values, they are removed from the dataset. To show the significance of the proposed method, a multiple comparison using the Hommel method, (Hommel, 1988), was performed to obtain the p-value. Table 4.8 shows the classification accuracy and p-values of each method. In addition, the highest classification accuracy of each method for each data set is shown in bold.

Table 4.8 shows that the classification accuracy of the proposed method is higher than that of RF and SVM, but lower than that of MLP. In addition, the proposed method failed to show significant differences against all the comparison methods. The proposed method is a generative model that represents joint probability distributions, while MLP is a functional discriminative model that takes features as inputs and class variable values as outputs. As a result, the proposed Bayesian network classifier was not superior to other classifiers in terms of accuracy. However, there is also an advantage that the proposed method is a generative model. The proposed method is a probabilistic model and has the advantage of being able to simultaneously estimate the probability distribution itself. For example, while MLP is difficult to handle missing values in input data, the proposed method can easily obtain the probability distribution marginalized by the missing variables. Therefore, the proposed method is more likely to show higher accuracy than MLP when classifying input data containing missing values. In addition, since the proposed method can asymptotically calculate the probability of class variables, it is easy to calculate the expected utility (loss) function in decision-making problems, and thus easy to construct a decision-making system. Furthermore, the proposed method has excellent explainability because it can show the causal relationship between the feature variables.

Table 4.8: Accuracies of respective classifiers

	dataset	variables	num of data	classes	RF	SVM	MLP	RAI-ANB
1	Balance	5	625	3	0.8289	0.9008	0.8565	**0.9167**
2	banknote	5	1372	2	0.8812	0.8819	**0.9665**	0.8812
3	Hayes-Roth	5	132	3	0.8022	0.8560	**0.8970**	0.8110
4	iris	5	150	3	0.8267	0.8133	**0.8819**	0.8267
5	lenses	5	24	3	**0.7333**	0.6500	0.7180	0.6333
6	Car	7	1728	4	0.9641	0.9688	**0.9707**	0.9398
7	liver	7	345	2	0.6146	0.6403	**0.9919**	0.6811
8	monk1	7	432	2	0.9560	0.9166	0.8106	**1.0000**
9	mux6	7	64	2	0.7976	0.5786	**0.9241**	0.8714
10	led7	8	3200	10	0.7288	**0.7369**	0.4922	0.7319
11	HTRU2	9	17898	2	0.9141	0.9112	**0.9482**	0.9084
12	Nursery	9	12960	3	**0.9891**	0.9860	0.8621	0.9398
13	pima	9	768	9	0.6875	0.6967	**0.7302**	0.7292
14	post	9	87	5	0.5736	0.7125	**0.8394**	0.6681
15	Breast Cancer	10	277	2	0.6534	0.7299	**0.7510**	0.7251
16	Breast Cancer Wisconsin	10	683	2	0.9664	0.9650	0.6325	**0.9766**
17	cmc	10	1473	3	0.4508	0.4895	**0.7571**	0.4807
18	glass	10	214	6	0.6214	0.5983	**0.8074**	0.6173
19	shuttle-small	10	5800	6	0.9721	0.9662	0.8500	**0.9722**
20	threeOf9	10	512	2	**0.9824**	0.8925	0.9115	0.9337
21	TicTac	10	958	2	**0.9133**	0.8831	0.8299	0.7798
22	magic	11	19020	2	0.7809	0.7807	**0.8267**	0.7803
23	Flare	11	1389	9	0.8179	**0.8431**	0.7313	0.8222
24	heart	14	270	2	0.7926	**0.8407**	0.7167	0.8222
25	wine	14	178	3	0.8935	0.9105	0.6561	**0.9265**
26	cleve	14	296	2	**0.8339**	0.8238	0.6518	0.8307
27	australian	15	690	2	0.8449	**0.8609**	0.8457	0.8464
28	crx	15	653	2	0.8638	**0.8653**	0.7877	0.8514
29	EEG	15	14980	2	0.7275	0.6955	**1.0000**	0.6834
30	Congressional	17	232	2	0.9524	0.9696	**1.0000**	0.9440
31	zoo	17	101	5	**0.9700**	0.9300	0.7881	**0.9700**
32	pendigits	17	10992	10	0.9406	0.9285	**0.9998**	0.9176
33	letter	17	20000	26	0.6471	0.5877	**0.9408**	0.5723
34	ClimateModel	19	540	2	0.9204	0.9148	0.7253	**0.9241**
35	ImageSegmentation	19	2310	7	**0.8229**	0.8134	0.6764	0.8139
36	lymphography	19	148	4	0.7910	0.7767	**0.9705**	0.8719
37	vehicle	19	846	4	0.6289	0.6348	**0.8170**	0.6015
38	hepatitis	20	80	2	0.8000	0.8375	**1.0000**	0.8750
39	german	21	1000	2	0.7330	0.7390	**1.0000**	0.7460
40	bank	21	30488	2	0.8803	0.8918	0.6205	**0.8940**
41	waveform-21	22	5000	3	0.7790	0.8132	**0.8168**	0.7922
42	Mushroom	22	5644	2	**1.0000**	**1.0000**	0.9203	**1.0000**
43	spect	23	263	2	0.8128	0.8021	**0.9709**	0.8207
44	kr-vs-kp	37	3196	2	0.9831	0.9384	**0.9950**	0.9524
45	Connect-4	43	67557	3	0.7901	0.7393	**0.8097**	0.7938
46	Flowmeters D	44	180	4	**0.8778**	0.8722	0.8833	0.8500
47	movement libras	91	360	15	0.6750	0.5361	**0.7306**	0.5333
48	dota2	117	102944	2	0.5314	0.5508	0.5425	**0.5957**
49	Musk1	167	476	2	0.7921	0.7604	**0.8380**	0.7986
50	Musk2	167	6598	2	0.9548	0.9406	**0.9767**	0.9615
51	Epileptic Seizure	179	11500	5	0.4207	0.4618	**0.4692**	0.3808
52	mfeat-fac	219	2000	10	0.4630	0.4245	**0.4675**	0.4650
53	semeion	257	1600	10	0.8789	**0.9360**	0.9272	0.8776
54	madelon	501	2000	2	0.5585	0.5860	0.5605	**0.5905**
55	HAPT	563	10929	12	0.8756	0.8866	**0.9222**	0.8805
56	HAR	563	10299	6	0.8880	0.9068	**0.9316**	0.8987
57	Parkinson's Disease	755	756	2	0.8389	0.4061	**0.9316**	0.7964
58	MNIST	785	70000	10	0.9695	**0.9731**	**0.9731**	0.9493
59	MicroMass	1300	360	10	0.9333	0.9222	**0.9722**	0.9556
	classification accuracy	average			0.8054	0.7945	**0.8275**	0.8069
		p-value			0.17619	0.17619	0.17619	-

Chapter 5

Conclusion

In this paper, we extend Honda's method to learning ANB, and propose a method for learning larger BNC than before. We also proved that the proposed method has asymptotic agreement for the ANB. Experiments on a benchmark dataset showed that RAI-GBN has low classification accuracy due to the low accuracy of the CI test when the number of data is small and the Markov blanket of the class variable is small. On the other hand, the proposed method stabilizes the classification accuracy by learning ANB, and the classification accuracy is almost same as that of the exact learning approach. We also showed that the proposed method can learn large networks with thousands of variables, which cannot be learned by the exact learning approach, and can achieve significantly higher classification accuracy than other BNC methods. We also compared the classification accuracy of the proposed method with that of common classifiers. As a result, the average classification accuracy of the proposed method was higher than the average classification accuracy of Random Forest and Support Vector Machine, but lower than the average classification accuracy of Deep Neural Network. However, the proposed method is a probabilistic model and has many advantages over the deep neural network, such as explainability of the model. For example, BNC is known to greatly improve the classification accuracy by performing model averaging (Cheng et al., 2002). Recently, it has been reported that the classification accuracy can be improved by combining ensemble learning (Aaomi, Sugahara, & Ueno, 2020). By using these methods, the classification accuracy of the proposed method is expected to be improved.

In addition to the above, other issues include the following. Large structure learning tends to reduce the accuracy of parameter classification and the reliability of CI tests due to the sparsity of the data. To address this problem, Isozaki et al. (2008, 2009) have proposed a parameter estimation method based on Minimum Free En-

ergy (MFE) and the CI test. By incorporating this MFE-based parameter estimation method and the CI test, we expect to improve the learning and classification accuracy in large structures.

Acknowledgments

I would like to express my sincere thanks to Professor Maomi Ueno, for his careful education, considerable encouragement and invaluable discussion. He also gave me the opportunities to present my research at conferences and submit my paper to a journal. I am grateful to Mr. Shota Sugahara for his constructive discussions in my proofs and experiments. Finally, I would like to thank all the members of Ueno, Kawano, Nishiyama, and Uto laboratories for their meaningful discussions and advice.

Bibliography

Aaomi, I., Sugahara, S., & Ueno, M. (2020). Model averaging Bayesian network classifier by ensemble learning. *IEICE TRANSACTIONS on Information and Systems*, *103*, 183–193.

Barlett, M., & Cussens, J. (2013). Advances in Bayesian Network Learning Using Integer Programming. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 182–191.

Buntine, W. (1991). Theory Refinement on Bayesian Networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 52–60, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Carvalho, A. M., Adão, P., & Mateus, P. (2013). Efficient Approximation of the Conditional Relative Entropy with Applications to Discriminative Learning of Bayesian Network Classifiers. *Entropy*, *15*(7), 2716–2735.

Carvalho, A. M., Roos, T., Oliveira, A. L., & Myllymäki, P. (2011). Discriminative Learning of Bayesian Networks via Factorized Conditional Log-Likelihood. *Journal of Machine Learning Research*, *12*, 2181–2210.

Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian Networks from Data: An Information-Theory BasedApproach. *Artificial Intelligence - AI*, *137*, 43–90.

Chickering, D. M. (2002). Optimal Structure Identification With Greedy Search.. *Journal of Machine Learning Research*, *3*, 507–554.

Cowell, R. (2009). Efficient maximum likelihood pedigree reconstruction. In *Theoretical Population Biology*, Vol. 76, pp. 285–291.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, *29*(2), 131–163.

Grossman, D., & Domingos, P. (2004). Learning Bayesian Network classifiers by maximizing conditional likelihood. In *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, pp. 361–368.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, *20*(3), 197–243.

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

Honda, K., Natori, K., Sugahara, S., Isozak, T., & Ueno, M. (2019). Learning huge Bayesian network structures using the transitivity. *IEICE TRANSACTIONS on Information and Systems*, *102*, 796–811.

Ide, J. S., Cozman, F. G., & Ramos, F. T. (2004). Generating Random Bayesian Networks with Constraints on Induced Width. In *Proceedings of the 16th european conference on artificial intelligence*, pp. 353–357. IOS Press.

Ide, J., & Cozman, F. (2002). Random generation of bayesian networks. In *Brazilian symposium on artificial intelligence*, pp. 366–376. Springer.

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

Isozaki, T., Kato, N., & Ueno, M. (2009). "Data temperature" in Minimum Free energies for Parameter Learning of Bayesian Networks.. *International Journal on Artificial Intelligence Tools*, *18*, 653–671.

Isozaki, T., & Ueno, M. (2009). Minimum Free Energy Principle for Constraint-Based Learning Bayesian Networks. In *Machine Learning and Knowledge Discovery in Databases*, pp. 612–627. Springer Berlin Heidelberg.

Jensen, F. V., & Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs* (2nd edition). Springer Publishing Company, Incorporated.

Koivisto, M., & Sood, K. (2004). Exact Bayesian Structure Discovery in Bayesian Networks. *Journal of Machine Learning Research*, *5*, 549–573.

Lichman, M. (2013). UCI Machine Learning Repository..

Ling, C. X., & Zhang, H. (2003). The Representational Power of Discrete Bayesian Networks. *J. Mach. Learn. Res.*, 709–721.

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

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

Mitchell, T. M. (1997). *Machine Learning* (1 edition). McGraw-Hill, Inc.

Natori, K., Uto, M., & Ueno, M. (2017). Consistent Learning Bayesian Networks with Thousands of Variables. In *Proceedings of Machine Learning Research*, Vol. 73, pp. 57–68.

Natori, K., Uto, M., Nishiyama, Y., Kawano, S., & Ueno, M. (2015). Constraint-Based Learning Bayesian Networks Using Bayes Factor. In *Proceedings of the Second International Workshop on Advanced Methodologies for Bayesian Networks - Volume 9505*, AMBN 2015, pp. 15–31. Springer-Verlag New York, Inc.

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

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465 – 471.

Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics*, *6*(2), 461–464.

Silander, T., & Myllymäki, P. (2006). A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *Proceedings of Uncertainty in Artificial Intelligence*, pp. 445–452.

Singth, A., & Moore, A. (2005). Finding optimal Bayesian networks by dynamic programming. Tech. rep., Technical Report, Carnegie Mellon University.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, and Search*. MIT press.

Steck, H., & Jaakkola, T. (2002). *On the dirichlet prior and Bayesian regularization.*, pp. 697–704. MIT Press.

Sugahara, S., & Ueno, M. (2020). Exact Learning Bayesian Network Classifier with Augmented Naive Bayes structure constraint. *IEICE TRANSACTIONS on Information and Systems*, *103*, 301–313.

Sugahara, S., Uto, M., & Ueno, M. (2018). Exact learning augmented naive Bayes classifier. In *International Conference on Probabilistic Graphical Models*, Vol. 72, pp. 439–450.

Tsamardinos, I., Brown, L., & Aliferis, C. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78.

Ueno, M. (2008). Learning likelihood-equivalence Bayesian networks using an empirical Bayesian approach. *Behaviormetrika*, *35*(2), 115–135.

Ueno, M. (2010). Learning Networks Determined by the Ratio of Prior and Data. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, p. 598–605, Arlington, Virginia, USA. AUAI Press.

Ueno, M. (2011). Robust learning Bayesian networks for prior belief. In *Proceedings of Uncertainty in Artificial Intelligence*, pp. 689–707.

Ueno, M., & Uto, M. (2012). Non-informative dirichlet score for learning bayesian networks. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pp. 331–338.

Yehezkel, R., & Lerner, B. (2009). Bayesian network structure learning by recursive autonomy identification. *Journal of Machine Learning Research*, *10*, 1527–1570.

Yuan, C., Lim, H., & Lu, T.-C. (2011). Most Relevant Explanation in Bayesian Networks. *Journal of Artificial Intelligence Research*, *42*(1), 309–352.