Recursive autonomy identification-based learning of augmented naive Bayes classifiers

Shouta Sugahara sugahara@ai.is.uec.ac.jp
Wakaba Kishida kishida@ai.lab.uec.ac.jp
Koya Kato kato@ai.lab.uec.ac.jp
Maomi Ueno ueno@ai.is.uec.ac.jp
The University of Electro-Communications, Tokyo, Japan

Abstract

Earlier reports have described classification accuracies of exactly learned augmented naive Bayes (ANB) classifiers. Those results indicate that a class variable with no parent has higher accuracy than those of other Bayesian network classifiers. Additionally, asymptotic estimation of the class posterior identical to that of the exactly learned Bayesian network is guaranteed to be achieved. Nevertheless, exact learning of large ANB is difficult because it entails an associated NP-hard problem that worsens as the number of variables increases. Recent reports have described that constraint-based learning methods with Bayes factor achieve larger network structures than when using traditional methods. This study proposes an efficient learning algorithm of an ANB classifier using recursive autonomy identification (RAI) with Bayes factor. A unique benefit of the proposed method is that the proposed method is guaranteed to accelerate execution of the RAI algorithm when the data follow an ANB structure. Numerical experiments were conducted to demonstrate the effectiveness of the proposed method.

Keywords: augmented naive Bayes; Bayesian networks; classification; structured learning

1. Introduction

A Bayesian network classifier (BNC) can be interpreted as a Bayesian network for which one node is a class variable and the other nodes are feature variables. Earlier reports have described 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) (Friedman et al., 1997; Carvalho et al., 2011, 2013; Grossman and Domingos, 2004). Recently, however, Sugahara et al. (2018); Sugahara and Ueno (2021) demonstrated experimentally that the BNC performance achieved by maximizing the ML is not necessarily worse than that achieved by maximizing CLL for large data. Unfortunately, their experiments also demonstrated that the classification accuracy of the structure maximizing the ML rapidly worsens as the sample size becomes small. They explained the reason: the class variable tends to have numerous parents when the sample size is small. Therefore, the conditional probability parameter estimation of the class variable becomes unstable because the number of parent configurations becomes large. Then the sample size for learning a parameter becomes sparse. This analysis suggests that exact learning BNC by maximizing the ML to have

no parents of the class variable might improve the classification accuracy. Consequently, they proposed exact learning augmented naive Bayes classifier (ANB), in which the class variable has no parent and in which all feature variables have the class variable as a parent. Additionally, they demonstrated the effectiveness of their method empirically. However, exact learning ANB has an associated NP-hard problem (Chickering, 1996): as the number of variables increases, the number of structure searches increases exponentially. Various algorithms for exact learning Bayesian networks have been developed, such as dynamic programming (Silander and Myllymäki, 2006), A^* search (Yuan et al., 2011), breadth-first branch and bound search (Malone et al., 2011), and integer programming (Cussens, 2012). Nevertheless, it cannot be applied to network structures with more than 60 variables.

However, 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 V-structures. Steck and Jaakkola (2002) 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)).

As described in this paper, we propose a constraint-based Learning ANB classifier using RAI with Bayes factor to learn large ANB classifier structures. 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 structure.

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

2. Bayesian network classifiers

2.1 Bayesian network

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

$$P(X_0, X_1, \ldots, X_n) = \prod_{i=0}^{n} P(X_i \mid \Pi_i, G). \tag{1}$$

where Π_i is the parent variable set of X_i. Letting θ_{ijk} be a conditional probability parameter of $X_i = k$ when the j-th instance of the parents of X_i is observed (We write $\Pi_i = j$), we define $\Theta = \{\theta_{ijk}\}$ $(i = 0, \cdots, n; j = 1, \cdots, q_{\Pi_i}; k = 1, \cdots, r_{X_i})$. A Bayesian network is a pair $B = (G, \Theta)$. Buntine (1991) assumed the Dirichlet prior and used an expected a posteriori (EAP) estimator $\hat{\theta}_{ijk}$ as

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{X_i=k,\Pi_i=j}}{\alpha_{ij} + N_{\Pi_i=j}}. \tag{2}$$

In that equation, $N_{X_i=k,\Pi_i=j}$ represents the number of samples of $X_i = k$ when $\Pi_i = j$, $N_{\Pi_i=j} = \sum_{k=1}^{r_{X_i}} N_{X_i=k,\Pi_i=j}$. In addition, α_{ijk} denotes the hyperparameters of the Dirichlet prior distributions (α_{ijk} is a pseudo-sample corresponding to $N_{X_i=k,\Pi_i=j}$); $\alpha_{ij} = \sum_{k=1}^{r_{X_i}} \alpha_{ijk}$.

The first learning task of the Bayesian network is to seek a structure G optimizing a given score. The most popular marginal likelihood (ML) score of Bayesian network (using a Dirichlet prior over model parameters) finds the maximum a posteriori (MAP) structure when we assume a uniform prior over structures, as described by Buntine (1991) and Heckerman et al. (1995). In addition, the Dirichlet prior is known as a distribution that ensures likelihood equivalence. This score is known as *Bayesian Dirichlet equivalence (BDe)* (Heckerman et al., 1995). Given no prior knowledge, *the Bayesian Dirichlet equivalence uniform (BDeu)*, as proposed earlier by Buntine (1991), is often used. Let $D = \{\mathbf{x}^1, \cdots, \mathbf{x}^d, \cdots, \mathbf{x}^N\}$ be training dataset and let each $\mathbf{x}^d$ be a tuple of the form $\langle x_0^d, x_1^d, \cdots, x_n^d \rangle$. For the analyses presented in this paper, we assume no missing data throughout. The BDeu score is represented as

$$P(D \mid G) = \prod_{i=0}^{n} \prod_{j=1}^{q_{\Pi_i}} \frac{\Gamma(\frac{\alpha}{q_{\Pi_i}})}{\Gamma(\frac{\alpha}{q_{\Pi_i}} + N_{\Pi_i=j})} \prod_{k=1}^{r_{X_i}} \frac{\Gamma(\frac{\alpha}{r_{X_i} q_{\Pi_i}} + N_{X_i=k,\Pi_i=j})}{\Gamma(\frac{\alpha}{r_{X_i} q_{\Pi_i}})}, \tag{3}$$

where α is a hyperparameter.

2.2 Bayesian network classifiers

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

$$\begin{aligned} \hat{c} &= \arg\max_{c \in \{1,\ldots,r_0\}} P(c \mid x_1, \ldots, x_n, G, \Theta) \\ &= \arg\max_{c \in \{1,\ldots,r_0\}} \prod_{j=1}^{q_0} \prod_{k=1}^{r_0} (\theta_{0jk})^{1_{0jk}} \times \prod_{i:X_i \in \mathbf{Ch}} \prod_{j=1}^{q_i} \prod_{k=1}^{r_{X_i}} (\theta_{ijk})^{1_{ijk}} \end{aligned} \tag{4}$$

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 4, 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 uses a general Bayesian network is called a General Bayesian Network (GBN). However, the most common score for BNC structures is the conditional log likelihood (CLL) of the class variable given all the feature variables (Friedman et al., 1997). Friedman et al. (1997) claimed that the structure maximizing CLL, called a discriminative model, provides more accurate classification than that maximizing the ML because the CLL reflects only the class variable posterior, whereas the ML reflects the posteriors of all the variables. Nevertheless, ML is known to have asymptotic consistency, which guarantees that the structure which maximizes the ML converges to the true structure when the sample size is sufficiently large. Sugahara and Ueno (2021) demonstrated experimentally that the BNC performance achieved by maximizing the ML is not necessarily worse than that achieved by maximizing CLL for large data. Unfortunately, their experiments also demonstrated that the classification accuracy of the structure maximizing the ML worsens rapidly as the sample size becomes small. They explained the reason: the class variable tends to have numerous parents when the sample size is small. Therefore, the conditional probability parameter estimation of the class variable becomes unstable because the number of parent configurations becomes large. Then the sample size for learning a parameter becomes sparse. To resolve this difficulty, they proposed exact-learning-augmented naive Bayes classifier (ANB) for which the class variable has no parent and for which all feature variables have the class variable as a parent. Their method is guaranteed to estimate the identical class posterior asymptotically to that of the exactly learned BN. They demonstrated the effectiveness of their method empirically. However, the exact learning approach is limited to learning dozens of variables. It cannot be applied to cases with numerous variables. Therefore, we propose an approach that can learn large BNCs.

3. Proposed Method

3.1 Constraint-based learning Bayesian networks based on Bayes factor

Constraint-based approaches relax computational costs and learn huge networks. Such approaches learn 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 adopting 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 V-structures. 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 the dependency when the true structure is independent) 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 (2002) proposed a conditional independence test with an asymptotic consistency: a Bayes factor with BDeu. For two variables $X, Y \in \mathbf{V}$ and a set

of conditional variables$\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, the log Bayes factor with BDeu for X and Y given $\mathbf{Z}$ is defined as

$$\log BF(X, Y \mid \mathbf{Z}) = \log LocalBDeu(X \mid \mathbf{Z}) - \log LocalBDeu(X \mid \mathbf{Z} \cup \{Y\}),$$

where

$$LocalBDeu(X \mid \mathbf{Z}) = \prod_{j=1}^{q_{\mathbf{Z}}} \frac{\Gamma(\frac{\alpha}{q_{\mathbf{Z}}})}{\Gamma(\frac{\alpha}{q_{\mathbf{Z}}} + N_{\mathbf{Z}=j})} \prod_{k=1}^{r_X} \frac{\Gamma(\frac{\alpha}{r_X q_{\mathbf{Z}}} + N_{X=k,\mathbf{Z}=j})}{\Gamma(\frac{\alpha}{r_X q_{\mathbf{Z}}})}.$$

When there exists a variable set $\mathbf{Z}$ such that $\log BF(X, Y \mid \mathbf{Z}) > 0$, then the edge between X and Y is deleted.

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.

3.2 Learning ANB using the RAI algorithm with the CI test using Bayes Factor

This section presents the algorithm of the constraint-based learning method of ANB with RAI algorithm. Let the graph be $G = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of variables in G and $\mathbf{E}$ is the set of edges in G. In addition, G has both directed and undirected edges. In addition, let $G_{ex} = (\mathbf{V}_{ex}, \mathbf{E}_{ex})$ be the subgraph partitioned by the RAI algorithm.

(1) Input data $\mathbf{D}$, initial order of CI tests $n_z = 1$, and initial graph $G_{ucf} = (\mathbf{V}_s, \mathbf{E}_s)$, which is a complete undirected graph consisting of all the feature variables.

(2) For all $X \in \mathbf{V}_s, Y \in \mathbf{V}_s \cup \mathbf{V}_{ex}, \mathbf{Z} \subseteq \mathbf{V}_s \cup \mathbf{V}_{ex} \setminus \{X_0\}, (|\mathbf{Z}| = n_z)$, when X and Y given $\mathbf{Z} \cup \{X_0\}$ are determined to be conditionally independent by CI tests, the edges between X and Y are removed.

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

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

(5) $n_z = n_z + 1$; Recursively invoke RAI on each subgraph.

(6) Add X_0 and the edges from X_0 to all the feature variables to the resulting structure.

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.

Moreover, the proposed method is guaranteed to estimate the true conditional probability of the class variable asymptotically although the proof is omitted due to limitations of space.

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, 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 Section 2, because 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.

4. Experiments

This section presents evaluation experiments conducted to underscore the effectiveness of the proposed method. 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 TAN method by maximizing aCLL (Carvalho et al., 2013).
- exact-GBN: Exact learning of GBN with BDeu score (Silander and Myllymäki, 2006).
- exact-ANB: Exact learning of ANB 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,

Table 1: Accuracies of the respective classifiers for small networks

	dataset	variable	number of data	classes	Naive Bayes	TAN	GBN-CMDL	BNC2P	TAN-aCLL	exact-GBN	exact-ANB	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	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	-
	calculation time (s)	average			0.00	2.58	30.53	21.11	10.05	1790.93	500.76	26.06	3.14
		standard error			0.00	0.16	25.50	12.87	6.77	895.76	252.69	20.90	1.26

the conditional probability parameters of the BNCs after structure learning were estimated using 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 1, "average" denotes the average classification accuracy of each method for all datasets. Also, "p-value" denotes the p-value obtained by multiple comparison. For "calculation time", "average" denotes the average computation time for structure learning of each method for all datasets. "Standard error" represents the standard error of the computation time of each method. Table 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 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 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 the 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 0.0.

Table 2: Number of Max parents and edges in the Markov blanket of the class variable for small networks

	dataset	Naive Bayes		TAN		exact-GBN		exact-ANB		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

In addition, because TAN can be computed in polynomial time, its computation time is shorter than those of the other methods (Friedman et al., 1997; Madden, 2009).

Table 1 also shows that the proposed method dynamically 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 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 only feature variables, 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 exact-GBN or exact-ANB. 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 calculation time of RAI-ANB is much shorter than that of either exact-GBN or exact-ANB.

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 3 shows the average accuracies and p-values of Hommel's tests. Table 4 presents the average number of edges in the Markov blanket of the class variable for each method.

From Table 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-GNB at the $p < 0.05$ significance level. Similarly to the results for small networks, the average computation time of the proposed method is shorter than that of RAI-GBN for the reason described earlier.

Table 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	average			0.6612	0.7042	0.5787	**0.7370**
		p-value			0.0044	0.0012	0.0015	-
	calculation time (s)	average			0.0	545.7	2002.1	1665.9
		standard error			0.0	434.6	972.2	1112.6

Table 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 Naive Bayes and TAN are lower than those of the proposed method for all datasets except for No. 3 and No. 5. Table 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 that of Naive Bayes.

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 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.

Table 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

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 presents the numbers of edges (NE), the number of decomposed structures (NDS), and the runtimes for RAI-GBN and RAI-ANB. 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, 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.

5. Conclusions

As described herein, we proposed an extension of constraint-based learning method using Bayes factor applied to the learning ANB. First, this study compared the classification accuracies of proposed methods with exact learning methods using small networks. Results indicate that the classification accuracy of the proposed method is nearly equivalent to that of the exact learning approach. Second, this study compared the classification accuracies of the proposed methods with BNCs using large networks that can not be learned using the exact learning methods. Results indicated that the classification accuracy of the proposed

method was significantly better than those obtained using the other BNC methods. Isozaki et al. (2008, 2009) proposed an effective learning Bayesian network method by adjusting the hyperparameter for small data. As a 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.

References

J. Abellán, M. Gómez-Olmedo, and S. Moral. Some variations on the pc algorithm. In *PGM*, pages 1–8, 2006.

W. Buntine. Theory Refinement on Bayesian Networks. In *UAI*, pages 52–60, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

A. M. Carvalho, T. Roos, A. L. Oliveira, and P. Myllymäki. Discriminative Learning of Bayesian Networks via Factorized Conditional Log-Likelihood. *JMLR*, 12:2181–2210, 2011.

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(7):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, 05 2002. doi: 10.1016/S0004-3702(02)00191-1.

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

J. Cussens. Bayesian network learning with cutting planes. In *UAI*, pages 153–160, 2012.

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 (2):131–163, 1997.

D. Grossman and P. Domingos. Learning Bayesian Network classifiers by maximizing conditional likelihood. In *ICML*, 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(3):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. *International Journal on Artificial Intelligence Tools*, 18:653–671, 10 2009. doi: 10.1142/S0218213009000342.

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*, pages 709–721, 2003.

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

B. M. 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 *UAI*, pages 479–488, 2011.

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, and M. Ueno. Consistent Learning Bayesian Networks with Thousands of Variables. In *AMBN*, volume 73, pages 57–68, 2017.

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

R. Y. Rohekar, Y. Gurwicz, S. Nisimov, G. Koren, and G. Novik. Bayesian structure learning by recursive bootstrap. In *NIPS*, pages 10546–10556, 2018.

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. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2019.10.003.

T. Silander and P. Myllymäki. A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In *UAI*, pages 445–452, 2006.

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

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

S. Sugahara and M. Ueno. Exact learning augmented naive bayes classifier. *Entropy*, 23(12), 2021. ISSN 1099-4300. doi: 10.3390/e23121703.

S. Sugahara, M. Uto, and M. Ueno. Exact learning augmented naive Bayes classifier. In *PGM*, volume 72, pages 439–450, 2018.

S. Sugahara, I. Aomi, and M. Ueno. Bayesian network model averaging classifiers by subbagging. In *PGM*, volume 138, pages 461–472, 2020.

S. Sugahara, I. Aomi, and M. Ueno. Bayesian network model averaging classifiers by subbagging. *Entropy*, 24(5), 2022. ISSN 1099-4300.

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

M. Ueno. Learning Networks Determined by the Ratio of Prior and Data. In *UAI*, pages 598–605, 2010. ISBN 9780974903965.

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

C. Yuan, H. Lim, and T.-C. Lu. Most Relevant Explanation in Bayesian Networks. *JAIR*, 42(1): 309–352, 2011.