Bayesian Network Model Averaging Classifiers by Subbagging

Shouta Sugahara SUGAHARA@AI.LAB.UEC.AC.JP

Itsuki Aomi AOMI@AI.LAB.UEC.AC.JP

Maomi Ueno UENO@AI.LAB.UEC.AC.JP

Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, 182-8585, Japan

Abstract

For classification problems, Bayesian networks are often used to infer a class variable when given feature variables. Earlier reports have described that the classification accuracy of Bayesian network structures achieved by maximizing the marginal likelihood (ML) is lower than that achieved by maximizing the conditional log likelihood (CLL) of a class variable given the feature variables. However, the performance of Bayesian network structures achieved by maximizing ML is not necessarily worse than that achieved by maximizing CLL for large data because ML has asymptotic consistency. As the sample size becomes small, however, the error of learning structures by maximizing the ML becomes rapidly large; it then degrades the classification accuracy. As a method to resolve this shortcoming, model averaging, which marginalizes the class variable posterior over all structures, has been proposed. However, the posterior standard error of the structures in the model averaging becomes large as the sample size becomes small; it subsequently degrades the classification accuracy. The main idea of this study is to improve the classification accuracy using the subbagging to reduce the posterior standard error of the structures in the model averaging. Moreover, to guarantee asymptotic consistency, we use the K-best method with the ML score. The experimentally obtained results demonstrate that our proposed method provides more accurate classification for small data than earlier methods do.

Keywords: Bayesian networks; classification; model averaging; structure learning.

1. Introduction

Bayesian network classifiers (BNC), which are special cases of Bayesian networks designed for classification problems, have yielded successful results in real-world applications. The most common score for learning Bayesian network structures is the marginal likelihood (ML) of a structure. The structure which maximizes ML is called a generative model, which represents the joint probability distribution of all variables. However, the most common score for BNC structures is conditional log likelihood (CLL) of the class variable given all the feature variables (Friedman et al., 1997; Grossman and Domingos, 2004; Carvalho et al., 2013). Friedman et al. (1997) claimed that the structure maximizing CLL, called a discriminative model, provides more accurate classification than that maximizing the ML. The reason is that the CLL only reflects 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. Therefore, Sugahara et al. (2018) demonstrated experimentally that the BNC performance achieved by maximizing the ML is not necessarily worse than that achieved by maximizing CLL for large data. However, their experiments also demonstrated that the classification accuracy of the structure maximizing the ML becomes rapidly worse as the sample size becomes small. They explained the reason as that 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 (ANB) classifier, in which the class variable has no parent and in which all feature variables have the class variable as a parent. Additionally, they empirically demonstrated the effectiveness of their method.

However, the fundamentally important reason for the problem is that the error of learning structures becomes large when the sample size becomes small. Model averaging, which marginalizes the class variable posterior over all structures, has been known as a method to alleviate this shortcoming (Madigan and Raftery, 1994; Chickering and Heckerman, 2000). However, the number of structures increases super-exponentially for the network size. Therefore, averaging all structures with numerous variables is computationally infeasible. The most common approach to this difficulty is the K-best method (Tian et al., 2010; Chen and Tian, 2014; He et al., 2016; Chen et al., 2015, 2016, 2018; Liao et al., 2018), which considers only the K-best scoring structures.

However, the posterior standard error of the structures in model averaging becomes large for a small sample size. It then decreases the classification accuracy. To reduce the posterior standard error, the resampling methods, such as the adaboost (adaptive boosting) (Freund and Schapire, 1997), the bagging (bootstrap aggregating) (Breiman, 1996), and subbagging (subsampling aggregating) (Bühlmann and Yu, 2002) are known. Also, Jing et al. (2008) proposed ensemble class variable prediction using adaboost. That study empirically demonstrated its effectiveness. Nevertheless, this method tends to cause overfitting difficulties for small amounts of data because adaboost tends to be sensitive to noisy data (Dietterich, 2000). Later, Rohekar et al. (2018) proposed B-RAI, a model averaging method with the bagging, based on the RAI algorithm (Yehezkel and Lerner, 2009), which learns a structure by recursively conducting conditional independence (CI) tests, edge direction and structure decomposition into smaller substructures. The B-RAI increases the number of models for the model averaging using multiple bootstrapped datasets. However, the B-RAI is inapplicable for the bagging to the posterior of the structures. Therefore, the posterior standard error of the structures is not expected to decrease. In addition, the CI tests of the B-RAI are not guaranteed to have asymptotic consistency. This engenders reduction of the computational costs but might degrade the classification accuracy for large data.

The main idea of this study is to improve the classification accuracy using the subbagging to reduce the posterior standard error of structures in model averaging. Moreover, to guarantee asymptotic consistency, we employ the K-best method with the ML score.

The proposed method is expected to present the following benefits. (1) The class variable posterior converges to the true value when the sample size is sufficiently large because it has asymptotic consistency. Also, (2) even for small data, the subbagging reduces the posterior standard error of the structures for the K-best method and improves the classification accuracy. To compare the respective classification performances of our method and earlier methods, we conduct experiments. Results of those experiments demonstrate that, for small data, our proposed method provides more accurate classification than the earlier methods do.

2. Bayesian Network Classifier

2.1 Bayesian network

Letting $\{X_0, X_1, \cdots, X_n\}$ be a set of $n+1$ discrete variables, then $X_i, (i = 0, \cdots, n)$ can take values in the set of states $\{1, \cdots, r_i\}$. One can write $X_i = k$ when observing that an X_i is state k. According to the Bayesian network structure G, the joint probabilities distribution is $P(X_0, X_1, \cdots, X_n) = \prod_{i=0}^{n} P(X_i \mid \Pi_i, G)$, 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 $\Theta = \{\theta_{ijk}\}$ $(i = 0, \cdots, n; j = 1, \cdots, q_i; k = 1, \cdots, r_i)$. A Bayesian network is a pair $B = (G, \Theta)$. In an earlier study, Buntine (1991) assumed the Dirichlet prior and used an expected a posteriori (EAP) estimator $\hat{\theta}_{ijk} = (\alpha_{ijk} + N_{ijk})/(\alpha_{ij} + N_{ij})$. In that equation, N_{ijk} represents the number of samples of $X_i = k$ when $\Pi_i = j$, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Additionally, α_{ijk} denotes the hyperparameters of the Dirichlet prior distributions (α_{ijk} is a pseudo-sample corresponding to N_{ijk}); $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.

The first learning task of the Bayesian network is to seek a structure G optimizing a given score. Let $D = \{\mathbf{u}^1, \cdots, \mathbf{u}^d, \cdots, \mathbf{u}^N\}$ be training dataset. Also, let each $\mathbf{u}^d$ be a tuple of the form $\langle x_0^d, x_1^d, \cdots, x_n^d \rangle$. The most popular marginal likelihood (ML) score, $P(D \mid G)$, of the Bayesian network finds the maximum a posteriori (MAP) structure G^* when we assume a uniform prior $P(G)$ over structures, as presented below.

$$G^* = \arg\max_G P(G \mid D) = \arg\max_G \frac{P(D \mid G)P(G)}{P(D)} = \arg\max_G P(D \mid G).$$

The ML has an asymptotic consistency (Haughton, 1988), i.e., the structure which maximizes ML converges to the true structure when the sample is large. 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. The BDeu score is represented as

$$P(D \mid G) = \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(\frac{\alpha}{q_i} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\frac{\alpha}{r_i q_i} + N_{ijk})}{\Gamma(\frac{\alpha}{r_i q_i})},$$

where α is a hyperparameter. Ueno (2008, 2010, 2011); Ueno and Uto (2012) demonstrated that learning structures is highly sensitive to α. As the best method to mitigate the influence of α for parameter estimation, he reported $\alpha = 1.0$.

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 $X_1, \cdots, X_n$ are feature variables. Given an instance $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$ for feature variables $X_1, \ldots, X_n$, the BNC B predicts the class variable's value by maximizing the posterior as $\hat{c} = \mathrm{argmax}_{c \in \{1, \cdots, r_0\}} P(c \mid \mathbf{x}, B)$.

However, Friedman et al.(1997) reported that the BNC maximizing ML can not optimize the classification performance. They proposed the sole use of the conditional log likelihood (CLL) of the class variable given the feature variables instead of the log likelihood for learning BNC structures.

Unfortunately, no closed-form formula exists for optimal parameter estimates to maximize CLL. Therefore, for each structure candidate, learning the network structure maximizing CLL requires some search methods such as gradient descent over the space of parameters. For that reason, exact learning network structures by maximizing CLL is computationally infeasible.

As a simple means of resolving this difficulty, Friedman et al. (1997) 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. Furthermore, they proposed the tree-augmented naive Bayes (TAN) classifier, 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.

In addition, Carvalho et al. (2011, 2013) proposed an approximate conditional log likelihood (aCLL) score, which is decomposable and computationally efficient. Letting G_{ANB} be an ANB structure, then we define $\Pi_i^* = \Pi_i \setminus \{X_0\}$ based on G_{ANB}. In addition, we let N_{ijck} be the number of samples of $X_i = k$ when $X_0 = c$ and $\Pi_i^* = j$ $(i = 1, \cdots, n; j = 1, \cdots, q_i^*; c = 1, \cdots, r_0; k = 1, \cdots, r_i)$, and let $N' > 0$ represent the number of pseudo-counts. Under several assumptions, aCLL can be represented as

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

where

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

The value of β is found using a Monte-Carlo method to approximate CLL. When the value of β is optimal, then aCLL is a minimum-variance unbiased approximation of CLL. They described that the classifier maximizing the approximated CLL provides better performance than that maximizing the approximated ML.

However, they stated no reason for why CLL outperformed ML. Differences of performance between ML and CLL in earlier studies might depend on the learning algorithms which were employed because they used not exact but approximate learning algorithms. Therefore, Sugahara et al. (2018) experimentally demonstrated that the BNC performance achieved by maximizing the ML is not necessarily worse than that achieved by maximizing CLL for small data. However, the classification accuracy of the structure maximizing the

ML becomes rapidly worse as the sample size becomes small. They explained the reason thusly: the class variable tends to have numerous parents for a small sample. Therefore, estimation of the conditional probability parameters 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 ANB because the class variable has no parent in ANB structures. Additionally, they empirically demonstrated the effectiveness of their method.

3. Model Averaging of Bayesian Network Classifiers

The less accurate classification of BNCs for small data results from learning structures errors. As a method to alleviate this shortcoming, model averaging, which marginalizes the class variable posterior over all structures, is reportedly effective (Madigan and Raftery, 1994; Chickering and Heckerman, 2000). Using model averaging, the class variable's value c is estimated as

$$\widehat{c} = \underset{c\in\{1,\cdots,r_0\}}{\arg\max}\ P(c \mid \mathbf{x}, D) = \underset{c\in\{1,\cdots,r_0\}}{\arg\max} \sum_{G\in\mathcal{G}} P(G \mid D)P(c \mid \mathbf{x}, G) = \underset{c\in\{1,\cdots,r_0\}}{\arg\max} \sum_{G\in\mathcal{G}} P(D \mid G)P(c \mid \mathbf{x}, G),$$

where $\mathcal{G}$ is a set of all structures. However, the number of structures increases super-exponentially for the network size. Therefore, averaging all the structures with numerous variables is computationally infeasible. The most common approach to resolving this problem is a K-best structures method (Tian et al., 2010; He et al., 2016; Chen et al., 2015, 2016, 2018; Liao et al., 2018), which considers only the K-best scoring structures. However, the K-best structures method finds the best K individual structures included in Markov equivalence classes, where the structures within each class represent the same set of conditional independence assertions and determine the same statistical model. To address the difficulty, Chen and Tian (2014) proposed the K-best EC method, which finds the K best equivalence classes directly. These methods have asymptotic consistency if they use an exact learning algorithm. Using the K-best scoring structures, $\{G^k\}_{k=1}^K$, the class variable posterior can be approximated as $P(c \mid \mathbf{x}, D) \approx \sum_{k=1}^K P(D \mid G^k)P(c \mid \mathbf{x}, G^k)$.

The posterior standard error of the structures in the model averaging becomes large as the sample size becomes small; it then decreases the classification accuracy. However, the resampling methods, such as the adaboost (Freund and Schapire, 1997) and the bagging (Breiman, 1996) are known to reduce the standard error of estimation. Actually, Jing et al. (2008) proposed the bAN_{mix} boosting method, which predicts the class variable using adaboost. Nevertheless, this method is not a model averaging method. It tends to cause overfitting for small data because the adaboost tends to be sensitive to noisy data (Dietterich, 2000).

Rohekar et al. (2018) proposed a model averaging method named B-RAI, based on the RAI algorithm (Yehezkel and Lerner, 2009), which learns the structure by sequential application of conditional independence (CI) tests, edge direction and structure decomposition into smaller substructures. This sequence of operations is performed recursively for each substructure, along with increasing order of the CI tests. In each level of recursion, the current structure is first refined by removing edges between variables that are independently

conditioned on a set of size n_z and directing the edges. Then, the structure is decomposed into ancestors and descendant groups. Each group is autonomous in that it includes the parents of its members (Yehezkel and Lerner, 2009). Furthermore, each autonomous group from the n_z-th recursion level is independently partitioned, resulting in a new level of n_z+1. Each such structure (a substructure over the autonomous set) is partitioned progressively (in a recursive manner) until a termination condition is satisfied (independence tests with condition set size n_z cannot be performed), at which point the resulting structure (a substructure) at that level is returned to its parent (the previous recursive call). Similarly, each group in its turn, at each recursion level, gathers back the structures (substructures) from the recursion level that followed it; it then returns itself to the recursion level that precedes it until the highest recursion level $n_z = 0$ is reached and the final structure is fully constructed. Consequently, the RAI constructs a tree in which each node represents a substructure and for which the level of the node corresponds to the maximal order of conditional independence that is encoded in the structure. Based on the RAI algorithm, B-RAI constructs a structure tree from which structures can be sampled. In essence, it replaces each node in the execution tree of the RAI with a bootstrap node. In the bootstrap node, for each autonomous group, s datasets are sampled with replacement from training data D. They calculate $\log[P(D \mid G)]$ for each leaf node in the tree (G is the structure in the leaf) using the BDeu score. For each autonomous group, given s sampled structures and their scores returned from s recursive calls, the B-RAI samples one of the s results proportionally to their (log) score. Finally, the sampled structures are merged. The sum of scores of all autonomous sets is the score of the merged structure.

However, B-RAI does not apply the bagging to the posterior of the structures. Therefore, the posterior standard error of the structures is not expected to decrease. In addition, the B-RAI is not guaranteed to have asymptotic consistency. This engenders reduction of the computational costs, but degradation of the classification accuracy.

4. Proposed Method

This section presents the proposed method, which improves the classification accuracy using resampling methods to reduce the posterior standard error of structures in model averaging. As described in section 3, exact model averaging over all structures is computationally infeasible. A well known solution for this problem is the K-best structures model averaging method with the BDeu score (Tian et al., 2010). However, this method finds the best K individual structures which include equivalent structures. To find the K best equivalence classes directly, we employ the K-best EC method with BDeu score (Chen and Tian, 2014).

The posterior standard error of the structures learned by the K-best EC method becomes large as the sample size becomes small. Then the classification accuracy decreases. As described previously, Jing et al. (2008) proposed a boosting method using adaboost. However, it tends to cause overfitting for small data because it is known to be sensitive to noise data (Dietterich, 2000). An alternative technique for the problem is bagging using random sampling with replacement. However, it is noteworthy that sampling with replacement might increase the standard error of estimation as the sample size becomes small because of the duplicated sampling (Rao, 1966). To avoid this difficulty, we use the subbagging

(Bühlmann and Yu, 2002), which is a modified bagging using random sampling without replacement.

The proposed method is expected to provide the following benefits. (1) Because the class variable posterior has asymptotic consistency, it converges to the true value when the sample size is sufficiently large. (2) Even for small data, the subbagging reduces the posterior standard error of the structures learned using the K-best EC method and improves the classification accuracy. The next section explains experiments conducted to compare the classification performances of the proposed method and earlier methods.

5. Experiments

This section presents experiments comparing the classification accuracy of the following 11 methods. (1) *NB*: Naive Bayes (2) *TAN* (Friedman et al., 1997): Tree augmented naive Bayes (3) *aCLL-TAN* (Carvalho et al., 2013): Exact learning TAN method by maximizing aCLL (4) *EBN*: Exact learning Bayesian network method by maximizing BDeu (5) *EANB*: Exact learning ANB method by maximizing BDeu (6) bAN_{mix} (Jing et al., 2008): Ensemble method using adaboost, which starts with the naive Bayes and greedily augments the current structure at iteration j with the j-th edge having the highest conditional mutual information (7) *Adaboost(EBN)* : Ensemble method of 10 structures learned using adaboost to *EBN* (8) *B-RAI* (Rohekar et al., 2018): Model averaging method over 100 structures sampled by the B-RAI with $s = 3$ (9) *KBestEC100* (Chen and Tian, 2014): K-best EC method using BDeu score with $K = 100$ (10) *Bagging(EBN)* : Ensemble method of 10 structures learned using the bagging to *EBN* (11) The proposed method with $K = 10$ and $T = 10$. Here, the classification accuracy represents the average percentage correct among all classifications from ten-fold cross validation. Although determination of hyperparameter α of BDeu is difficult, we used $\alpha = 1.0$, which allows the data to reflect the estimated parameters to the greatest degree possible (Ueno, 2008, 2010, 2011; Ueno and Uto, 2012). We used EAP estimators with $\alpha_{ijk} = 1/(r_i q_i)$ as conditional probability parameters of the respective classifiers. Using the proposed method and *Bagging(EBN)*, the size of the sampled data is 90% of the training data. We used 26 classification benchmark datasets from the UCI repository. Continuous variables were discretized into two bins using the median value as cut-off. Furthermore, data with missing values were removed from the datasets. Through this section, we define “small datasets” as the datasets with less than 1000 sample size, and define “large datasets” as the datasets with 1000 or more sample size.

To confirm the significant differences of the proposed method from other methods, we applied multiple Hommel tests (Hommel, 1988), which are used as a standard in machine learning studies (Demšar, 2006). Table 1 presents the classification accuracy and p-values obtained using Hommel tests. The results show that, among the methods explained above, the proposed method yields the best average accuracy. Moreover, the proposed method outperforms almost all model selection methods, except for *EANB*, at the $p < 0.10$ significance level. Particularly *NB*, *TAN*, and *aCLL-TAN* provide lower classification accuracy than the proposed method does for the No.1, No.20, and No.24 datasets. The reason is that those methods have the small upper bound of maximum number of parents. Such a small upper bound is known to cause poor representational power of the structure (Ling and Zhang, 2003). The classification accuracy of *EBN* is the same or almost identical to that of the

				(1) Classification accuracy											(2) Parents		(3) APSES	
No.	Datasets	Sample size	Variables	NB	TAN	aCLL-TAN	EBN	EANB	bAN_{mix}	Adaboost (EBN)	B-RAI	Bagging (EBN)	KBest EC100	The proposed method	EBN	The proposed method	KBest EC100	The proposed method
1	lenses	24	5	0.6250	0.7083	0.7083	0.8125	**0.8750**	0.6667	0.8125	0.8500	0.8333	0.8333	0.8333	0.9000	1.3700	0.0631	**0.0425**
2	mux6	64	7	0.5469	0.6094	0.5938	0.4531	0.5469	0.5938	0.4531	0.3238	0.6094	0.4219	**0.6250**	5.7000	4.6790	0.0625	**0.0600**
3	post	87	9	0.6552	0.6322	0.5977	0.7126	0.7126	0.6552	0.7126	**0.7139**	0.7126	0.7126	0.7126	0.0000	0.0240	0.0817	**0.0547**
4	zoo	101	17	**0.9901**	0.9406	0.9505	0.9426	0.9604	**0.9901**	0.9406	0.9435	0.9604	0.9505	0.9505	3.7000	4.3050	**0.0599**	0.0600
5	HayesRoth	132	5	0.8106	0.6439	0.6742	0.6136	**0.8333**	0.6970	0.6136	0.6143	0.6136	0.7803	0.7727	3.0000	2.4550	0.0600	0.0600
6	iris	150	5	0.7133	**0.8267**	0.8200	**0.8267**	0.8067	**0.8267**	0.8200	0.8133	**0.8267**	0.8200	**0.8267**	1.8000	1.8920	0.0686	**0.0564**
7	wine	178	14	0.9270	0.9213	0.9157	0.9438	0.9270	0.9326	0.9213	0.8941	**0.9551**	0.9438	0.9438	1.7000	1.4000	0.0702	**0.0545**
8	glass	214	10	0.5421	0.5467	**0.6215**	0.5607	0.5280	0.5981	0.5701	0.5470	0.5701	0.5748	0.5748	0.4000	0.6800	0.0691	**0.0600**
9	CVR	232	17	0.9095	0.9526	0.9224	0.9612	0.9526	0.9310	0.9655	0.9697	**0.9698**	0.9655	**0.9698**	0.9000	1.4170	0.0789	**0.0504**
10	heart	270	14	0.8296	0.8333	0.8148	0.8296	**0.8444**	0.8333	0.8074	0.7611	0.8407	0.8333	0.8370	1.7000	1.5400	0.0722	**0.0600**
11	BreastCancer	277	10	0.7365	0.7220	0.6968	0.7076	0.6751	0.7148	**0.7509**	0.6888	0.7004	0.7329	0.7220	0.7000	0.8230	0.0677	**0.0600**
12	cleve	296	14	0.8311	0.8243	**0.8446**	0.8074	0.8142	0.8176	0.7939	0.7771	0.8108	0.8176	0.8176	1.9000	1.6870	0.0722	**0.0547**
13	liver	345	7	0.6464	0.6609	0.6522	0.5768	0.6058	**0.6638**	0.5971	0.5995	0.6174	0.6261	0.6232	0.0000	0.1910	0.0697	**0.0600**
14	threeOf9	512	10	0.8008	0.8691	0.8906	0.8691	0.8672	0.8789	0.9063	0.7598	0.8906	**0.9434**	0.9023	5.0000	3.8540	0.0600	0.0600
15	crx	653	16	0.8392	0.8515	0.8453	0.8392	**0.8622**	0.8331	0.8591	0.8590	0.8499	0.8484	0.8499	1.2000	1.0810	0.0600	0.0600
16	Australian	690	15	0.8348	0.8290	0.8478	0.8565	0.8580	0.8333	**0.8638**	0.8493	0.8464	0.8478	0.8464	1.0000	1.1360	0.0685	**0.0600**
17	pima	768	9	0.7057	0.7188	0.7031	0.7253	0.7188	0.7083	0.7240	0.7123	0.7227	**0.7331**	0.7266	1.6000	1.0900	0.0649	**0.0600**
18	TicTacToe	958	10	0.6889	0.7599	0.7192	0.8549	0.8445	0.7505	**0.9123**	0.6994	0.8466	0.8486	0.8518	1.6000	0.3960	0.0674	**0.0600**
19	banknote	1372	5	0.8433	**0.8819**	0.8761	0.8812	0.8812	0.8754	0.8776	0.8812	0.8812	0.8812	0.8812	0.0000	0.6890	0.0600	0.0600
20	Solar Flare	1389	11	0.7804	0.7970	0.8200	**0.8431**	**0.8431**	0.8143	**0.8431**	0.8409	**0.8431**	**0.8431**	**0.8431**	0.8000	0.9120	0.0693	**0.0600**
21	CMC	1473	10	0.4644	0.4725	0.4650	0.4549	0.4270	**0.4779**	0.4399	0.4100	0.4521	0.4616	0.4487	0.9000	0.8230	0.0600	0.0600
22	led7	3200	8	0.7288	0.7309	**0.7347**	0.7288	0.7288	0.7300	0.7288	0.7228	0.7284	0.7303	0.7309	0.6000	0.9540	0.0651	**0.0600**
23	shuttle-small	5800	10	0.9383	0.9567	0.9538	0.9693	**0.9716**	0.9681	0.9662	0.9659	0.9693	0.9693	0.9693	2.0000	2.1150	0.0600	0.0600
24	EEG	14980	15	0.5774	0.6298	0.6138	0.6844	0.6895	0.6031	**0.6906**	0.6450	0.6881	0.6885	0.6899	0.5000	0.4710	0.0600	0.0600
25	HTRU2	17898	9	0.8966	**0.9141**	**0.9141**	**0.9141**	**0.9141**	0.9102	0.9073	0.9066	**0.9141**	**0.9141**	**0.9141**	1.5000	1.6230	0.0600	**0.0550**
26	MAGICGT	19020	11	0.7447	0.7769	0.7656	0.7859	**0.7879**	0.7734	0.7849	0.7827	0.7859	0.7871	0.7860	0.0000	0.4710	0.0600	0.0600
Average accuracy				0.7541	0.7696	0.7678	0.7752	0.7875	0.7722	0.7793	0.7512	0.7861	0.7888	**0.7942**	1.5038	1.4645	0.0660	**0.0581**
p-values				0.0019	0.0033	0.0093	0.0025	> 0.1000	0.0139	0.0394	0.0002	0.0629	> 0.1000	-	-	-	0.0001	-

Table 1: (1) Classification accuracies of each BNC, (2) average numbers of the class variable's parents in the structures of the *EBN* and those of the proposed method, and (3) average posterior standard errors of structures (APSES) of the *KBestEC100* and those of the proposed method.

proposed method for large datasets such as No.20, No.23, No.25, and No.26 datasets because both methods have asymptotic consistency. However, the classification accuracy of the proposed method is equal to or greater than that of *EBN* for small datasets from No.1 to No.15. As described previously, the classification accuracy of *EBN* is worse than that of the model averaging methods because the error of learning structure by *EBN* becomes large as the sample size becomes small.

Although *EANB* has lower computational costs than *EBN* and although all the compared model averaging methods do, no significant difference was found between the *EANB* and the proposed method. In fact, *EANB* can represent any joint probability distribution when the sample size is sufficiently large (Ling and Zhang, 2003). Therefore, the classification accuracy of *EANB* is the same or almost identical as that of the proposed method for large datasets, such as No.20, No.24 and No.25 datasets. For almost small datasets such as the datasets from No.6 to No.9 and from No.11 to No.13, the proposed method provides higher classification accuracy than *EANB* does because the error of learning ANB structures becomes large. However, for the No.4 and No.5 datasets, the classification accuracy of *EANB* is much higher than that obtained using the proposed method. To analyze this phenomenon, we investigate the average number of the class variable's parents in the structures learned by *EBN* and that by the proposed method. The results displayed in "Parents" of Table 1 illustrate that the average number of the class variable's parents in the structures learned by *EBN* and that by the proposed method tends to be large in the No.4 and No.5 datasets. Consequently, the estimation of conditional probability parameters of the class variable becomes unstable because the number of parent configurations becomes large. Then the sample size for learning a parameter becomes sparse. Actually, the ANB constraint prevents numerous parents of the class variable. Moreover, it improves the classification accuracy.

The proposed method outperforms almost all model averaging methods, except for *KBestEC100*, at the $p < 0.10$ significance level. The bAN_{mix} provides much lower accuracy than the proposed method does in No.1, No.20, and No.24 datasets because it has the small upper bound of a maximum number of parents, similar to *NB*, *TAN*, and *aCLL-TAN*. For almost all large datasets, the classification accuracy of the proposed method is higher than that of *B-RAI* because the proposed method has an asymptotic consistency, whereas *B-RAI* does not. The proposed method provides higher classification accuracy than *Adaboost(EBN)* does for small datasets, such as No.5 and No.10 datasets, because *Adaboost(EBN)* tends to cause overfitting, as described in section 3. The classification accuracy of *Bagging(EBN)* is much worse than that of the proposed method in the No.5 dataset because the error of learning structures using each sampled data becomes large as the sample size becomes small. The proposed method alleviates this difficulty somewhat using model averaging for sampled data.

Although the proposed method does not significantly outperform *KBestEC100*, it provides higher average accuracy than *KBestEC100* does. To demonstrate the advantage of the proposed method for small data, we compare the posterior standard error of the structures learned using the proposed method with that learned by *KBestEC100*. We estimate the posterior standard error of structures learned by the *KBestEC100* as explained below.

1. Generate 10 random structures $\{G_m\}_{m=1}^{10}$.
2. Sample 10 datasets, $\{\widetilde{D}_i\}_{i=1}^{10}$, with replacement from the training dataset D, where $|\widetilde{D}_i| = |D|$.
3. Compute the posteriors $P(G_m \mid \widetilde{D}_i) \approx P(\widetilde{D}_i \mid G_m)/\sum_{m'=1}^{10} P(\widetilde{D}_i \mid G_{m'}), (m = 1, \cdots, 10; i = 1, \cdots, 10)$.
4. Estimate the standard error of the posteriors $P(G_m \mid D), (m = 1, \cdots, 10)$ as

$$\sqrt{\frac{1}{10(10-1)} \sum_{i=1}^{10} \left\{ P(G_m \mid \widetilde{D}_i) - \frac{1}{10} \sum_{j=1}^{10} P(G_m \mid \widetilde{D}_j) \right\}^2}. \tag{1}$$

We estimate the posterior standard error of structures learned using the proposed method as listed below.

1. Generate 10 random structures $\{G_m\}_{m=1}^{10}$.
2. Sample 10 datasets, $\{\widetilde{D}_{ti}\}_{i=1}^{10}$, with replacement from each bootstrapped dataset D_t, where $|\widetilde{D}_{ti}| = |D_t|$.
3. Compute the posteriors $P(G_m \mid \widetilde{D}_i) \approx \frac{1}{T} \sum_{t=1}^{T} [P(\widetilde{D}_{it} \mid G_m)/\sum_{m'=1}^{10} P(\widetilde{D}_{it} \mid G_{m'})], (m = 1, \cdots, 10; i = 1, \cdots, 10)$.
4. Estimate the standard error of the posteriors $P(G_m \mid D), (m = 1, \cdots, 10)$ using formula (1).

Average posterior standard errors over 10 structures $\{G_m\}_{m=1}^{10}$ of the proposed methods and those of the *KBestEC100* are presented in "APSES" of Table 1. We obtain the significance values for these results obtained using the Wilcoxon signed-rank test. The p-values of the test are presented at the bottom of Table 1. The results demonstrate that the APSES of the proposed method is significantly lower than that of the *KBestEC100*.

Moreover, we investigate the relation between the APSES and the training data sample size. As presented in Figure 1, the APSES of *KBestEC100* tends to become large as the sample size becomes small, although the APSES of the proposed method does not become large as the sample size becomes small. Particularly the proposed method provides higher classification accuracy than *KBestEC100* does when the APSES of the proposed method are lower than those of the *KBestEC100*, such as those of No.3, No.6, and No.9 datasets. Consequently, the proposed method reduces the posterior standard error of the structures. It therefore improves the classification accuracy.

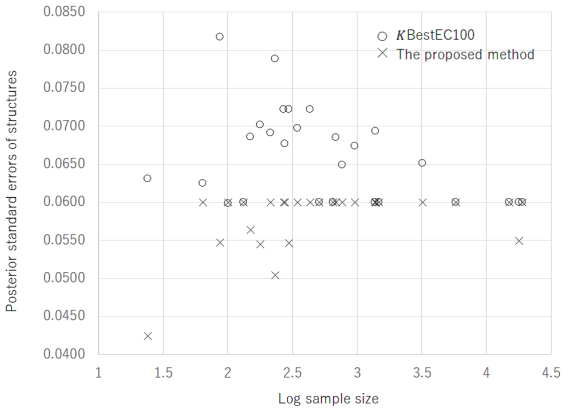

Figure 1: Average posterior standard errors of structures (APSES) of the *KBestEC100* and those of the proposed method.

6. Conclusions

As described herein, we improve the K-best method classification accuracy using the sub-bagging to reduce the posterior standard error of the structures. Our experiments demonstrate that the proposed method provides more accurate classification than the K-best EC method does for small data. Even for large data, the proposed method provides highly accurate classification because it has asymptotic consistency. However, results show that the classification accuracy of the *EANB* is comparable to that of the proposed method, although the *EANB* has lower computational costs than the proposed method does. In practice, *EANB* might be more useful than the proposed method for learning large classifiers.

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.

References

P. Bühlmann and B. Yu. Analyzing bagging. *Ann. Statist.*, 30(4):927–961, 08 2002. doi: 10.1214/aos/1031689014.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug. 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350.

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

A. M. Carvalho, T. Roos, A. L. Oliveira, and P. Myllymäki. Discriminative Learning of Bayesian Networks via Factorized Conditional Log-Likelihood. *Journal of Machine Learning Research*, 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.

E. Y.-J. Chen, A. Choi, and A. Darwiche. Learning Bayesian networks with non-decomposable scores. In *Graph Structures for Knowledge Representation and Reasoning*, pages 50–71, 2015. ISBN 978-3-319-28702-7.

E. Y.-J. Chen, A. C. Choi, and A. Darwiche. Enumerating equivalence classes of Bayesian networks using ec graphs. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 591–599, 2016.

E. Y.-J. Chen, A. Darwiche, and A. Choi. On pruning with the MDL score. *International Journal of Approximate Reasoning*, 92:363 – 375, 2018. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2017.10.023.

Y. Chen and J. Tian. Finding the k-best equivalence classes of Bayesian network structures for model averaging. *Proceedings of the National Conference on Artificial Intelligence*, 4:2431–2438, 2014.

D. M. Chickering and D. Heckerman. A comparison of scientific and engineering criteria for Bayesian model selection. *Statistics and Computing*, 10(1):55–62, 2000. doi: 10.1023/A:1008936501289.

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

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000. doi: 10.1023/A:1007607513941.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

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 *Proceedings, Twenty-First International Conference on Machine Learning*, pages 361–368, 2004.

D. M. A. Haughton. On the choice of a model to fit data from an exponential family. *Annals of Statistics*, 16(1):342–355, 1988. doi: 10.1214/aos/1176350709.

R. He, J. Tian, and H. Wu. Structure learning in Bayesian networks of a moderate size by efficient sampling. *Journal of Machine Learning Research*, 17(101):1–54, 2016.

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. doi: 10.1109/ICTAI.2008.56.

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, 2009.

Y. Jing, V. Pavlović, and J. M. Rehg. Boosted Bayesian network classifiers. *Machine Learning*, 73 (2):155–184, 2008.

Z. Liao, C. Sharma, J. Cussens, and P. van Beek. Finding all Bayesian network structures within a factor of optimal. In *The Thirty-third AAAI Conference on Artificial Intelligence*, 2018.

C. X. Ling and H. Zhang. The representational power of discrete Bayesian networks. *J. Mach. Learn. Res.*, 3:709–721, 2003.

D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

J. N. K. Rao. On the comparison of sampling with and without replacement. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute*, 34(2):125–138, 1966. ISSN 03731138.

R. Y. Rohekar, Y. Gurwicz, S. Nisimov, G. Koren, and G. Novik. Bayesian structure learning by recursive bootstrap. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 10546–10556, 2018.

S. Sugahara, M. Uto, and M. Ueno. Exact learning augmented naive Bayes classifier. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, volume 72, pages 439–450, 2018.

J. Tian, R. He, and L. Ram. Bayesian model averaging using the k-best Bayesian network structures. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, page 589–597, 2010. ISBN 9780974903965.

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

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

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

M. Ueno and M. Uto. Non-informative dirichlet score for learning bayesian networks. *Proceedings of the 6th European Workshop on Probabilistic Graphical Models, PGM 2012*, pages 331–338, 01 2012.

R. Yehezkel and B. Lerner. Bayesian network structure learning by recursive autonomy identification. *J. Mach. Learn. Res.*, 10:1527–1570, 2009. ISSN 1532-4435.