Applied Intelligence (2022) 52:15496–15516
https://doi.org/10.1007/s10489-022-03356-z

Stochastic optimization for bayesian network classifiers

Yi Ren[1] · LiMin Wang[2] · XiongFei Li[2] · Meng Pang[3] · JunYang Wei[1]

Accepted: 7 February 2022 / Published online: 16 March 2022

Abstract
How to reduce the complexity of network topology and make the learned joint probability distribution fit data are two important but inconsistent issues for learning Bayesian network classifier (BNC). By transforming one single high-order topology into a set of low-order ones, ensemble learning algorithms can include more hypothesis implicated in training data and help achieve the tradeoff between bias and variance. Resampling from training data can vary the results of member classifiers of the ensemble, whereas the potentially lost information may bias the estimate of conditional probability distribution and then introduce insignificant rather than significant dependency relationships into the network topology of BNC. In this paper, we propose to learn from training data as a whole and apply heuristic search strategy to flexibly identify the significant conditional dependencies, and then the attribute order is determined implicitly. Random sampling is introduced to make each member of the ensemble "unstable" and fully represent the conditional dependencies. The experimental evaluation on 40 UCI datasets reveals that the proposed algorithm, called random Bayesian forest (RBF), achieves remarkable classification performance compared to the extended version of state-of-the-art out-of-core BNCs (e.g., SKDB, WATAN, WAODE, SA2DE, SASA2DE and IWAODE).

Keywords Bayesian network classifiers · Ensemble learning · Stochastic optimization · Random sampling

1 Introduction

Classification is a fundamental issue in machine learning and data analysis that requires to learn a classifier or a function, which can assign the right class labels to different instances represented by an attribute vector [1]. Bayesian network classifiers (BNCs) [2] describe data in the form of directed acyclic graph (DAG) in which nodes represent the attributes in a given domain and edges connecting the respective nodes indicate the dependencies between these attributes. However, learning a full Bayesian network classifier is very time consuming and quickly becomes an NP-hard problem with the increasing number of attributes [3].

In consequence, how to learn restricted BNCs has attracted a lot of research interest in the past decades [4–7]. Numerous supervised BNCs have been proposed, such as Naive Bayes (NB) [8, 9], tree-augmented Naive Bayes (TAN) [10] and k-dependence Bayes (KDB) [11, 12]. However, these BNCs can only represent a limited number of conditional dependencies, which are always the most significant [13]. Information-theoretic metrics, e.g., mutual information (MI) and conditional mutual information (CMI), are commonly applied to roughly quantify the mutual or conditional dependence between the attributes. However, due to the limitation in structure complexity and computation complexity, the biased estimate of high-order conditional probability may result in poor performance especially when processing small data.

To address this issue, researchers propose to learn ensemble of classifiers [14–16] which combines multiple learning models' decisions to classify new examples by (weighted) voting. Ensembles are more likely to include more hypothesis, and often perform better than the single classifiers that make them up. Ensemble learning does not require the learned BNC to represent high-order

✉ LiMin Wang
wanglim@jlu.edu.cn

[1] College of Software, Jilin University, ChangChun 130012, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, ChangChun 130012, China

[3] College of Computer Science and Technology, Jilin University, ChangChun 130012, China

dependencies, and the network topology of each member is relatively simpler, thus high-confidence estimate of low-order conditional probability helps approximate the true target function [17]. Boosting [18] and bagging [19] are the two most popular ensemble learning approaches, each of them trains classifiers with different subsets of the training data [20].

The key issue for constructing an ensemble is how to vary the results of the member classifiers, and keep the covariance between the (varied) classifiers low, but not raise the bias of the base classifier [21]. The instances from the training dataset may not follow the same probability distribution, thus the (conditional) dependency relationships between variables may vary greatly for different instances, and these relationships should be fully represented by committee members of the ensemble. Boosting and bagging vary the learned (conditional) dependency relationships by resampling, whereas that may potentially lose information implicit in the data as a whole. The estimate of conditional probability distribution will be biased and when applied to compute MI or CMI, insignificant rather than significant dependency relationships may be introduced to the network topology of BNC.

According to Bayes theorem and chain rule of the joint probability, BNC assumes predictive attribute X_j as the candidate parent of attribute X_i and represents conditional dependence between them in the network topology only when $j < i$ holds. That is, for attribute order $\{X_1, X_2, \cdots, X_n\}$ given implicitly or explicitly, attribute X_j should appear before X_i. For different instances, predictive attributes may take specific values, thus the attribute order and the dependency relationships may vary greatly. Although significant dependency relationships appear much more often than insignificant ones, the latter may also appear for specific instances.

Breiman [19] reveals that ensemble learning improves classification accuracy for "unstable" learning procedure where small changes in the training set may result in large changes in models. To systematically create multiple BNCs using the same dataset, we propose to apply stochastic optimization to make the learned BNCs unstable. Different BNCs can represent dependency relationships from different aspects and then generalize their classification in complementary ways. The main contributions and innovations of this paper can be highlighted as follows:

- Following the principles of stochastic optimization, we propose to learn the attribute order and conditional dependencies by applying random sampling. The resulting highly scalable algorithm, called RBF (random Bayesian forest), combines the low variance of ensemble learning with the low bias of high-dependence topology.
- We compare the classification performance of RBF with extended version of other state-of-the-art BNCs (e.g., SKDB, WATAN, WAODE, SA2DE, SASA2DE and IWAODE) on 40 datasets, ranging in size from 5 to 60 attributes and 57 to 164,860 instances. We show that our algorithm shows competitive classification performance in terms of zero-one loss, root mean squared error(RMSE), bias-variance decomposition and Friedman test.

The paper is organized as follows: we review the background and provide a brief introduction to ensemble learning in Section 2. Section 3 describes in detail our base learning algorithm, called random Bayesian classifier (RBC), and then the ensemble of RBC, called random Bayesian forest (RBF). Section 4 presents the experimental evaluation and compares the performance of RBF with related BNCs. Section 5 draws the conclusions.

2 Related work

2.1 Bayesian network classifier

The Bayesian network (BN) defines a pair $(\mathcal{G}, \Theta)$ that encodes a joint probability distribution over a set of attributes $\mathbf{X} = \{X_1, \ldots, X_n\}$ and class variable Y. It consists of two parts: (1) a DAG $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ whose nodes $\mathcal{V}$ represent attributes, and edges $\mathcal{E}$ represent attribute dependencies, and (2) the parameters Θ which quantifies the network topology. BNC approximates joint probability distribution with a factorization according to a BN. Given a specific instance $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, BNC $\mathcal{B}$ assigns the maximum a posteriori (MAP) class (simply y^*) to $\mathbf{x}$ by

$$\begin{aligned} y^* &= \arg\max_{y\in Y} P_{\mathcal{B}}(y|\mathbf{x}) = \arg\max_{y\in Y} \frac{P_{\mathcal{B}}(\mathbf{x}, y)}{P_{\mathcal{B}}(\mathbf{x})} \propto \arg\max_{y\in Y} P_{\mathcal{B}}(\mathbf{x}, y) \\ &= \arg\max_{y\in Y} P(y) \prod_{i=1}^{n} P(x_i|\pi_i^{\mathcal{B}}, y) \end{aligned} \tag{1}$$

where $\pi_i^{\mathcal{B}}$ represents the parent attributes of X_i in $\mathcal{B}$. Thus the problem of learning posterior probability $P_{\mathcal{B}}(y|\mathbf{x})$ for classification turns to be the problem of learning joint probability $P_{\mathcal{B}}(\mathbf{x}, y)$ for data fitting. Equation (1) implicitly requires to learn an attribute order first so that attribute X_i can only select parents from attributes before it in the order. To make the network topology of BNC fit data, each factor (i.e., $P(x_i|\pi_i^{\mathcal{B}}, y)$) in (1) should help maximize the estimate of $P_{\mathcal{B}}(\mathbf{x}, y)$. For full BNC, the i-th attribute may have at most $i-1$ candidate parents, whereas high-order dependencies will lead to biased estimate of conditional probability. We take the dataset `magic` [1] from the UCI repository of machine learning as an example for

[1] https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope

experimental study. Dataset `magic` contains the result of simulated registration of high energy gamma particles in an atmospheric Cherenkov telescope with 19,020 instances, 10 attributes and 2 class labels.

Figure 1(a)∼1(f) respectively visualize the distribution of values of $P(x_2|x_i, y)$ ($i = 1, 3, 4, 7, 9$, or 10). The Y-axis in Fig. 1 represents the values of $P(x_2|x_i, y)$ sorted in descending order when $x_2 = 4$ and $y = 1$, the X-axis represents the index number of values of X_i, and the dotted line represents the value of $P(x_2|y)$. As shown in Fig. 1, for different values of X_i, there exist instances corresponding to $P(x_2|y) > P(x_2|x_i, y)$ although they may appear less often. That is, X_i may be independent from X_2 and thus the lower-order probability $P(x_2|y)$ is more reasonable than higher-order probability $P(x_2|x_i, y)$ in some cases.

NB is the simplest BNC with a strong independence assumption that all the attributes are independent given class variable Y, thus NB doesn't need to learn the attribute order and conditional dependencies. However, the conditional independence assumption of NB may hold only while dealing with sparsely distributed datasets, so its estimates of conditional probabilities are often suboptimal. In contrast, KDB provides a highly scalable learning approach that alleviates some of NB's independence assumption by allowing each non-class attribute to have up to k parents. KDB first computes mutual information $I(X_i; Y)$ to sort attributes. Each attribute X_i can have at most k parent attributes with the highest values of $I(X_i; X_j|Y)$ according to the current topology. Flexible KDB (FKDB) [22] takes an efficient heuristic learning strategy to sort attributes by comparing conditional entropy. To control the structure complexity, FKDB selects a subset of π_i, which can minimize the conditional entropy of attribute X_i, and learns significant causal relationships from data. Selective KDB (SKDB) [23] evaluates and selects candidate parents for X_i from all possible attribute subsets $\{x_1, \cdots, x_{i-1}\}$, and chooses the value of k up to the maximum capacity available. Thus it is highly scalable and achieves a good tradeoff between structure complexity and classification accuracy.

2.2 Ensemble learning

As discussed above, the same conditional probability distribution may fit different instances to different extents. Ensemble learning improves the performance of single learners by training multiple learners to encode possible probability distributions and then combine them. To learn ensemble of BNCs and combine the decisions of committee members, Webb et al. [24] propose the averaged one-dependence estimators (AODE), which respectively chooses each attribute as the parent of all the other attributes, and then averages all superparent one-dependence estimators (SPODEs). Jiang et al. [25] propose the weighted AODE (WAODE) to assign different weights to SPODEs. Kong et al. [26] propose the averaged tree-augmented one-dependence estimators (ATODE), which adds the augmented edges for each SPODE by identifying causality between attributes. Jiang et al. [27] propose the averaged tree augmented naive Bayes (ATAN), which selects each attribute as the root node to build a one-dependence maximum weighted spanning tree, and then averages all of the spaning TAN classifiers. Hellman et al. [28] introduce the ensembled continuous Bayesian networks (ECBNs), which predict values for continuous random variables and discover salient dependence relationships. Geiger and Heckerman [29] propose the Bayesian multinet to learn a single network for different partitions of the class label, and then use multiple networks to encode asymmetric independence assertions.

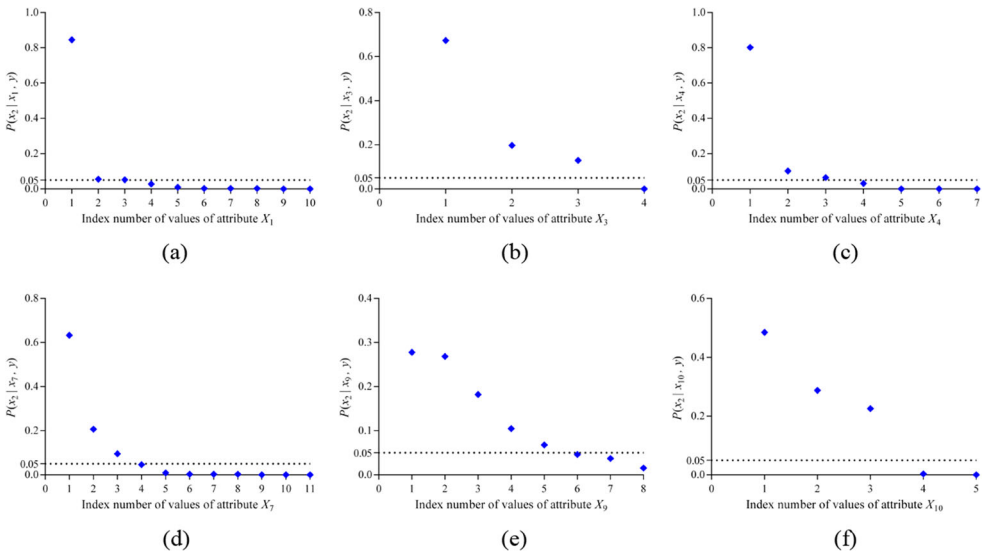


Fig. 1 The distributions of values of $P(x_2|x_i, y)$ on dataset `magic`

Bagging and boosting are popular ensemble learning approaches that combine arbitrary number of base-learners, and are applicable to different machine learning algorithms. By manipulating the training data given to a "base" learning algorithm, bagging [19] uses the bootstrap [30] (i.e., sampling with replacement) to broaden "independency" among the component classifiers [31]. The outputs of the subclassifiers are finally combined by averaging or voting. Boosting [18, 32] iteratively trains subclassifiers on weighted training data. A "weak" classifier (e.g., NB) is built first and then a succession of classifiers are built iteratively. The data points misclassified by the previous classifier are given more weight. AdaBoost [33] is the most popular boosting algorithm.

Diversity in training data or learning procedure helps to build diverse learners, and randomness provides an efficient and effective way to introduce diversity. Ho [34] proposes random subspace method (RSM) to randomly select a subset of attributes. Kunwar et al. [35] propose Random Bayesian Network (RBN) which trains on a different set of training samples (Bagging) and attribute set (RSM). Ma et al. [36] propose Bagging-MultiTAN to respectively train different TAN classifiers on different training subsets. Breiman [37] proposes Random Forest (RF) which uses bagging to aggregate multiple decision trees that are each grown using a process that involves a stochastic element to increase diversity.

3 Random bayesian forest

3.1 Our motivation and structural learning framework

Information-theoretic metrics, e.g., mutual information $I(X_i; Y)$ or conditional mutual information $I(X_i; X_j|Y)$ defined as follows, are widely applied to measure (conditional) dependency relationships for BNC learning.

$$\begin{cases} I(X_i; Y) = \sum\limits_{X_i, Y} P(x_i, y) \log \dfrac{P(x_i, y)}{P(x_i)P(y)} = \sum\limits_{X_i, Y} P_i \log \delta_i \\ I(X_i; X_j|Y) = \sum\limits_{X_i, X_j, Y} P(x_i, x_j, y) \log \dfrac{P(x_i, x_j|y)}{P(x_i|y)P(x_j|y)} = \sum\limits_{X_i, X_j, Y} P_{ij} \log \delta_{ij} \end{cases} \tag{2}$$

$I(X_i; Y)$ measures the direct mutual dependence between predictive attribute X_i and class variable Y, and thus it can also measure the extents to which X_i is effective for classification. For restricted BNCs, all the predictive attributes depend on a common class variable Y, and augmented edges measured by $I(X_i; X_j|Y)$ are added to represent conditional dependence between attributes only. Given specific instance $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$, if $P(x_i, y) > P(x_i)P(y)$ then x_i and y are mutually dependent rather than independent in term of probability theory. The higher the value of δ_i, the stronger the probability-theoretic dependence between x_i and y. As shown in (2), significant mutual dependence needs to satisfy two requirements: stronger probability-theoretic dependence measured by δ_i, and higher probability measured by P_i when the dependence happens. Similarly, significant conditional dependence also needs to satisfy two requirements: stronger probability-theoretic conditional dependence measured by δ_{ij}, and higher probability measured by P_{ij} when the conditional dependence happens.

Given an attribute order $\{X_1, X_2, \cdots, X_n\}$, implicitly or explicitly, X_i is the candidate parent for attributes $\{X_{i+1}, \cdots, X_n\}$ in the order. If the first few attributes in the order are relatively independent from the rest of attributes, less conditional dependencies will be introduced for them and that may bias the estimates of conditional probabilities. To achieve the bias-variance tradeoff for structure learning, we propose to apply heuristic search strategy to flexibly learn the significant conditional dependencies, and then the attribute order is determined implicitly.

The learning framework of proposed RBF is depicted in Fig. 2. RBF is an ensemble of RBCs which are obtained by applying stochastic optimization. For given dataset $\mathcal{D}$ after preprocessing, RBC selects one attribute X_r as the root node which is assumed to be dependent on class variable only, and then it will be added to the network topology $\mathcal{G}$. Then RBC recursively selects the next attribute X_i and adds it to $\mathcal{G}$. X_i is assumed to be dependent on all the attributes already in $\mathcal{G}$, or all the attributes in $\mathcal{G}$ are candidate parents of X_i. Due to the restriction in structure complexity, X_i can only selects limited number of attributes as its parents Π_i. Thus during the learning procedure, RBC needs to select X_r, X_i and Π_i, and RBC selects by performing random sampling based on the probability distribution. After that RBCs vote for the most possible class label.

3.2 RBF learning algorithm

3.2.1 Random selection of root node X_r

For full BNC there exists directed edge $X_j \rightarrow X_i$ between attributes X_j and X_i when $j < i$. That is, the root attribute X_r is the only one that is dependent on class variable Y, whereas the other attributes are dependent on X_r to different extents. Thus similar to KDB, mutual information $I(X_i; Y)$ is introduced to measure the significance of attribute X_i. Higher value of $I(X_i; Y)$ corresponds to stronger mutual

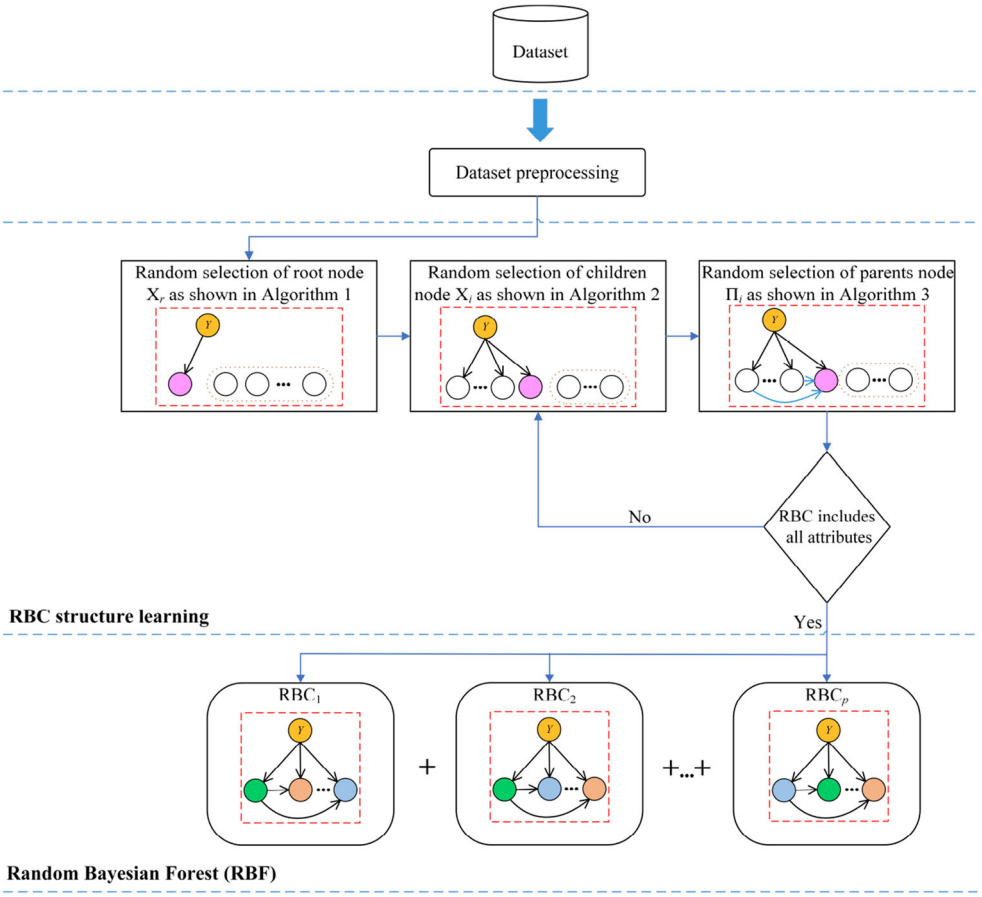


Fig. 2 The learning framework of RBF

dependence related to attribute X_i. Thus X_i with the highest value of $I(X_i; Y)$ is considered in priority as the candidate root attribute. We first normalize the values of $I(X_i; Y)$ for different attributes and transform them into the form of probability. Then the attributes are listed in descending order of the normalized probability. To introduce diversity and mitigate the negative effect of overfitting, as described in Algorithm 1, we perform random sampling based on the probability distribution to select root attribute from the list, and then add it to the network topology. The root attribute is also the first candidate parent attribute for the other attributes.

Algorithm 1 SelectRoot($\mathcal{D}$).

Input: Training dataset $\mathcal{D}$
Output: Root node X_r
1 Calculate mutual information $I(X_i; Y)(1 \leq i \leq n)$ from $\mathcal{D}$ for all the attributes;
2 $\triangle I \leftarrow$ normalize($I(X_i; Y)$) // Normalize $I(X_i; Y)$ into a probability distribution that sums to 1
3 $X_r \sim \triangle I$ // Select attribute X_r by random sampling from a multinomial distribution
4 **return** Attribute X_r.

3.2.2 Random selection of children node X_i

The attributes already in the network topology $\mathcal{G}$ constitute candidate parents for the newly added children attributes, or the children attributes should have strong conditional dependence on the parent attributes. Thus children correlation criterion $CCC(X_i|\mathcal{G})$ is introduced to select the children attribute as follows.

$$CCC(X_i|\mathcal{G}) = \sum_{X_j \in \mathcal{G}, i \neq j} I(X_i; X_j|Y) \tag{3}$$

The values of $CCC(X_i|\mathcal{G})(X_i \notin \mathcal{G})$ are normalized first and transformed into the form of probability. Then the corresponding attributes are sorted into list $\triangle CCC$ in descending order of the normalized probability. As described in Algorithm 2, we perform random sampling to select children node from the list, and then add it to the network topology. But note that, if the candidate children attributes are assumed to be independent from the attributes already in $\mathcal{G}$, i.e., $CCC(X_i|\mathcal{G}) = 0(X_i \notin \mathcal{G})$, then one of them will be randomly selected as the children attribute.

Algorithm 2 SelectChildren($\mathcal{G}$, $\mathbf{X}$).

Input: The network topology $\mathcal{G}$ and attribute set $\mathbf{X}$.
Output: New children attribute X_i and independent attribute set $\mathcal{Q}_I$.

Let $\mathcal{Q}$ be a list of all the attributes in $\mathcal{G}$, and $\hat{\mathcal{Q}} = \mathbf{X} - \mathcal{Q}$;
Calculate $CCC(X_j|\mathcal{G})$ for each attribute X_i in $\hat{\mathcal{Q}}$;
Remove attribute X_j from $\hat{\mathcal{Q}}$ when $CCC(X_j|\mathcal{G}) = 0$, and add it to $\mathcal{Q}_I$;
if $\hat{\mathcal{Q}}.size > 0$ **then**
    $\triangle CCC \leftarrow$ normalize(CCC) // Normalize CCC into a probability distribution that sums to 1
    $X_i \sim \triangle CCC$ // Perform random sampling from a multinomial distribution
else
    $X_i \sim \mathcal{Q}_I$//Randomly select X_i from $\mathcal{Q}_I$.
end
return attribute X_i and independent set $\mathcal{Q}_I$.

3.2.3 Random selection of parents node Π_i

Due to the restriction in computational complexity and structure complexity, each children attribute X_i can only select limited number of parent attributes, or more precisely, at most k parents for k-dependence BNC. Thus conditional mutual information $I(X_i; X_j|Y)$ is introduced to select parent attributes for X_i as follows. Similar to the learning procedure described in Algorithm 2, Algorithm 3 applies random sampling to select parents from attributes in $\mathcal{G}$. For k-dependence topology, Algorithm 3 will perform random sampling and select at most k parents for attribute X_i. Corresponding directed edges will be added to $\mathcal{G}$. If X_i is independent from all the attributes in $\mathcal{G}$ then its parents will be selected randomly.

Algorithm 3 SelectParents($\mathcal{G}$, $\mathcal{Q}_I$, X_i, k).

Input: The network topology $\mathcal{G}$, independent attribute set $\mathcal{Q}_I$, children attribute X_i and k.
Output: The parent set Π_i of attribute X_i.

Let $\mathcal{Q}$ be a list of all the attributes in $\mathcal{G}$ and $\Pi_i = \emptyset$;
Calculate $I(X_i; X_j|Y)(X_j \in \mathcal{Q})$ for each pair of attributes;
if $X_i \notin \mathcal{Q}_I$ **then**
    for $c = 1$ *to* $min(|\mathcal{Q}|, k)$ **do**
        $\triangle I \leftarrow$ normalize $I(X_i; X_j|Y)$ //Normalize $I(X_i; X_j|Y)$ into a probability distribution that sums to 1;
        $X_p \sim \triangle I$// Perform random sampling from a multinomial distribution;
        Let $\Pi_i = \Pi_i \cup X_p$;
    end
else
    for $c = 1$ *to* $min(|\mathcal{Q}|, k)$ **do**
        $X_p \sim \mathcal{Q}$ // Randomly select attribute from $\mathcal{Q}$;
        Let $\Pi_i = \Pi_i \cup X_p$;
    end
end
return the parent set Π_i.

3.2.4 RBF learning algorithm

The network topology $\mathcal{G}$ is represented in the form of tree and contains three parts: root node, branch node (including parent attribute or children attribute) and directed edge between them. The proposed algorithm, called random Bayesian classifier (RBC), applies random sampling to select them at different learning phases. Algorithm 4 describes the learning procedure of RBC.

Algorithm 4 RBC learning algorithm.

Input: Training dataset $\mathcal{D}$ with attribute set $\mathbf{X} = \{X_1, \cdots, X_n, Y\}$ and k.
Output: RBC model $\mathcal{G} = (\mathcal{V}, \mathcal{T})$, where $\mathcal{V}$ represents node set and $\mathcal{T}$ represents directed edge set.

$\mathcal{V} = \{Y\}, \mathcal{T} = \emptyset$;
$X_r \leftarrow$ **SelectRoot**($\mathcal{D}$); // See Algorithm 1
$\mathcal{V} = \mathcal{V} \cup \{X_r\}, \mathcal{T} = \mathcal{T} \cup \{Y \rightarrow X_r\}$;
repeat
    $X_i, \mathcal{Q}_I \leftarrow$ **SelectChildren**($\mathcal{G}$, $\mathbf{X}$); // See Algorithm 2
    $\Pi_i \leftarrow$ **SelectParents**($\mathcal{G}$, $\mathcal{Q}_I$, X_i, k); // See Algorithm 3
    $\mathcal{V} = \mathcal{V} \cup \{X_i\}, \mathcal{T} = \mathcal{T} \cup \{Y \rightarrow X_i, \Pi_i \rightarrow X_i\}$;
until RBC *includes all attributes*;
return RBC.

Introducing randomness to the learning procedure of BNC will build an "unstable" topology, and that helps avoid overfitting and reduce variance. On the other hand, the training data is just a sample from the complete dataset, that may result in potentially lost information implicated. One single BNC cannot encode by coincidence the most significant dependency relationships in its topology, that would result in suboptimal classification performance.

Ensemble of classifiers performs better than its committee members on average. Different attribute orders and augmented edges will form different BNCs. It also shows that the classification performance of different BNCs varies and, in some cases, varies greatly. Randomness helps to independently learn RBCs which describe the true Bayesian network from different aspects. The wrong prediction from BNC A may be corrected by BNC B. In this paper, we adopt RBC as the base algorithm and then the different classifiers are obtained by applying stochastic optimization. After that they vote for the most possible class label. Algorithm 5 gives the general learning framework for the ensemble of RBC.

Introducing randomness to the learning procedure of BNC will build an "unstable" topology, and that helps avoid overfitting and reduce variance. On the other hand, the training data is just a sample from the complete dataset, that may result in potentially lost information implicated. One single BNC cannot encode by coincidence the most significant dependency relationships in its topology, that would result in suboptimal classification performance.

Ensemble of classifiers performs better than its committee members on average. Different attribute orders and

augmented edges will form different BNCs. It also shows that the classification performance of different BNCs varies and, in some cases, varies greatly. Randomness helps to independently learn RBCs which describe the true Bayesian network from different aspects. The wrong prediction from BNC A may be corrected by BNC B. In this paper, we adopt RBC as the base algorithm and then the different classifiers are obtained by applying stochastic optimization. After that they vote for the most possible class label. Algorithm 5 gives the general learning framework for the ensemble of RBC.

Algorithm 5 RBF learning algorithm.

Input: Training set $\mathcal{D}$; the number of RBCs p; parameter k.
Output: An RBF model.

```
RBF = ∅.
for m = 1 to p do
    RBC_m=RBC(D, X and k); //See Algorithm 4
    Compute the conditional probability table (CPT) according
    to the network topology of RBC_m learned from D.
    RBF = RBF ∪ RBC_m.
end
return RBF.
```

3.3 Classification process and complexity analysis

For subclassifier RBC_m, given the testing instance $\mathbf{x}$, RBC_m assigns the MAP class to $\mathbf{x}$ by

$$y_m^* = \arg\max_{y \in Y} P(y) \prod_{i=1}^{n} P(x_i | \pi_i^{\text{RBC}_m}, y) \tag{4}$$

After training a set of learners, ensemble learning combines multiple learning models' predictions appropriately. Thus, in practice, the class-membership prediction produced by RBF are simply voted as follows:

$$y^* = \arg\max\{F(y_1), F(y_2), \cdots, F(y_q)\} \tag{5}$$

where y^* is the predicted class label, $F(y_i)$ is the number of models whose prediction results are y_i in all models and q is the number of class labels.

During the training phase, the time complexity of forming the frequency table from which the probability estimates by RBF is $\mathcal{O}(tn^2)$, where t and n respectively denote the number of training instances and the number of attributes. Calculating $I(X_i; Y)$ and $I(X_i; X_j|Y)$ respectively need $\mathcal{O}(cnv)$ and $\mathcal{O}(c(nv)^2)$ time, where v is the maximum number of values of discrete attributes and c is the number of classes. The procedure of randomly selecting the root attribute needs $\mathcal{O}(n)$ time. The time complexity of randomly selecting attributes and conditional dependencies is $\mathcal{O}(n(n^2 + kn))$, where k is user-defined. Finally, the computational complexity of RBF is $\mathcal{O}(tn^2 + cnv + c(nv)^2 + p(n + n(n^2 + kn)))$, where p is the number of RBCs. During the testing phase, classifying a test instance using RBF only needs $\mathcal{O}(pnck)$ time.

4 Experimental results

We perform experiments on 40 benchmark datasets from the UCI repository of machine learning [38] and summarize the characteristics of datasets in Table 1, including the name of dataset, the number of instances, attributes, and classes. The results of RBF using MDL (Minimum Description Length)

Table 1 Datasets

No.	Dataset	Instance	Attribute	Class
1	labor	57	16	2
2	labor-negotiations	57	16	2
3	zoo	101	16	7
4	echocardiogram	131	6	2
5	lymphography	148	18	4
6	hepatitis	155	19	2
7	wine	178	13	3
8	autos	205	25	7
9	sonar	208	60	2
10	new-thyroid	215	5	3
11	soybean-large	307	35	19
12	ionosphere	351	34	2
13	dermatology	366	34	6
14	house-votes-84	435	16	2
15	cylinder-bands	540	39	2
16	chess	551	39	2
17	syncon	600	60	6
18	soybean	683	35	19
19	crx	690	15	2
20	breast-cancer-w	699	9	2
21	anneal	898	38	6
22	tic-tac-toe	958	9	2
23	vowel	990	13	11
24	german	1000	20	2
25	car	1728	6	4
26	segment	2310	19	7
27	kr-vs-kp	3196	36	2
28	dis	3772	29	2
29	hypo	3772	29	4
30	sick	3772	29	2
31	phoneme	5438	7	50
32	satellite	6435	36	6
33	thyroid	9169	29	20
34	Electrical-Grid	10000	13	2
35	nursery	12960	8	5
36	magic	19020	10	2
37	adult	48842	14	2
38	shuttle	58000	9	7
39	connect-4	67557	42	3
40	localization	164860	5	11

Table 2 A summary table of the statistics employed

Statistics employed	Description
Zero-one loss	Zero-one loss [44] is one of the most commonly used metrics to evaluate the classification performance. Supposing y and $\hat{y}$ are the true class label and that generated by a learning algorithm, respectively, given M unlabeled test instances, the zero-one loss function is defined as $\xi(y,\hat{y}) = \frac{\sum_{i=1}^{M} 1-\varrho(y_i,\hat{y_i})}{M}$, (6) where $\varrho(y_i,\hat{y_i}) = 1$ if $y_i = \hat{y_i}$ and 0 otherwise.
RMSE	RMSE [45] measures how well calibrated the probability estimates are. The RMSE is defined as follows, $RMSE = \sqrt{\frac{1}{s}\sum_{i=1}^{s}(1-P(\hat{y}\|\mathbf{x}))^2}$, (7) where s is the sum of training instances.
Bias and variance	The bias-variance decomposition provides a different perspective on the error of learned classifiers [46]. The bias of a classifier can measure the difference between systematic predictions and true response, and the variance of a classifier can measure the variability or randomness of its predictions. $bias^2 = \frac{1}{2}\sum_{\hat{y},y\epsilon Y}[P(\hat{y}\|\mathbf{x}) - P(y\|\mathbf{x})]^2$, (8) and $variance = \frac{1}{2}[1-\sum_{\hat{y}\epsilon Y} P(\hat{y}\|\mathbf{x})^2]$, (9)
Friedman and Bonferroni-Dunn test	The Friedman test [47] is a non-parametric test and explores the statistical significance of multiple algorithms over multiple data sets. It computes as follows: $\mathcal{F}_F = \frac{(D-1)\mathcal{X}_F^2}{D(t-1)-\mathcal{X}_F^2}$ (10) where $\mathcal{X}_F^2 = \frac{12D}{t(t+1)}\sum_{i=1}^{t} R_i^2 - 3D(t+1)$ (11) where t, D and R_i respectively represent the number of algorithms, the number of datasets and the average rank of the i-th algorithm. The null hypothesis of the Friedman test will be rejected if there exists significant difference among algorithms, then the Bonferroni-Dunn test will be performed to further analyze the difference by comparing critical difference(CD). We assess the difference between the algorithms to be significant if the corresponding average ranks greater than the CD [48]. The value of CD can be computed as follows: $\mathrm{CD} = q_\alpha\sqrt{\frac{t(t+1)}{6D}}$, (12) where q_α are the critical values that are calculated by dividing the values in the row for the infinite degree of freedom of the table of Studentized range statistics ($\alpha = 0.05$) by $\sqrt{2}$.

discretization [39] to preprocess numerical attributes. The missing values in the datasets are processed into a distinct value in all cases, m-estimation ($m = 1$) [40] is used for base probability estimation. Each algorithm is processed with 10 rounds of 10-fold cross validation. The following algorithms are introduced for comparison with our proposed RBF:

· SKDB [23], selective KDB with $k = 5$.

· WATAN [27], weighted averaged TAN.

· WAODE [25], weighted AODE.

· SA2DE [41], selective A2DE which uses both MI and CMI to directly rank the attributes.

· SASA2DE [42], sample-based attribute selection A2DE whose sample size is 50k.

· IWAODE [43], instance-based weighting AODE.

The primary loss functions are zero-one loss, RMSE, bias and variance, and the detailed results in Tables 8, 9, 10 and 11 are shown in the Appendix. We use the Win/Draw/Loss (WDL) records where each cell's W/D/L in the table indicates that one classifier is better than another on W datasets, equal on D datasets and worse on L datasets to show the classification performance. A summary table of the statistics employed is shown in Table 2.

To illustrate how to determine the number of committee members of RBF, in Fig. 3 we present learning curves for all the datasets (which are described in Table 1). As can be seen, lower bias delivered by large number of committee members and lower variance by random sampling result in lower error for RBF, while as the number increases to some extent the zero-one loss doesn't change significantly. When RBF selects 30 classifiers as an ensemble it can substantially reduce error across all the datasets, thus we take 30 as the default number of members for RBF.

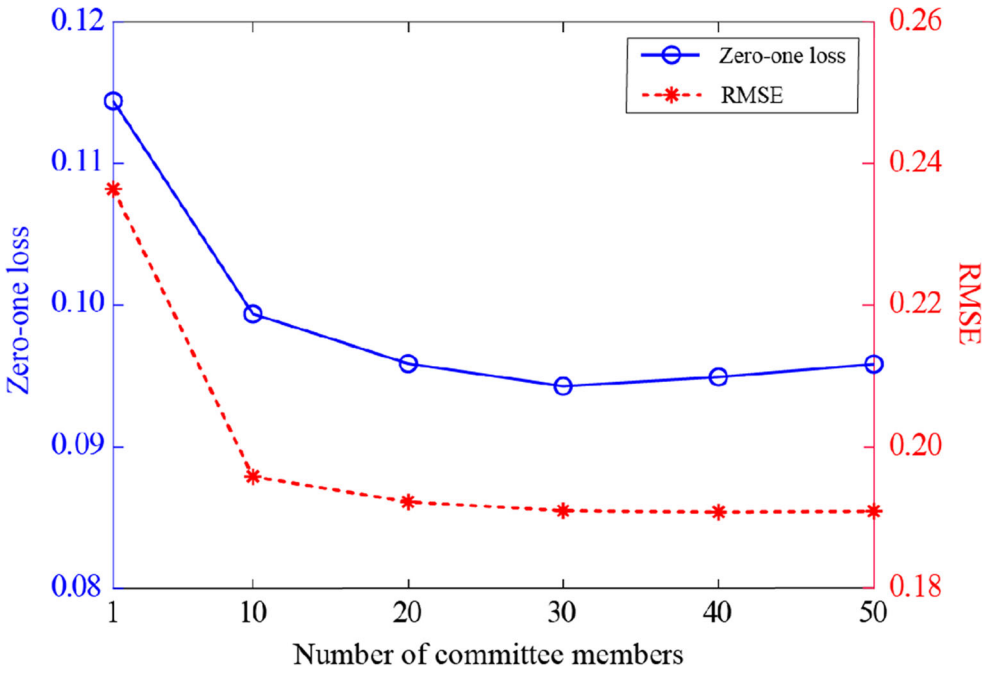


Fig. 3 The changes in RMSE and zero-one loss as the number of committee members of the ensemble increases

4.1 Diversity of the classifier generated by RBF

To use ensembling well, we need good performing learners with lower correlation. Dietterich [49] has measures of dispersion of an ensemble and notes that more accurate ensembles have larger dispersion. Breiman's [37] results indicate that the generalization error of random forests depends on the strength of the individual trees in the forest and the correlation between them.

In order to investigate the diversity of classifiers generated by RBF, statisticians have developed several measures of agreement (or disagreement) between classiers. The most widely used measure is the Kappa statistic (K statistic) [49]. We show the diversity of the classifiers on some datasets by k-error diagrams, which help visualize the accuracy and diversity of the individual classifiers constructed by the RBF classifier.

Given two classifiers g_1 and g_2. Suppose there are W classes, and let H be a $W \times W$ square array such that H_{ij} contains the number of test examples assigned to class i by g_1 and into class j by g_2. The K statistic is defined as:

$$k = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2}, \tag{13}$$

where Θ_1 be an estimate of the probability that the two classifiers agree, and Θ_2 be an estimate of the probability that the two classifiers agree by chance. They are respectively defined as follows:

$$\Theta_1 = \frac{\sum_{i=1}^{W} H_{ii}}{m}, \tag{14}$$

$$\Theta_2 = \sum_{i=1}^{W}(\sum_{j=1}^{W} \frac{H_{ij}}{m} \cdot \sum_{j=1}^{W} \frac{H_{ji}}{m}), \tag{15}$$

where m is the total number of test examples. $k = 0$ when the agreement of the two classifiers equals that expected by chance, and $k = 1$ when the two classifiers agree on every example. Negative values occur when agreement is weaker expected by chance—that is, there is systematic disagreement between the classifiers.

We choose 12 datasets (`hepatitis`, `sonar`, `chess`, `car`, `kr-vs-kp`, `dis`, `hypo`, `nursery`, `magic`, `adult`, `connect-4` and `localization`) from Table 1. We run RBF on every dataset and obtain 30 classifiers. For each pair of classifiers, we compute their K statistic value according to equation (13). We then construct a scatter plot in which each point corresponds to a pair of classifiers. Its x coordinate is the diversity value (k) and its y coordinate is the mean accuracy of the classifiers. Figure 4 shows the mean accuracy and K statistic value between every two classifiers. The classifiers generated by RBF have larger diversity on some datasets (e.g. `chess`, `kr-vs-kp`, `dis` and `localization`), but have smaller diversity on some other datasets (e.g. `hypo`, `nursery` and `connect-4`). In some datasets (e.g. `hepatitis` and `sonar`), some quite agreement classifiers are generated. However, there exist the diversity in the majority of classifiers.

4.2 Comparison in terms of zero-one loss and RMSE

Tables 3 and 4 show WDL records summarizing the relative zero-one loss and RMSE of the different algorithms. The

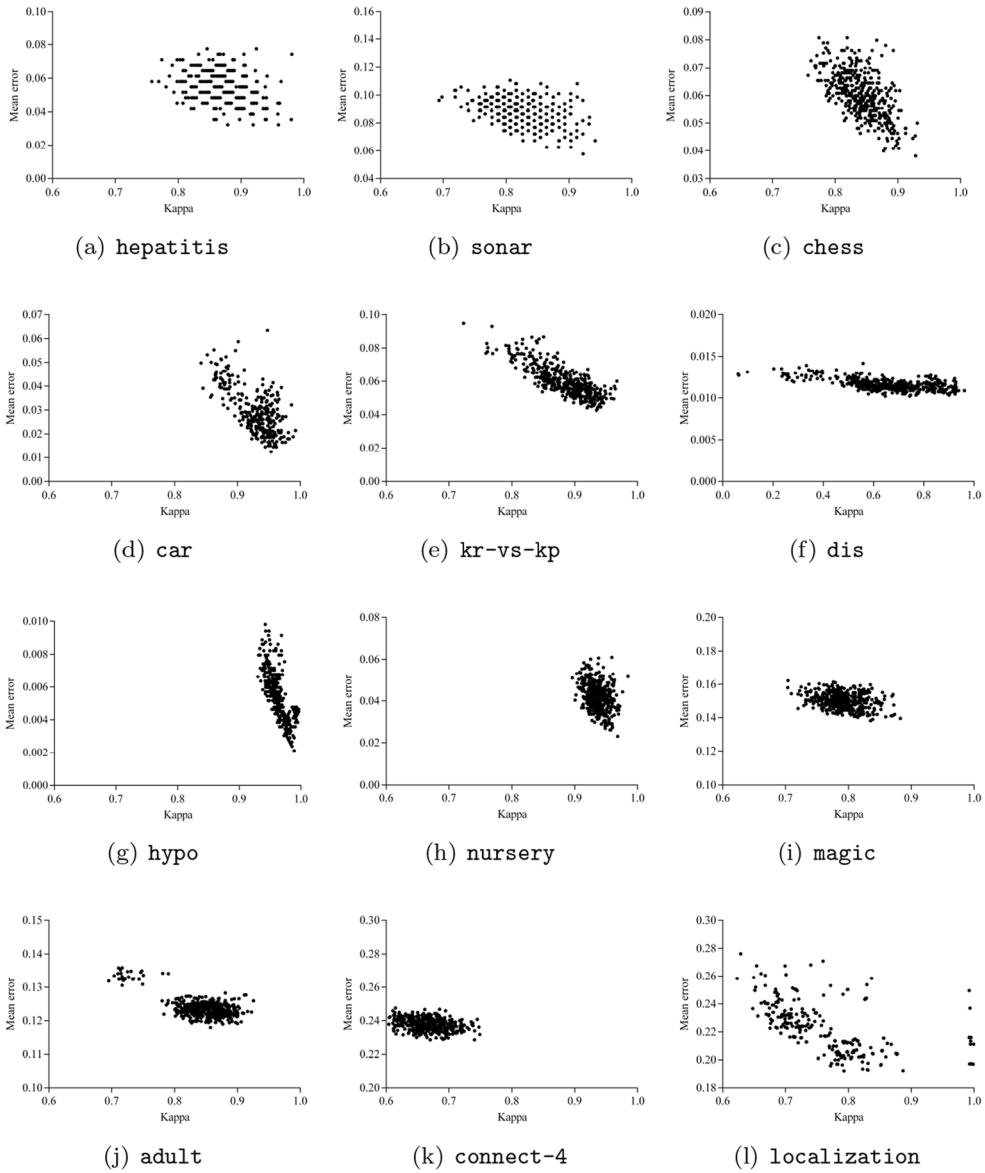


Fig. 4 the k-error diagrams of the classifiers generated by RBF on twelve datasets

Table 3 Win/Draw/Loss records of zero-one loss on all, small and large datasets

	SKDB			WATAN		
	All	Small	Large	All	Small	Large
RBF	31/5/4	23/2/0	8/3/4	31/7/2	18/5/2	13/2/0
p	**<0.0001**	**<0.0001**	0.3877	**<0.0001**	**0.0004**	**0.0002**
	WAODE			SA2DE		
	All	Small	Large	All	Small	Large
RBF	24/10/6	14/6/5	10/4/1	28/8/4	22/1/2	6/7/2
p	**0.0014**	0.0636	**0.0117**	**<0.0001**	**<0.0001**	0.2891
	SASA2DE			IWAODE		
	All	Small	Large	All	Small	Large
RBF	20/15/5	18/4/3	2/11/2	26/9/5	15/7/3	11/2/2
p	**0.0041**	**0.0015**	1.3750	**0.0002**	**0.0075**	**0.0225**

Table 4 Win/Draw/Loss of RMSE on all, small and large datasets

	SKDB			WATAN		
	All	Small	Large	All	Small	Large
RBF	25/11/4	19/5/1	6/6/3	24/12/4	14/9/2	10/3/2
p	**0.0001**	**<0.0001**	0.5078	**0.0002**	**0.0042**	**0.0386**
	WAODE			SA2DE		
	All	Small	Large	All	Small	Large
RBF	15/21/4	10/11/4	5/10/0	25/11/4	18/5/2	7/6/2
p	**0.0192**	0.1796	0.0625	**0.0001**	**0.0004**	0.1797
	SASA2DE			IWAODE		
	All	Small	Large	All	Small	Large
RBF	15/20/5	14/8/3	1/12/2	18/17/5	10/10/5	8/7/0
p	**0.0414**	**0.0127**	1.0000	**0.0106**	0.3018	**0.0078**

p value following each WDL record is the outcome of a two-tailed binomial sign test and represents the probability that RBF would obtain the observed or more extreme ratio of wins to losses. We assess a difference is significant if $p \leq 0.05$, and all such *p* values are changed to boldface corresponding tables.

As can be seen from Tables 3 and 4, RBF achieves the advantage in terms of zero-one loss and RMSE over all the other BNCs, and the difference between them is statistically significant on all datasets. Generally, variants of AODE, e.g., WAODE, SA2DE, SASA2DE and IWAODE, assume different independence assumptions for different SPODE members and the complementary characteristic help fully represent all possible conditional dependencies. In contrast, SKDB and WATAN apply conditional mutual information to identify dependency relationships, which may be information-theoretic rather than probability-theoretic significant. The experimental results show that our heuristic search and random sample strategies provide high accuracy. For example, RBF beats SKDB on 31 datasets whereas it beats IWAODE on 26 datasets in terms of zero-one loss. RMSE-wise, RBF beats SKDB on 25 datasets whereas it beats IWAODE on 18 datasets.

The complexity of problem domains makes ever more urgent the need for scaling-up of existing learning algorithms to deal with datasets of different sizes. To clarify the effectiveness of heuristic search strategy and random sampling, we categorize datasets in terms of their sizes. For example, datasets with less than 2,000 instances (25 datasets) and more than 2,000 instances (15 datasets) are denoted as small size and large size respectively. On smaller datasets RBF has a better zero-one loss performance and RMSE than other BNCs, and most achieved a significant advantage. On larger datasets RBF has significantly better zero-one loss than WATAN (13 wins and 0 loss), WAODE (10 wins and 1 loss), IWAODE (11 wins and 2 losses). RBF

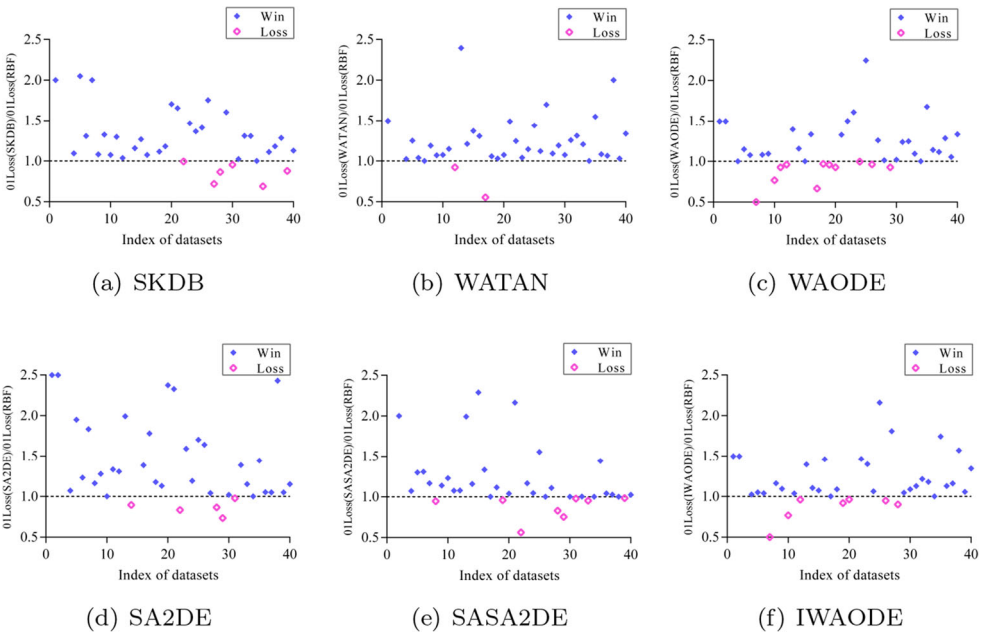


Fig. 5 Scatter plot of comparisons in terms of zero-one loss

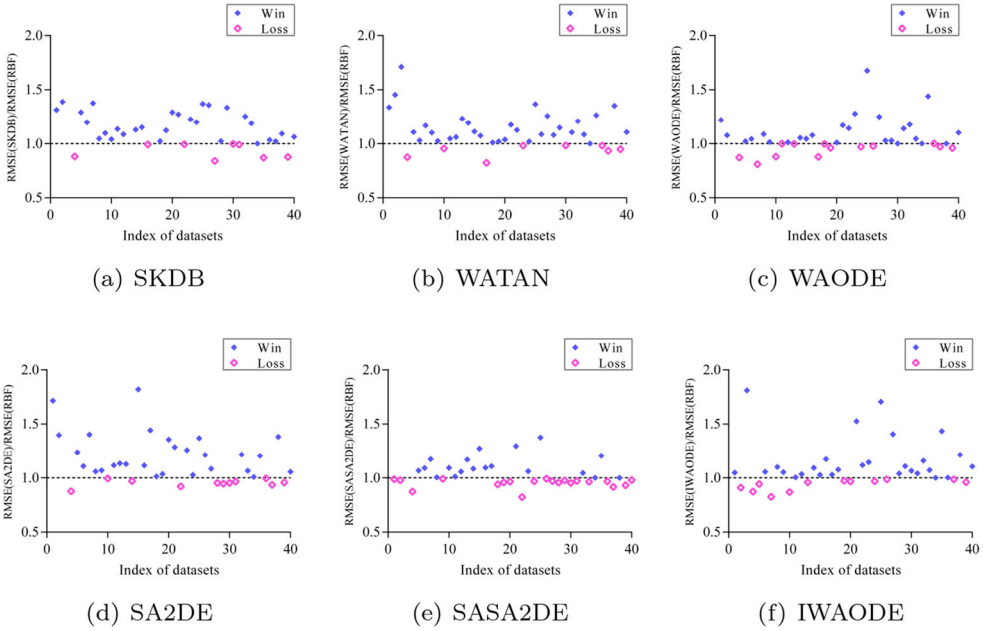


Fig. 6 Scatter plot of comparisons in terms of RMSE

is comparable to SKDB, SA2DE and SASA2DE in terms of zero-one loss and RMSE on large datasets. We claim that RBF's improved performance on small size datasets is very encouraging.

To further illustrate that heuristic search strategy and random sampling are powerful methods to improve the performance of ensemble models. Figures 5 and 6 respectively present the scatter plots of the comparison results of RBF against other BNCs in terms of zero-one loss and RMSE, and the dotted line means that RBF performs almost the same as the alternative BNC. Note that some outlier points are removed for significance analysis. We can observe that most data points are located above the dotted line, that means RBF performs better than other BNCs much more often, and the advantages are obvious and significant.

4.3 Comparison in terms of bias-variance decomposition

Tables 5 and 6 respectively show the W/D/L comparison results using bias and variance. In general, lower bias means that model can capture fine detail in the training data. But this low bias may has potentially higher variance, which leads to greater changes in the topology learned from sample to sample. SKDB and WATAN try to fully represent the most significant conditional dependencies and build more robust topologies, that often result low bias and high variance. In contrast, variants of AODE inherit the tradeoff between bias and variance due to its impractical independence assumptions for different SPODE members and ensemble learning strategy. Significant and insignificant conditional dependencies are indiscriminately represented in the SPODE members. RBF also reduces bias by applying ensemble learning strategy, whereas it reduces variance by applying random sampling to randomly select conditional dependencies from all possible significant ones. Compared to SKDB, RBF achieves the advantage in terms of bias although less often (19 wins and 11 losses). Variance-wise, high-dependence BNCs may have high variance, leading to overfitting training data. Thus RBF obtains lower variance significantly more often than SKDB (32 wins and 4 losses). Compared with WATAN, RBF has a more significant advantage in bias (20 wins and 8 losses) and variance (31 wins and 6 losses). RBF achieves lower bias and variance more often than variants of AODE, except for IWAODE in variance. The reason might be that IWAODE applies weighting approach to improve the estimates of conditional probabilities while retaining the basic topologies of all SPODEs, and the weaker independence assumptions help avoid overfitting.

From Tables 5 and 6, when dealing with small datatsets, RBF is comparable to other BNCs in terms of bias. RBF has significantly better variance performance than SKDB (20 wins and 2 losses), WATAN (21 wins and 3 losses) and SA2DE (18 wins and 4 losses). IWAODE is a low variance high bias learner, it should be suitable for small data. This can be seen in Table 6 where IWAODE has significantly better variance than RBF (18 wins and 4 losses) on small datasets. When dealing with large datasets, most of the results are not significant, except for WATAN (9 wins and 1 loss) in terms of bias and SKDB (12 wins and 2 losses) and SA2DE (11 wins and 2 losses) in terms of variance.

Table 5 Win/Draw/Loss comparison results of bias on all, small and large datasets

	SKDB			WATAN		
	All	Small	Large	All	Small	Large
RBF	19/10/11	15/4/6	4/6/5	20/12/8	11/7/7	9/5/1
p	0.2005	0.0784	1.0000	**0.0357**	0.4807	**0.0215**
	WAODE			SA2DE		
	All	Small	Large	All	Small	Large
RBF	14/9/17	7/5/13	7/4/4	17/7/16	13/2/10	4/5/6
p	0.7201	0.2632	0.5488	1.0000	0.6776	0.7539
	SASA2DE			IWAODE		
	All	Small	Large	All	Small	Large
RBF	11/9/20	8/6/11	3/3/9	17/13/10	8/9/8	9/4/2
p	0.1496	0.6476	0.1460	0.2478	1.1964	0.0654

4.4 Difference among all the classifiers

The average ranks of the algorithms obtained by applying the Friedman test with respect to zero-one loss, RMSE, bias and variance are shown in Table 7. The Friedman statistic F_F is distributed according to the F distribution with $t-1 = 6$ and $(t-1)(D-1) = 234$ degrees of freedom. The critical value of $F(6, 234)$ for $\alpha = 0.05$ is 2.14. At the bottom of Table 7, we could see that the F_F statistics for zero-one loss, RMSE, bias and variance are 50.5600, 40.1800, 19.3100 and 73.7900 respectively. Therefore, we can reject the null hypothesis, indicating that there are significant differences among those 7 algorithms.

In order to further explore the significant difference among algorithms, we perform the Bonferroni-Dunn test and show the comparison results in terms of zero-one loss, RMSE, bias and variance in Fig. 7, where the middle line corresponds to the average level of different algorithms. For $\alpha = 0.05$ with 7 algorithms and 40 datasets, q_α is 2.638 and the value of CD is 1.274. The CD interval is marked to the left and right of the average rank of RBF.

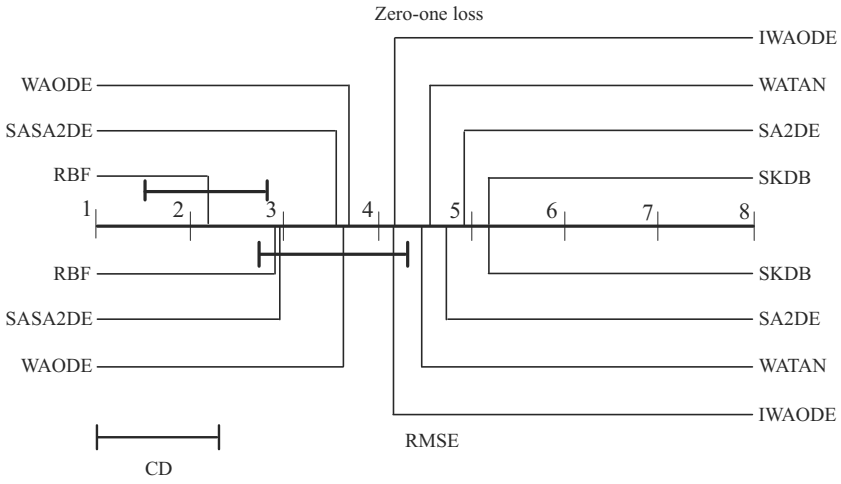
As can be seen from the Fig. 7, RBF enjoys a significant advantage over other algorithms in terms of zero-one loss. RBF ranks first in terms of RMSE whereas it doesn't have a significantly higher score than SASA2DE, WAODE and IWAODE. With respect to bias, the performance of RBF is comparable to other BNCs. With respect to variance, the performance of RBF is comparable to SASA2DE, WAODE and IWAODE, significantly better than WATAN, SA2DE and SKDB.

5 Conclusions

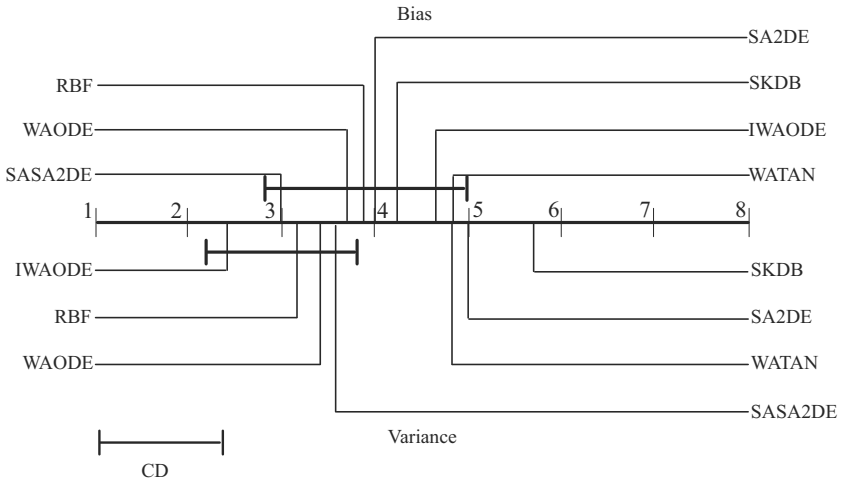
Ensemble learning can lead to performance improvement for "unstable" learning algorithms. State-of-the-art approaches, e.g., Bagging and Boosting, use resampling from the training set to produce very different models. To mitigate the negative effect caused by biased estimate of probability distributions, we propose to apply heuristic search strategy and random sampling to randomly select strong dependency relationships. In return for this extra random sampling during training, the proposed algorithm, RBF, provides well-calibrated posterior class probability estimates and always improves classification accuracy by reducing the structure complexity. RBF reduces bias by applying ensemble learning strategy and reduces variance by applying random sampling, thus it achieves the tradeoff

Table 6 Win/Draw/Loss comparison results of variance on all, small and large datasets

	SKDB			WATAN		
	All	Small	Large	All	Small	Large
RBF	32/4/4	20/3/2	12/1/2	31/3/6	21/1/3	10/2/3
p	**<0.0001**	**0.0001**	**0.0129**	**<0.0001**	**0.0003**	0.0923
	WAODE			SA2DE		
	All	Small	Large	All	Small	Large
RBF	18/5/17	9/5/11	9/2/4	29/5/6	18/3/4	11/2/2
p	0.7283	0.8238	0.2668	**0.0001**	**0.0043**	**0.0225**
	SASA2DE			IWAODE		
	All	Small	Large	All	Small	Large
RBF	19/11/10	13/6/6	6/5/4	10/6/24	4/3/18	6/3/6
p	0.1360	0.1671	0.7539	**0.0243**	**0.0043**	1.2256



(a) zero-one loss and RMSE



(b) bias and variance

Fig. 7 The comparison results of the Bonferroni-Dunn test in terms of zero-one, RMSE, bias and variance on 40 datasets. CD = 1.274

between bias and variance. As shown in the experimental results, RBF attains lower error than the other out-of-core BNCs considered.

RBF provides an efficient and effective solution to potentially large problem space in learning Bayesian network classifier (BNC). This solution allows it to capture and take advantage of the additional fine-detail that is inherent in very large data while its efficiency makes it feasible to deploy. We take the dataset `localization`[2] from the UCI repository of machine learning as an example. Dataset `localization` contains recordings of five people performing different activities and each person wore four sensors (tags) while performing the same scenario five times. Dataset `localization` has 164,860 instances, 5 attributes (Sequence Name, Tag identificator, x coordinate of the tag, y coordinate of the tag and z coordinate of the tag) and 11 class labels (walking,falling, 'lying down', lying, 'sitting down', sitting, 'standing up from lying', 'on all fours', 'sitting on the ground', 'standing up from sitting' and 'standing up from sitting on the ground'). As shown in Fig. 4, the classifiers generated by RBF have significant diversity on dataset `localization`. RBF significantly reduces misclassification rate (0.2672) compared to SKDB (0.3013), WATAN (0.3575), WAODE (0.3566), SA2DE (0.3078), SASA2DE (0.2735) and IWAODE (0.3593). Future directions for research include:

Table 7 Average ranks of the algorithms

Algorithm	zero-one loss	RMSE	bias	variance
RBF	2.1875	2.9000	3.7625	3.1500
SASA2DE	3.5500	2.9500	2.9750	3.5625
WAODE	3.6875	3.6250	3.6875	3.4000
IWAODE	4.1750	4.1625	4.6375	2.4000
WATAN	4.5500	4.4625	4.7250	4.8125
SA2DE	4.9125	4.7250	3.9875	4.9875
SKDB	4.9375	5.1750	4.2250	5.6875
F_F statistic	50.5600	40.1800	19.3100	73.7900

[2] https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity

1. Weighting approaches to combining the predictions from committee members of the ensemble in a more reasonable and efficient way;

2. More appropriate loss functions to tackle RBF's tendency to overfit or underfit training sets;

3. Customized selection of edges and numbers of parents for different attributes in a discriminative manner.

Appendix

Table 8 Experimental results of zero-one loss

Dataset	SKDB	WATAN	WAODE	SA2DE	SASA2DE	IWAODE	RBF
labor	0.0702	0.0526	0.0526	0.0877	**0.0175**	0.0526	0.0351
labor-negotiations	0.1053	0.1053	0.0526	0.0877	0.0702	0.0526	**0.0351**
zoo	0.0396	0.0198	0.0297	0.0297	0.0297	0.0198	**0.0000**
echocardiogram	0.3511	0.3282	**0.3206**	0.3435	0.3435	0.3282	**0.3206**
lymphography	0.2770	0.1689	0.1554	0.2635	0.1757	0.1419	**0.1351**
hepatitis	0.2194	0.1742	0.1806	0.2065	0.2194	0.1742	**0.1677**
wine	0.0674	0.0337	**0.0169**	0.0618	0.0393	**0.0169**	0.0337
autos	0.1951	0.2146	0.1951	0.2098	**0.1707**	0.2098	0.1805
sonar	0.2740	0.2212	0.2260	0.2644	0.2356	0.2260	**0.2067**
new-thyroid	0.0651	0.0651	**0.0465**	0.0605	0.0744	**0.0465**	0.0605
soybean-large	0.1140	0.1010	**0.0814**	0.1173	0.0945	0.0912	0.0879
ionosphere	0.0769	**0.0684**	0.0712	0.0969	0.0798	0.0712	0.0741
dermatology	0.0792	0.0328	0.0191	0.0273	0.0273	0.0191	**0.0137**
house-votes-84	0.0506	0.0529	0.0506	**0.0391**	0.0506	0.0483	0.0437
cylinder-bands	0.2278	0.2463	**0.1796**	0.5704	0.4111	0.1926	**0.1796**
chess	0.0762	0.0926	0.0944	0.0980	0.0944	0.1034	**0.0708**
syncon	0.0567	**0.0083**	0.0100	0.0267	0.0150	0.0150	0.0150
soybean	0.0556	0.0527	**0.0483**	0.0586	0.0556	0.0542	0.0498
crx	0.1696	0.1478	0.1377	0.1623	0.1377	**0.1319**	0.1435
breast-cancer-w	0.0658	0.0415	**0.0358**	0.0916	0.0401	0.0372	0.0386
anneal	0.0111	0.0100	0.0089	0.0156	0.0145	0.0178	**0.0067**
tic-tac-toe	0.1806	0.2265	0.2724	0.1514	**0.1023**	0.2662	0.1816
vowel	0.1778	0.1263	0.1949	0.1929	0.1414	0.1697	**0.1212**
german	0.3290	0.2760	**0.2400**	0.2870	0.2520	0.2560	0.2410
car	0.0556	0.0567	0.0885	0.0671	0.0613	0.0851	**0.0394**
segment	0.0615	0.0394	0.0338	0.0576	0.0351	**0.0333**	0.0351
kr-vs-kp	**0.0329**	0.0776	0.0576	0.0476	0.0507	0.0826	0.0457
dis	0.0122	0.0154	0.0143	0.0122	**0.0117**	0.0127	0.0141
hypo	0.0175	0.0130	0.0101	**0.0080**	0.0082	0.0114	0.0109
sick	**0.0228**	0.0257	0.0244	0.0244	0.0239	0.0260	0.0239
phoneme	0.1909	0.2345	0.2308	**0.1822**	0.1824	0.2104	0.1865
satellite	0.1206	0.1207	0.1148	0.1276	0.0922	0.1117	**0.0920**
thyroid	0.0784	0.0723	0.0655	0.0690	**0.0570**	0.0706	0.0599
Electrical-Grid	**0.0002**	**0.0002**	**0.0002**	**0.0002**	**0.0002**	**0.0002**	**0.0002**
nursery	**0.0291**	0.0654	0.0708	0.0609	0.0609	0.0735	0.0422
magic	0.1718	0.1674	0.1762	0.1625	0.1609	0.1744	**0.1545**
adult	0.1532	0.1380	0.1445	0.1360	0.1331	0.1502	**0.1296**
shuttle	0.0009	0.0014	0.0009	0.0017	**0.0007**	0.0011	**0.0007**
connect-4	**0.2007**	0.2354	0.2406	0.2397	0.2245	0.2409	0.2283
localization	0.3013	0.3575	0.3566	0.3078	0.2735	0.3593	**0.2672**

The value in boldface indicates the classifier with the best performance

Table 9 Experimental results of RMSE

Dataset	SKDB	WATAN	WAODE	SA2DE	SASA2DE	IWAODE	RBF
labor	0.2067	0.2105	0.1920	0.2700	**0.1555**	0.1649	0.1575
labor-negotiations	0.2655	0.2778	0.2057	0.2672	0.1871	**0.1739**	0.1913
zoo	0.0884	0.0614	0.0724	0.0746	0.0841	0.0650	**0.0359**
echocardiogram	0.4928	0.4890	**0.4872**	0.4901	0.4889	0.4881	0.5598
lymphography	0.3156	0.2705	0.2496	0.3027	0.2607	**0.2304**	0.2446
hepatitis	0.4243	0.3645	0.3695	0.3926	0.3863	0.3743	**0.3545**
wine	0.1668	0.1416	**0.0983**	0.1701	0.1423	0.1001	0.1214
autos	0.2203	0.2320	0.2290	0.2230	0.2117	0.2317	**0.2106**
sonar	0.4426	0.4130	0.4091	0.4308	**0.3999**	0.4246	0.4036
new-thyroid	0.1899	0.1740	0.1605	0.1815	0.1995	**0.1584**	0.1826
soybean-large	0.0978	0.0902	**0.0860**	0.0962	0.0871	0.0866	0.0862
ionosphere	0.2674	0.2613	0.2489	0.2791	0.2604	0.2546	**0.2463**
dermatology	0.1433	0.0850	0.0688	0.0778	0.0806	**0.0661**	0.0690
house-votes-84	0.2066	0.2181	0.1927	**0.1773**	0.1984	0.1998	0.1830
cylinder-bands	0.4427	0.4277	0.4016	0.7005	0.4898	0.3952	**0.3848**
chess	**0.2399**	0.2594	0.2603	0.2692	0.2643	0.2835	0.2417
syncon	0.1326	**0.0503**	0.0537	0.0882	0.0678	0.0629	0.0612
soybean	0.0662	0.0654	0.0646	0.0656	**0.0609**	0.0697	0.0648
crx	0.3760	0.3415	0.3219	0.3469	**0.3207**	0.3259	0.3351
breast-cancer-w	0.2373	0.1904	0.1855	0.2491	**0.1772**	0.1776	0.1838
anneal	0.0582	0.0538	0.0536	0.0589	0.0593	0.0699	**0.0458**
tic-tac-toe	0.3551	0.4023	0.4085	0.3288	**0.2937**	0.3992	0.3575
vowel	0.1567	**0.1254**	0.1633	0.1606	0.1355	0.1463	0.1278
german	0.5146	0.4373	0.4161	0.4409	0.4158	**0.4157**	0.4294
car	0.1621	0.1617	0.1983	0.1621	0.1627	0.2019	**0.1184**
segment	0.1210	0.0968	**0.0870**	0.1081	0.0885	0.0879	0.0892
kr-vs-kp	**0.1573**	0.2358	0.2343	0.2034	0.1819	0.2635	0.1876
dis	0.1041	0.1098	0.1046	**0.0969**	0.0972	0.1058	0.1018
hypo	0.0840	0.0723	0.0647	**0.0596**	0.0615	0.0698	0.0630
sick	0.1447	0.1426	0.1452	0.1383	**0.1382**	0.1547	0.1452
phoneme	0.0756	0.0844	0.0871	**0.0737**	0.0742	0.0795	0.0764
satellite	0.1917	0.1849	0.1800	0.1862	0.1598	0.1774	**0.1531**
thyroid	0.0813	0.0742	0.0715	0.0729	**0.0660**	0.0734	0.0685
Electrical-Grid	**0.0141**	**0.0141**	**0.0141**	0.0142	**0.0141**	**0.0141**	**0.0141**
nursery	**0.0953**	0.1385	0.1577	0.1321	0.1321	0.1572	0.1096
magic	0.3646	0.3461	0.3526	0.3507	**0.3409**	0.3534	0.3527
adult	0.3361	0.3076	0.3197	0.3079	**0.3018**	0.3250	0.3296
shuttle	0.0143	0.0177	**0.0131**	0.0181	**0.0131**	0.0159	**0.0131**
connect-4	**0.3062**	0.3315	0.3356	0.3349	0.3257	0.3359	0.3499
localization	0.2010	0.2095	0.2087	0.2000	**0.1854**	0.2093	0.1894

The value in boldface indicates the classifier with the best performance

Table 10 Experimental results of bias

Dataset	SKDB	WATAN	WAODE	SA2DE	SASA2DE	IWAODE	RBF
labor	0.0316	**0.0142**	0.0200	0.0668	0.0342	0.0205	0.0205
labor-negotiations	0.0584	0.0653	**0.0268**	0.0874	0.0584	**0.0268**	0.0368
zoo	0.0585	**0.0270**	0.0273	0.0339	0.0342	0.0282	0.0288
echocardiogram	0.3002	0.2658	**0.2572**	0.3005	0.3033	0.2840	0.2714
lymphography	0.1310	0.0978	0.0951	0.2833	**0.0820**	0.0857	0.1014
hepatitis	0.1727	0.1684	0.1655	**0.1555**	0.1590	0.1749	0.1737
wine	0.0569	0.0531	0.0381	**0.0315**	0.0378	0.0317	0.0417
autos	0.2265	0.2269	0.2115	0.2590	**0.1740**	0.2034	0.1960
sonar	0.1675	0.1646	0.1722	0.1659	**0.1588**	0.1694	0.1604
new-thyroid	0.0356	0.0332	**0.0263**	0.0396	0.0375	0.0304	0.0358
soybean-large	0.1137	0.1151	**0.0655**	0.1019	0.0797	0.0811	0.1131
ionosphere	0.0940	0.0823	0.0751	0.0816	**0.0738**	0.0881	0.0890
dermatology	0.0693	0.0263	**0.0061**	0.0087	0.0167	0.0065	0.0278
house-votes-84	0.0304	0.0393	0.0406	**0.0273**	0.0276	0.0493	0.0406
cylinder-bands	0.1942	0.2193	**0.1501**	0.5514	0.4352	0.1711	0.1689
chess	0.1229	0.1398	0.1286	**0.1074**	0.1190	0.1397	0.1325
syncon	0.0553	0.0202	**0.0180**	0.0275	0.0299	0.0336	0.0314
soybean	0.0617	0.0521	**0.0503**	0.0551	0.0529	0.0693	0.0689
crx	0.1234	0.1148	0.0953	0.1257	0.1202	**0.0904**	0.1090
breast-cancer-w	0.0302	0.0349	0.0327	0.0685	0.0238	**0.0234**	0.0248
anneal	**0.0067**	0.0194	0.0194	0.0126	0.0167	0.0181	0.0072
tic-tac-toe	0.1266	0.1742	0.2104	0.1003	**0.0832**	0.1994	0.1401
vowel	**0.1556**	0.1842	0.1811	0.1773	0.1732	0.2249	0.1719
german	0.2187	0.2046	**0.2036**	0.2164	0.2081	0.2112	0.2041
car	0.0494	0.0478	0.0633	0.0426	0.0401	0.0599	**0.0317**
segment	0.0518	0.0489	**0.0357**	0.0372	0.0408	0.0436	0.0434
kr-vs-kp	**0.0283**	0.0700	0.0518	0.0426	0.0422	0.0763	0.0459
dis	0.0176	0.0194	0.0179	0.0179	0.0170	**0.0168**	0.0193
hypo	0.0089	0.0119	0.0078	0.0063	**0.0059**	0.0080	0.0084
sick	0.0202	0.0206	0.0216	0.0200	**0.0194**	0.0220	0.0238
phoneme	0.1585	0.1982	0.2172	0.1571	**0.1323**	0.1829	0.1546
satellite	0.0850	0.0945	0.0902	0.0901	**0.0767**	0.0884	0.0823
thyroid	0.0533	0.0584	0.0561	0.0499	**0.0488**	0.0648	0.0569
Electrical-Grid	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
nursery	0.0397	0.0522	0.0616	0.0525	0.0466	0.0658	**0.0345**
magic	**0.1244**	0.1252	0.1541	0.1260	0.1303	0.1595	0.1267
adult	**0.1193**	0.1312	0.1387	0.1233	0.1238	0.1437	0.1234
shuttle	0.0009	0.0009	**0.0006**	0.0008	0.0007	0.0007	**0.0006**
connect-4	**0.1636**	0.2253	0.2237	0.2114	0.2040	0.2255	0.2153
localization	**0.2004**	0.3105	0.3068	0.2232	0.2134	0.3126	0.2014

The value in boldface indicates the classifier with the best performance

Table 11 Experimental results of variance

Dataset	SKDB	WATAN	WAODE	SA2DE	SASA2DE	IWAODE	RBF
labor	0.0842	0.0542	**0.0221**	0.1068	0.0447	0.0268	0.0268
labor-negotiations	0.1258	0.1347	**0.0626**	0.1021	0.1047	**0.0626**	0.0789
zoo	0.0627	0.0548	**0.0424**	0.0539	0.0597	0.0445	0.0470
echocardiogram	0.1393	**0.1226**	0.1335	0.1274	0.1340	0.1277	0.1402
lymphography	0.1547	0.1084	0.0478	0.2045	0.0894	**0.0408**	0.0680
hepatitis	0.0606	0.0571	0.0541	0.0739	0.0625	0.0486	**0.0459**
wine	0.0702	0.0486	0.0246	0.0464	0.0317	**0.0141**	0.0193
autos	0.1897	0.1687	0.1503	0.2057	0.1819	**0.1363**	0.1466
sonar	0.1252	0.1165	0.1003	0.1051	0.1107	**0.0929**	0.1106
new-thyroid	0.0362	**0.0203**	0.0244	0.0351	0.0358	**0.0203**	0.0248
soybean-large	0.0814	0.1084	0.0855	0.1070	0.0791	**0.0738**	0.0800
ionosphere	0.0684	0.0399	0.0368	0.0491	0.0338	**0.0238**	0.0298
dermatology	0.0766	0.0483	0.0242	0.0307	0.0357	**0.0189**	0.0386
house-votes-84	0.0144	0.0172	0.0083	0.0168	0.0138	0.0079	**0.0070**
cylinder-bands	0.0753	0.0762	0.1010	**0.0014**	0.0026	0.0828	0.0867
chess	0.0427	0.0504	**0.0364**	0.0510	0.0444	0.0379	0.0380
syncon	0.0452	0.0217	0.0230	0.0315	0.0176	**0.0164**	0.0186
soybean	0.0344	0.0589	0.0334	0.0409	0.0313	**0.0290**	0.0372
crx	0.0709	0.0500	0.0264	0.0426	**0.0207**	0.0240	0.0445
breast-cancer-w	0.0449	0.0385	0.0128	0.0388	0.0273	**0.0122**	0.0233
anneal	0.0194	0.0158	0.0161	0.0201	0.0171	**0.0103**	0.0142
tic-tac-toe	0.1608	0.0819	0.0604	0.0997	0.0588	**0.0529**	0.0674
vowel	**0.2177**	0.2361	0.2310	0.2376	0.2232	0.2463	0.2287
german	0.1285	0.1017	0.0765	0.0890	0.0820	**0.0692**	0.0818
car	0.0403	0.0374	0.0427	**0.0329**	0.0340	0.0430	0.0355
segment	0.0486	0.0290	0.0255	0.0339	0.0235	**0.0204**	0.0244
kr-vs-kp	0.0130	0.0152	0.0119	0.0134	**0.0098**	0.0185	0.0101
dis	0.0020	**0.0004**	0.0021	0.0019	0.0028	0.0036	0.0007
hypo	0.0072	0.0063	0.0056	0.0049	**0.0047**	0.0068	0.0052
sick	0.0041	0.0048	0.0057	0.0047	**0.0037**	**0.0037**	0.0047
phoneme	**0.0769**	0.1541	0.1311	0.0870	0.0940	0.1270	0.1084
satellite	0.0449	0.0368	0.0364	0.0592	0.0317	0.0325	**0.0216**
thyroid	0.0352	0.0253	0.0239	0.0300	0.0219	**0.0202**	0.0203
Electrical-Grid	**0.0001**	**0.0001**	**0.0001**	**0.0001**	**0.0001**	**0.0001**	**0.0001**
nursery	0.0345	0.0167	0.0111	0.0167	0.0169	**0.0104**	0.0150
magic	0.0532	0.0490	**0.0289**	0.0488	0.0429	0.0291	0.0418
adult	0.0427	0.0165	0.0113	0.0188	0.0151	**0.0109**	0.0137
shuttle	0.0004	0.0004	0.0004	0.0004	**0.0003**	**0.0003**	**0.0003**
connect-4	0.0489	**0.0149**	0.0215	0.0341	0.0279	0.0209	0.0174
localization	0.1299	0.0594	0.0632	0.1122	0.0832	**0.0577**	0.0884

The value in boldface indicates the classifier with the best performance

Acknowledgements This work is supported by the National Key Research and Development Program of China (No.2019YFC1804804), Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (No.KLIGIP-2021A04), and the Scientific and Technological Developing Scheme of Jilin Province (No.20200201281JC).

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2-3):131–163
2. Jiang LX, Zhang LG, Li CQ, Wu J (2019) A Correlation-Based feature weighting filter for naive bayes. IEEE Trans Knowl Data Eng 31(2):201–213
3. Chickering DM, Heckerman D, Meek C (2004) Large-sample learning of Bayesian networks is NP-hard. J Mach Learn Res 5:1287–1330
4. Jiang LX, Zhang LG, Yu LJ, Wang DH (2019) Class-specific attribute weighted naive Bayes. Pattern Recogn 88:321–330
5. Wang LM, Zhang S, Mammadov M, Li K, Zhang XH (2021) Semi-supervised weighting for averaged one-dependence estimators. Applied Intelligence
6. Liu Y, Wang LM, Mammadov M, Chen SL, Wang GJ, Qi SK, Sun MH (2021) Hierarchical Independence Thresholding for learning Bayesian network classifiers. Knowl-Based Syst, p 212
7. Liu Y, Wang LM, Mammadov M (2020) Learning semi-lazy Bayesian network classifier under the c.i.i.d assumption. Knowl-Based Syst, p 208
8. Jiang LX, Li CQ, Wang SS, Zhang LG (2016) Deep feature weighting for naive Bayes and its application to text classification. Eng Appl Artif Intell 52:26–39
9. Jiang LX, Zhang H, Cai ZH (2009) A novel bayes model: hidden naive bayes. IEEE Trans Knowl Data Eng 21(10):1361–1371
10. Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. IEEE Trans Inf Theory 14(3):462–467
11. Sahami M (1996) Learning limited dependence bayesian classifiers. Knowledge Discovery in Databases 96(1):335–338
12. Wang LM, Zhang XH, Li K, Zhang S (2021) Semi-supervised learning for k-dependence Bayesian classifiers. Applied Intelligence
13. Wang LM, Chen SL, Mammadov M (2018) Target Learning: A Novel Framework to Mine Significant Dependencies for Unlabeled Data. In: Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 106–117
14. Herr HD, Krzysztofowicz R (2019) Ensemble Bayesian forecasting system Part II: Experiments and properties. J Hydrol 575:1328–1344
15. Aridas CK, Kotsiantis SB, Vrahatis MN (2016) Increasing Diversity in Random Forests Using Naive Bayes. In: Proceedings of the 12th International Conference on Artificial Intelligence Applications and Innovations, pp 75–86
16. Ho TK (1995) Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, pp 278–282
17. Wang LM, Chen P, Chen SL, Sun MH (2021) A novel approach to fully representing the diversity in conditional dependencies for learning Bayesian network classifier. Intelligent Data Analysis 25(1):35–55
18. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13rd International Conference on Machine Learning, pp 148–156
19. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
20. Adeva JJG, Beresi UC, Calvo RA (2005) Accuracy and diversity in ensembles of text categorisers. CLEI Electronic Journal 9(1):1–12
21. Zhang H, Petitjean F, Buntine W (2020) Bayesian network classifiers using ensembles and smoothing. Knowl Inf Syst 62(9):3457–3480
22. Wang LM, Wang GJ, Duan ZY, Lou H, Sun MH (2019) Optimizing the Topology of Bayesian Network Classifiers by Applying Conditional Entropy to Mine Causal Relationships Between Attributes. IEEE Access 7:134271–134279
23. Martinez AM, Webb GI, Chen SL, Zaidi NA (2016) Scalable learning of bayesian network classifiers. J Mach Learn Res 17(1):1515–1549
24. Webb GI, Boughton JR, Wang ZH (2005) Not so naive bayes: aggregating one-dependence estimators. Mach Learn 58(1):5–24
25. Jiang LX, Zhang H, Cai ZH, Wang DH (2012) Weighted average of one-dependence estimators. Journal of Experimental & Theoretical Artificial Intelligence 24(2):219–230
26. Kong H, Shi XH, Wang LM, Liu Y, Mammadov M (2021) Averaged tree-augmented one-dependence estimators. Appl Intell 51(7):4270–4286
27. Jiang LX, Cai ZH, Wang DH, Zhang H (2012) Improving Tree augmented Naive Bayes for class probability estimation. Knowl-Based Syst 26:239–245
28. Hellman S, McGovern A, Xue M (2012) Learning ensembles of Continuous Bayesian Networks: An application to rainfall prediction. In: Proceedings of 2012 Conference on Intelligent Data Understanding, pp 112–117
29. Geiger D, Heckerman D (1996) Knowledge representation and inference in similarity networks and Bayesian multinets. Artif Intell 82(1-2):45–74
30. Davison AC, Hinkley DV, Young GA (2003) Recent developments in bootstrap methodology. Stat Sci 18(2):141–157
31. Bryll R, Gutierrez-Osuna R, Quek F (2003) Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recogn 36(6):1291–1302
32. Jing YS, Pavlovic V, Rehg JM (2008) Boosted Bayesian network classifiers. Mach Learn 73(2):155–184
33. Ratsch G, Onoda T, Muller KR (2001) Soft margins for AdaBoost. Mach Learn 42(3):287–320
34. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
35. Kunwar R, Pal U, Blumenstein M (2014) Semi-Supervised Online Bayesian Network Learner for Handwritten Characters Recognition. In: Proceedings of the 22nd International Conference on Pattern Recognition, pp 3104–3109
36. Ma SC, Shi HB (2004) Tree-augmented Naive Bayes ensembles. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, pp 1497–1502
37. Breiman L (2001) Random forests. Machine Learning 45(1):5–32

38. Murphy PM, Aha DW (1994) UCI Repository of Machine Learning Databases, Available online: http://www.ics.uci.edu/mlearn/MLRepository.html
39. Kumar V, Heikkonen J, Rissanen J, Kaski K (2006) Minimum description length denoising with histogram models. IEEE Transactions on Signal Processing 54(8):2922–2928
40. Cestnik B (1990) Estimating probabilities: A crucial task in machine learning. In: Proceedings of the 9th European Conference on Artificial Intelligence, pp 147–149
41. Chen SL, Martinez AM, Webb GI, Wang LM (2017) Selective anDE for large data learning: a low-bias memory constrained approach. Knowl Inf Syst 50(2):475–503
42. Chen SL, Martinez AM, Webb GI, Wang LM (2017) Sample-based Attribute Selective anDE for Large Data. IEEE Trans Knowl Data Eng 29(1):172–185
43. Duan ZY, Wang LM, Chen SL, Sun MH (2020) Instance-based weighting filter for superparent one-dependence estimators. Knowl-Based Syst 203:106085
44. Kohavi R, Wolpert DH (1996) Bias plus variance decomposition for zero-one loss functions. In: Proceedings of the 13th International Conference on Machine Learning, pp 275–283
45. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688
46. Salles T, Rocha L, Goncalves M (2020) A bias-variance analysis of state-of-the-art random forest text classifiers. ADAC 15(2):379–405
47. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
48. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30
49. Dietterich TG (2000) An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Mach Learn 40(2):139–157

Yi Ren received the B.Sc. degree from Changchun University of Technology, China, in 2020 and he is currently a postgraduate student in the College of Software, Jilin University, China. His research interests include Bayesian network and data analysis.

LiMin Wang received the Ph.D. degree in computer science from Jilin University, China, in 2005. He is currently a professor in the College of Computer Science and Technology, Jilin University, China. He has authored or co-authored more than 60 academic articles in reputed peer-reviewed international journals and conferences. His research interests include machine learning, data mining, decision making and Bayesian network. He has supervised many M.Sc. and Ph.D. students in the above-mentioned fields. He has also been involved with reviewing and organizing different workshops, seminars, and training sessions on different technologies.

XiongFei Li (Member, IEEE) received the B.S. degree in computer software from Nanjing University, in 1985, the M.S. degree in computer software from the Chinese Academy of Sciences, in 1988, and the Ph.D. degree in communication and information system from Jilin University, in 2002. Since 1988, he has been a member of the faculty of the Computer Science and Technology, Jilin University, Changchun, China. He is currently a Professor of computer software and theory with Jilin University. He has authored more than 60 research articles. His research interests include data mining, intelligent network, image processing, and analysis.

Meng Pang received the B.Sc. degree from Shanxi University, China, in 2020 and he is currently a postgraduate student in the College of Computer Science and Technology, Jilin University, China. His research interests include Bayesian network and data analysis.

JunYang Wei received the B.Sc. degree from North China Electric Power University, China, in 2019 and she is currently a postgraduate student in the College of Software, Jilin University, China. Her research interests include machine learning, big data mining, and Bayesian networks.