SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2025, Vol. 68, Iss. 3, 132106:1–132106:25
https://doi.org/10.1007/s11432-022-3728-2

Attribute grouping-based naive Bayesian classifier

Yulin HE[1,2], Guiliang OU[2],
Philippe FOURNIER-VIGER[2] & Joshua Zhexue HUANG[1,2*]

[1]*Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China*
[2]*College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China*

Received 20 June 2022/Revised 31 October 2022/Accepted 25 February 2023/Published online 11 February 2025

Abstract The naive Bayesian classifier (NBC) is a supervised machine learning algorithm having a simple model structure and good theoretical interpretability. However, the generalization performance of NBC is limited to a large extent by the assumption of attribute independence. To address this issue, this paper proposes a novel attribute grouping-based NBC (AG-NBC), which is a variant of the classical NBC trained with different attribute groups. AG-NBC first applies a novel effective objective function to automatically identify optimal dependent attribute groups (DAGs). Condition attributes in the same DAG are strongly dependent on the class attribute, whereas attributes in different DAGs are independent of one another. Then, for each DAG, a random vector functional link network with a SoftMax layer is trained to output posterior probabilities in the form of joint probability density estimation. The NBC is trained using the grouping attributes that correspond to the original condition attributes. Extensive experiments were conducted to validate the rationality, feasibility, and effectiveness of AG-NBC. Our findings showed that the attribute groups chosen for NBC can accurately represent attribute dependencies and reduce overlaps between different posterior probability densities. In addition, the comparative results with NBC, flexible NBC (FNBC), tree augmented Bayes network (TAN), gain ratio-based attribute weighted naive Bayes (GRAWNB), averaged one-dependence estimators (AODE), weighted AODE (WAODE), independent component analysis-based NBC (ICA-NBC), hidden naive Bayesian (HNB) classifier, and correlation-based feature weighting filter for naive Bayes (CFW) show that AG-NBC obtains statistically better testing accuracies, higher area under the receiver operating characteristic curves (AUCs), and fewer probability mean square errors (PMSEs) than other Bayesian classifiers. The experimental results demonstrate that AG-NBC is a valid and efficient approach for alleviating the attribute independence assumption when building NBCs.

Keywords naive Bayesian classifier, attribute independence assumption, attribute grouping, dependent attribute group, posterior probability, class-conditional probability

Citation He Y L, Ou G L, Fournier-Viger P, et al. Attribute grouping-based naive Bayesian classifier. Sci China Inf Sci, 2025, 68(3): 132106, https://doi.org/10.1007/s11432-022-3728-2

1 Introduction

The naive Bayesian classifier (NBC) [1] is a classical classification algorithm and is among the top 10 algorithms in the fields of data mining and machine learning [2]. The NBC [3] is based on the principle of minimum error probability or maximum posterior probability and uses the Bayes' formula to assess the probability that a sample belongs to a given class, i.e., the posterior probability. The NBC is widely used because it involves simple implementation, of its simple implementation process, easy-to-calculate probabilities, and low training complexity. In addition, it can be applied for binary and multi-class classification. Prior research has shown that the NBC can deliver excellent performance in various domains, such as document classification [4], image classification [5], and intrusion detection [6]. In the last decade, the growing need for processing big data has further stimulated the research interest in improving the application performance of NBC as it is fast and easy to deploy in distributed environments [7–9].

A key factor influencing the performance of NBCs is the validity of attribute independence assumption. The attribute independence assumption states that for any given data set, any pair of condition attributes are independent of the decision attribute. This assumption is crucial in NBC because it simplifies the calculation of posterior probabilities, i.e., the likelihood that a class is observed given the attribute values of a sample.

The NBC computes an approximation as the product of marginal probabilities based on the attribute independence assumption rather than calculating a posterior probability as a joint probability. This

* Corresponding author (email: zx.huang@szu.edu.cn)

approach simplifies the probability calculations and enables the good estimations of probabilities even with small training sets. However, the prediction performance of NBC can be substantially impaired if the independence assumption is violated. Hence, several studies on NBC have focused on how to relax the independence assumption to further enhance the generalization performance of NBC. Improvements have been made mainly in two areas: reducing attribute dependence by modifying the structure of NBC (i.e., network structure-based dependence) and using a joint probability density function (p.d.f.) estimation to represent attribute dependence. Some representative studies are summarized as follows.

• **Network structure-based dependence.** Cooper and Herskovits [10] designed Bayesian belief network (BBN) which is represented with a directed acyclic graph (DAG). In this DAG, the nodes represent attributes and arcs between nodes indicate attribute dependencies. Friedman et al. [11] proposed the tree augmented naive Bayes (TAN), which introduced the mechanism of dashed lines-based attribute independences and solid lines-based attribute dependences in a BBN. Keogh and Pazzani [12] designed the super parent-based TAN (SP-TAN), which allows each node has at most one non-class parent node. Webb et al. [13] proposed an averaged one-dependence estimator (AODE) by averaging one-dependence classifiers. AODE is an ensemble classifier that uses attribute selection to construct a series of one-dependent classifiers to obtain a simpler Bayesian network structure. A weighted AODE (WAODE) [14] was developed to further improve AODE by assigning different weights to the one-dependence classifiers of AODE. Pernkopf et al. [15] presented a maximum margin Bayesian network (MM-BN) classifier, which integrates the idea of maximum margin into structure learning for Bayesian networks. Cvejić [16] gave an improved lower bound for Bayesian network learning. Jiang et al. [17] developed a hidden naive Bayes (HNB) classifier, which utilizes the weighted sum of two-attribute dependencies to represent multiple-attribute dependence, where the weights are assigned based on the mutual information between two condition attributes. In addition, the studies have recently proposed two excellent weighted NBC models, namely correlation-based feature weighting filter for naive Bayes (CFW) [18] and class-specific attribute weighted naive Bayes [19] by exploring dependencies between condition attributes and the class attribute in a discriminative and adaptive way.

• **Joint p.d.f. estimation-based dependence.** Joint p.d.f. estimation-based dependence representations directly use multivariate p.d.f. estimation techniques to model an unknown conditional probability in Bayesian classifiers when compared to network structure-based dependence representation. They do not represent attribute dependence with an indirect and complex network structure. Pérez et al. [20] proposed a kernel-based Bayesian network (KBN), which is an improved flexible NBC (FNBC) [21] based on the multivariate kernel density estimation technique. Based on the joint p.d.f. estimation, Wang et al. [22] and He et al. [23] designed a non-naive Bayesian classifier (NNBC) with an optimal bandwidth parameter selection strategy. Unlike KBN, the NNBC minimizes the mean integrated squared error between the true joint p.d.f. and the estimated joint p.d.f. so as to determine the optimal bandwidths. Geng et al. [24] designed a model-free Bayesian classifier that does not build a network structure rather than using the nearest neighbor strategy to obtain information about the sample space and joint p.d.f.

Researchers have also developed many other NBC improvements, such as linear programming-based Bayesian network [25], global Bayesian network [26], semi-lazy Bayesian network classifier [27], k-dependence Bayesian classifiers [28, 29], decreasingly NBC [30], multinomial NBC [31], and weighted NBC with accurate ranking [32].

The aforementioned Bayesian methods improve NBC classification performance to some extent, but the resulting models are often complex and these methods have a high time complexity. In addition, these methods do not perform a deep exploration of training data as they were designed to improve the model. An analysis of these approaches reveals that they retain the independence assumption of NBC and its importance in obtaining good prediction performance. Some scholars have tried exploring the intrinsic characteristics of training data so as to satisfy the attribute independence assumption at the data level rather than at the model level. Feature extraction is commonly used to transform data with attribute dependence into data with reduced attribute dependence. Bressan and Vitria [33] used class-conditional independent component analysis (CC-ICA) [34] to improve the classification performance of NBC. Qin et al. [35] improved the classification performance of NBC by utilizing ICA. Fan and Poh [36] assessed the influence of three feature extraction methods, i.e., principal component analysis (PCA) [37], ICA [38], and CC-ICA, on the prediction performances of NBC. They reported that PCA, ICA, and CC-ICA can slightly increase the testing accuracies of NBC on the selected data sets. Jayanthi and Sasikala [39] performed web link spam detection by training the NBC based on website attributes extracted with PCA. Zhang et al. [40] used PCA to extract key attributes in network data and then trained the NBC

to conduct network intrusion detection.

According to the aforementioned literatures, current NBC improvements primarily focus on model-driven and data-driven methods. Model-driven methods attempt to model attribute dependence by weakening dependent attributes or strengthening independent attributes, whereas data-driven methods attempt to transform the original attribute set into a new one with independent attributes. However, neither of these two methods can adequately compensate for the gap between "model universality" and "data applicability". In other words, NBC construction and data processing are two separate processes that do not consider attribute dependence and independence simultaneously. In the present paper, we intend to train the NBC model such that posterior probability can be calculated adaptively and attribute dependence and independence are individually treated based on a sophisticated exploration of intrinsic characteristics of a given data set. Accordingly, we designed a novel attribute grouping-based NBC (AG-NBC) method by considering both a functional method and structural data. AG-NBC applies an attribute grouping strategy to form a number of dependent attribute groups (DAGs), which are mutually independent as much as possible. It does not represent attribute dependence with a complex model structure and does not use feature extraction to obtain attribute independence. The main contributions of this paper are threefold:

(1) An effective objective function was designed to automatically identify optimal DAGs. Condition attributes in the same DAG have a strong dependence with respect to the class attribute, while attributes from different DAGs are independent of each other.

(2) The joint probability of attributes in a specific DAG was modeled using a random vector functional link (RVFL) network with fast training speed and good capability to represent attribute dependence. A simple NBC was constructed by integrating RVFL networks trained on multiple DAGs, which can be regarded as reduced attributes corresponding to the original condition attributes.

(3) Extensive experiments were conducted by comparing AG-NBC with nine state-of-the-art Bayesian classifiers (i.e., NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, and CFW) and RVFL network. The experimental results on accuracy, area under the receiver operating characteristic curve (AUC), and probability mean square errors (PMSE) show that AG-NBC has better prediction capability, lower classification risk, and higher probability estimation quality compared to other studied classifiers.

The remainder of this paper is organized as follows. Section 2 describes the basic principles of FNBC, AODE, and HNB. Section 3 introduces the main concepts of the proposed AG-NBC method. Section 4 reports experimental results and provides statistical analysis. Finally, a conclusion is drawn in Section 5, and research directions for future work are discussed.

2 Preliminaries

Let there be a classification data set $\mathbb{D}$ that contains $\mathcal{N}$ samples, where each sample has $\mathcal{D}$ condition attributes. All samples are divided into $\mathcal{M}$ different classes and there are $\mathcal{N}_m$ samples that belong to the m-th ($m = 1, 2, \ldots, \mathcal{M}$) class. The data set that consists only of samples from the m-th class is represented as

$$\mathbb{D}^{(m)} = \left\{ \left(x_n^{(m)}, y_n^{(m)} \right) \,\middle|\, x_n^{(m)} = \left(x_{n1}^{(m)}, x_{n2}^{(m)}, \ldots, x_{n\mathcal{D}}^{(m)} \right), y_n^{(m)} = w_m, n = 1, 2, \ldots, \mathcal{N}_m \right\},$$

where $\mathbb{D} = \bigcup_{m=1}^{\mathcal{M}} \mathbb{D}^{(m)}$, $\sum_{m=1}^{\mathcal{M}} \mathcal{N}_m = \mathcal{N}$, $x_{nd}^{(m)} \in \Re$, $d = 1, 2, \ldots, \mathcal{D}$ is the attribute value of the d-th condition attribute A_d and the class label set is $\{w_1, w_2, \ldots, w_{\mathcal{M}}\}$. Based on the data set $\mathbb{D}$, the principles of the flexible naive Bayesian classifier (FNBC) [21], AODE [13] and the HNB classifier [17] are presented.

2.1 FNBC

Assume that there is a new sample $x = (x_1, x_2, \ldots, x_{\mathcal{D}})$. NBC determines its class label by the following equation:

$$y = \underset{m=1,2,\ldots,\mathcal{M}}{\arg\max}\, P(w_m \,|x\,) = \underset{m=1,2,\ldots,\mathcal{M}}{\arg\max}\, \frac{P(x\,|w_m\,)\,P(w_m)}{P(x)} \propto \underset{m=1,2,\ldots,\mathcal{M}}{\arg\max}\, P(x\,|w_m\,)\,P(w_m), \tag{1}$$

where $P(w_m|x)$ is the posterior probability, $P(w_m)$ is the prior probability, $P(x|w_m)$ is the class-conditional probability, and $P(x)$ is the total probability. Generally, the prior probability in (1) can be calculated as

$$P(w_m) = \frac{\mathcal{N}_m}{\mathcal{N}}. \tag{2}$$

The key of training an NBC is to calculate the conditional probability based on the attribute independence assumption as

$$P(x|w_m) = P(x_1, x_2, \ldots, x_{\mathcal{D}}|w_m) = \prod_{d=1}^{\mathcal{D}} P(x_d|w_m). \tag{3}$$

John and Langley [21] constructed an FNBC which uses the kernel density estimation technique [41] to calculate the term of $P(x_d|w_m)$ in (3) as

$$P(x_d|w_m) \propto p(x_d|w_m) = \frac{1}{\mathcal{N}_m} \sum_{n=1}^{\mathcal{N}_m} \frac{1}{\sqrt{2\pi}h} \exp\left[-\frac{1}{2}\left(\frac{x_d - x_{nd}^{(m)}}{h}\right)^2\right], \tag{4}$$

where $p(x_d|w_m)$ is the estimated probability density function (p.d.f.) value of x_d based on the d-th attribute values $x_{1d}^{(m)}$, $x_{2d}^{(m)}$, $\ldots$, $x_{\mathcal{N}_m,d}^{(m)}$ corresponding to samples from the m-th class and $h > 0$ ($h = \frac{1}{\sqrt{\mathcal{N}_m}}$ in [21]) is the bandwidth parameter. The attribute independence assumption limits the prediction performance of FNBC to a certain degree. Consequently, some improved strategies have been proposed to relax this strong assumption, e.g., the AODE and the HNB classifier.

2.2 AODE

The AODE approximates the posterior probability $P(w_m|x)$ by constructing multiple ODE as

$$P(w_m|x) = \frac{P(w_m, \mathrm{x})}{P(x)} \propto P(w_m, \mathrm{x}) = P(w_m, x_d) P(x|w_m, x_d) \tag{5}$$

for any condition attribute[1] x_d , $d \in \{1, 2, \ldots, \mathcal{D}\}$, where the term of $P(x|w_m, x_d)$ can be calculated as

$$P(x|w_m, x_d) = \prod_{i=1}^{\mathcal{D}} P(x_i|w_m, x_d). \tag{6}$$

Combining (5) and (6) and dividing by the number of inference attributes, AODE's inference model is obtained by

$$\begin{aligned} P(w_m|x) &= \frac{\sum_{\substack{d=1 \\ \sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{nd}^{(m)} = x_d] \geqslant \delta}}^{\mathcal{D}} [P(w_m, x_d) \prod_{i=1}^{\mathcal{D}} P(x_i|w_m, x_d)]}{|\{d|d \in \{1, 2, \ldots, \mathcal{D}\} \wedge \sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{nd}^{(m)} = x_d] \geqslant \xi\}|} \\ &= \frac{\sum_{\substack{d=1 \\ \sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{nd}^{(m)} = x_d] \geqslant \delta}}^{\mathcal{D}} [P(w_m, x_d) \prod_{i=1}^{\mathcal{D}} \frac{P(x_i, x_d, w_m)}{P(w_m, x_d)}]}{|\{d|d \in \{1, 2, \ldots, \mathcal{D}\} \wedge \sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{nd}^{(m)} = x_d] \geqslant \xi\}|}, \end{aligned} \tag{7}$$

where $\mathbb{I}(\cdot)$ is an indicator function which takes 1 if $\cdot$ is true and 0 otherwise, the threshold ξ is set to 30 in [13], $P(w_m, x_d)$ and $P(x_i, x_d, w_m)$ can be respectively estimated with Laplace correction as

$$P(w_m, x_d) = \frac{\sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{nd}^{(m)} = x_d] + 1}{\mathcal{N}_m + \mathcal{M} \times n_d} \tag{8}$$

and

$$P(x_i, x_d, w_m) = \frac{\sum_{n=1}^{\mathcal{N}_m} \mathbb{I}[x_{ni}^{(m)} = x_i, x_{nd}^{(m)} = x_d] + 1}{\mathcal{N}_m + \mathcal{M} \times n_i \times n_d}, \tag{9}$$

where n_i, $i = 1, 2, \ldots, \mathcal{D}$ is the number of different discrete values in the i-th condition attribute of the training data set.

1) Here, x_d represents a discrete attribute. Continuous-attribute values should be firstly discretized to obtain discrete-attribute values when using AODE and HNB to perform classification.

2.3 HNB classifier

HNB assumes that for any condition attribute, all other condition attributes are deemed as its parent nodes and thus used in the following (10) to express the conditional probability as

$$P\left(x\left|w_m\right.\right)=\prod_{d=1}^{\mathcal{D}}P\left(x_d\left|\mathrm{Par}\left(x_d\right),w_m\right.\right)=\prod_{d=1}^{\mathcal{D}}\sum_{\substack{i=1\\i\neq d}}^{\mathcal{D}}\left[\alpha_{di}\times P\left(x_d\left|x_i,w_m\right.\right)\right],\tag{10}$$

where

$$\mathrm{Par}\left(x_d\right)=\begin{cases}\left\{x_2,x_3,\ldots,x_{\mathcal{D}}\right\}, & \text{if } d=1,\\ \left\{x_1,x_2,\ldots,x_{\mathcal{D}-1}\right\}, & \text{if } d=\mathcal{D},\\ \left\{x_1,\ldots,x_{d-1},x_{d+1},\ldots,x_{\mathcal{D}}\right\}, & \text{otherwise}\end{cases}\tag{11}$$

is the parent node set of the d-th condition attribute and α_{di} is the dependence weight. The calculation of dependence weights is very important to the construction of HNB classifiers. Jiang et al. [17] used the conditional mutual information to represent the dependence between the d-th condition attribute and the i-th condition attribute as

$$\alpha_{di}=\frac{I\left(x_d,x_i\left|w_m\right.\right)}{\sum_{\substack{j=1\\j\neq d}}^{\mathcal{D}}\left[I\left(x_d,x_j\left|w_m\right.\right)\right]},\tag{12}$$

where

$$\begin{aligned}I\left(x_d,x_i\left|w_m\right.\right)&=\sum_{x_d,x_i}P\left(x_d,x_i,w_m\right)\log_2\frac{P\left(x_d,x_i\left|w_m\right.\right)}{P\left(x_d\left|w_m\right.\right)P\left(x_i\left|w_m\right.\right)}\\&=\sum_{x_d,x_i}P\left(x_d,x_i,w_m\right)\log_2\frac{P\left(x_d,x_i,w_m\right)}{P\left(w_m\right)P\left(x_d\left|w_m\right.\right)P\left(x_i\left|w_m\right.\right)}\end{aligned}\tag{13}$$

is the conditional mutual information. Jiang et al. [17] used the M-estimation to calculate the terms of $P\left(w_m\right)$ and $P\left(x_d\left|w_m\right.\right)$ in (13) and $P\left(x_d\left|x_i,w_m\right.\right)$ in (10) as

$$P\left(w_m\right)=\frac{\mathcal{N}_m+\frac{1}{\mathcal{M}}}{\mathcal{N}+1},\tag{14}$$

$$P\left(x_d\left|w_m\right.\right)=\frac{\sum_{n=1}^{\mathcal{N}_m}\mathbb{I}[x_{nd}^{(m)}=x_d]+\frac{1}{n_d}}{\mathcal{N}_m+1},\tag{15}$$

and

$$P\left(x_d,\left|x_i,w_m\right.\right)=\frac{\sum_{n=1}^{\mathcal{N}_m}\mathbb{I}[x_{nd}^{(m)}=x_d,x_{ni}^{(m)}=x_i]+\frac{1}{n_d}}{\sum_{n=1}^{\mathcal{N}_m}\mathbb{I}[x_{ni}^{(m)}=x_i]+1}.\tag{16}$$

3 AG-NBC

This section presents the proposed AG-NBC algorithm, which is based on a novel attribute grouping strategy. The generation mechanism of DAGs is first explained and then the training and prediction processes of AG-NBC are presented.

3.1 Determination of DAGs

From a training set perspective, a set of DAGs $G_1,G_2,\ldots,G_{\mathcal{T}}$ is a partition of the original condition attribute set $A=\{A_1,\ A_2,\ \ldots,\ A_{\mathcal{D}}\}$, where $\mathcal{T}$ is the number of DAGs and $\mathcal{D}_t$ is the number of original condition attributes in the DAG $G_t=\{A_1^{(t)},A_2^{(t)},\ldots,A_{\mathcal{D}_t}^{(t)}\}$, $t=1,2,\ldots,\mathcal{T}$. DAGs satisfy three conditions:

(1) $\bigcup_{t=1}^{\mathcal{T}}G_t=A$ and $\sum_{t=1}^{\mathcal{T}}\mathcal{D}_t=\mathcal{D}$.

(2) For any $t_i,t_j\in\{1,2,\ldots,\mathcal{T}\}$ and $t_i\neq t_j$, $G_{t_i}\cap G_{t_j}=\emptyset$.

(3) For any $l\in\{1,2,\ldots,\mathcal{D}_t\}$, there exists $d\in\{1,2,\ldots,\mathcal{D}\}$ such that $A_l^{(t)}=A_d$.

Authors of this paper have observed that condition attributes of a same DAG typically have a strong dependence to each other while those from different DAGs generally have a weak dependence. Inspired by subspace clustering techniques [42, 43], an objective function is designed as

$$\begin{aligned} L\left(U,V,R,G\right) = &\sum_{m=1}^{\mathcal{M}}\sum_{n=1}^{\mathcal{N}} p_{nm}\sum_{d=1}^{\mathcal{D}} u_{md}(x_{nd}-z_{md})^2 + \varepsilon_1\sum_{m=1}^{\mathcal{M}}\sum_{d=1}^{\mathcal{D}} u_{md}^2 \\ &+\sum_{t=1}^{\mathcal{T}}\sum_{d=1}^{\mathcal{D}} g_{dt}\sum_{m=1}^{\mathcal{M}} r_{mt}(u_{md}-v_{mt})^2 + \varepsilon_2\sum_{m=1}^{\mathcal{M}}\sum_{t=1}^{\mathcal{T}} r_{mt}\ln r_{mt} \end{aligned} \tag{17}$$

to determine the optimal DAGs.

• $P=(p_{nm})_{\mathcal{N}\times\mathcal{M}}$ is a sample membership degree (SMD) matrix, where

$$p_{nm} = \begin{cases} 1, & \text{if } y_n = w_m, \\ 0, & \text{if } y_n \neq w_m \end{cases} \tag{18}$$

represents the membership degree that the n-th sample x_n belongs to the m-th class w_m.

• $Z=(z_{md})_{\mathcal{M}\times\mathcal{D}}$ is the sample class center (SCC) matrix, where

$$z_{md} = \frac{\sum_{n=1}^{\mathcal{N}_m} x_{nd}^{(m)}}{\mathcal{N}_m} \tag{19}$$

represents the d-th attribute value of the m-class center.

• $U=(u_{md})_{\mathcal{M}\times\mathcal{D}}$ is an attribute contribution degree (ACD) matrix, $u_{md}\in(0,1)$ representing the contribution degree of the d-th condition attribute A_d to the m-th class w_m. For any $m\in\{1,2,\ldots,\mathcal{M}\}$, there exists

$$\sum_{d=1}^{\mathcal{D}} u_{md} = 1. \tag{20}$$

• $V=(v_{mt})_{\mathcal{M}\times\mathcal{T}}$ is a group contribution degree (GCD) matrix, where $v_{mt}\in\Re$ represents the contribution degree of the t-th DAG G_t on the m-th class w_m.

• $R=(r_{mt})_{\mathcal{M}\times\mathcal{T}}$ is a group importance degree (GID) matrix, where $r_{mt}>0$ represents the importance degree of the t-th DAG G_t on the m-th class w_m. For any $t\in\{1,2,\ldots,\mathcal{T}\}$, there exists

$$\sum_{m=1}^{\mathcal{M}} r_{mt} = 1. \tag{21}$$

• $G=(g_{dt})_{\mathcal{D}\times\mathcal{T}}$ is an attribute membership degree (AMD) matrix,

$$g_{dt} = \begin{cases} 1, & \text{if } A_d \in G_t, \\ 0, & \text{if } A_d \notin G_t \end{cases} \tag{22}$$

that represents the membership degree that the d-th condition attribute A_d belongs to the t-th DAG G_t.

• $\varepsilon_1>0$ and $\varepsilon_2<0$ are two regularization factors.

For subspace clustering, attribute groups (AGs) that can bring about good clustering results are determined in an interactive [42] or automatic [43] way. The essence of AGs in subspace clustering is to explore the homogeneous relations among data attributes by assigning weights to them according to their contributions to the generation of clusters: the attributes in an AG have the same weights, while different weights are assigned to attributes from different AGs. Similarly to how subspace clustering generates clusters, AG-NBC tries to identify potential attribute relations to train an NBC. The differences between subspaces and DAGs are in their usage and in the generation principle: (1) subspaces are intended for clustering, while DAGs are used for classification and (2) the generation of subspaces is untraceable due to the loss of class information, whereas identification through DAGs can be achieved using specific class information. The optimal ACD matrix U, GCD matrix V, GID matrix R, and AMD matrix G can be calculated iteratively by minimizing the objective function of (17).

We briefly discuss why minimizing the objective function shown in (17) can lead to optimal DAGs. Eq. (17) mainly considers two kinds of membership information regarding sample membership to class (i.e., the first term in (17)) and attribute membership to group (i.e., the third term in (17)) and two types of weight optimizations about attribute contribution to class (i.e., the second term in (17)) and group importance to class, i.e., the fourth term in (17). The second term is used to distinguish the different attribute contributions to classification by assigning weights for condition attributes and the fourth term is used to quantize the group importance by considering the different attribute contributions. The optimal DAGs should not only minimize the distance between samples and class centers (i.e., the first term) but also the distance between attributes and group centers, i.e., the third term. Thus, we try to find optimal DAGs by minimizing the designed objective function as shown in (17). The updating rules of u_{md}, v_{mt}, r_{mt}, and g_{dt} are derived as follows.

• Updating rule of GCD v_{mt}. By calculating the first derivative of L with respect to v_{mt}, the equation

$$\sum_{d=1}^{\mathcal{D}} 2g_{dt}r_{mt}\left(w_{md}-v_{mt}\right)=0 \tag{23}$$

can be obtained and further simplified as

$$v_{mt}=\frac{\sum_{d=1}^{\mathcal{D}} g_{dt}u_{md}}{\sum_{d=1}^{\mathcal{D}} g_{dt}}. \tag{24}$$

• Updating rule of AMD g_{dt}. Eq. (17) indicates that L is a linear function with respect to g_{dt}, where the value of g_{dt} is calculated as

$$g_{dt}=\begin{cases}1, & \text{if } t=\underset{s=1,2,\ldots,\mathcal{T}}{\arg\min}\left\{\sum_{m=1}^{\mathcal{M}} r_{ms}(u_{md}-v_{ms})^2\right\},\\ 0, & \text{otherwise,}\end{cases} \tag{25}$$

so that the value of L reaches the minimum for other parameters.

• Updating rule of ACD u_{md}. The Lagrangian function by considering both (17) and (20) is constructed as

$$\tilde{L}_1\left(U,V,R,G\right)=L\left(U,V,R,G\right)+\sum_{m=1}^{\mathcal{M}}\left[\lambda_m\left(\sum_{d=1}^{\mathcal{D}} u_{md}-1\right)\right], \tag{26}$$

where $\lambda_1,\lambda_2,\ldots,\lambda_{\mathcal{M}}$ are Lagrange multipliers. The first derivatives of $\tilde{L}_1$ with respect to u_{md} and λ_m are

$$\frac{\partial\tilde{L}_1}{\partial u_{md}}=\sum_{n=1}^{\mathcal{N}} p_{nm}(x_{nd}-z_{md})^2+2\varepsilon_1 u_{md}+2\sum_{t=1}^{\mathcal{T}} g_{dt}r_{mt}\left(u_{md}-v_{mt}\right)+\lambda_m=0 \tag{27}$$

and

$$\frac{\partial\tilde{L}_1}{\partial\lambda_m}=\sum_{d=1}^{\mathcal{D}} u_{md}-1=0. \tag{28}$$

From (27), the following equation is derived:

$$u_{md}=\frac{2Q_{md}^{(3)}-Q_{md}^{(1)}-\lambda_m}{2[\varepsilon_1+Q_{md}^{(2)}]}, \tag{29}$$

where

$$Q_{md}^{(1)}=\sum_{n=1}^{\mathcal{N}} p_{nm}(x_{nd}-z_{md})^2, \tag{30}$$

$$Q_{md}^{(2)}=\sum_{t=1}^{\mathcal{T}} g_{dt}r_{mt}, \tag{31}$$

and

$$Q_{md}^{(3)} = \sum_{t=1}^{\mathcal{T}} g_{dt} r_{mt} v_{mt}. \tag{32}$$

Substituting (29) into (28) yields

$$\lambda_m = \frac{\sum_{d=1}^{\mathcal{D}} \frac{2Q_{md}^{(3)} - Q_{md}^{(1)}}{2[\varepsilon_1 + Q_{md}^{(2)}]} - 1}{\sum_{d=1}^{\mathcal{D}} \frac{1}{2[\varepsilon_1 + Q_{md}^{(2)}]}}. \tag{33}$$

Then, the iterative expression of u_{md} is obtained by substituting (33) into (29) as

$$u_{md} = \frac{2Q_{md}^{(3)} - Q_{md}^{(1)} - \frac{\sum_{d=1}^{\mathcal{D}} \frac{2Q_{md}^{(3)} - Q_{md}^{(1)}}{2[\varepsilon_1 + Q_{md}^{(2)}]} - 1}{\sum_{d=1}^{\mathcal{D}} \frac{1}{2[\varepsilon_1 + Q_{md}^{(2)}]}}}{2[\varepsilon_1 + Q_{md}^{(2)}]}. \tag{34}$$

• Updating rule of GID r_{mt}. The Lagrangian function by considering both (17) and (21) is constructed as

$$\tilde{L}_2(U, V, R, G) = L(U, V, R, G) + \sum_{t=1}^{\mathcal{T}} \left[\tau_t \left(\sum_{m=1}^{\mathcal{M}} r_{mt} - 1 \right) \right], \tag{35}$$

where $\tau_1, \tau_2, \ldots, \tau_{\mathcal{T}}$ are Lagrange multipliers. The first derivatives of $\tilde{L}_2$ with respect to v_{mt} and τ_t are

$$\frac{\partial \tilde{L}_2}{\partial r_{mt}} = \sum_{d=1}^{\mathcal{D}} g_{dt} (u_{md} - v_{mt})^2 + \varepsilon_2 (\ln r_{mt} + 1) + \tau_t = 0 \tag{36}$$

and

$$\frac{\partial \tilde{L}_2}{\partial \tau_t} = \sum_{m=1}^{\mathcal{M}} r_{mt} - 1 = 0. \tag{37}$$

From (36), it is derived that

$$r_{mt} = \exp\left[- \left(1 + \frac{Q_{mt}^{(4)} + \tau_t}{\varepsilon_2} \right) \right], \tag{38}$$

where

$$Q_{mt}^{(4)} = \sum_{d=1}^{\mathcal{D}} g_{dt} (u_{md} - v_{mt})^2. \tag{39}$$

Substituting (38) into (37) yields

$$\sum_{m=1}^{\mathcal{M}} \exp\left[- \left(1 + \frac{Q_{mt}^{(4)} + \tau_t}{\varepsilon_2} \right) \right] = 1. \tag{40}$$

Moreover, the expression of τ_t can be derived as

$$\tau_t = \varepsilon_2 \ln \left[\sum_{m=1}^{\mathcal{M}} \exp\left[- \left(1 + \frac{Q_{mt}^{(4)}}{\varepsilon_2} \right) \right] \right]. \tag{41}$$

Then, we obtain the expression of v_{mt} by substituting (41) into (38) as

$$r_{mt} = \exp\left[- \left(1 + \frac{Q_{mt}^{(4)} + \varepsilon_2 \ln \left[\sum_{m=1}^{\mathcal{M}} \exp\left[-\left(1 + \frac{Q_{mt}^{(4)}}{\varepsilon_2} \right) \right] \right]}{\varepsilon_2} \right) \right]. \tag{42}$$

Algorithm 1 Optimization algorithm of objective function $L(U,V,R,G)$.

Input: The given data set $X=(x_{nd})_{\mathcal{N}\times\mathcal{D}}$, SMD matrix $P=(p_{nm})_{\mathcal{N}\times\mathcal{M}}$, SCC matrix $Z=(z_{md})_{\mathcal{M}\times\mathcal{D}}$, regularization factors ε_1 and ε_2, stopping threshold $\delta>0$.
Output: The optimal ACD $\bar{u}_{md}$, GCD $\bar{v}_{mt}$, GID $\bar{r}_{mt}$, AMD $\bar{g}_{dt}$.
Initialize the ACD $\bar{u}_{md}^{(0)}=\frac{1}{\mathcal{D}}$, $m=1,2,\ldots,\mathcal{M}$, $d=1,2,\ldots,\mathcal{D}$;
Initialize the AMD $\bar{g}_{dt}^{(0)}=\begin{cases}1, & \text{if } t=1,\\ 0, & \text{otherwise,}\end{cases}$ $d=1,2,\ldots,\mathcal{D}$, $t=1,2,\ldots,\mathcal{T}$;
Calculate the GCD $\bar{v}_{mt}^{(0)}$ according to (24);
Calculate the GID $\bar{r}_{mt}^{(0)}$ according to (42);
Initialize the iteration number $\mathcal{I}=-1$;
repeat
  $\mathcal{I}\leftarrow\mathcal{I}+1$;
  Calculate the value $L^{(\mathcal{I})}$ of the objective function (17) based on $\bar{u}_{md}^{(\mathcal{I})}$, $\bar{v}_{mt}^{(\mathcal{I})}$, $\bar{r}_{mt}^{(\mathcal{I})}$, $\bar{g}_{dt}^{(\mathcal{I})}$;
  Update the ACD $\bar{u}_{md}^{(\mathcal{I}+1)}$ according to (34) based on $\bar{v}_{mt}^{(\mathcal{I})}$, $\bar{r}_{mt}^{(\mathcal{I})}$, $\bar{g}_{dt}^{(\mathcal{I})}$;
  Update the GCD $\bar{v}_{mt}^{(\mathcal{I}+1)}$ according to (24) based on $\bar{u}_{md}^{(\mathcal{I}+1)}$ and $\bar{g}_{dt}^{(\mathcal{I})}$;
  Update $\bar{g}_{dt}^{(\mathcal{I}+1)}$ according to (25) based on $\bar{u}_{md}^{(\mathcal{I}+1)}$, $\bar{v}_{mt}^{(\mathcal{I}+1)}$, $\bar{r}_{mt}^{(\mathcal{I})}$;
  Update the GID $\bar{r}_{mt}^{(\mathcal{I}+1)}$ according to (42) based on $\bar{u}_{md}^{(\mathcal{I}+1)}$, $\bar{v}_{mt}^{(\mathcal{I}+1)}$, $\bar{g}_{dt}^{(\mathcal{I}+1)}$;
  Calculate the value $L^{(\mathcal{I}+1)}$ of the objective function (17) based on $\bar{u}_{md}^{(\mathcal{I}+1)}$, $\bar{v}_{mt}^{(\mathcal{I}+1)}$, $\bar{r}_{mt}^{(\mathcal{I}+1)}$, $\bar{g}_{dt}^{(\mathcal{I}+1)}$;
until $|L^{(\mathcal{I}+1)}-L^{(\mathcal{I})}|<=\delta$
Obtain the optimal ACD $\bar{u}_{md}=\bar{u}_{md}^{(\mathcal{I})}$, GCD $\bar{v}_{mt}=\bar{v}_{mt}^{(\mathcal{I})}$, GID $\bar{r}_{mt}=\bar{r}_{mt}^{(\mathcal{I})}$, AMD $\bar{g}_{dt}=\bar{g}_{dt}^{(\mathcal{I})}$.

3.2 Training AG-NBC

Training AG-NBC is done as follows. It can be observed in (24), (25), (34), and (42) that v_{mt} is a function of u_{md} and r_{mt} is a function of v_{mt} and u_{md} for the given g_{dt}. Thus, the values of u_{md}, $m=1,2,\ldots,\mathcal{M}$, $d=1,2,\ldots,\mathcal{D}$ need to be determined first. After the updating rules of all parameters are found, Algorithm 1 applies the optimization procedures for ACD u_{md}, GCD v_{mt}, GID r_{mt}, AMD g_{dt}, $t=1,2,\ldots,\mathcal{T}$ based on the given data set $X=(x_{nd})_{\mathcal{N}\times\mathcal{D}}$, SMD matrix $P=(p_{nm})_{\mathcal{N}\times\mathcal{M}}$, and SCC matrix $Z=(z_{md})_{\mathcal{M}\times\mathcal{D}}$. The next paragraphs present Algorithm 1 in more details.

• The term of $\sum_{m=1}^{\mathcal{M}}\sum_{t=1}^{\mathcal{T}}r_{mt}\ln r_{mt}$ in the objective function (17) ensures a greater diversity for r_{mt}s than that of $\sum_{m=1}^{\mathcal{M}}\sum_{t=1}^{\mathcal{T}}r_{mt}^2$ and leads to clearer attribute grouping. For example, there are two parameters $\alpha_1,\alpha_2\in[0,1]$, which satisfy $\alpha_1+\alpha_2=1$. The optimal α_1 and α_2 that minimize the following two objective functions

$$\begin{cases} l_1(\alpha_1,\alpha_2)=-(\alpha_1\ln\alpha_1+\alpha_2\ln\alpha_2),\\ l_2(\alpha_1,\alpha_2)=\alpha_1^2+\alpha_2^2,\end{cases}\tag{43}$$

are $(\alpha_1,\alpha_2)=(0,1)$ or $(\alpha_1,\alpha_2)=(1,0)$ for $l_1(\alpha_1,\alpha_2)$ and $(\alpha_1,\alpha_2)=\left(\frac{1}{2},\frac{1}{2}\right)$ for $l_2(\alpha_1,\alpha_2)$. It can be found that the optimal parameters of $l_1(\alpha_1,\alpha_2)$ have greater diversities than the optimal parameters of $l_2(\alpha_1,\alpha_2)$.

• The terms of $\sum_{d=1}^{\mathcal{D}}u_{md}(x_{nd}-z_{md})^2$ and $\sum_{m=1}^{\mathcal{M}}r_{mt}(u_{md}-v_{mt})^2$ in (17) represent sample classification and attribute grouping, respectively. u_{md} measures the contribution degree of the d-th attribute of sample $x_n=(x_{n1},x_{n2},\ldots,x_{n\mathcal{D}})$ to the current classification result. The more important the d-th attribute is, the larger the value of u_{md} is. r_{mt} measures the importance degree of the t-th attribute group to the correct classification result when the d-th attribute is included in the t-th attribute group, where v_{mt} is the attribute center of samples in the m-th class.

• The computational complexities of determining the ACD u_{md}, GCD v_{mt}, GID r_{mt}, AMD g_{dt} are $O(\mathcal{MD})$, $O(\mathcal{MT})$, $O(\mathcal{MT})$, and $O(\mathcal{DT})$, respectively. Hence, the computational complexity of Algorithm 1 is

$$O(\mathcal{I}(\mathcal{MD}+\mathcal{MT}+\mathcal{MT}+\mathcal{DT}))=O(\mathcal{I}(\mathcal{M}(\mathcal{D}+2\mathcal{T})+\mathcal{DT}))\approx O(\mathcal{IDT}),\tag{44}$$

when $\mathcal{M}\ll\mathcal{DT}$ holds for a given data set.

After DAGs $G_1,G_2,\ldots,G_{\mathcal{T}}$ are identified by Algorithm 1, the AG-NBC is constructed as follows.

Similarly to the NBC, the class label of a new sample $x = (x_1, x_2, \ldots, x_{\mathcal{D}})$ can be calculated as

$$\begin{aligned} P(w_m|x) &= P(w_m|x_1, x_2, \ldots, x_{\mathcal{D}}) \\ &= P(w_m|G_1(x), G_2(x), \ldots, G_{\mathcal{T}}(x)) \\ &= \frac{P(G_1(x), G_2(x), \ldots, G_{\mathcal{T}}(x)|w_m)P(w_m)}{P(G_1(x), G_2(x), \ldots, G_{\mathcal{T}}(x))} \\ &= \frac{[\prod_{t=1}^{\mathcal{T}} P(G_t(x)|w_m)]P(w_m)}{P(G_1(x), G_2(x), \ldots, G_{\mathcal{T}}(x))} \\ &= \frac{[\prod_{t=1}^{\mathcal{T}} \frac{P(w_m|G_t(x))P(G_t(x))}{P(w_m)}]P(w_m)}{\prod_{t=1}^{\mathcal{T}} P(G_t(x))} \\ &= \frac{\prod_{t=1}^{\mathcal{T}} P(w_m|G_t(x))}{[P(w_m)]^{\mathcal{T}-1}}, \end{aligned} \tag{45}$$

where $G_t(x) = \{x_1^{(t)}, x_2^{(t)}, \ldots, x_{\mathcal{D}_t}^{(t)}\}$, $t = 1, 2, \ldots, \mathcal{T}$ represents the attribute values of the new sample x in the t-th DAG. To compute $P\left(w_m\,|G_t(x)\right)$ in (45) with a low computational complexity, an RVFL network [44–46] is used to model the joint probability of attributes in a specific DAG. The RVFL network is a special single hidden-layer feed-forward neural network, where the input layer is directly connected to both the hidden layer and output layer, i.e., the connections between the input layer and output layer are added in a three-layer full-linked feed-forward neural network. For the RVFL network, weights of connections between the input layer and hidden layers are randomly assigned, while weights of connections between the hidden layer and output layer and between the input layer and output layer are analytically calculated. RVFL networks have extremely fast training speed as they do not perform time-consuming weight adjustments.

For the given training data set $\mathbb{D}$ with DAGs $G_1, G_2, \ldots, G_{\mathcal{T}}$, the RVFL network corresponding to the t-th DAG is denoted as RVFL$_t$. It is trained based on the data set

$$\mathbb{D}_t = \left\{ \left(x_n^{[t]}, y_n \right) \middle| x_n^{[t]} = \left(x_{n1}^{[t]}, x_{n2}^{[t]}, \ldots, x_{n\mathcal{D}_t}^{[t]} \right), y_n = (y_{n1}, y_{n2}, \ldots, y_{n\mathcal{M}}), n = 1, 2, \ldots, \mathcal{N} \right\} \tag{46}$$

having $\mathcal{D}_t$ dependent attributes in G_t. The predicted output for the new sample x corresponding to RVFL$_t$ is represented as

$$\begin{aligned} y^{(t)} &= \left(y_1^{(t)}, y_2^{(t)}, \ldots, y_{\mathcal{M}}^{(t)} \right) \\ &= \left(h_1^{(t)}, h_2^{(t)}, \ldots, h_{\mathcal{L}}^{(t)}, x_1^{(t)}, x_2^{(t)}, \ldots, x_{\mathcal{D}_t}^{(t)} \right) V^{(t)}, \end{aligned} \tag{47}$$

where

$$y_m^{(t)} = P\left(w_m\,|G_t(x)\right) \in [0, 1] \tag{48}$$

is regarded as the posterior probability, and $\mathcal{L}$ is the number of hidden-layer nodes in the RVFL$_t$,

$$h_l^{(t)} = \frac{1}{1 + \exp(-s_l^{(t)})} \tag{49}$$

is the output of the l-th ($l = 1, 2, \ldots, \mathcal{L}$) hidden-layer node,

$$\left(s_1^{(t)}, s_2^{(t)}, \ldots, s_{\mathcal{L}}^{(t)} \right) = \left(x_1^{(t)}, x_2^{(t)}, \ldots, x_{\mathcal{D}_t}^{(t)} \right) W^{(t)} \tag{50}$$

is the hidden-layer input vector, $W^{(t)} = (\alpha_{dl}^{(t)})_{\mathcal{D}_t \times \mathcal{L}}$ is the input-layer weight matrix which is initialized with random numbers, and $V^{(t)} = (\beta_{lm}^{(t)})_{\mathcal{L} \times \mathcal{M}}$ is the output-layer weight matrix which is calculated as

$$V^{(t)} = \left[\begin{array}{cc} H^{(t)} & X^{(t)} \end{array} \right]^{\dagger} Y^{(t)}, \tag{51}$$

$[H^{(t)}\ X^{(t)}]^{\dagger}$ is the pseudo-inverse matrix of $[H^{(t)}\ X^{(t)}]$, where

$$H^{(t)} = \frac{1}{1 + \exp(-X^{[t]} W^{(t)})} \tag{52}$$

Table 1 Description of the thirty benchmark data sets.

Data set	Sample	Attribute	Class	$\mathcal{T}$	Number of samples in each class	$\mathcal{L}$
Arrhythmia	1385	279	4	5	(336, 332, 355, 362)	300
Australian	690	14	2	2	(383, 307)	50
Bp	198	33	2	3	(47, 151)	50
Cleveland	297	13	5	3	(160, 54, 35, 35, 13)	50
Coil2000	9822	85	2	4	(9236, 586)	100
Contraceptive	1473	9	3	2	(629, 333, 511)	50
Glass	214	9	6	2	(70, 76, 17, 29, 13 , 9)	50
Heart	270	13	2	3	(120, 150)	50
Hepatitis	280	19	2	3	(113, 167)	50
Ionosphere	351	32	2	3	(126, 225)	50
Letter	20000	16	26	3	(789, 766, 736, 805, 768, 775, 773, 734, 755, 747, 739, 761, 792, 783, 753, 803, 783, 758, 748, 796, 813, 764, 752, 787, 786, 734)	50
Libras	360	90	15	5	(24, 24, 24, 24 ,24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24)	100
Muskroom	476	166	2	5	(207, 269)	200
Optdigits	5620	64	10	4	(554, 571, 557, 572, 568, 558, 558, 566, 554, 562)	100
Page_small	547	10	5	2	(486, 36, 4, 9, 12)	50
Parkinsons	195	22	2	4	(147, 48)	50
Penbase	10992	16	10	3	(1143, 1143, 1144, 1055, 1144, 1055, 1056, 1142, 1055, 1055)	50
Ring	7400	20	2	3	(3664, 3736)	50
Satimage	6435	36	6	4	(1533, 703, 1358, 626, 707, 1508)	50
Segment	2310	19	7	3	(330, 330, 330, 330, 330, 330, 330)	50
Sonar	208	60	2	4	(97, 111)	100
Spambase	4597	57	2	3	(2785, 1812)	100
Spectf	267	44	2	4	(212, 55)	50
Texture	5500	40	11	4	(500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 500)	50
Thyroid	215	21	3	3	(150, 35, 50)	50
Twonorm	7400	20	2	3	(3703, 3697)	50
Wdbc	569	30	2	4	(357, 212)	50
Wine	178	13	3	2	(59, 71, 48)	50
WineQR	1599	11	6	2	(10, 53, 681, 638, 199, 18)	50
WineQW	489	11	6	2	(1, 15, 153, 218, 83, 19)	50

is the hidden-layer output matrix corresponding to the training data $X^{[t]} = (x_{nd}^{[t]})_{\mathcal{N}\times\mathcal{D}_t}$. For the construction of the AG-NBC, dependence exists for attributes in the same DAG and thus the RVFL network is used to capture this type of dependence, while the different DAGs are independent of each other and the product of the output of the different RVFL networks are considered as the joint probability.

4 Experimental settings and results

This section describes a series of experiments that were carried out to evaluate the feasibility, rationality, and effectiveness of the proposed AG-NBC. Thirty mixed-attribute data sets were utilized, which were obtained from the KEEL software [47] website.

A description of these data sets is provided in Table 1[2)]. All data sets can be obtained from the AG-NBC Data Sets folder of our BaiDuPan online storage space[3)] using the extraction code "tyc9". The experiments were run on a workstation equipped with an Intel Quadcore 3.00 GHz i5-9400 CPU and 16 GB of main memory.

4.1 Feasibility validation of AG-NBC

The first experiment assessed the feasibility of AG-NBC, and in particular the convergence of Algorithm 1. This includes the convergence of ACD, GCD, GID, and AMD. Two data sets, i.e., parkinsons

2) For each data set, the optimal number of hidden-layer nodes in the RVFL network and the number of DAGs are determined using cross-validation. For some given $\mathcal{T}$ and $\mathcal{L}$ values, the optimal DAGs are calculated using Algorithm 1 by optimizing the objective function shown in (17).

3) https://pan.baidu.com/s/1Ix6s6RDK-ZrTR31zEoiThg.

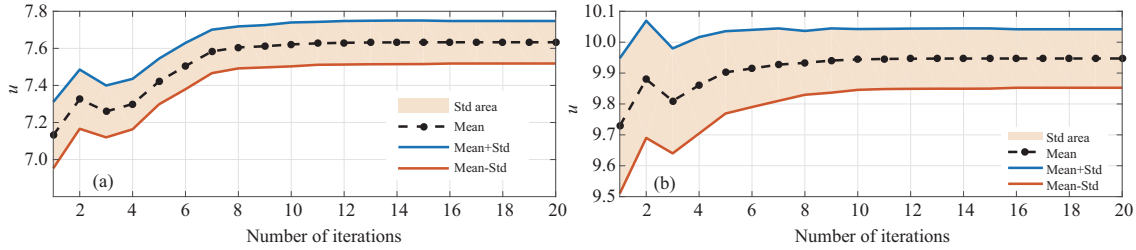

Figure 1 (Color online) Convergence of ACD u_{md}. (a) Parkinsons data set; (b) spectf data set.

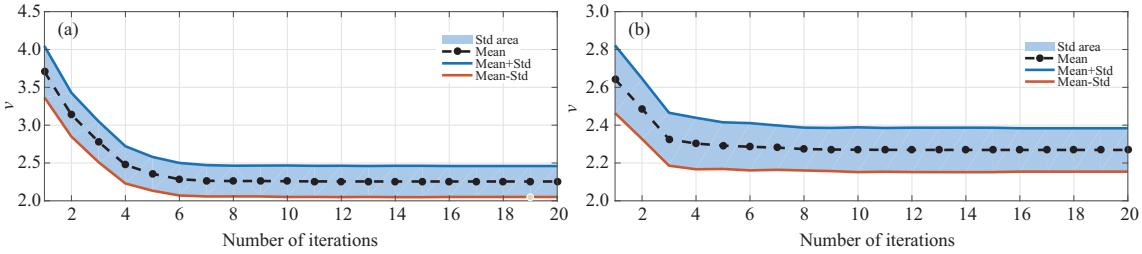

Figure 2 (Color online) Convergence of GCD v_{mt}. (a) Parkinsons data set; (b) spectf data set.

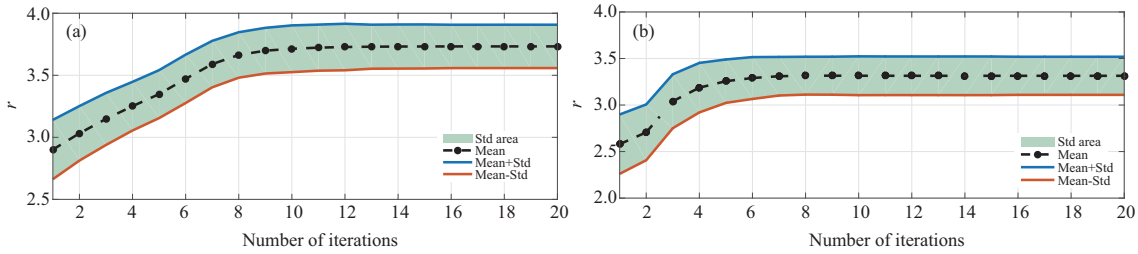

Figure 3 (Color online) Convergence of GID r_{mt}. (a) Parkinsons data set; (b) spectf data set.

and spectf are selected as examples in this experiment to train Algorithm 1, where the regularization factors are $\varepsilon_1 = 1$, $\varepsilon_2 = -1$ and the stopping threshold is set to $\delta = 0.001$. For ACD, GCD, GID, and AMD, the variation trends of

$$u = \sum_{m=1}^{\mathcal{M}} \sum_{d=1}^{\mathcal{D}} |u_{md}|, \tag{53}$$

$$v = \sum_{m=1}^{\mathcal{M}} \sum_{t=1}^{\mathcal{T}} |v_{mt}|, \tag{54}$$

$$r = \sum_{m=1}^{\mathcal{M}} \sum_{t=1}^{\mathcal{T}} |r_{mt}|, \tag{55}$$

and

$$g_t = \sum_{d=1}^{\mathcal{D}} g_{dt}, t = 1, 2, \ldots, \mathcal{T} \tag{56}$$

are validated by increasing the iteration count. Independent training was conducted ten times for ACD, GCD, and GID by randomly selecting 70% samples from the original data sets. For AMD, the whole data set was used to obtain the final attribute groups. Figures 1–4 present the convergences of ACD, GCD, GID, and AMD, respectively.

It can be clearly seen that ACD, GCD, GID, and AMD converge to a stable value as the iteration number increases. ACD (Figure 1) and GID (Figure 3) first increase and then converge, while GCD (Figure 2) first decreases and then converges. For the parkinsons ($\mathcal{T} = 4$) and spectf ($\mathcal{T} = 3$) data sets, the optimal DAGs in Figure 4, which are determined by Algorithm 1, are

$$\{11, 14, 16\}, \{1, 3, 17, 19\}, \{2, 15, 18, 20, 22\}, \{4, 5, 6, 7, 8, 9, 10, 12, 13, 21\}$$

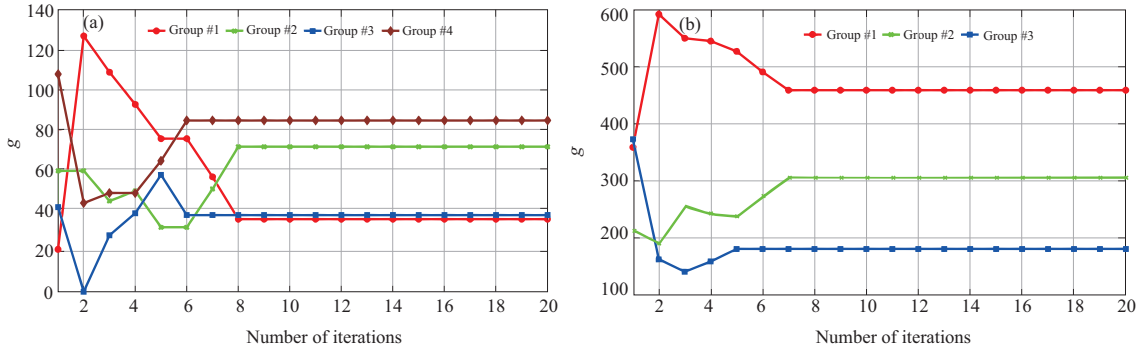

Figure 4 (Color online) Convergence of AMD g_{dt}. (a) Parkinsons data set; (b) spectf data set.

and

$$\{2,7,14,22,27,28,37\},\{1,3,6,8,9,11,12,17,18,19,20,21,23,31,32,34,35,38\},$$
$$\{4,5,10,13,15,16,24,25,26,29,30,33,36,39,40,41,42,43,44\},$$

respectively. These experimental results demonstrate the feasibility of using the designed objective function and updating rules for ACD, GCD, GID, and AMD. It indicates that Algorithm 1 can find the optimal DAGs for the given data set.

4.2 Rationality validation of AG-NBC

The second experiment is a rationality validation, which was carried out to confirm whether the optimal attribute groups contain dependent attributes and if the classification risk is reduced. Two data sets are used as example in this experiment, namely segment and contraceptive. For Algorithm 1, the regularization factors are $\varepsilon_1=1$, $\varepsilon_2=-1$ and the stopping threshold is $\delta=0.001$. For the RVFL network, the input-layer weights are initialized with uniform random numbers in the $[-1,1]$ interval and the number of hidden-layer nodes is 50.

For the segment data set, two optimal DAGs #1 $\{6,7,8,9\}$ and #2 $\{10,11,12,13,15,17\}$ are selected to verify the result of attribute dependence exploration in AG-NBC construction, where the optimal DAGs are determined with Algorithm 1 by optimizing the objective function shown in (17). The dependence heatmaps corresponding to DAGs #1 and #2 are presented in Figures 5(a) and (b), and are used to present the attribute dependence in the same DAG and attribute independence in different DAGs. The dependence heatmap corresponding to attributes in DAGs #1 and #2 is presented in Figure 5(c). The dependence degree between two condition attributes is measured using the mutual information which is calculated with the sklearn.metrics.mutual_info_score[4] package of the scikit-learn machine learning library. It can be observed in Figure 5 that the dependence degree between attributes in the same DAG is obviously larger than the dependence degree between attributes from different DAGs. The experimental results indicate that the proposed attribute grouping strategy is effective to find dependent attributes and separate independent attributes.

On the contraceptive data set, NBC and AG-NBC were trained and tested using 70% and 30% of the samples, respectively. The classification risks of NBC and AG-NBC are compared based on the predicted outputs of NBC and AG-NBC for testing samples. Assume that the posterior probabilities of NBC and AG-NBC are

$$\left\{P_n^{(\mathrm{NBC})}\middle|P_n^{(\mathrm{NBC})}=\left(P_{n1}^{(\mathrm{NBC})},P_{n2}^{(\mathrm{NBC})},P_{n3}^{(\mathrm{NBC})}\right),n=1,2,\ldots,\mathcal{N}'\right\}$$

and

$$\left\{P_n^{(\text{AG-NBC})}\middle|P_n^{(\text{AG-NBC})}=\left(P_{n1}^{(\text{AG-NBC})},P_{n2}^{(\mathrm{NBC})},P_{n3}^{(\text{AG-NBC})}\right),n=1,2,\ldots,\mathcal{N}'\right\},$$

where $\mathcal{N}'=442$ is the number of testing samples for the contraceptive data set. Let $p_i^{(\mathrm{NBC})}(\cdot)$ and $p_i^{(\text{AG-NBC})}(\cdot)$, $i=1,2,3$ represent the probability density functions (p.d.f.s) of

$$\left\{P_{1i}^{(\mathrm{NBC})},P_{2i}^{(\mathrm{NBC})},\ldots,P_{\mathcal{N}'i}^{(\mathrm{NBC})}\right\}\text{ and }\left\{P_{1i}^{(\text{AG-NBC})},P_{2i}^{(\text{AG-NBC})},\ldots,P_{\mathcal{N}'i}^{(\text{AG-NBC})}\right\},$$

4) https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html.

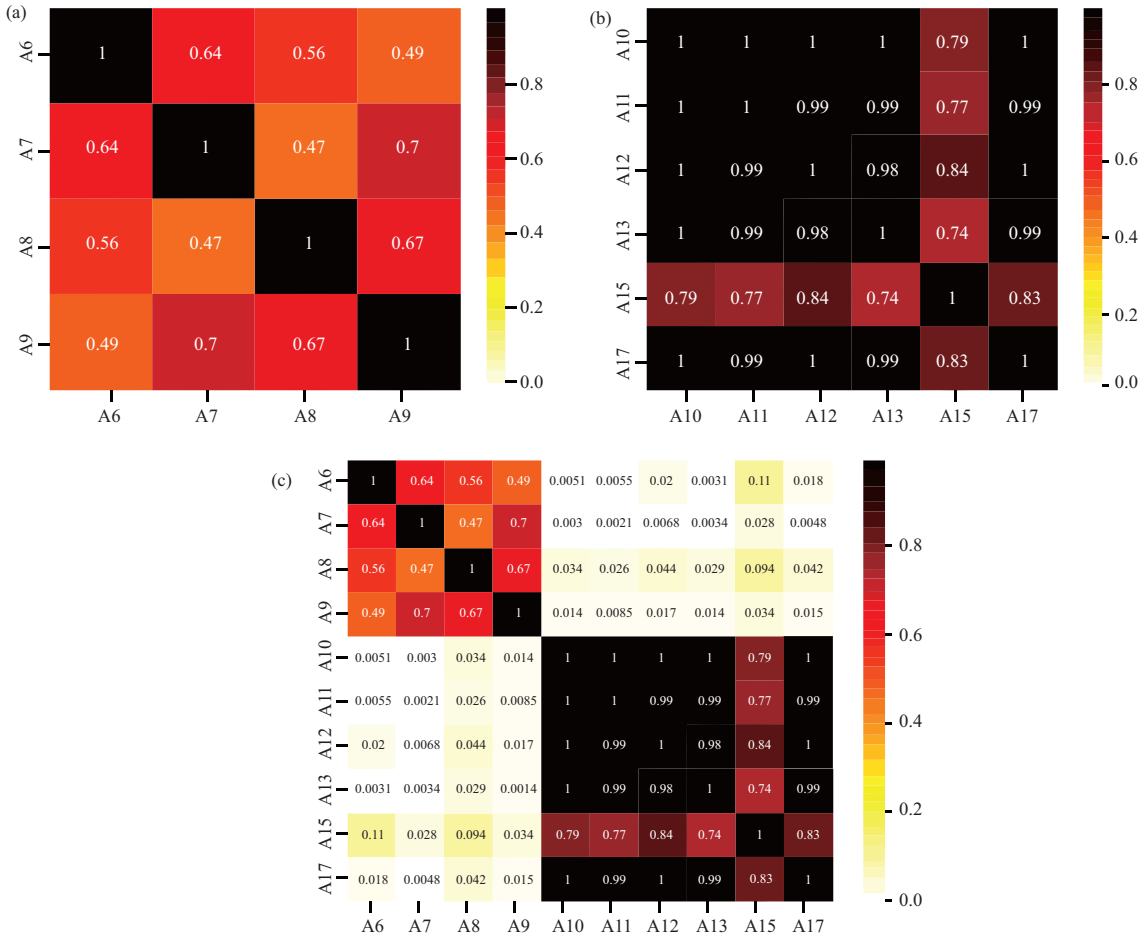

Figure 5 (Color online) Dependence heatmaps for the segment data set. (a) Heatmap of attribute group $\{6, 7, 8, 9\}$; (b) heatmap of attribute group $\{10, 11, 12, 13, 15, 17\}$; (c) heatmap of attributes in DAGs #1 and #2.

which are modeled with normal distributions with means

$$\mu_i^{(\mathrm{NBC})} = \frac{\sum_{n=1}^{\mathcal{N}'} P_{ni}^{(\mathrm{NBC})}}{\mathcal{N}'}, \tag{57}$$

$$\mu_i^{(\mathrm{AG\text{-}NBC})} = \frac{\sum_{n=1}^{\mathcal{N}'} P_{ni}^{(\mathrm{AG\text{-}NBC})}}{\mathcal{N}'}, \tag{58}$$

and standard deviations (Stds)

$$\sigma_i^{(\mathrm{NBC})} = \sqrt{\frac{\sum_{n=1}^{\mathcal{N}'} \left[P_{ni}^{(\mathrm{NBC})} - \mu_i^{(\mathrm{NBC})}\right]^2}{\mathcal{N}' - 1}}, \tag{59}$$

$$\sigma_i^{(\mathrm{AG\text{-}NBC})} = \sqrt{\frac{\sum_{n=1}^{\mathcal{N}'} \left[P_{ni}^{(\mathrm{AG\text{-}NBC})} - \mu_i^{(\mathrm{AG\text{-}NBC})}\right]^2}{\mathcal{N}' - 1}}. \tag{60}$$

For NBC and AG-NBC, the classification risks between class $\#i$ and class $\#j$, $i, j \in \{1, 2, 3\}$ are measured with the areas of intersection regions corresponding to p.d.f.s $p_i^{(\mathrm{NBC})}(\cdot)$, $p_j^{(\mathrm{NBC})}(\cdot)$ and $p_i^{(\mathrm{AG\text{-}NBC})}(\cdot)$, $p_j^{(\mathrm{AG\text{-}NBC})}(\cdot)$ as

$$\mathrm{Risk}_{ij}^{(\mathrm{NBC})} = \int_{-\infty}^{+\infty} \min\left\{p_i^{(\mathrm{NBC})}(x), p_j^{(\mathrm{NBC})}(x)\right\} \mathrm{d}x, \tag{61}$$

and

$$\mathrm{Risk}_{ij}^{(\mathrm{AG\text{-}NBC})} = \int_{-\infty}^{+\infty} \min\left\{p_i^{(\mathrm{AG\text{-}NBC})}(x), p_j^{(\mathrm{AG\text{-}NBC})}(x)\right\} \mathrm{d}x, \tag{62}$$

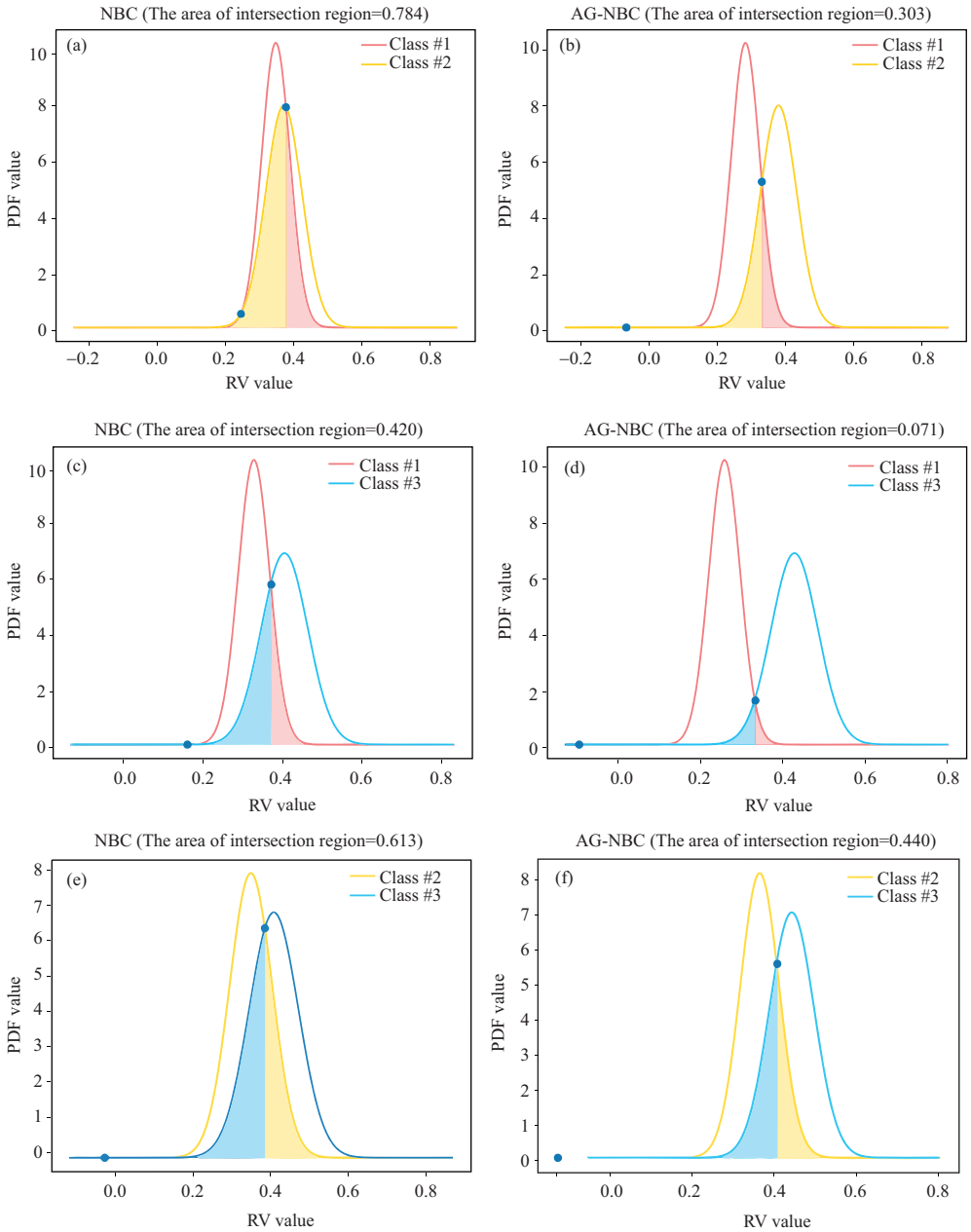

Figure 6 (Color online) Classification risk for the contraceptive data set. (a) NBC's risk between class #1 and class #2; (b) AG-NBC's risk between class #1 and class #2; (c) NBC's risk between class #1 and class #3; (d) AG-NBC's risk between class #1 and class #3; (e) NBC's risk between class #2 and class #3; (f) AG-NBC's risk between class #2 and class #3.

respectively. It can be observed in Figure 6 that the classification risks of AG-NBC are obviously smaller than the classification risks of NBC, e.g., the classification risk of AG-NBC corresponding to class #1 and class #3 is 0.071, while the classification risk of NBC corresponding to class #1 and class #3 is 0.420. It indicates that AG-NBC can provide clearer classification results than NBC for the given testing samples. These experimental results demonstrate the rationality of constructing an NBC using DAGs that are independent from each other. Moreover, using RVFL to model the joint p.d.f. for the dependent attributes in a single DAG is effective.

4.3 Effectiveness validation of AG-NBC

In the third experiment, the effectiveness of AG-NBC is evaluated by comparing its classification performance with that of NBC, FNBC [21], tree augmented Bayes network (TAN) [11], gain ratio-based attribute weighted naive Bayes (GRAWNB) [32], AODE [13], WAODE [14], independent component

analysis-based NBC (ICA-NBC) [36], HNB classifier [17], and CFW [18]. In addition, an ablation experiment is also conducted, where the RVFL network is directly trained based on the original data set to calculate the posterior probability $P(w_m|x)$ without the usage of DAGs. The number of hidden-layer nodes in the ablation RVFL network is twice as many as the number of input-layer nodes. The discrete attributes are processed with the one-hot encoding technique when constructing the ablation RVFL network. All classification algorithms are implemented using the Python programming language.

Thirty KEEL data sets presented in Table 1 are utilized to calculate the testing accuracies, area under the ROC curves (AUCs) [48], and PMSEs [22] of these 11 classification algorithms, where the AUC assesses the misclassification cost and the PMSE measures the estimation quality of posterior probability. For each data set, training and testing are done 10 times using a random partition of samples, i.e., 70% samples to train the different classification algorithms and the remaining 30% to test their generalization capabilities. All Bayesian classification algorithms including AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network are trained and tested based on the same training and testing data sets. In addition, the discretization function pandas.cut[5)] with 10 bins is used to transform the continuous-attribute values into discrete-attribute values for the training and testing of NBC, TAN, AODE, WAODE, HNB, and CFW. Pandas is a powerful and flexible open source data analysis tool for the Python programming language. The parameter settings of Algorithm 1 are set as in the aforementioned experiments. For RVFL networks in AG-NBC and ablation study, the input-layer weights are initialized with uniform random numbers in the $[-1, 1]$ range.

The comparative results of testing accuracies, AUCs, and PMSEs for AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network are summarized in Tables 2–4. It can be clearly seen that AG-NBC obtains the highest average accuracy (i.e., 0.811), highest average AUC (i.e., 0.881), and smallest average PMSE (i.e., 0.324) on the 30 benchmark data sets. Three statistical analyses were conducted based on the experimental results shown in Tables 2–4. First, the two-tailed t-test [49] under the significance level 0.05 was applied to calculate the win\tie\lose counts of AG-NBC and compare them to those of the other classification algorithms. The results are presented in Table 5. Second, it is found that the number of wins for AG-NBC for the 30 data sets is at least $\frac{30}{2} + 1.96 \times \frac{\sqrt{30}}{2} \approx 20$ for a significance level of 0.05. Hence, it can be said that AG-NBC has significantly better testing performances than the other classification algorithms under a significance level of 0.05. Third, the critical difference (CD) value [49] was calculated as

$$\mathrm{CD} = 3.219 \times \sqrt{\frac{11 \times (11+1)}{6 \times 30}} \approx 2.756, \tag{63}$$

for eleven classification algorithms and thirty data sets. In Figure 7, we can see an interval of one CD value to the left and right of the average rank of AG-NBC. Any algorithm with a rank outside this area is significantly different from AG-NBC. The statistical results reveal the following facts.

(1) AG-NBC obtains significantly better testing accuracies than the other 10 classification algorithms. The win records of AG-NBC versus NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL are 30, 26, 26, 28, 27, 26, 27, 29, 28, and 25, respectively. The number of AG-NBC wins is all more than the critical value of 20. Figure 7(a) shows that AG-NBC obtains the smaller average rank value and there is no classification algorithm in the CD interval of AG-NBC. It indicates that the superiority of AG-NBC for testing accuracy is statistically obvious in comparison with the other 10 classification algorithms.

(2) Compared with NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, and CFW, AG-NBC obtains significantly better testing AUCs. The win records of AG-NBC versus the other 9 Bayesian algorithms are more than the critical value 20, i.e., 24, 24, 23, 26, 23, 25, 21, 24, and 21. Figure 7(b) shows that AG-NBC obtains the smaller average rank value and there are only ICA-NBC and CFW in the CD interval of AG-NBC. The main reason why the AUC of AG-NBC is not obviously different from the AUCs of ICA-NBC and CFW is that attribute grouping plays an equal role as the feature extraction and feature weighting strategies to reduce the classification cost of the constructed NBCs.

(3) AG-NBC gets significantly better testing PMSEs than TAN, AODE, WAODE, HNB, and RVFL. The win records of AG-NBC versus them are 21, 20, 20, 21, and 20, respectively. It is a very interesting statistical result as shown in Figure 7(c) that AG-NBC cannot obtain obviously better probability qualities

5) https://pandas.pydata.org/docs/reference/api/pandas.cut.html?highlight=cut#pandas.cut.

Table 2 Comparison of the testing accuracies corresponding to AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network.

Data set	AG-NBC	NBC	FNBC	TAN	GRFWNB	AODE	WAODE	ICA-NBC	HNB	CFW	RVFL
Arrhythmia	**0.625±0.026**	0.576±0.065	0.616±0.044	0.584±0.044	0.575±0.051	0.575±0.043	0.577±0.037	0.579±0.035	0.592±0.023	0.566±0.079	0.617±0.018
Australian	**0.865±0.024**	0.854±0.025	0.865±0.014	0.852±0.013	0.858±0.017	0.861±0.018	0.859±0.016	0.855±0.023	0.865±0.021	0.859±0.019	0.859±0.019
Bp	**0.814±0.022**	0.713±0.048	0.705±0.043	0.710±0.095	0.741±0.052	0.783±0.024	0.783±0.019	0.765±0.027	0.718±0.101	0.733±0.015	0.720±0.028
Cleveland	0.595±0.021	0.555±0.054	0.582±0.036	0.541±0.032	0.577±0.022	0.597±0.053	0.588±0.048	0.591±0.019	0.573±0.014	0.579±0.034	**0.601±0.015**
Coil2000	**0.932±0.004**	0.853±0.011	0.866±0.010	0.865±0.009	0.898±0.005	0.878±0.007	0.889±0.013	0.916±0.010	0.882±0.008	0.885±0.009	0.928±0.005
Contraceptive	0.503±0.017	0.479±0.021	0.483±0.016	0.508±0.015	0.483±0.025	0.479±0.019	0.477±0.020	**0.510±0.013**	0.474±0.016	0.487±0.030	0.501±0.019
Glass	**0.623±0.056**	0.489±0.064	0.496±0.045	0.496±0.042	0.533±0.059	0.514±0.049	0.527±0.026	0.515±0.016	0.489±0.043	0.531±0.054	0.535±0.046
Heart	0.819±0.036	0.814±0.043	0.819±0.038	0.820±0.031	0.815±0.033	0.818±0.022	0.811±0.031	0.814±0.024	0.803±0.035	**0.824±0.039**	0.810±0.032
Hepatitis	0.875±0.034	0.864±0.044	0.843±0.065	0.854±0.077	0.875±0.047	0.896±0.060	0.891±0.023	**0.902±0.017**	0.867±0.061	0.875±0.041	0.895±0.030
Ionosphere	0.899±0.030	0.888±0.033	**0.905±0.023**	0.880±0.020	0.892±0.021	0.884±0.025	0.887±0.028	0.891±0.015	0.888±0.029	0.883±0.033	0.899±0.030
Letter	**0.600±0.007**	0.586±0.014	0.583±0.008	0.597±0.003	0.589±0.013	0.586±0.004	0.583±0.006	0.586±0.016	0.599±0.004	0.588±0.015	0.591±0.010
Libras	**0.705±0.075**	0.601±0.027	0.615±0.046	0.608±0.025	0.623±0.013	0.628±0.033	0.634±0.037	0.649±0.032	0.652±0.041	0.647±0.018	0.636±0.068
Muskroom	**0.817±0.045**	0.805±0.023	0.804±0.008	0.808±0.017	0.807±0.031	0.814±0.023	0.810±0.025	0.810±0.032	0.809±0.027	0.811±0.033	0.791±0.041
Optdigits	0.932±0.006	0.910±0.007	0.911±0.008	0.911±0.005	0.913±0.005	0.906±0.008	0.906±0.011	0.927±0.006	0.909±0.007	0.921±0.004	**0.939±0.003**
Page_small	**0.930±0.015**	0.910±0.015	0.901±0.015	0.892±0.017	0.889±0.028	0.898±0.019	0.901±0.014	0.913±0.018	0.906±0.017	0.903±0.012	0.899±0.013
Parkinsons	**0.853±0.035**	0.776±0.042	0.774±0.059	0.798±0.061	0.783±0.040	0.805±0.049	0.812±0.042	0.813±0.039	0.798±0.031	0.815±0.058	0.824±0.027
Penbase	**0.869±0.009**	0.859±0.006	0.850±0.003	0.859±0.007	0.857±0.004	0.862±0.006	0.860±0.007	0.845±0.010	0.859±0.006	0.863±0.003	0.858±0.011
Ring	0.955±0.002	0.954±0.009	**0.964±0.003**	0.947±0.003	0.953±0.001	0.950±0.003	0.949±0.002	0.951±0.007	0.950±0.002	0.947±0.006	0.946±0.014
Satimage	**0.826±0.015**	0.796±0.004	0.796±0.003	0.798±0.008	0.811±0.007	0.801±0.006	0.789±0.005	0.803±0.008	0798±0.004	0.779±0.006	0.785±0.005
Segment	**0.969±0.007**	0.917±0.005	0.893±0.005	0.915±0.001	0.901±0.008	0.908±0.002	0.901±0.002	0.894±0.009	0.919±0.004	0.905±0.012	0.920±0.014
Sonar	**0.771±0.053**	0.716±0.061	0.717±0.041	0.724±0.042	0.745±0.024	0.743±0.054	0.748±0.044	0.751±0.038	0.737±0.041	0.740±0.048	0.743±0.055
Spambase	**0.886±0.008**	0.747±0.006	0.791±0.009	0.799±0.011	0.778±0.007	0.808±0.007	0.815±0.006	0.801±0.004	0.826±0.005	0.806±0.006	0.856±0.009
Spectf	**0.803±0.033**	0.728±0.059	0.745±0.041	0.777±0.041	0.779±0.056	0.735±0.055	0.744±0.058	0.742±0.032	0.796±0.033	0.786±0.044	0.758±0.033
Texture	**0.946±0.004**	0.789±0.010	0.784±0.013	0.804±0.011	0.833±0.006	0.819±0.014	0.817±0.018	0.836±0.011	0.811±0.009	0.842±0.004	0.812±0.004
Thyroid	**0.935±0.005**	0.933±0.004	**0.935±0.006**	0.934±0.006	0.929±0.004	0.927±0.005	0.928±0.008	0.929±0.007	0.931±0.006	0.929±0.006	0.919±0.006
Twonorm	0.969±0.004	0.967±0.005	**0.971±0.001**	0.966±0.004	0.957±0.006	0.963±0.004	0.963±0.003	0.961±0.004	0.967±0.002	0.962±0.002	0.947±0.005
Wdbc	0.925±0.027	0.910±0.019	0.874±0.025	**0.935±0.013**	0.922±0.015	0.932±0.014	**0.935±0.013**	0.920±0.013	0.927±0.020	0.916±0.011	0.918±0.027
Wine	0.979±0.024	0.969±0.043	0.972±0.021	0.976±0.019	0.972±0.022	0.970±0.021	0.973±0.017	**0.980±0.022**	0.978±0.012	0.972±0.027	0.974±0.026
WineQR	0.574±0.026	0.570±0.019	0.575±0.019	0.576±0.028	0.577±0.033	0.572±0.011	0.573±0.013	0.581±0.018	0.567±0.018	**0.585±0.022**	0.580±0.019
WineQW	**0.534±0.029**	0.520±0.041	0.517±0.018	0.516±0.014	0.520±0.018	0.529±0.018	0.521±0.019	0.529±0.026	0.507±0.017	0.523±0.024	0.519±0.030
Average	**0.811±0.023**	0.768±0.028	0.772±0.023	0.775±0.024	0.780±0.022	0.781±0.023	0.782±0.020	0.785±0.018	0.780±0.022	0.783±0.024	0.789±0.022

Table 3 Comparison of the testing AUCs corresponding to AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network.

Data set	AG-NBC	NBC	FNBC	TAN	GRFWNB	AODE	WAODE	ICA-NBC	HNB	CFW	RVFL
Arrhythmia	**0.690±0.025**	0.635±0.031	0.669±0.047	0.659±0.046	0.647±0.052	0.638±0.025	0.643±0.027	0.644±0.048	0.627±0.027	0.631±0.049	0.688±0.025
Australian	0.919±0.013	0.910±0.017	0.914±0.017	0.891±0.020	0.890±0.018	0.891±0.020	0.891±0.026	**0.927±0.027**	0.890±0.018	0.911±0.022	0.910±0.013
Bp	0.718±0.068	0.570±0.075	0.666±0.076	0.712±0.047	0.676±0.061	0.668±0.090	0.694±0.068	0.707±0.043	**0.739±0.063**	0.692±0.061	0.711±0.068
Cleveland	0.666±0.024	0.625±0.037	0.672±0.032	0.613±0.023	0.685±0.040	0.671±0.041	0.669±0.046	0.681±0.047	0.676±0.040	**0.692±0.034**	0.689±0.043
Coil2000	0.721±0.016	0.710±0.020	0.712±0.016	0.704±0.017	0.716±0.014	**0.741±0.015**	0.737±0.017	0.726±0.018	0.722±0.015	0.729±0.019	0.718±0.015
Contraceptive	**0.727±0.015**	0.676±0.016	0.660±0.018	0.668±0.014	0.664±0.019	0.669±0.021	0.667±0.028	0.676±0.029	0.674±0.013	0.674±0.014	0.706±0.018
Glass	0.872±0.031	0.810±0.044	0.822±0.032	0.839±0.020	0.821±0.023	0.851±0.039	0.869±0.037	0.822±0.024	0.825±0.028	0.866±0.059	**0.877±0.019**
Heart	**0.901±0.017**	0.871±0.022	0.892±0.016	0.879±0.038	0.876±0.047	0.888±0.028	0.893±0.024	0.873±0.026	0.880±0.028	0.880±0.028	0.899±0.016
Hepatitis	0.876±0.075	0.891±0.069	0.846±0.142	0.895±0.068	0.841±0.105	0. 868±0.098	0.848±0.089	0.895±0.079	0.864±0.055	0.853±0.102	**0.924±0.075**
Ionosphere	**0.969±0.010**	0.940±0.022	0.933±0.020	0.927±0.024	0.930±0.026	0.935±0.028	0.924±0.021	0.937±0.031	0.931±0.023	0.940±0.016	0.963±0.013
Letter	0.943±0.002	0.950±0.002	0.953±0.001	**0.956±0.003**	0.954±0.001	0.951±0.001	0.949±0.002	0.948±0.005	0.950±0.001	0.955±0.003	0.951±0.003
Libras	**0.956±0.012**	0.940±0.018	0.941±0.011	0.941±0.012	0.950±0.014	0.949±0.012	0.944±0.012	0.945±0.013	0.941±0.015	0.955±0.012	0.947±0.015
Muskroom	0.906±0.024	**0.914±0.020**	0.904±0.035	0.913±0.017	0.881±0.021	0.897±0.024	0.886±0.027	0.912±0.024	0.898±0.027	0.907±0.028	0.913±0.018
Optdigits	0.994±0.001	0.992±0.001	0.992±0.002	0.991±0.002	0.992±0.002	0.991±0.001	0.992±0.001	0.991±0.001	0.992±0.001	0.994±0.001	**0.995±0.001**
Page_small	**0.891±0.032**	0.806±0.043	0.824±0.038	0.838±0.026	0.849±0.027	0.845±0.029	0.848±0.031	0.834±0.027	0.841±0.035	0.859±0.024	0.858±0.028
Parkinsons	**0.889±0.039**	0.852±0.045	0.819±0.057	0.825±0.057	0.858±0.030	0.862±0.046	0.848±0.044	0.837±0.049	0.853±0.041	0.853±0.041	0.857±0.038
Penbase	0.988±0.002	0.985±0.001	0.978±0.001	0.978±0.002	0.980±0.002	0.981±0.005	0.985±0.006	0.988±0.001	0.985±0.001	0.987±0.001	**0.989±0.002**
Ring	0.987±0.002	0.987±0.001	0.987±0.002	0.987±0.001	0.989±0.002	0.990±0.001	0.990±0.002	0.986±0.002	0.994±0.001	**0.993±0.001**	0.987±0.003
Satimage	0.949±0.002	**0.955±0.002**	0.954±0.002	0.952±0.003	0.954±0.002	0.954±0.003	0.954±0.003	0.952±0.003	0.954±0.003	0.954±0.003	0.950±0.001
Segment	**0.989±0.002**	0.984±0.003	0.983±0.004	0.986±0.002	0.985±0.003	0.985±0.002	0.984±0.003	0.987±0.004	0.985±0.002	0.984±0.003	0.988±0.003
Sonar	**0.849±0.046**	0.736±0.041	0.789±0.045	0.803±0.073	0.811±0.056	0.806±0.047	0.811±0.049	0.833±0.075	0.817±0.046	0.825±0.049	0.842±0.040
Spambase	**0.957±0.008**	0.807±0.008	0.897±0.007	0.897±0.007	0.891±0.007	0.900±0.005	0.912±0.008	0.902±0.009	0.896±0.007	0.921±0.009	0.938±0.019
Spectf	0.826±0.040	0.761±0.064	0.770±0.070	**0.864±0.042**	0.757±0.076	0.731±0.055	0.737±0.052	0.742±0.039	0.731±0.055	0.768±0.025	0.757±0.033
Texture	**0.997±0.000**	0.961±0.002	0.960±0.002	0.961±0.002	0.968±0.001	0.967±0.003	0.967±0.002	0.963±0.004	0.967±0.002	0.975±0.004	0.967±0.001
Thyroid	**0.796±0.009**	0.732±0.008	0.747±0.018	0.723±0.007	0.739±0.013	0.731±0.009	0.751±0.011	0.763±0.017	0.794±0.018	0.744±0.015	0.778±0.001
Twonorm	0.996±0.001	0.993±0.002	**0.997±0.000**	0.994±0.001	0.994±0.002	0.994±0.001	0.994±0.001	0.995±0.001	0.994±0.000	**0.997±0.000**	0.996±0.001
Wdbc	**0.981±0.011**	0.964±0.005	0.942±0.010	0.959±0.013	0.963±0.011	0.960±0.011	0.962±0.012	0.968±0.016	0.964±0.009	**0.981±0.013**	0.961±0.010
Wine	**0.999±0.001**	**0.999±0.001**	**0.999±0.001**	**0.999±0.001**	**0.999±0.001**	0.992±0.006	0.998±0.001	**0.999±0.001**	0.998±0.002	0.995±0.004	**0.999±0.001**
WineQR	0.729±0.026	0.698±0.038	0.686±0.040	0.702±0.016	0.712±0.028	0.676±0.042	0.703±0.046	0.711±0.018	0.716±0.040	0.719±0.035	**0.749±0.020**
WineQW	0.737±0.031	0.667±0.053	0.679±0.039	0.656±0.037	0.712±0.029	0.726±0.025	0.693±0.028	0.689±0.027	0.633±0.023	0.707±0.019	**0.744±0.033**
Average	**0.881±0.020**	0.844±0.024	0.853±0.027	0.857±0.021	0.856±0.025	0.857±0.026	0.858±0.024	0.861±0.026	0.858±0.029	0.865±0.026	0.875±0.019

Table 4 Comparison of the testing PMSEs corresponding to AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network.

Data set	AG-NBC	NBC	FNBC	TAN	GRFWNB	AODE	WAODE	ICA-NBC	HNB	CFW	RVFL
Arrhythmia	0.572±0.051	0.633±0.037	0.627±0.042	0.602±0.037	0.602±0.045	0.589±0.067	0.582±0.059	0.572±0.042	0.594±0.056	0.607±0.078	**0.561±0.042**
Australian	0.251±0.012	**0.218±0.023**	**0.218±0.026**	0.308±0.056	0.316±0.047	0.315±0.058	0.319±0.028	0.272±0.048	0.310±0.047	0.223±0.031	0.287±0.013
Bp	**0.335±0.039**	0.535±0.103	0.559±0.109	0.471±0.042	0.510±0.081	0.502±0.068	0.522±0.049	0.428±0.078	0.485±0.092	0.387±0.078	0.479±0.033
Cleveland	0.529±0.017	0.621±0.040	0.532±0.043	0.566±0.051	0.551±0.054	0.547±0.040	**0.523±0.047**	0.542±0.047	0.561±0.041	0.538±0.046	0.558±0.016
Coil2000	**0.109±0.005**	0.212±0.011	0.209±0.014	0.189±0.007	0.155±0.004	0.167±0.005	0.166±0.007	0.223±0.010	0.171±0.012	0.162±0.009	0.234±0.004
Contraceptive	**0.577±0.006**	0.633±0.022	0.658±0.020	0.601±0.008	0.649±0.011	0.645±0.024	0.642±0.021	0.644±0.029	0.655±0.011	0.637±0.018	0.603±0.007
Glass	0.685±0.016	0.620±0.040	0.597±0.031	**0.596±0.053**	0.606±0.047	0.612±0.033	0.642±0.042	0.633±0.019	0.615±0.031	0.620±0.071	0.647±0.018
Heart	0.287±0.022	0.320±0.049	0.326±0.050	**0.279±0.060**	0.305±0.082	0.287±0.046	0.293±0.052	0.316±0.049	0.303±0.061	0.300±0.061	0.296±0.023
Hepatitis	**0.237±0.087**	0.245±0.100	0.276±0.131	0.271±0.127	0.285±0.131	0.299±0.147	0.263±0.134	0.256±0.129	0.278±0.061	0.265±0.107	0.248±0.089
Ionosphere	**0.179±0.019**	0.200±0.037	0.194±0.040	0.188±0.038	0.197±0.067	0.216±0.053	0.202±0.047	0.189±0.037	0.196±0.061	0.197±0.043	0.205±0.018
Letter	0.526±0.001	0.534±0.005	**0.541±0.003**	0.533±0.006	0.536±0.005	0.532±0.006	0.537±0.003	0.523±0.009	0.537±0.007	0.529±0.004	0.522±0.005
Libras	0.634±0.015	0.739±0.077	**0.745±0.075**	0.684±0.073	0.693±0.072	0.671±0.073	0.683±0.069	0.703±0.072	0.709±0.067	0.693±0.083	0.689±0.016
Muskroom	**0.251±0.028**	0.295±0.052	0.278±0.080	0.268±0.039	0.319±0.053	0.274±0.061	0.281±0.047	0.273±0.056	0.282±0.050	0.297±0.040	0.278±0.030
Optdigits	**0.149±0.002**	0.164±0.012	0.157±0.011	0.166±0.012	0.162±0.012	0.160±0.010	0.161±0.014	0.159±0.016	0.171±0.011	0.163±0.014	0.165±0.003
Page_small	**0.114±0.018**	0.150±0.029	0.133±0.023	0.137±0.033	0.126±0.020	0.127±0.021	0.128±0.027	0.147±0.017	0.121±0.023	0.117±0.039	0.117±0.016
Parkinsons	**0.230±0.056**	0.482±0.084	0.502±0.086	0.497±0.117	0.502±0.056	0.472±0.101	0.468±0.142	0.589±0.049	0.493±0.100	0.472±0.099	0.436±0.049
Penbase	**0.227±0.009**	0.329±0.008	0.271±0.007	0.267±0.014	0.254±0.009	0.248±0.008	0.257±0.010	0.301±0.035	0.240±0.006	0.239±0.008	0.238±0.010
Ring	0.165±0.007	0.078±0.004	0.080±0.004	0.078±0.005	0.058±0.006	0.065±0.005	**0.044±0.004**	0.059±0.006	0.055±0.003	0.063±0.004	0.114±0.008
Satimage	0.394±0.002	0.384±0.012	0.390±0.014	0.387±0.013	0.391±0.010	0.383±0.018	0.397±0.018	0.385±0.016	0.382±0.015	**0.380±0.011**	0.403±0.004
Segment	0.167±0.011	0.154±0.015	0.157±0.015	0.149±0.013	0.147±0.012	0.154±0.019	0.157±0.016	0.149±0.011	0.155±0.017	**0.143±0.010**	0.148±0.015
Sonar	**0.316±0.052**	0.536±0.078	0.575±0.063	0.460±0.121	0.445±0.100	0.487±0.086	0.491±0.073	0.504±0.069	0.456±0.102	0.406±0.077	0.445±0.055
Spambase	**0.187±0.011**	0.336±0.008	0.322±0.013	0.284±0.012	0.289±0.008	0.252±0.006	0.267±0.008	0.284±0.007	0.266±0.010	0.278±0.005	0.265±0.017
Spectf	**0.281±0.063**	0.546±0.081	0.520±0.077	0.623±0.114	0.517±0.113	0.437±0.055	0.443±0.072	0.335±0.067	0.304±0.030	0.379±0.059	0.393±0.059
Texture	0.384±0.013	0.383±0.010	0.373±0.014	0.388±0.009	0.374±0.015	0.378±0.006	0.334±0.009	**0.293±0.012**	0.371±0.013	0.362±0.010	0.388±0.012
Thyroid	0.174±0.004	0.132±0.006	0.131±0.005	0.124±0.005	0.126±0.005	0.129±0.010	0.123±0.006	**0.122±0.007**	0.121±0.007	0.129±0.007	0.128±0.005
Twonorm	0.114±0.003	0.052±0.006	0.052±0.004	0.053±0.003	0.048±0.004	0.046±0.003	0.057±0.006	0.061±0.003	0.043±0.003	**0.041±0.002**	0.113±0.004
Wdbc	**0.128±0.017**	0.183±0.028	0.233±0.036	0.219±0.033	0.181±0.037	0.143±0.035	0.154±0.038	0.163±0.027	0.194±0.033	0.136±0.047	0.139±0.019
Wine	0.191±0.023	**0.044±0.028**	0.046±0.035	0.051±0.015	0.068±0.023	0.106±0.046	0.089±0.044	0.078±0.021	0.074±0.022	0.057±0.012	0.142±0.025
WineQR	0.666±0.004	0.593±0.021	0.597±0.012	0.580±0.016	0.583±0.024	0.605±0.012	0.604±0.026	0.613±0.027	0.577±0.034	**0.573±0.021**	0.609±0.008
WineQW	0.656±0.021	0.585±0.048	0.594±0.031	0.602±0.037	**0.583±0.038**	0.588±0.029	0.607±0.034	0.619±0.047	0.596±0.023	0.585±0.033	0.637±0.027
Average	**0.324±0.021**	0.365±0.036	0.361±0.037	0.354±0.036	0.352±0.066	0.347±0.038	0.348±0.038	0.349±0.036	0.344±0.035	0.333±0.038	0.350±0.022

Table 5 Statistical analysis results on win\tie\lose corresponding to comparisons in Tables 2–4.

Win\tie\lose	AG-NBC vs. NBC	AG-NBC vs. FNBC	AG-NBC vs. TAN	AG-NBC vs. GRAWNB	AG-NBC vs. AODE
Accuracy	30\0\0	26\1\3	26\0\4	28\0\2	27\0\3
AUC	24\2\4	24\2\4	23\2\5	26\1\3	23\2\5
PMSE	19\0\11	18\0\12	21\0\9	19\0\11	20\1\9
Win\tie\lose	AG-NBC vs. WAODE	AG-NBC vs. ICA-NBC	AG-NBC vs. HNB	AG-NBC vs. CFW	AG-NBC vs. RVFL
Accuracy	26\0\4	27\1\2	29\0\1	28\0\2	25\1\4
AUC	25\0\5	21\2\7	24\0\6	21\2\7	17\2\11
PMSE	20\0\10	18\1\11	21\0\9	19\0\11	20\0\10

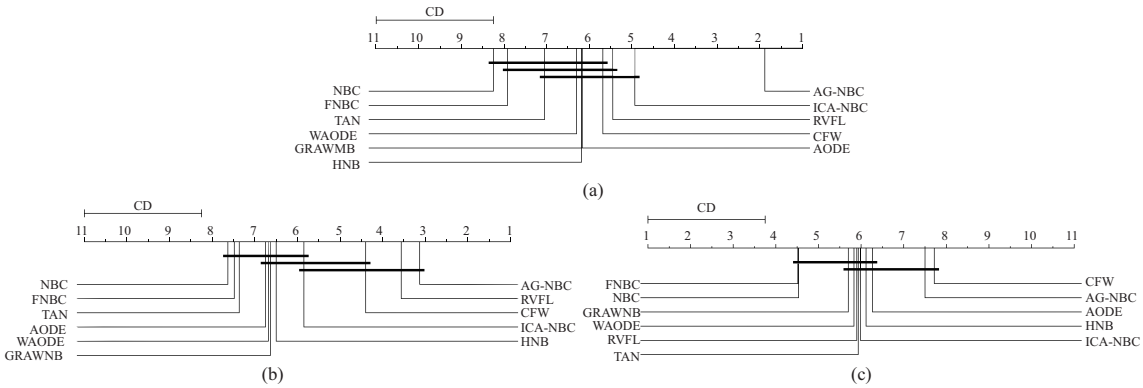

Figure 7 CD diagrams corresponding to comparisons in Tables 2–4. (a) CD diagram of accuracy; (b) CD diagram of AUC; (c) CD diagram of PMSE.

Figure 8 (Color online) Comparison of the training times corresponding to AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and RVFL network. (a) Training time on muskroom data set; (b) training time on segment data set.

than NBC, FNBC, GRAWNB, ICA-NBC, and CFW. We can find that TAN, AODE, WAODE, HNB, and RVFL all use additional structures to model the unknown joint probabilities, while NBC, FNBC, GRAWNB, ICA-NBC, and CFW directly calculate the joint probabilities with the products of marginal probabilities. The usage of additional model structures increases the classification uncertainty and further reduces the probability estimation quality.

In order to further demonstrate the efficiency of AG-NBC, we provide an additional experiment to compare the training times of AG-NBC, NBC, FNBC, TAN, GRAWNB, AODE, WAODE, ICA-NBC, HNB, CFW, and the RVFL network on two example data sets. The comparative results of training times on the muskroom and segment data sets are presented in Figure 8. We can see that training AG-NBC is faster than ICA-NBC, HNB, and CFW on muskroom, and than ICA-NBC and HNB on segment. In particular, AG-NBC shows the competitive advantage on training time with feature extraction-based and feature weighting-based NBCs on the high-dimensional muskroom data set. It indicates that AG-NBC has the potential to handle the high-dimensional classification problem. In fact, the generation process of DAGs terminates quickly in a small number of iterations. This conclusion is also supported by the

Table 6 Comparison of the testing accuracies between random attribute groups and DAGs when constructing NBC.

Data set	DAGs	Partition 1	Partition 2	Partition 3	Partition 4	Partition 5	Partition 6	Partition 7	Partition 8	Average	AG-NBC
Arrhythmia	(127,21,6,17,108)	0.597±0.028	0.618±0.026	0.631±0.027	0.628±0.029	0.584±0.038	0.599±0.029	0.616±0.028	0.614±0.029	0.611±0.029	0.625±0.026
Australian	(5,9)	0.850±0.025	0.870±0.024	0.851±0.027	0.875±0.021	0.856±0.023	0.842±0.026	0.837±0.026	0.848±0.021	0.854±0.024	0.865±0.024
Bp	(6,6,21)	0.733±0.027	0.750±0.025	0.833±0.028	0.800±0.037	0.767±0.026	0.778±0.055	0.755±0.047	0.777±0.040	0.774±0.034	0.814±0.022
Cleveland	(7,3,3)	0.541±0.036	0.568±0.061	0.573±0.054	0.540±0.043	0.568±0.046	0.550±0.033	0.580±0.044	0.556±0.046	0.560±0.045	0.595±0.021
Coil2000	(15,6,34,30)	0.927±0.007	0.931±0.009	0.937±0.010	0.940±0.004	0.919±0.013	0.929±0.006	0.926±0.009	0.931±0.008	0.930±0.008	0.932±0.004
Contraceptive	(3,6)	0.473±0.018	0.477±0.019	0.492±0.014	0.519±0.011	0.484±0.014	0.513±0.018	0.472±0.015	0.495±0.016	0.491±0.016	0.503±0.017
Glass	(3,7)	0.606±0.026	0.631±0.067	0.579±0.069	0.608±0.043	0.626±0.058	0.625±0.043	0.611±0.077	0.600±0.060	0.610±0.055	0.623±0.056
Heart	(2,3,8)	0.814±0.026	0.810±0.042	0.814±0.028	0.821±0.042	0.805±0.042	0.810±0.049	0.813±0.021	0.820±0.025	0.813±0.026	0.819±0.036
Hepatitis	(4,11,4)	0.867±0.058	0.891±0.056	0.879±0.073	0.879±0.054	0.861±0.081	0.883±0.058	0.850±0.033	0.846±0.067	0.870±0.060	0.875±0.034
Ionosphere	(22,4,6)	0.883±0.027	0.891±0.031	0.890±0.024	0.870±0.058	0.875±0.015	0.889±0.034	0.865±0.046	0.862±0.033	0.878±0.035	0.899±0.030
Letter	(5,7,4)	0.602±0.008	0.583±0.006	0.604±0.014	0.591±0.011	0.601±0.012	0.589±0.015	0.585±0.006	0.597±0.009	0.594±0.010	0.600±0.007
Libras	(12,20,17,18,23)	0.683±0.062	0.693±0.078	0.687±0.071	0.702±0.067	0.706±0.070	0.697±0.068	0.674±0.061	0.714±0.059	0.695±0.067	0.705±0.075
Muskroom	(34,31,23,38,40)	0.823±0.051	0.806±0.047	0.816±0.058	0.821±0.069	0.802±0.058	0.807±0.054	0.811±0.049	0.804±0.057	0.811±0.055	0.817±0.045
Optdigits	(30,20,4,10)	0.927±0.008	0.931±0.004	0.929±0.006	0.926±0.007	0.935±0.006	0.923±0.009	0.941±0.003	0.926±0.003	0.930±0.006	0.932±0.006
Page_small	(2,8)	0.925±0.008	0.916±0.018	0.921±0.018	0.915±0.016	0.917±0.016	0.921±0.016	0.916±0.018	0.924±0.011	0.919±0.015	0.930±0.015
Parkinsons	(3,3,11,5)	0.868±0.042	0.836±0.048	0.859±0.028	0.832±0.041	0.832±0.047	0.857±0.028	0.861±0.031	0.851±0.033	0.850±0.037	0.853±0.035
Penbase	(5,6,5)	0.865±0.012	0.861±0.006	0.868±0.011	0.866±0.012	0.867±0.011	0.857±0.012	0.864±0.014	0.858±0.016	0.863±0.012	0.869±0.009
Ring	(4,4,12)	0.919±0.007	0.925±0.007	0.925±0.005	0.927±0.004	0.938±0.006	0.927±0.005	0.928±0.005	0.933±0.006	0.928±0.006	0.955±0.002
Satimage	(8,8,9,11)	0.772±0.009	0.798±0.010	0.781±0.018	0.783±0.016	0.804±0.015	0.817±0.013	0.829±0.019	0.805±0.016	0.798±0.015	0.826±0.015
Segment	(1,2,16)	0.886±0.014	0.883±0.017	0.879±0.012	0.909±0.015	0.915±0.016	0.937±0.008	0.949±0.011	0.951±0.012	0.913±0.013	0.969±0.007
Sonar	(25,24,4,7)	0.779±0.045	0.771±0.040	0.737±0.054	0.752±0.025	0.765±0.032	0.784±0.019	0.775±0.027	0.769±0.045	0.767±0.036	0.771±0.053
Spambase	(41,8,8)	0.886±0.010	0.880±0.011	0.879±0.012	0.883±0.013	0.876±0.011	0.878±0.012	0.881±0.007	0.883±0.009	0.881±0.010	0.886±0.008
Spectf	(9,9,14,12)	0.789±0.045	0.815±0.037	0.781±0.044	0.809±0.037	0.799±0.037	0.806±0.032	0.778±0.032	0.795±0.041	0.797±0.038	0.803±0.033
Texture	(4,12,4,17)	0.935±0.006	0.943±0.006	0.930±0.005	0.938±0.003	0.941±0.003	0.937±0.006	0.935±0.007	0.945±0.002	0.938±0.005	0.946±0.004
Thyroid	(3,3,13)	0.929±0.004	0.937±0.005	0.931±0.006	0.931±0.004	0.932±0.003	0.927±0.008	0.931±0.008	0.936±0.003	0.932±0.005	0.935±0.005
Twonorm	(2,4,14)	0.951±0.004	0.948±0.003	0.950±0.002	0.975±0.004	0.956±0.003	0.969±0.003	0.970±0.003	0.959±0.001	0.960±0.003	0.969±0.004
Wdbc	(4,3,14,9)	0.914±0.021	0.941±0.014	0.949±0.013	0.950±0.015	0.923±0.020	0.919±0.012	0.921±0.012	0.927±0.013	0.931±0.015	0.925±0.027
Wine	(3,10)	0.991±0.012	0.959±0.020	0.980±0.013	0.969±0.019	0.961±0.037	0.989±0.012	0.983±0.013	0.987±0.012	0.977±0.017	0.979±0.024
WineQR	(4,7)	0.577±0.013	0.579±0.016	0.583±0.021	0.566±0.020	0.568±0.021	0.581±0.022	0.565±0.027	0.561±0.018	0.573±0.020	0.574±0.026
WineQW	(5,6)	0.511±0.035	0.507±0.054	0.529±0.030	0.522±0.026	0.509±0.033	0.530±0.030	0.518±0.035	0.532±0.036	0.520±0.035	0.534±0.029

feasibility experiments presented in Figures 1–4. We can see that Algorithm 1 reached convergence in about 8 iterations. Above all, the experimental results and statistical analysis show that AG-NBC can obtain higher testing accuracy with a lower misclassification cost and a higher probability estimation quality. The experimental results demonstrate that the construction of NBC by using the proposed attribute grouping strategy is effective to relieve the attribute independence assumption of NBC without substantially modifying its simple model structure.

Finally, an additional experiment is conducted to compare the prediction performances of random attribute groups (RAGs) and DAGs to construct an NBC. The main purpose of this experiment is to demonstrate the utility of DAGs while also providing a feasible direction for future research on AG-NBC construction. Based on the optimal number of attribute groups and number of attributes in each attribute group, a random partition of $\mathcal{D}$ condition attributes into $\mathcal{T}$ attribute groups was prepared for NBC training. For each data set, RAGs-based NBC was repeatedly trained eight times based on the same data partition as for the DAGs-based NBC training mentioned above. The experimental results are summarized in Table 6. It is observed that (1) DAGs-based NBC obtains better prediction performances than RAGs-based NBC in terms of average testing accuracy and (2) attribute groups that are more efficient than the DAGs proposed in this work do exist. The experimental results not only validate the effectiveness of DAGs determined with Algorithm 1 by optimizing the objective function shown in (17) but also suggest that forming more attribute groups can help NBC in obtaining better prediction performances than DAGs.

5 Conclusion and future work

In this study, an AG-NBC was proposed to alleviate the attribute independence assumption of the NBC. An effective objective function was designed, which considers both sample classification and attribute grouping to determine optimal DAGs. Dependencies among attributes of the same DAG were modeled using the RVFL network, which has fast training speed and accurate dependence expression. The outputs from different RVFL networks are directly multiplied together as the posterior probability of NBC. Our findings showed that AG-NBC has better classification performance than the other studied state-of-the-art Bayesian classifiers and RVFL network trained on the original data set.

As future work, the authors plan to (1) explore more efficient attribute grouping strategies to construct improved NBC, (2) implement AG-NBC on the random sample partition [50] data representation model so as to deal with large-scale classification problems in a distributed environment, (3) use the joint probability density function estimation technique [22] to handle attribute dependence within each DAG to construct a semi-naive Bayesian classifier with low computation complexity and simple model structure, and (4) use the proposed method to deal with practical high-dimensional text classification and sentiment analysis problems.

Acknowledgements This paper was supported by National Natural Science Foundation of China (Grant No. 61972261), Natural Science Foundation of Guangdong Province (Grant No. 2314050006683), Key Basic Research Foundation of Shenzhen (Grant Nos. JCYJ2022081810, 0205012), and Basic Research Foundation of Shenzhen (Grant No. JCYJ20210324093609026).

References

1 Bishop C. Pattern Recognition and Machine Learning. Berlin: Springer, 2007
2 Wu X, Kumar V, Quinlan J R, et al. Top 10 algorithms in data mining. Knowl Inf Syst, 2008, 14: 1–37
3 Rish I. An empirical study of the naive Bayes classifier. In: Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 2001. 41–46
4 Ting S L, Ip W H, Tsang A H. Is naive Bayes a good classifier for document classification. Int J Softw Eng Appl, 2011, 5: 37–46
5 McCann S, Lowe D G. Local naive Bayes nearest neighbor for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012. 3650–3656
6 Mukherjee S, Sharma N. Intrusion detection using naive Bayes classifier with feature reduction. Procedia Tech, 2012, 4: 119–128
7 Liu B, Blasch E, Chen Y, et al. Scalable sentiment classification for big data analysis using naive Bayes classifier. In: Proceedings of the IEEE International Conference on Big Data, 2013. 99–104
8 Prabhat A, Khullar V. Sentiment classification on big data using naive Bayes and logistic regression. In: Proceedings of the International Conference on Computer Communication and Informatics, 2017. 1–5
9 Sun N, Sun B, Lin J D, et al. Lossless pruned naive Bayes for big data classifications. Big Data Res, 2018, 14: 27–36
10 Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn, 1992, 9: 309–347
11 Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn, 1997, 29: 131–163
12 Keogh E J, Pazzani M J. Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. In: Proceedings of the International Workshop on Artificial Intelligence and Statistics, Florida, 1999. 1–6
13 Webb G I, Boughton J R, Wang Z. Not so naive Bayes: aggregating one-dependence estimators. Mach Learn, 2005, 58: 5–24

14 Jiang L X, Zhang H, Cai Z H, et al. Weighted average of one-dependence estimators. J Exp Theor Artif Intell, 2012, 24: 219–230
15 Pernkopf F, Wohlmayr M, Tschiatschek S. Maximum margin Bayesian network classifiers. IEEE Trans Pattern Anal Mach Intell, 2012, 34: 521–532
16 Fan X, Yuan C. An improved lower bound for Bayesian network structure learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2015
17 Jiang L X, Zhang H, Cai H Z H. A novel Bayes model: hidden naive Bayes. IEEE Trans Knowl Data Eng, 2009, 21: 1361–1371
18 Jiang L X, Zhang L G, Li C Q, et al. A correlation-based feature weighting filter for naive Bayes. IEEE Trans Knowl Data Eng, 2019, 31: 201–213
19 Jiang L X, Zhang L G, Yu L J, et al. Class-specific attribute weighted naive Bayes. Pattern Recogn, 2019, 88: 321–330
20 Pérez A, Larrañaga P, Inza I. Bayesian classifiers based on kernel density estimation: flexible classifiers. Int J Approx Reason, 2009, 50: 341–362
21 John G, Langley P. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Quebec, 1995. 1–8
22 Wang X Z, He Y L, Wang D D. Non-naive Bayesian classifiers for classification problems with continuous attributes. IEEE Trans Cybern, 2014, 44: 21–39
23 He Y L, Wang R, Kwong S, et al. Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis. Inform Sci, 2014, 259: 252–268
24 Geng Z Q, Meng Q C, Bai J, et al. A model-free Bayesian classifier. Inform Sci, 2019, 482: 171–188
25 Bartlett M, Cussens J. Integer linear programming for the Bayesian network structure learning problem. Artif Intell, 2017, 244: 258–271
26 Gao T, Fadnis K, Campbell M. Local-to-global Bayesian network structure learning. In: Proceedings of the International Conference on Machine Learning, 2017. 1193–1202
27 Liu Y, Wang L M, Mammadov M. Learning semi-lazy Bayesian network classifier under the c.i.i.d assumption. Knowl-Based Syst, 2020, 208: 106422
28 Wang L M, Zhao H Y. Learning a flexible K-dependence Bayesian classifier from the chain rule of joint probability distribution. Entropy, 2015, 17: 3766–3786
29 Wang L M, Zhang X H, Li K, et al. Semi-supervised learning for K-dependence Bayesian classifiers. Appl Intell, 2022, 52: 3604–3622
30 Webb G I, Boughton J R, Zheng F, et al. Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive Bayesian classification. Mach Learn, 2012, 86: 233–272
31 Wong T T, Tsai H C. Multinomial naïve Bayesian classifier with generalized Dirichlet priors for high-dimensional imbalanced data. Knowl-Based Syst, 2021, 228: 107288
32 Zhang H, Sheng S L. Learning weighted naive Bayes with accurate ranking. In: Proceedings of the 4th IEEE International Conference on Data Mining, 2004. 567–570
33 Bressan M, Vitria J. Improving naive Bayes using class-conditional ICA. In: Proceedings of the Ibero-American Conference on Artificial Intelligence, 2002. 1–10
34 Bressan M, Guillamet D, Vitria J. Using an ICA representation of high dimensional data for object recognition and classification. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. 1004–1009
35 Qin F, Ren S L, Cheng Z K, et al. Bayes classification model based on ICA. Comput Eng Des, 2007, 28: 4873–4877
36 Fan L, Poh K L. A comparative study of PCA, ICA and class-conditional ICA for naive Bayes classifier. In: Proceedings of the International Work-Conference on Artificial Neural Networks, 2007. 16–22
37 Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometr Intell Lab Syst, 1987, 2: 37–52
38 Comon P. Independent component analysis, a new concept? Signal Process, 1994, 36: 287–314
39 Jayanthi S K, Sasikala S. Naive Bayesian classifier and PCA for web link spam detection. Comput Sci Telecommun, 2014, 41: 3–15
40 Zhang B, Liu Z Y, Jia Y G, et al. Network intrusion detection method based on PCA and Bayes algorithm. Secur Commun Netw, 2018, 2018: 1–11
41 Parzen E. On estimation of a probability density function and mode. Ann Math Statist, 1962, 33: 1065–1076
42 Chen X J, Ye Y M, Xu X F, et al. A feature group weighting method for subspace clustering of high-dimensional data. Pattern Recogn, 2012, 45: 434–446
43 Gan G, Ng M K P. Subspace clustering with automatic feature grouping. Pattern Recogn, 2015, 48: 3703–3713
44 Igelnik B, Pao Y H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE Trans Neural Netw, 1995, 6: 1320–1329
45 Zhang L, Suganthan P N. A survey of randomized algorithms for training neural networks. Inform Sci, 2016, 364: 146–155
46 Zhang L, Suganthan P N. A comprehensive evaluation of random vector functional link networks. Inform Sci, 2016, 367: 1094–1105
47 Alcala-Fdez J, Fernandez A, Luengo J, et al. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multi-Valued Logic Soft Comput, 2011, 17: 255–287
48 Hand D J, Till R J. A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach Learn, 2001, 45: 171–186
49 Demar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res, 2006, 7: 1–30
50 Salloum S, Huang J Z, He Y. Random sample partition: a distributed data model for big data analysis. IEEE Trans Ind Inf, 2019, 15: 5846–5854

Appendix A

Table A1 Acronyms.

Abbreviation	Full name
NBC	Naive Bayesian classifier
AG-NBC	Attribute grouping-based naive Bayesian classifier
FNBC	Flexible naive Bayesian classifier
TAN	Tree augmented Bayes network
SP-TAN	Super parent-based tree augmented Bayes network
GRAWNB	Gain ratio-based attribute weighted naive Bayes
AODE	Averaged one-dependence estimator
WAODE	Weighted averaged one-dependence estimator
ICA-NBC	Independent component analysis-based naive Bayesian classifier
HNB	Hidden naive Bayesian
CFW	Correlation-based feature weighting filter for naive Bayes
AUC	Area under the receiver operating characteristic curve
PMSE	Probability mean square error
p.d.f.	Probability density function
KBN	Kernel-based Bayesian network
NNBC	Non-naive Bayesian classifier
PCA	Principal component analysis
ICA	Independent component analysis
CC-ICA	Class-conditional independent component analysis
AG	Attribute group
DAG	Dependent attribute group
RVFL	Random vector functional link
SMD	Sample membership degree
SCC	Sample class center
ACD	Attribute contribution degree
GCD	Group contribution degree
GID	Group importance degree
AMD	Attribute membership degree
CD	Critical difference

Table A2 Notations.

Mathematical symbol	Meaning
$\mathbb{D}$	Classification data set
$\mathcal{N}$	Number of samples in $\mathbb{D}$
$\mathcal{D}$	Number of sample's condition attributes
$\mathcal{M}$	Number of classes in $\mathbb{D}$
$\mathbb{D}^{(m)}$	Sample set corresponding to the m-th class
$\mathcal{N}_m$	Number of samples of the m-th class.
A_d	the d-th Condition attribute of data set $\mathbb{D}$
$\{w_1, w_2, \ldots, w_{\mathcal{M}}\}$	Class set of $\mathbb{D}$
$x_n^{(m)} = (x_{n1}^{(m)}, x_{n2}^{(m)}, \ldots, x_{n\mathcal{D}}^{(m)})$	The n-th sample belonging to the m-th class
$y_n^{(m)}$	Class label of $x_n^{(m)}$
$x_{nd}^{(m)}$	The d-th condition attribute value of $x_n^{(m)}$
$x = (x_1, x_2, \ldots, x_{\mathcal{D}})$	New sample with $\mathcal{D}$ condition attributes
y	Class label of x
$P(w_m \mid x)$	Posterior probability
$P(w_m)$	Prior probability
$P(x \mid w_m)$	Class-conditional probability
$P(x)$	Total probability
h	Bandwidth of kernel density estimator
$\mathbb{I}(\cdot)$	Indicator function
$\mathrm{Par}(x_d)$	Parent node set of the d-th condition attribute
α_{di}	Dependence weight between the i-th and d-th condition attribute
$I(x_d, x_i \mid w_m)$	Conditional mutual information
G_t	The t-th dependent attribute group
$\mathcal{D}_t$	Number of condition attributes in the t-th DAG
$\mathcal{T}$	Number of DAGs
$A_d^{(t)}$	The d-th condition attribute in the t-th DAG
$L(U, V, R, G)$	Objective function to obtain the optimal DAGs
$P = (p_{nm})_{\mathcal{N}\times\mathcal{M}}$	Sample membership degree matrix
$Z = (z_{md})_{\mathcal{M}\times\mathcal{D}}$	Sample class center matrix
$U = (u_{md})_{\mathcal{M}\times\mathcal{D}}$	Attribute contribution degree matrix
$V = (v_{mt})_{\mathcal{M}\times\mathcal{T}}$	Group contribution degree matrix
$R = (r_{mt})_{\mathcal{M}\times\mathcal{T}}$	Group importance degree matrix
$G = (g_{dt})_{\mathcal{D}\times\mathcal{T}}$	Attribute membership degree matrix
ε_1, ε_2	Regularization factors
$\mathbb{D}_t$	Sample set corresponding to DAG G_t
$x_n^{[t]} = (x_{n1}^{[t]}, x_{n2}^{[t]}, \ldots, x_{n\mathcal{D}_t}^{[t]})$	Input of RVFL network corresponding to the n-th sample in $\mathbb{D}_t$
$y_n = (y_{n1}, y_{n2}, \ldots, y_{n\mathcal{M}})$	Output of RVFL network corresponding to the n-th sample in $\mathbb{D}_t$
RVFL_t	RVFL network trained based on $\mathbb{D}_t$
$\mathcal{L}$	Number of hidden-layer nodes in RVFL_t
$W^{(t)} = (\alpha_{dl}^{(t)})_{\mathcal{D}_t\times\mathcal{L}}$	Input-layer weight matrix of RVFL_t
$V^{(t)} = (\beta_{lm}^{(t)})_{\mathcal{L}\times\mathcal{M}}$	Output-layer weight matrix of RVFL_t
$s_l^{(t)}$	The l-th hidden-layer node input of RVFL_t
$h_l^{(t)}$	The l-th hidden-layer node output of RVFL_t
$P_n^{(\mathrm{NBC})}$	Posterior probability calculated by NBC for the n testing sample
$P_n^{(\mathrm{AG\text{-}NBC})}$	Posterior probability calculated by AG-NBC for the n testing sample
$p_i^{(\mathrm{NBC})}(\cdot)$	Probability density function corresponding to NBC's posterior probabilities
$p_i^{(\mathrm{AG\text{-}NBC})}(\cdot)$	Probability density function corresponding to AG-NBC's posterior probabilities
$\mu_i^{(\mathrm{NBC})}$	Mean of NBC's probability density function
$\mu_i^{(\mathrm{AG\text{-}NBC})}$	Mean of AG-NBC's probability density function
$\sigma_i^{(\mathrm{NBC})}$	Standard deviation of NBC's probability density function
$\sigma_i^{(\mathrm{AG\text{-}NBC})}$	Standard deviation of AG-NBC's probability density function
$\mathrm{Risk}_{ij}^{(\mathrm{NBC})}$	Classification risk of NBC between the i-th and j-th classes
$\mathrm{Risk}_{ij}^{(\mathrm{AG\text{-}NBC})}$	Classification risk of AG-NBC between the i-th and j-th classes