SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2022, Vol. 65 222101:1–222101:14
https://doi.org/10.1007/s11432-020-3277-0

Attribute augmented and weighted naive Bayes

Huan ZHANG[1], Liangxiao JIANG[1,2]* & Chaoqun LI[3]

[1]*School of Computer Science, China University of Geosciences, Wuhan* 430074*, China;*
[2]*Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai,* 200240*, China;*
[3]*School of Mathematics and Physics, China University of Geosciences, Wuhan* 430074*, China*

Received 5 November 2020/Revised 6 February 2021/Accepted 13 May 2021/Published online 17 November 2022

Abstract Numerous enhancements have been proposed to mitigate the attribute conditional independence assumption in naive Bayes (NB). However, almost all of them only focus on the original attribute space. Due to the complexity of real-world applications, we argue that the discriminative information provided by the original attribute space might be insufficient for classification. Thus, in this study, we expect to discover some latent attributes beyond the original attribute space and propose a novel two-stage model called attribute augmented and weighted naive Bayes (A^2WNB). At the first stage, we build multiple random one-dependence estimators (RODEs). Then we use each built RODE to classify each training instance in turn and define the predicted class labels as its latent attributes. At last, we construct the augmented attributes by concatenating the latent attributes with the original attributes. At the second stage, to alleviate the attribute redundancy, we optimize the augmented attributes' weights by maximizing the conditional log-likelihood (CLL) of the built model. Extensive experimental results show that A^2WNB significantly outperforms NB and all the other existing state-of-the-art competitors.

Keywords naive Bayes, attribute augmentation, attribute weighting, attribute conditional independence, classification

Citation Zhang H, Jiang L X, Li C Q. Attribute augmented and weighted naive Bayes. Sci China Inf Sci, 2022, 65(12): 222101, https://doi.org/10.1007/s11432-020-3277-0

1 Introduction

Supervised classification is an essential and crucial task in data mining. Over the years, Bayesian network classifiers have received much attention due to the explicit model interpretability and the surprising classification performance. Among numerous Bayesian network classifiers, naive Bayes (NB) is a basic but remarkably effective classifier, assuming that all attributes are fully independent given the class. Given a test instance $\boldsymbol{x}$ represented by an attribute value vector $\langle a_1, a_2, \ldots, a_m \rangle$, NB uses (1) to predict its class label.

$$\hat{c}(\boldsymbol{x}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m} \theta_{A_j = a_j | c}, \tag{1}$$

where m is the number of attributes, a_j is the jth attribute value of $\boldsymbol{x}$, C is the collection of all possible class labels c, π_c is the prior probability of the class c and $\theta_{A_j=a_j|c}$ is the conditional probability of $A_j = a_j$ given the class c, which can be estimated by

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1}{n + q}, \tag{2}$$

$$\theta_{A_j=a_j|c} = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j)\delta(c_i, c) + 1}{\sum_{i=1}^{n} \delta(c_i, c) + n_j}, \tag{3}$$

where n is the number of training instances, q is the number of classes, n_j is the number of values for the jth attribute A_j, c_i is the class label of the ith training instance, a_{ij} is the jth attribute value of the ith training instance, and the indicator function $\delta(x, y)$ is one if $x = y$ and zero otherwise.

* Corresponding author (email: ljiang@cug.edu.cn)

Although the structure is extremely simple, NB has already exhibited surprising classification performance and continues to be one of the top 10 algorithms in data mining [1]. Nonetheless, the attribute conditional independence assumption might not hold true in reality. As a result, numerous enhancements have been proposed to relax this assumption, which could be categorized into structure extension [2–5], instance selection [6–8], instance weighting [9–11], attribute selection [12–14], attribute weighting [15–17], and fine tuning [18–20].

To our knowledge, however, almost all the existing enhancements only focus on the original attribute space [21, 22], i.e., the human-defined attribute space. But in real-world applications, classification problems are usually very complex and are multifactor functioning processes. We argue that the original attribute space might be incomplete and the discriminative information provided by them might be insufficient for classification. Therefore, in this study, we expect to discover some latent attributes beyond the original attribute space and propose a novel two-stage model called attribute augmented and weighted NB (A^{2}WNB). In A^{2}WNB, we first learn some discriminative but not nameable latent attributes and then build an attribute weighted NB wrapper to alleviate the attribute redundancy. Specifically, at the first stage, we build multiple random one-dependence estimators (RODEs) in the original attribute space. Then we use each built RODE to classify each training instance in turn and define the predicted class labels as its latent attributes. At last, we construct the augmented attributes by concatenating the latent attributes with the original attributes. At the second stage, to alleviate the attribute redundancy, we optimize the augmented attributes' weights by maximizing the conditional log-likelihood (CLL) of the built model. The experimental results on a collection of 69 benchmark datasets from the University of California at Irvine (UCI) repository validate the super performance of our proposed A^{2}WNB.

To sum up, the main contributions of our work include:

- We develop a new general framework, i.e., attribute augmentation and weighting, for classification. Firstly, we propose to use attribute augmentation to improve the discriminative ability of the original attribute space. Then, in order to mitigate the increased risk of attribute redundancy, we propose to use attribute weighting to optimize the weight vector for the augmented attributes.
- We propose a novel two-stage model called attribute augmented and weighted NB (A^{2}WNB). We first define the predicted class labels of the built multiple RODEs as the latent attributes for each training instance. Then we concatenate them with the original attributes to construct the augmented attributes. At last, we use the augmented attributes to build an attribute weighted NB wrapper by maximizing the CLL. To our knowledge, this is the first study to discover latent attributes beyond the original attribute space for improving NB.
- We conduct extensive experiments on a collection of 69 UCI benchmark datasets to validate the effectiveness of A^{2}WNB by comparing it with some related state-of-the-art competitors. Besides, we also conduct an ablation study and sensitivity analysis for A^{2}WNB.
- We conduct a case study to make a thorough analysis to show how each part in A^{2}WNB takes effect, and we also give some possible practical applications of the proposed A^{2}WNB.

The organization of this paper is as follows. Section 2 reviews some related algorithms with regard to this study. Section 3 proposes our A^{2}WNB model. Section 4 presents the experimental setup and results. Section 5 concludes this study and outlines the main directions for future work.

2 Related work

In the last few years, numerous enhancements have been proposed to mitigate the attribute conditional independence assumption in NB [8, 23, 24]. Among them, structure extension is a direct and effective approach to representing attribute dependencies explicitly by adding some directed arcs [25, 26]. For this kind of approach, how to find each attribute's parent node is a crucial problem. Friedman et al. [2] singled out tree-augmented naive Bayes (TAN). In TAN, the attribute dependencies could be represented using a tree-like structure. Specifically, the class node directly points to all attribute nodes and each attribute node only has at most one parent node from the other attributes. Furthermore, to avoid the structure learning in TAN, Webb et al. [3] proposed another model called averaged one-dependence estimators (AODE). In AODE, each attribute is in turn regarded as the parent node of all the other attributes. AODE relaxes the attribute conditional independence assumption by averaging all of the qualified one-dependence estimators, meanwhile without incurring the high time complexity. After that, Jiang [27] proposed another model called RODEs. In RODE, each attribute has at most one parent node from the other attributes, and this parent node is randomly selected from $\log_2 m$ (where m is the number of attributes) attributes with the maximal conditional mutual information. Finally, based on the ensemble learning idea, they proposed to learn an ensemble of RODEs, and their class membership probabilities for a given test instance are directly averaged to make a prediction. In their study, the ensemble size is set to the number of attributes m. In summary, RODE improves the classification performance better than the standard NB, and at the same time conducts randomness in one-dependence estimators.

Different from structure extension which adds some directed arcs to represent attribute dependencies explicitly, attribute weighting has been confirmed more flexible and effective by assigning a continuous weight to each attribute based on their predictive ability [28,29]. Existing attribute weighting approaches could be classified into two broad categories: filters and wrappers [30, 31]. Filters employ general data characteristics to calculate the attribute weights, regarding it as a preprocessing step before building the classifier. Wrappers optimize the attribute weights iteratively according to the performance feedback from the built classifier. For filters, how to calculate the weight of each attribute is a crucial issue. Zhang and Sheng [32] proposed a gain ratio-based attribute weighted NB (GRAWNB) model, which assumes that the weight of an attribute is relevant to its gain ratio, i.e., an attribute with a higher gain ratio should be assigned a larger weight and vice versa. Besides, Hall [15] proposed a decision tree-based attribute weighted NB (DTAWNB) model, which first builds an ensemble of unpruned decision tree from the training instances with random samples, then each attribute weight is assigned inversely proportional to its minimum depth in the decision tree. Recently, Jiang et al. [17] proposed a correlation-based attribute (feature) weighted NB (CFWNB) model, which considers both the attribute-class relevance and the attribute-attribute redundancy. For wrappers, how to search for an optimal weight vector is a key issue. Zaidi et al. [16] proposed an attribute weighted NB model called weighting attributes to alleviate NB's independence assumption (WANBIA). WANBIA utilizes gradient descent searches to optimize the weight vector. The objective function is to maximize the CLL or minimize the mean squared error (MSE), and thus two different versions are created, which are denoted by $\text{WANBIA}^{\text{CLL}}$ and $\text{WANBIA}^{\text{MSE}}$, respectively. Recently, class-specific attribute weighting [33] and attribute value weighting [34] have been proposed as two more fine-grained attribute weighting paradigms, which discriminatively assign different weights to different classes and attribute values, respectively.

3 Attribute augmented and weighted naive Bayes (A^2WNB)

3.1 Motivation

As discussed above, numerous enhancements have been proposed to mitigate the attribute conditional independence assumption in NB. However, almost all of them only focus on the original attribute space. But real-world applications could be rather complicated, and are often influenced by innumerable factors [35, 36]. Due to the restriction of human cognitive abilities, sometimes it is essentially unrealistic to extract enough discriminative attributes for classification, even for the most intelligent domain experts. To be more specific, the original attributes that human defined are often those nameable attributes such as age, sex, and birth date. However, discriminative attributes for classification are sometimes hard to be described explicitly. We argue that the incompleteness of the original attribute space accounts for the poor performance of the classifier. Therefore, in this study, we develop a new general framework, i.e., attribute augmentation and weighting, for classification. Firstly, we use attribute augmentation to discover some latent attributes to improve the discriminative ability of the original attribute space. Then, in order to mitigate the increased risk of attribute redundancy, we use attribute weighting to assign different weights to different attributes. Based on this general framework, we propose a novel two-stage model called attribute augmented and weighted NB (A^2WNB). In summary, A^2WNB can have a more powerful discriminative ability by augmenting the original attribute space, yet at the same time without increasing the risk of attribute redundancy.

3.2 Overall framework of A^2WNB

Before introducing the overall framework of A^2WNB, we first define three important concepts: the original attributes, the latent attributes, and the augmented attributes. Assuming a training instance has m attributes, the original attributes are just the human-defined attributes represented by the attribute value vector $\langle a_1, a_2, \ldots, a_m \rangle$. These attributes are usually extracted by feature engineering and are nameable

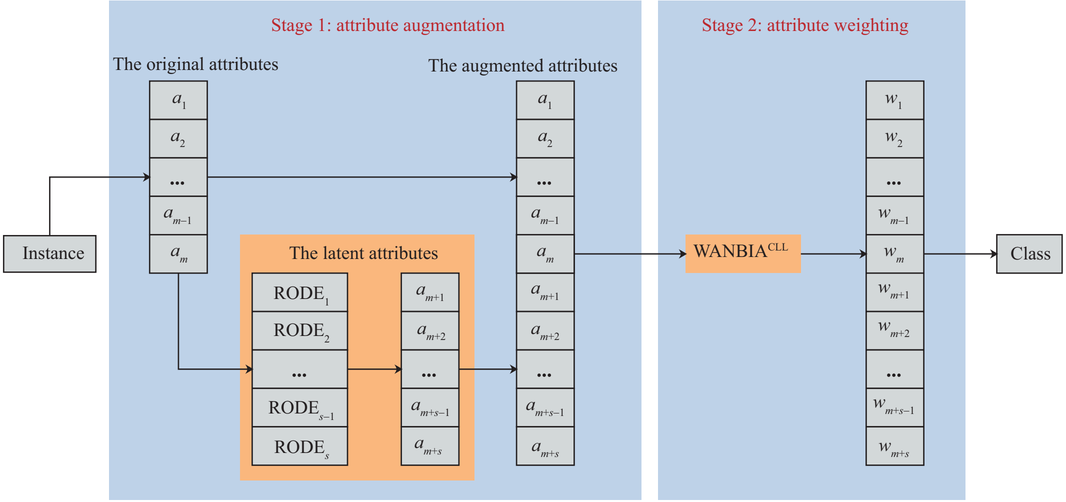

Figure 1 (Color online) The overall framework of A^{2}WNB.

attributes. Accordingly, we define the latent attributes as the automatically learned attributes represented by another attribute value vector $\langle a_{m+1}, a_{m+2}, \dots, a_{m+s}\rangle$. These attributes are discovered by self-learning and are usually unnameable attributes. The third concept is the augmented attributes represented by the attribute value vector $\langle a_1, \dots, a_m, a_{m+1}, \dots, a_{m+s}\rangle$, which is the concatenation between the original attributes and the latent attributes. The augmented attributes extend the original attribute space and thus could provide more sufficient discriminative information for classification.

The overall framework of A^{2}WNB is described detailedly in Figure 1. From Figure 1, we can see clearly that A^{2}WNB is a two-stage model. At the first stage, we build s RODEs in the original attribute space. Then we use each RODE to classify each training instance in turn and define the predicted class labels as its latent attributes. Next, we construct the augmented attributes by concatenating the latent attributes with the original attributes to extend the attribute space. At the second stage, we optimize the augmented attributes' weights by an attribute weighted NB wrapper WANBIA$^{\text{CLL}}$ [16] to mitigate the attribute redundancy in the augmented attributes.

Please note that there are two reasons that we choose RODE as the base classifier at the first stage: (1) RODE conducts randomness in one-dependence estimators, thus we could construct multiple different RODEs to learn the latent attributes. (2) The individual RODE is generally more competitive than the existing Bayesian network classifiers with randomness such as RSNB [37], because RODE extends the structure of NB and uses directed arcs to represent attribute dependencies explicitly, but RSNB is an attribute selection-based model which only selects a subset of attributes to alleviate the attribute conditional independence assumption in NB. Likewise, WANBIA$^{\text{CLL}}$ is chosen as the base classifier at the second stage also due to two points: (1) WANBIA$^{\text{CLL}}$ could effectively mitigate the attribute redundancy by learning a discriminative weight between 0 and 1 to each attribute. (2) In general, WANBIA$^{\text{CLL}}$ could obtain a more remarkable classification performance and is much faster to train than other existing attribute weighted NB wrappers such as DEAWNB [38], in which sophisticated evolution computation processes are conducted to learn attribute weights and thus is very time-consuming.

Now, there are two crucial problems remaining to be solved: one is how to build the RODEs and learn the latent attributes, the other is how to build the WANBIA$^{\text{CLL}}$ on the augmented attributes. We will discuss these two problems in the next two subsections, respectively.

3.3 Stage 1: attribute augmentation

In this subsection, we first describe the detailed processes of building a single RODE. To begin with, conditional mutual information is an important concept needed to be introduced. Let A_i, A_j, and C be three variables, the conditional mutual information between A_i and A_j given C can be defined by (4). Moreover, the average conditional mutual information CMI_{avg} of all these attributes can be calculated by (5).

$$\mathrm{CMI}(A_i; A_j \mid C) = \sum_{a_i, a_j, c} F(a_i, a_j, c) \log \frac{F(a_i, a_j, c)F(c)}{F(a_i, c)F(a_j, c)}, \tag{4}$$

$$\mathrm{CMI}_{\mathrm{avg}} = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j=1 \wedge j \neq i}^{m} \mathrm{CMI}\left(A_i; A_j \mid C\right), \tag{5}$$

where $F(\cdot)$ is the frequency with which a combination of terms appears in the training data.

The first step in building a single RODE is to calculate $\mathrm{CMI}(A_i; A_j|C)$ by (4) for each pair of attributes A_i and A_j. Next, for each attribute A_j, we find the $\log_2 m$ attributes with the maximal $\mathrm{CMI}(A_i; A_j|C)$. Then, we randomly select one of them A_{jp} as the parent of A_j. If the selected $\mathrm{CMI}(A_{jp}; A_j|C)$ is no less than the $\mathrm{CMI}_{\mathrm{avg}}$, A_{jp} is regarded the attribute parent of A_j. Otherwise, A_j has no attribute parent. From the processes above, we can see that the single RODE conducts randomness in one-dependence estimators because the parent of A_j is randomly selected. Thus we could repeat the above processes to construct s RODEs independently. After that, each training instance $\boldsymbol{y}$ is classified by each built RODE in turn using

$$\hat{c}(\boldsymbol{y}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m} \theta_{A_j = a_j | a_{jp}, c}, \tag{6}$$

where π_c can be estimated by (2), and if A_j has no parent node, $\theta_{A_j=a_j|c}$ can be estimated by (3), otherwise $\theta_{A_j=a_j|a_{jp},c}$ can be estimated by

$$\theta_{A_j=a_j|a_{jp},c} = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j)\delta(a_{ijp}, a_{jp})\delta(c_i, c) + 1}{\sum_{i=1}^{n} \delta(a_{ijp}, a_{jp})\delta(c_i, c) + n_j}, \tag{7}$$

where a_{ij} is the jth attribute value of the ith training instance, a_{jp} is the attribute value of the parent node of A_j, and a_{ijp} is the attribute value of a_{jp} of the ith training instance.

It is evident that each instance could get s predicted class labels from the s RODEs. For each instance, the rth predicted class label is defined as the rth latent attribute of this instance. At last, we concatenate the s latent attributes with the m original attributes and call the $s + m$ attributes as the augmented attributes. We believe that the augmented attributes can improve the classification performance because they can provide more powerful discriminative ability beyond the original attribute space. However, yet at the same time, the increased risk of attribute redundancy might weaken the classification performance to some extent. Therefore, in Subsection 3.4, we will build an attribute weighted NB wrapper to further enhance the classification performance.

3.4 Stage 2: attribute weighting

In this subsection, we will employ an attribute weighted NB wrapper called WANBIA$^{\mathrm{CLL}}$ [16] to mitigate the attribute redundancy in the augmented attributes. In WANBIA$^{\mathrm{CLL}}$, each attribute will be assigned a continuous weight between 0 and 1. Then the weights are incorporated into the naive Bayesian classification formula. Unlike the original WANBIA$^{\mathrm{CLL}}$ algorithm whose weight vector size is m, the weight vector size of A^{2}WNB is $m + s$, because m of them are the original attributes, and the other s are the latent attributes. Here, given a test instance $\boldsymbol{x}$, A^{2}WNB utilizes (8) to predict its class label.

$$\hat{c}(\boldsymbol{x}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m+s} \theta_{A_j=a_j|c}^{w_j}, \tag{8}$$

where π_c can be estimated by (2), $\theta_{A_j=a_j|c}$ can be estimated by (3), and w_j represents the weight of A_j.

The original WANBIA algorithm utilizes gradient descent searches to optimize the attribute weights. The objective function is either maximizing the CLL or minimizing the MSE. Motivated by this research, in this study, we also employ gradient descent searches to optimize the weight vector of the augmented attributes. For simplicity, here we only take the objective function by maximizing the CLL. But when we change the objective function to minimize the MSE, similar conclusions could be drawed. Following WANBIA$^{\mathrm{CLL}}$, we also employ the L-BFGS-M algorithm [39] to optimize the weight vector. In A^{2}WNB, all of the empirical parameters utilized in the optimization procedure are consistent with WANBIA$^{\mathrm{CLL}}$. For detail, the CLL objective function can be defined as

$$\mathrm{CLL}(\boldsymbol{w}) = \log \hat{P}(C|\mathcal{D}, \boldsymbol{w}) = \sum_{i=1}^{|\mathcal{D}|} \log \hat{P}(c_i|\boldsymbol{y}_i, \boldsymbol{w}) = \sum_{i=1}^{|\mathcal{D}|} \log \frac{\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w})}{\sum_{c'} \gamma_{c'\boldsymbol{y}_i}(\boldsymbol{w})}, \tag{9}$$

where $\mathcal{D}$ is the training dataset, and

$$\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w}) = \pi_c \prod_j \theta_{a_j|c}^{w_j}. \tag{10}$$

Before calculating the gradient of $\mathrm{CLL}(\boldsymbol{w})$ with respect to w_j, we could first calculate the gradient of $\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w})$ with respect to w_j as follows:

$$\begin{aligned} \frac{\partial}{\partial w_j} \gamma_{c\boldsymbol{y}_i}(\boldsymbol{w}) &= \left(\pi_c \prod_{j' \neq j} \theta_{a_{j'}|c}^{w_{j'}}\right) \frac{\partial}{\partial w_j} \theta_{a_j|c}^{w_j} \\ &= \left(\pi_c \prod_{j' \neq j} \theta_{a_{j'}|c}^{w_{j'}}\right) \theta_{a_j|c}^{w_j} \log\left(\theta_{a_j|c}\right) \\ &= \gamma_{c\boldsymbol{y}_i}(\boldsymbol{w}) \log\left(\theta_{a_j|c}\right). \end{aligned} \tag{11}$$

Then, the gradient of $\mathrm{CLL}(\boldsymbol{w})$ with respect to w_j can be represented as follows:

$$\begin{aligned} \frac{\partial}{\partial w_j} \mathrm{CLL}(\boldsymbol{w}) &= \frac{\partial}{\partial w_j} \sum_{\boldsymbol{y}_i \in \mathcal{D}} \left(\log\left(\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w})\right) - \log\left(\sum_{c'} \gamma_{c'\boldsymbol{y}_i}(\boldsymbol{w})\right)\right) \\ &= \sum_{\boldsymbol{y}_i \in \mathcal{D}} \left(\frac{\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w}) \log(\theta_{a_j|c})}{\gamma_{c\boldsymbol{y}_i}(\boldsymbol{w})} - \frac{\sum_{c'} \gamma_{c'\boldsymbol{y}_i}(\boldsymbol{w}) \log(\theta_{a_j|c'})}{\sum_{c'} \gamma_{c'\boldsymbol{y}_i}(\boldsymbol{w})}\right) \\ &= \sum_{\boldsymbol{y}_i \in \mathcal{D}} \left(\log(\theta_{a_j|c}) - \sum_{c'} \hat{P}(c'|\boldsymbol{y}_i, \boldsymbol{w}) \log(\theta_{a_j|c'})\right). \end{aligned} \tag{12}$$

3.5 Time complexity analysis

In summary, the entire learning algorithm for our A^{2}WNB can be partitioned into training (A^{2}WNB-Training) and classification (A^{2}WNB-Classification) algorithms. They are briefly depicted by Algorithms 1 and 2, respectively. According to Algorithm 1, to build a single RODE, we must first calculate the conditional mutual information for each pair of attributes and calculate the average conditional mutual of all these attributes, which takes the time complexity of $O(qm^2v^2)$, where v is the average number of values for all attributes, namely $v = \frac{1}{m}\sum_{j=1}^{m} n_j$. Then, we need to find the parent of each attribute, which takes the time complexity of $O(m \log_2^2 m)$. At last, we estimate the prior and conditional probabilities, which take the time complexity of $O(nm^2)$. In summary, the time complexity of training s RODEs is $O(s(nm^2 + qm^2v^2 + m \log_2^2 m))$. Besides, the execution time for classifying all the training instances using the built s RODEs is proportional to $sqnm$. Therefore, the whole time complexity of constructing the latent attributes is $O(snm^2 + sqm^2v^2 + sm \log_2^2 m + sqnm)$. If we only take the highest order term, the time complexity of constructing the latent attributes is $O(sqm^2v^2)$. In reality, this process of building s RODEs can be addressed parallelly. In this manner, the time complexity of constructing the latent attributes can be reduced to $O(qm^2v^2)$. According to Algorithm 2, when we classify a test instance, the time complexity of constructing the latent attributes is $O(sqm)$. When we build s RODEs parallelly, the time complexity can also be reduced to $O(qm)$.

4 Experiments and results

4.1 Experimental setting and benchmark data

In this subsection, we conducted a series of experiments to investigate the performance of our proposed A^{2}WNB. We first compared A^{2}WNB with the two most related algorithms WANBIA$^{\mathrm{CLL}}$ [16] and

Algorithm 1 A^{2}WNB-training(D)

```
Input: $D$-a training dataset.
Output: $\boldsymbol{R}$-the built $s$ RODEs, $\boldsymbol{w}$-the attribute weight vector.
  // Stage 1: attribute augmentation
for each pair of attributes $A_i$ and $A_j$ $(i, j = 1, 2, \ldots, m$ and $i \neq j)$ do
    Calculate CMI($A_i; A_j \mid C$) by Eq. (4);
end for
Calculate the average conditional mutual information $\mathrm{CMI}_{\mathrm{avg}}$ by Eq. (5);
for $r = 1$ to $s$ do
    for the original attribute $A_j$ $(j = 1, 2, \ldots, m)$ do
        Find the $\log_2 m$ attributes with the maximal CMI($A_i; A_j \mid C$);
        Select one of them $A_{jp}$ as the parent of $A_j$;
        if the selected CMI($A_{jp}; A_j \mid C$) is no less than the $\mathrm{CMI}_{\mathrm{avg}}$ then
            $A_{jp}$ is regarded as the attribute parent of $A_j$;
        else
            $A_j$ has no attribute parent;
        end if
    end for
    Build the $r$th RODE $R_r$;
end for
for each training instance $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, n)$ from $D$ do
    for each RODE $R_r$ $(r = 1, 2, \ldots, s)$ do
        Predict the class label $c_r$ of $\boldsymbol{y}_i$ using the built $r$th RODE $R_r$ by Eq. (6);
        Define the value of $c_r$ as the $r$th latent attribute value, i.e., the $(m + r)$th attribute value $\boldsymbol{a}_{m+r}$, of $\boldsymbol{y}_i$;
    end for
end for
  // Stage 2: attribute weighting
for each attribute $A_j$ $(j = 1, \ldots, m, m + 1, \ldots, m + s)$ do
    Estimate the prior probability $\pi_c$ of each class $c$ by Eq. (2);
    Estimate the conditional probability $\theta_{A_j = a_j|c}$ of each attribute value $a_j$ given the class $c$ by Eq. (3);
    Initialize all the augmented attributes' weights in the weight vector to 1.0;
    Invoke the optimization procedure L-BFGS-M to optimize the CLL objective function using Eqs. (9)–(12) and obtain the
    optimized weight vector $\boldsymbol{w}$;
end for
return $\boldsymbol{R}$ and $\boldsymbol{w}$.
```

Algorithm 2 A^{2}WNB-classification($\boldsymbol{R}$, $\boldsymbol{w}$, $\boldsymbol{x}$)

```
Input: $\boldsymbol{R}$-the built $s$ RODEs, $\boldsymbol{w}$-the attribute weight vector, $\boldsymbol{x}$-a test instance.
Output: $\hat{c}(\boldsymbol{x})$-the predicted class label of $\boldsymbol{x}$.
for each RODE $R_r$ $(r = 1, 2, \ldots, s)$ do
    Predict the class label $c_r$ of $\boldsymbol{x}$ using the built $r$th RODE $R_r$ by Eq. (6);
    Define the value of $c_r$ as the $r$th latent attribute value, i.e., the $(m + r)$th attribute value $\boldsymbol{a}_{m+r}$, of $\boldsymbol{x}$;
end for
Estimate the prior probability $\pi_c$ of each class $c$ by Eq. (2);
Estimate the conditional probability $\theta_{A_j = a_j|c}$ of each attribute value $a_j$ given the class $c$ by Eq. (3);
Predict the class label $\hat{c}(\boldsymbol{x})$ of $\boldsymbol{x}$ by Eq. (8) using the attribute weight vector $\boldsymbol{w}$;
return $\hat{c}(\boldsymbol{x})$.
```

RODE [27]. Next, we compared A^{2}WNB with a state-of-the-art attribute weighted NB algorithm called CFWNB [17] and a representative ensemble learning algorithm called random forest (RF) [40]. Finally, we also compared the classification performance of A^{2}WNB with the standard NB. Our experiments were conducted on a Windows machine with Waikato environment for knowledge analysis (WEKA) platform [41], and the software version of WEKA we used is 3.6.7[1]. We implemented A^{2}WNB, RODE, and CFWNB algorithms on the WEKA platform. Here, the default number of the latent attributes s in A^{2}WNB is set to the number of attributes m $(s = m)$. The ensemble size of RODEs is also set to the number of attributes m. In addition, we used the existing RF and NB algorithms on the WEKA platform and the source code of WANBIA$^{\mathrm{CLL}}$ was kindly provided by the original authors.

Our experiments were conducted on a collection of 69 benchmark classification datasets from the UCI repository, which are the same as the peer work proposed our most related competitor WANBIA$^{\mathrm{CLL}}$ [16] and represent a wide range of domains and data characteristics. Please note that, although the paper [16] used 73 datasets, to save time, we deleted the largest four datasets of them (i.e., "Poker-hand", "Covertype", "Census-Income (KDD)", "Localization") because they all had more than 100000 instances and thus the running speed was too slow. As for data preprocessing, in our experiments, all the missing attribute values were replaced by the unsupervised attribute filter ReplaceMissingValues in the WEKA

[1] https://sourceforge.net/projects/weka/files/weka-3-6/3.6.7.

platform, in which all missing attribute values were replaced with the modes of the nominal attribute values and the means of the numerical attribute values from the available data. Then, since all the compared algorithms could only deal with discrete value data, we applied the unsupervised filter Discretize in WEKA to discretize the numeric attributes, in which all numerical attributes are discretized into nominal attributes with equal-width ten bins.

4.2 Experimental results and analysis

We presented the detailed classification accuracy of each algorithm on each dataset in Table 1. All of the results were obtained via 10 runs of 10-fold cross validation. The highest and the second highest classification accuracy on each dataset is highlighted in bold and underlined, respectively. For each row, a field marked with • and ○ implies that the classification accuracy of A^{2}WNB statistically upgrades or degrades compared with its competitors when paired two-tailed t-tests with a $p = 0.05$ significance level are conducted [42]. The averages and the Win/Tie/Lose ($W/T/L$) values are summarized at the bottom of the table. The average (arithmetic mean) of each algorithm across all datasets provides a gross indicator of the relative performance in addition to the other statistics. Each entry's $W/T/L$ in the table implies that, compared to its competitors, A^{2}WNB wins on W datasets, ties on T datasets, and loses on L datasets.

Yet at the same time, based on the classification accuracy results presented in Table 1, we employed the Wilcoxon signed-ranks test [43] to thoroughly compare each pair of algorithms. Table 2 summarizes the detailed comparison results. In Table 2, • indicates that the algorithm in the column improves the algorithm in the corresponding row, and ○ indicates that the algorithm in the row improves the algorithm in the corresponding column. The lower diagonal level of significance is $\alpha = 0.05$, and the upper diagonal level of significance is $\alpha = 0.1$.

Observed from all these comparison results, we can draw the conclusion that A^{2}WNB is generally the best among all the competitors including WANBIA$^{\rm CLL}$, RODE, CFWNB, RF, and NB. This fully verifies the universal applicability of A^{2}WNB for a wide range of domains and data characteristics. Now, we summarize the highlights as:

• **The improvement of averaged classification accuracy.** The averaged classification accuracy of A^{2}WNB on 69 datasets is 82.30%, which is remarkably higher than that of WANBIA$^{\rm CLL}$ (81.12%), RODE (81.49%), CFWNB (79.79%), RF (81.13%), and NB (79.36%).

• **The effectiveness of attribute augmentation (ablation study).** Compared to WANBIA$^{\rm CLL}$, A^{2}WNB notably wins on 23 datasets and loses only on 4 datasets. Therefore, the results fully suggest that attribute augmentation (stage 1 in A^{2}WNB) is very effective to improve the classification performance.

• **The effectiveness of attribute weighting (ablation study).** Compared to RODE, A^{2}WNB significantly wins on 16 datasets and loses only on 1 dataset. The results strongly prove that attribute weighting (stage 2 in A^{2}WNB) is also powerful to enhance the classification performance.

• **The superiority than other state-of-the-art attribute weighted NB algorithm.** Compared to CFWNB, A^{2}WNB significantly wins on 32 datasets and loses only on 3 datasets. This suggests that the classification performance of A^{2}WNB is much better than the existing state-of-the-art attribute weighted NB algorithm.

• **The superiority than other state-of-the-art ensemble learning algorithm.** Compared to RF, A^{2}WNB significantly wins on 17 datasets and loses on 11 datasets. This suggests that A^{2}WNB is also competitive with the existing representative ensemble learning algorithm.

• **The superiority than the standard NB.** Compared to the standard NB, A^{2}WNB significantly wins on 33 datasets and loses only on 3 datasets. This suggests that the proposed attribute augmentation and weighted model is indeed very effective.

• **The significance on the Wilcoxon signed-ranks test.** Based on the Wilcoxon signed-ranks test results in Table 2, we could easily find that whatever the significance level is $\alpha = 0.05$ or $\alpha = 0.1$, our proposed A^{2}WNB significantly outperforms the standard NB and all the other existing state-of-the-art competitors including WANBIA$^{\rm CLL}$, RODE, CFWNB, and RF. In a word, A^{2}WNB is really promising for classification.

Meanwhile, according to the results in Table 1, we observed that A^{2}WNB can achieve the highest and the second highest classification accuracy on 40 datasets, which fully demonstrated the superperformance of A^{2}WNB. Yet at the same time, we have also noticed that compared to the most related competitor WANBIA$^{\rm CLL}$, A^{2}WNB still loses on a few datasets, such as "Habermans". We can see that on such

Table 1 Classification accuracy comparisons for A^{2}WNB versus WANBIA$^{\text{CLL}}$, RODE, CFWNB, RF, and NB

Dataset	A^{2}WNB	WANBIA$^{\text{CLL}}$	RODE	CFWNB	RF	NB
Abalone	**54.86**±2.25	54.38±2.03	54.14±2.12	52.78±2.04 ●	53.69±2.10	52.26±2.04 ●
Adult	**84.70**±0.42	84.62±0.46	83.46±0.53 ●	82.34±0.49 ●	82.63±0.41 ●	81.97±0.49 ●
Annealing	90.61±2.61	90.61±2.77	88.60±2.65 ●	90.04±2.44	**91.27**±2.95	88.16±3.06 ●
Audiology	73.95±6.23	74.84±6.99	69.08±6.16 ●	70.10±6.86	**75.95**±8.23	71.40±6.37
AutoImports	79.45±8.16	74.61±9.08	78.43±9.59	65.25±10.98 ●	**83.26**±8.21	63.97±11.35 ●
BalanceScale	87.33±3.48	**91.44**±1.30 ○	87.01±3.00	90.34±2.03 ○	76.61±4.01 ●	**91.44**±1.30 ○
BreastCancer	69.10±7.60	71.46±7.39	71.27±6.35	72.88±7.29	69.17±6.80	**72.94**±7.71
CarEvaluation	**94.00**±1.93	85.46±2.56 ●	93.30±2.24	76.36±1.75 ●	93.18±1.86	85.46±2.56 ●
Chess	90.93±4.12	87.59±4.09 ●	89.84±4.12	85.94±4.12 ●	**91.23**±3.86	87.03±4.18 ●
Connect-4	75.76±0.36	72.74±0.34 ●	75.37±0.33 ●	71.52±0.32 ●	**80.07**±0.44 ○	72.13±0.37 ●
Contact-lenses	74.83±26.00	**76.17**±25.54	70.00±29.01	73.17±25.17	75.67±29.05	**76.17**±25.54
CMC	53.05±4.03	**53.65**±3.63	52.59±3.90	53.17±3.56	46.50±3.73 ●	50.74±3.93
CreditScreening	85.04±4.24	85.51±3.85	85.07±4.08	**85.96**±3.90	83.72±4.08	84.74±3.83
Cylinder	**81.37**±4.71	78.43±5.37	81.30±4.87	77.46±5.21 ●	73.30±5.00 ●	75.52±5.02 ●
Dermatology	97.70±2.36	**98.47**±1.79	97.59±2.41	97.90±2.19	95.27±3.27 ●	97.46±2.37
Echocardiogram	64.91±11.40	70.93±11.85	63.69±9.80	70.57±10.49	64.70±10.90	**71.30**±11.94
German	75.31±3.48	75.94±3.73	75.85±3.01	**76.16**±3.52	73.81±3.78	75.93±3.87
Glass	**60.38**±9.12	57.91±10.37	59.52±8.26	56.84±9.55	59.52±8.41	57.69±10.07
Habermans	69.90±6.96	**75.30**±5.86 ○	67.97±7.32	73.32±3.32	68.63±6.09	**75.30**±5.86 ○
HeartDisease	81.36±7.43	83.07±6.60	80.96±7.17	**84.59**±6.59	77.52±7.46	83.44±6.27
Hepatitis	84.50±8.93	84.32±9.17	84.65±9.20	**85.03**±9.23	81.77±7.89	84.06±9.91
HorseColic	73.10±7.63	74.93±7.02	73.70±6.06	**75.23**±6.42	73.03±5.68	74.45±7.24
HouseVotes84	**95.44**±2.93	95.20±2.90	94.74±3.14	91.72±3.90 ●	95.13±2.95	89.98±3.93 ●
Hungarian	81.70±6.02	82.86±5.54	82.29±5.57	**83.85**±6.08	79.09±6.62	83.68±5.97
Hypothyroid	93.47±0.49	**93.57**±0.53	93.48±0.52	93.49±0.53	92.38±0.82 ●	92.79±0.73 ●
Ionosphere	92.00±4.11	91.28±4.16	**92.57**±4.30	91.48±4.15	90.43±5.10	90.86±4.33
Iris	94.00±6.03	**96.27**±4.67	94.13±5.86	95.53±5.85	94.87±5.35	94.33±6.79
Kr-vs-kp	95.47±1.20	93.38±1.30 ●	93.71±1.39 ●	93.59±1.32 ●	**98.87**±0.61 ○	87.79±1.91 ●
Labor	92.37±11.46	94.90±8.73	91.90±11.69	91.97±10.39	88.37±12.24	**96.70**±7.27
LED	73.59±4.18	**73.75**±4.42	73.25±4.17	73.49±4.34	73.23±4.17	**73.75**±4.41
Letter	84.39±0.70	70.68±0.80 ●	83.60±0.69 ●	70.87±0.87 ●	**89.84**±0.74 ○	70.09±0.93 ●
LiverDisorders	65.98±7.82	64.07±7.55	**66.21**±7.08	63.72±5.93	62.46±7.83	64.15±7.45
LungCancer	49.25±25.65	49.08±27.85	46.17±24.74	**54.92**±24.68	46.92±25.42	51.92±25.59
Lymphography	84.88±9.05	85.70±9.75	84.22±9.23	84.76±9.02	80.02±9.39	**85.97**±8.88
MAGIC	**81.31**±0.83	80.23±0.80 ●	79.38±0.85 ●	76.90±0.75 ●	81.20±0.80	74.75±0.72 ●
Mushrooms	99.99±0.03	99.62±0.20 ●	99.90±0.11 ●	98.93±0.37 ●	**100.00**±0.00	95.52±0.78 ●
Musk1	**95.04**±3.67	93.95±3.63	94.47±3.92	84.32±6.82 ●	87.02±6.05 ●	81.82±7.16 ●
Musk2	96.02±1.30	**99.97**±0.08 ○	95.05±0.79	87.17±1.41 ●	97.88±0.80 ○	82.06±1.64 ●
Nettalk	77.70±1.44	74.84±1.64 ●	74.28±1.46 ●	72.86±1.48 ●	**83.84**±1.39 ○	71.80±1.87 ●
New-Thyroid	**93.48**±4.91	92.32±4.64	82.97±4.68 ●	90.97±5.07	93.31±5.62	91.72±5.05
Nursery	92.53±1.39	89.43±1.42 ●	91.57±1.43 ●	87.75±1.63 ●	**95.14**±1.14 ○	89.43±1.42 ●
OpticalDigits	**96.11**±0.77	93.72±1.03 ●	96.00±0.84	92.68±1.13 ●	86.65±1.46 ●	92.25±1.07 ●
PageBlocks	93.43±0.74	92.43±0.81 ●	93.19±0.85	92.36±0.84 ●	**93.70**±0.76	92.31±0.92 ●
PenDigits	**97.01**±0.48	88.19±1.03 ●	96.68±0.48 ●	87.49±0.98 ●	96.82±0.53	87.07±1.05 ●
Pima	76.70±4.77	76.68±4.86	76.39±4.92	**76.72**±4.59	72.59±4.67 ●	75.68±4.85
Pioneer	96.91±0.57	92.41±0.81 ●	96.24±0.60 ●	91.57±0.88 ●	**98.20**±0.44 ○	90.17±1.02 ●
Postoperative	63.00±11.61	68.56±8.36	65.56±8.72	**69.78**±6.72	59.78±11.69	68.22±8.51
PrimaryTumor	**47.31**±5.63	47.20±6.31	46.88±5.65	45.99±5.68	39.68±5.96 ●	47.20±6.02
Promoter	85.42±10.41	91.53±9.87	84.92±11.35	**92.55**±7.77 ○	79.05±13.28	90.14±9.59
Satellite	88.85±1.03	83.03±1.44 ●	88.62±1.00	81.05±1.57 ●	**88.89**±1.07	80.93±1.58 ●
Segment	94.60±1.43	92.13±1.72 ●	93.97±1.54 ●	90.27±1.69 ●	**95.56**±1.41	89.03±1.66 ●
Sick	98.04±0.69	97.39±0.83 ●	97.89±0.76	97.44±0.81 ●	**98.09**±0.66	96.78±0.91 ●
Sign	70.31±1.34	61.56±1.34 ●	70.62±1.36	59.17±1.22 ●	**80.99**±1.09 ○	61.11±1.37 ●
Sonar	**82.24**±8.09	76.91±9.56	82.04±8.33	75.27±10.16 ●	72.87±11.46 ●	76.35±9.94
Spambase	**86.95**±1.43	86.23±1.48 ●	85.83±1.53 ●	84.59±1.63 ●	86.57±1.63	84.66±1.38 ●
Splice	93.27±2.08	95.82±1.08 ○	95.55±1.07 ○	**96.12**±1.02 ○	88.13±2.81 ●	95.41±1.18 ○
Statlog	94.01±0.51	92.97±0.26 ●	93.38±1.40	92.24±0.26 ●	**94.33**±0.25 ○	88.87±0.35 ●
Syncon	98.08±1.81	97.35±2.00	**98.17**±1.80	97.02±2.08	91.53±3.94 ●	96.88±2.23
Teaching	54.98±12.25	54.25±11.49	52.21±13.11	55.23±11.79	**58.77**±13.03	54.25±11.75
Thyroid	93.82±0.39	93.80±0.37	**93.87**±0.38	93.78±0.38	93.19±0.55 ●	93.68±0.42
Tic-Tac-Toe	77.01±3.90	72.38±4.14 ●	76.85±3.23	69.05±4.49 ●	**91.76**±2.66 ○	69.64±4.40 ●
Vehicle	**72.05**±3.89	64.42±3.83 ●	71.60±4.13	61.52±3.51 ●	70.57±4.08	61.03±3.48 ●
Volcanoes	65.72±3.13	66.26±2.55	64.34±3.32	**66.63**±2.07	61.18±2.97 ●	66.28±2.60
Vowel	88.05±3.69	68.63±4.41 ●	87.29±3.35	68.31±4.68 ●	**90.38**±3.33	66.09±4.78 ●
Wall-following	90.00±1.28	86.11±1.40 ●	86.12±1.57 ●	83.80±1.55 ●	**94.58**±0.91 ○	80.33±1.62 ●
Waveform-5000	**83.87**±1.82	83.12±1.63	83.59±1.74	81.19±1.42 ●	77.52±1.93 ●	79.97±1.46 ●
Wine	95.22±5.19	96.57±4.00	94.88±5.34	**96.74**±3.76	93.29±5.52	96.52±3.88
Yeast	57.55±3.64	**58.81**±3.27	56.15±3.81	56.60±3.26	51.69±3.65 ●	58.56±3.32
Zoo	**97.13**±5.53	95.35±5.51	96.63±5.51	94.86±6.52	91.35±9.28	93.98±7.14
Average	82.30	81.12	81.49	79.79	81.13	79.36
W/T/L	–	23/42/4	16/52/1	32/34/3	17/41/11	33/33/3

Table 2 Classification accuracy comparisons of Wilcoxon tests

Algorithm	$\mathrm{A^2WNB}$	$\mathrm{WANBIA^{CLL}}$	RODE	CFWNB	RF	NB
$\mathrm{A^2WNB}$	–	○	○	○	○	○
$\mathrm{WANBIA^{CLL}}$	●	–		○		○
RODE	●		–	○		○
CFWNB	●	●	●	–		○
RF	●				–	
NB	●	●	●	●		–

datasets, the classification accuracies of RODE are much lower than that of NB, which means the learned latent attributes in A^{2}WNB might be inaccurate on such datasets. Therefore, the attribute augmentation stage has a great influence on the final model, and only when the latent attributes accurately reflect some discriminative information, the classification performance of the final model can be improved. Conversely, if the latent attributes are useless or even have some negative effects, then the final classification performance will not be improved, or even be decreased accordingly.

Finally, we observed the sensitivity of our proposed A^{2}WNB with different s values, i.e., the number of the latent attributes, such as $m/3$, $m/2$, $2m$, and $3m$, where m is the number of the original attributes. Table 3 shows the summary comparison results. From them, we can see that A^{2}WNB ($s = m$) is slightly better than A^{2}WNB ($s = m/3$), almost comparative with A^{2}WNB ($s = m/2$) as well as A^{2}WNB ($s = 2m$), and only slightly worse than A^{2}WNB ($s = 3m$). These results indicate that the performance of A^{2}WNB is not sensitive to the values of the parameter s as long as it is not too small (generally no less than $m/2$). This makes it a very attractive alternative to those algorithms, which require fine-tuning of the parameters to achieve good results.

4.3 Case study

Moreover, to thoroughly analyze how each part in A^{2}WNB takes effect, we picked out the dataset "Nursery"[2)] in Table 1 to conduct another group of experiments to observe the performance of A^{2}WNB. This dataset was originally developed to rank applications for nursery schools. Based on the domain knowledge, experts extracted 8 discriminative attributes, which were all discrete value data and each of them took on 2–5 attribute values. The detailed description of them was listed in Table 4. The dataset contains 12980 instances, and the instances can be classified into 5 classes, whose values are listed in the last line of Table 4. According to the description in [44], at that time, the acceptance committee was lack of unifying evaluation methodology and the data they gathered did not provide adequate information to make the decision. That is to say, the discriminative information provided by the original attribute space is insufficient. Therefore, our proposed A^{2}WNB model is very suitable for solving this problem.

Here, we use the same experimental platform and the same experimental settings as we described in Subsection 4.1. In this group of experiments, we compare the classification accuracy of A^{2}WNB with its three variants denoted as A^{2}WNB-R, A^{2}WNB-O, and A^{2}NB, respectively. Each of them removes or replaces a specific part in A^{2}WNB to observe the effectiveness of this part. Specifically, A^{2}WNB-R is a variant that replaces the latent attributes learning part in A^{2}WNB by randomly adding m attributes from the original attribute space. Therefore, we compare their performances to validate the effectiveness of the latent attributes learning part in A^{2}WNB. Next, A^{2}WNB-O is another variant we designed that removes the original attributes in the attribute weighting stage. In other words, the attribute augmentation stage in A^{2}WNB-O is the same as that in A^{2}WNB, but in the attribute weighting stage, A^{2}WNB-O only optimizes the latent attributes' weights instead of the augmented attributes. This variant can also be called the standard stacking combine strategy. We compare their performances to verify whether the original attributes are helpful to improve the classification accuracy. Finally, we compare the performance of A^{2}WNB with another variant designed by ourselves called A^{2}NB, in which the attribute augmentation stage is the same as A^{2}WNB, but then directly builds an NB model by the augmented attributes rather than an attribute weighted NB model. Since A^{2}NB removes the attribute weighting stage, we compare their performances to explore that whether attribute weighting is necessary in A^{2}WNB.

Their detailed classification accuracy comparison results are shown in Figure 2(a). Please note that, due to a certain degree of randomness in the attribute augmentation stage, the classification performance of A^{2}WNB in Figure 2(a) (92.44%) is a little different from that in Table 1 (92.53%), but we can still

[2)] http://archive.ics.uci.edu/ml/datasets/Nursery.

Table 3 Classification accuracy comparisons for A^2WNB with different values of the latent attributes number s

Dataset	A^2WNB	A^2WNB	A^2WNB	A^2WNB	A^2WNB
	$(s = m)$	$(s = m/3)$	$(s = m/2)$	$(s = 2m)$	$(s = 3m)$
Abalone	54.86±2.25	54.76±2.31	54.83±2.20	54.68±2.10	54.66±2.19
Adult	84.70±0.42	84.67±0.43	84.69±0.43	84.73±0.39	84.73±0.43
Annealing	90.61±2.61	90.47±2.65	90.47±2.51	90.85±2.73	90.71±2.71
Audiology	73.95±6.23	73.11±6.59	73.38±6.45	73.03±6.38	73.68±6.91
AutoImports	79.45±8.16	79.08±8.91	79.08±9.80	79.77±9.08	79.75±8.57
BalanceScale	87.33±3.48	85.58±3.39	86.50±3.12	87.29±3.83	87.31±3.79
BreastCancer	69.10±7.60	69.92±7.62	68.99±7.80	68.58±7.01	68.86±7.15
CarEvaluation	94.00±1.93	92.92±2.81	93.48±2.45	94.27±1.84	94.48±1.79
Chess	90.93±4.12	90.27±4.21	90.73±3.84	91.11±4.02	91.24±4.05
Connect-4	75.76±0.36	75.48±0.38 ●	75.58±0.36 ●	75.88±0.36	75.91±0.32 ○
Contact-lenses	74.83±26.00	67.50±29.34	72.17±28.63	71.17±27.81	72.33±28.55
CMC	53.05±4.03	52.83±3.84	53.07±4.06	53.30±3.95	53.20±3.90
CreditScreening	85.04±4.24	84.99±4.12	84.77±4.01	84.87±4.00	84.84±4.24
Cylinder	81.37±4.71	81.24±4.85	81.69±4.77	81.48±5.01	81.59±4.96
Dermatology	97.70±2.36	97.67±2.32	97.68±2.25	97.73±2.37	97.86±2.42
Echocardiogram	64.91±11.40	64.58±10.59	62.99±12.01	63.21±12.24	63.88±12.82
German	75.31±3.48	74.90±3.54	75.48±3.45	75.44±3.51	75.70±3.46
Glass	60.38±9.12	59.90±8.81	60.19±8.54	60.97±7.83	61.23±8.98
Habermans	69.90±6.96	69.90±6.96	69.90±6.96	69.90±6.96	69.90±6.96
HeartDisease	81.36±7.43	80.15±7.31	81.15±7.15	81.19±7.15	81.28±6.85
Hepatitis	84.50±8.93	83.55±9.96	84.21±9.57	83.75±10.18	83.43±10.14
HorseColic	73.10±7.63	72.42±7.55	72.72±8.20	73.51±7.31	73.84±6.87
HouseVotes84	95.44±2.93	95.47±2.82	95.35±3.17	95.51±3.09	95.74±2.95
Hungarian	81.70±6.02	81.97±6.09	81.29±5.98	82.39±5.73	81.57±6.22
Hypothyroid	93.47±0.49	93.45±0.50	93.44±0.52	93.47±0.56	93.45±0.56
Ionosphere	92.00±4.11	92.43±4.47	92.08±4.24	92.17±4.23	92.17±4.01
Iris	94.00±6.03	94.40±6.27	94.27±6.14	93.73±5.67	93.87±5.74
Kr-vs-kp	95.47±1.20	95.02±1.17	95.14±1.21	95.70±1.02	95.75±1.12
Labor	92.37±11.46	92.00±11.96	91.17±11.74	91.67±11.75	91.87±11.23
LED	73.59±4.18	73.65±4.01	73.36±4.23	73.53±4.08	73.57±4.29
Letter	84.39±0.70	83.15±0.73 ●	83.80±0.78 ●	84.79±0.73 ○	85.01±0.72 ○
LiverDisorders	65.98±7.82	64.44±8.34	65.82±7.66	65.97±7.65	65.92±7.57
LungCancer	49.25±25.65	48.83±27.40	47.25±24.36	45.00±24.07	45.08±26.49
Lymphography	84.88±9.05	83.82±8.60	84.84±8.74	84.56±8.57	85.10±8.84
MAGIC	81.31±0.83	81.17±0.85	81.18±0.80	81.40±0.77	81.41±0.79
Mushrooms	99.99±0.03	99.96±0.08	99.98±0.06	99.99±0.04	99.99±0.04
Musk1	95.04±3.67	95.15±3.56	95.10±3.39	94.52±4.06	94.27±4.02
Musk2	96.02±1.30	99.92±0.13 ○	99.92±0.13 ○	95.15±0.76	95.31±0.76
Nettalk	77.70±1.44	76.77±1.56 ●	77.20±1.48	77.93±1.52	77.96±1.55
New-Thyroid	93.48±4.91	92.93±5.06	93.25±4.79	93.35±4.77	93.44±4.97
Nursery	92.53±1.39	91.38±1.54 ●	92.10±1.54	92.83±1.54	92.82±1.50 ○
OpticalDigits	96.11±0.77	96.03±0.73	96.04±0.78	96.05±0.85	95.74±1.07
PageBlocks	93.43±0.74	93.37±0.81	93.39±0.81	93.54±0.77	93.58±0.82
PenDigits	97.01±0.48	96.67±0.50 ●	96.82±0.51	97.17±0.44	97.22±0.45
Pima	76.70±4.77	76.58±4.88	76.26±4.61	76.36±4.48	76.31±4.67
Pioneer	96.91±0.57	96.75±0.54	96.85±0.54	97.03±0.53	97.04±0.56
Postoperative	63.00±11.61	63.89±10.64	61.56±12.77	60.67±12.77	61.67±11.86
PrimaryTumor	47.31±5.63	47.49±6.71	47.32±5.94	47.67±6.20	47.58±5.89
Promoter	85.42±10.41	86.74±10.78	85.49±11.93	85.29±10.91	84.85±11.24
Satellite	88.85±1.03	88.77±1.07	88.78±1.01	88.91±0.99	88.92±1.04
Segment	94.60±1.43	94.45±1.68	94.40±1.40	94.62±1.50	94.67±1.45
Sick	98.04±0.69	98.04±0.73	97.98±0.74	98.08±0.72	98.07±0.80
Sign	70.31±1.34	69.07±1.65 ●	69.93±1.38	70.70±1.23	70.75±1.23
Sonar	82.24±8.09	81.90±7.67	82.15±8.19	82.48±8.11	82.62±8.01
Spambase	86.95±1.43	86.82±1.36	86.86±1.42	86.89±1.47	86.92±1.43
Splice	93.27±2.08	93.67±2.06	93.72±2.14	92.78±2.87	95.41±1.15 ○
Statlog	94.01±0.51	93.56±0.64	93.60±0.61	94.18±0.32	94.20±0.26
Syncon	98.08±1.81	98.22±1.71	98.12±1.74	98.10±1.76	98.20±1.83
Teaching	54.98±12.25	53.93±11.61	54.59±12.87	54.00±12.03	54.40±12.14
Thyroid	93.82±0.39	93.84±0.39	93.82±0.36	93.83±0.38	93.84±0.38
Tic-Tac-Toe	77.01±3.90	75.72±3.61	75.87±3.85	76.76±3.53	77.27±3.70
Vehicle	72.05±3.89	71.08±4.09	71.56±4.02	72.06±3.98	71.99±4.08
Volcanoes	65.72±3.13	65.72±3.13	65.72±3.13	65.72±3.13	65.72±3.13
Vowel	88.05±3.69	86.52±3.27	86.92±3.51	88.86±3.35	89.10±3.38
Wall-following	90.00±1.28	89.67±1.29	89.79±1.34	90.13±1.34	90.13±1.25
Waveform-5000	83.87±1.82	83.92±1.59	83.92±1.60	84.02±1.67	84.00±1.62
Wine	95.22±5.19	95.61±5.24	95.77±4.56	95.17±5.45	95.34±4.70
Yeast	57.55±3.64	57.12±3.45	57.36±3.72	57.49±3.83	57.61±3.85
Zoo	97.13±5.53	96.83±5.63	96.04±5.97	96.83±5.63	97.12±5.16
Average	82.31	81.95	82.05	82.13	82.26
$W/T/L$	–	6/62/1	2/66/1	0/68/1	0/65/4

Table 4 Description of the attributes' information on the "Nursery" dataset

Attribute ID	Attribute name	Attribute value number	Attribute values
1	Parents	3	Usual, pretentious, great_pret
2	Has_nurs	5	Proper, less_proper, improper, critical, very_crit
3	Form	4	Complete, completed, incomplete, foster
4	Children	4	1, 2, 3, more
5	Housing	3	Convenient, less_conv, critical
6	Finance	2	Convenient, inconv
7	Social	3	Nonprob, slightly_prob, problematic
8	Health	3	Recommended, priority, not_recom
–	Class	5	Not_recom, recommend, very_recom, priority, spec_prior

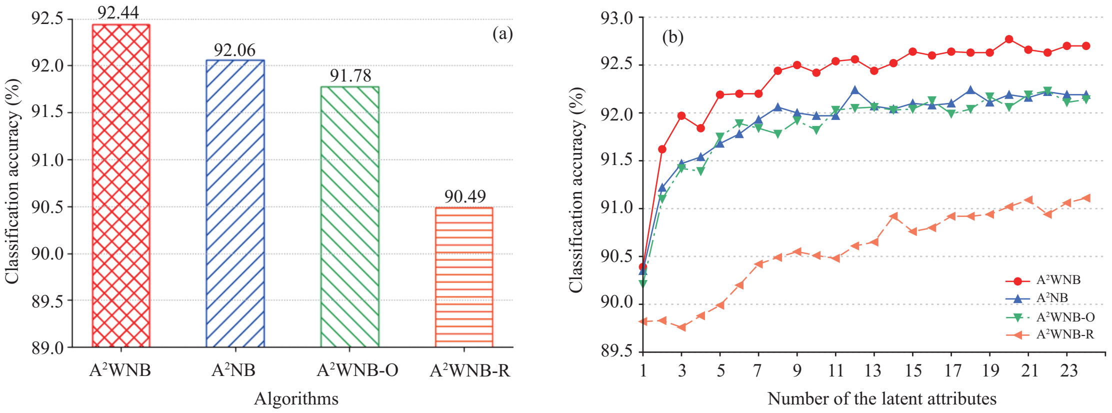

Figure 2 (Color online) Comparison of A^{2}WNB versus A^{2}NB, A^{2}WNB-O, and A^{2}WNB-R on the "Nursery" dataset. (a) Classification accuracy; (b) number of the latent attributes.

find that the classification performance of A^{2}WNB (92.44%) is best, and is remarkably higher than that of A^{2}NB (92.06%), A^{2}WNB-O (91.78%), and A^{2}WNB-R (90.49%). From these results, we can draw the following conclusions: (1) The classification accuracy of A^{2}WNB is evidently higher than that of A^{2}WNB-R, which verifies that the good performance of A^{2}WNB is caused not only by the increasement of attribute dimension, but also resulted from the superiority of the latent attributes learning strategy used in A^{2}WNB. If we randomly add some attributes to augment the original attributes, the built model cannot achieve the same high classification performance as A^{2}WNB. (2) Since the classification accuracy of A^{2}WNB is higher than that of A^{2}WNB-O, this proves that the original attribute space is also useful for improving the performance. Meanwhile, this suggests that our proposed attribute augmentation and weighting framework is more powerful than the standard stacking combine framework. (3) The classification accuracy of A^{2}WNB is higher than that of A^{2}NB. This suggests that the attribute weighting stage is really effective to mitigate the attribute redundancy risk.

Then, to thoroughly show how the latent attributes take effect step by step, we also compare the classification accuracy of A^{2}WNB with its three variants under different numbers of the latent attributes, and the detailed results are shown in Figure 2(b). From the results, we can easily find that no matter what the number of latent attributes is, the classification performance of A^{2}WNB is always better than its three variants, which fully verifies the superiority of A^{2}WNB once again. Also, there are some other useful findings as follows: (1) Generally speaking, as the number of latent attributes increases, the classification performances of A^{2}WNB and its three variants are all improved. Therefore, attribute augmentation is really an effective approach to solving the problems where the discriminative information provided by the original attribute space is insufficient. (2) The performances of A^{2}NB and A^{2}WNB-O are close and are both better than that of A^{2}WNB-R. This suggests the superiority of the latent attributes learning strategy used in A^{2}WNB once again.

Finally, to conduct a deeper case study, we observed the importance of original and latent attributes by "select attributes" window in explorer of the WEKA platform. Since the dataset "Nursery" has 8 original attributes and 8 latent attributes, here we denote original attributes' ID as 1–8, and denote latent attributes' ID as 9–16. Next we employ "ChiSquaredAttributeEval", "GainRatioAttributeEval", and "ReliefFAttributeEval" in WEKA to rank the attribute importance of original and latent attributes.

Table 5 The detailed ranking results sorted in descending order of attribute importance based on different methods

Method	Importance ranking (attribute ID)
ChiSquaredAttributeEval	9, 12, 13, 14, 11, 15, 10, 16, 8, 2, 1, 5, 7, 4, 6, 3
GainRatioAttributeEval	12, 9, 11, 10, 14, 13, 15, 16, 8, 2, 1, 7, 5, 6, 4, 3
ReliefFAttributeEval	14, 13, 12, 15, 9, 8, 10, 11, 16, 2, 1, 5, 7, 4, 6, 3

Table 5 shows the detailed ranking results sorted in descending order of attribute importance. From the results, we can see that latent attributes are generally more important than original attributes. Therefore, these results fully validate that the attribute augmentation stage in A^{2}WNB is really effective, and the latent attributes can indeed provide more sufficient discriminative information for classification.

5 Conclusion and future work

In this study, we argue that sometimes the original attribute space is insufficient for classification due to the difficulty of extracting all the discriminative attributes in real-world applications, and thus we develop a new general framework, i.e., attribute augmentation and weighting, for classification. Based on this framework, we propose a novel two-stage model called attribute augmented and weighted NB (A^{2}WNB). In A^{2}WNB, we first build multiple RODEs and then use each built RODE to classify each instance in turn and define the predicted class labels as its latent attributes. Next, we construct the augmented attributes by concatenating the latent attributes with the original attributes, and at last, we optimize the augmented attributes' weights by maximizing the CLL to avoid the attribute redundancy. We conduct an ablation study and compare A^{2}WNB with other existing state-of-the-art competitors to validate its superiority. Besides, we also conduct a case study to thoroughly validate the effectiveness of each part in A^{2}WNB. In summary, our work provides a novel and attractive model that could be applied to a broad range of practical applications such as disease diagnosis and credit risk assessment.

How to discover the latent attributes to improve discriminative ability is a meaningful problem. Currently, we only use the predicted class label of each built RODE to learn the latent attributes, which is effective yet may cause a relatively high time complexity. Exploring other efficient methods to learn the latent attributes is the main direction for our future work. In addition, applying the general attribute augmentation and weighting framework to improve some other classification models is another interesting topic for our future work.

Acknowledgements The work was supported by National Natural Science Foundation of China (Grant No. U1711267), Fundamental Research Funds for the Central Universities (Grant No. CUGGC03), and Foundation of Key Laboratory of Artificial Intelligence, Ministry of Education, China (Grant No. AI2020002).

References

1 Wu X, Kumar V, Quinlan J R, et al. Top 10 algorithms in data mining. Knowl Inf Syst, 2008, 14: 1–37

2 Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn, 1997, 29: 131–163

3 Webb G I, Boughton J R, Wang Z H. Not so naive Bayes: aggregating one-dependence estimators. Mach Learn, 2005, 58: 5–24

4 Jiang L X, Zhang H, Cai Z H. A novel Bayes model: hidden naive Bayes. IEEE Trans Knowl Data Eng, 2009, 21: 1361–1371

5 Qiu C, Jiang L X, Li C Q. Not always simple classification: learning superparent for class probability estimation. Expert Syst Appl, 2015, 42: 5433–5440

6 Kohavi R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996. 202–207

7 Frank E, Hall M A, Pfahringer B. Locally weighted naive bayes. In: Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, 2003. 249–256

8 Wang S S, Jiang L X, Li C Q. Adapting naive Bayes tree for text classification. Knowl Inf Syst, 2015, 44: 77–89

9 Jiang L X, Wang D H, Cai Z H. Discriminatively weighted naive bayes and its application in text classification. Int J Artif Intell Tools, 2012, 21: 1250007

10 Jiang L X, Qiu C, Li C Q. A novel minority cloning technique for cost-sensitive learning. Int J Patt Recogn Artif Intell, 2015, 29: 1551004

11 Xu W Q, Jiang L X, Yu L J. An attribute value frequency-based instance weighting filter for naive Bayes. J Exp Theor Artif Intell, 2019, 31: 225–236

12 Langley P, Sage S. Induction of selective bayesian classifiers. In: Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence, 1994. 399–406

13 Chen S, Martínez A M, Webb G I. Highly scalable attribute selection for averaged one-dependence estimators. In: Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2014. 86–97

14 Chen S, Webb G I, Liu L, et al. A novel selective naïve Bayes algorithm. Knowl-Based Syst, 2020, 192: 105361

15 Hall M. A decision tree-based attribute weighting filter for naive Bayes. Knowl-Based Syst, 2007, 20: 120–126

16 Zaidi N A, Cerquides J, Carman M J, et al. Alleviating naive bayes attribute independence assumption by attribute weighting. J Mach Learn Res, 2013, 14: 1947–1988

17 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

18 Hindi K E. Fine tuning the naïve Bayesian learning algorithm. AI Commun, 2014, 27: 133–141

19 Diab D M, Hindi K E. Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. Appl Soft Comput, 2017, 54: 183–199

20 Hindi K E, Aljulaidan R R, AlSalman H. Lazy fine-tuning algorithms for naïve Bayesian text classification. Appl Soft Comput, 2020, 96: 106652

21 Chen S L, Martinez A M, Webb G I, et al. Sample-based attribute selective An DE for large data. IEEE Trans Knowl Data Eng, 2017, 29: 172–185

22 Zhang H, Jiang L X, Yu L J. Attribute and instance weighted naive Bayes. Pattern Recogn, 2021, 111: 107674

23 Duan Z Y, Wang L M, Chen S L, et al. Instance-based weighting filter for superparent one-dependence estimators. Knowl-Based Syst, 2020, 203: 106085

24 Zhang H, Petitjean F, Buntine W. Bayesian network classifiers using ensembles and smoothing. Knowl Inf Syst, 2020, 62: 3457–3480

25 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

26 Long Y G, Wang L M, Duan Z Y, et al. Robust structure learning of Bayesian network by identifying significant dependencies. IEEE Access, 2019, 7: 116661

27 Jiang L X. Random one-dependence estimators. Pattern Recogn Lett, 2011, 32: 532–539

28 Wu J, Pan S R, Zhu X Q, et al. Self-adaptive attribute weighting for naive Bayes classification. Expert Syst Appl, 2015, 42: 1487–1502

29 Jiang L X, Li C Q, Wang S S, et al. Deep feature weighting for naive Bayes and its application to text classification. Eng Appl Artifi Intell, 2016, 52: 26–39

30 Lee C H. A gradient approach for value weighted classification learning in naive Bayes. Knowl-Based Syst, 2015, 85: 71–79

31 Lee C H. An information-theoretic filter approach for value weighted classification learning in naive Bayes. Data Knowl Eng, 2018, 113: 116–128

32 Zhang H, Sheng S L. Learning weighted naive bayes with accurate ranking. In: Proceedings of the 4th International Conference on Data Mining, 2004. 567–570

33 Jiang L X, Zhang L G, Yu L J, et al. Class-specific attribute weighted naive Bayes. Pattern Recogn, 2019, 88: 321–330

34 Zhang H, Jiang L X, Yu L J. Class-specific attribute value weighting for naive Bayes. Inf Sci, 2020, 508: 260–274

35 Mahmoudi A, Yaakub M R, Bakar A A. The relationship between online social network ties and user attributes. ACM Trans Knowl Discov Data, 2019, 13: 26

36 Ali S, Shakeel M H, Khan I, et al. Predicting attributes of nodes using network structure. ACM Trans Intell Syst Technol, 2021, 12: 21

37 Jiang L X, Cai Z H, Zhang H, et al. Not so greedy: randomly selected naive Bayes. Expert Syst Appl, 2012, 39: 11022–11028

38 Wu J, Cai Z H. Attribute weighting via differential evolution algorithm for attribute weighted naive bayes (WNB). J Comput Inform Syst, 2011, 7: 1672–1679

39 Zhu C Y, Byrd R H, Lu P, et al. Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization. ACM Trans Math Softw, 1997, 23: 550–560

40 Breiman L. Random forests. Mach Learn, 2001, 45: 5–32

41 Witten I H, Frank E, Hall M A. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Amsterdam: Elsevier, 2011

42 Bengio Y, Nadeau C. Inference for the generalization error. Mach Learn, 2003, 52: 239–281

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

44 Olave M, Rajkovic V, Bohanec M. An application for admission in public school systems. Expert Syst Public Admin, 1989, 1: 145–160