Information Sciences 508 (2020) 260–274

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Class-specific attribute value weighting for Naive Bayes

Huan Zhang[a], Liangxiao Jiang[a,b,*], Liangjun Yu[c]

[a] Department of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China
[c] Department of Computer Science, Hubei University of Education, Wuhan 430205, China

ARTICLE INFO

Article history:
Received 8 April 2019
Revised 2 August 2019
Accepted 28 August 2019
Available online 28 August 2019

Keywords:
Naive Bayes
Attribute value weighting
Class-specific attribute values
Weight optimization
Classification

ABSTRACT

Naive Bayes (NB) is one of the top 10 data mining algorithms. However, its assumption of conditional independence rarely holds true in real-world applications. To alleviate this assumption, numerous attribute weighting approaches have been proposed. However, few of these simultaneously pay attention to the horizontal granularity of attribute values and vertical granularity of class labels. In this study, we propose a new paradigm for fine-grained attribute weighting, named class-specific attribute value weighting. For each class, this approach discriminatively assigns a specific weight to each attribute value. We refer to the resulting improved model as class-specific attribute value weighted NB (CAVWNB). In CAVWNB, the class-specific attribute value weight matrix is learned by either maximizing the conditional log-likelihood (CLL) or minimizing the mean squared error (MSE). Thus, two versions are proposed, which we denote as CAVWNB^{CLL} and CAVWNB^{MSE}, respectively. Extensive experimental results on a large number of datasets show that both CAVWNB^{CLL} and CAVWNB^{MSE} significantly outperform NB and all the other existing state-of-the-art attribute weighting approaches used for comparison.

1. Introduction

Classification is one of the most fundamental and significant tasks in data mining [18], and it is widely employed for text [29] and image [13] classification. When all the relevant prior probabilities are given, naive Bayes (NB) is often employed for classification learning, owing to its easy construction and surprising effectiveness [7,11,43].

Given a test instance $\mathbf{x}$ represented by an attribute value vector $\langle a_1, a_2, \ldots, a_m \rangle$, NB utilizes Eq. (1) to predict its class label.

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

where m is the number of attributes, a_j is the jth attribute value of $\mathbf{x}$, 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 the following m-estimation:

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

* Corresponding author.
E-mail address: ljiang@cug.edu.cn (L. Jiang).

https://doi.org/10.1016/j.ins.2019.08.071
0020-0255/

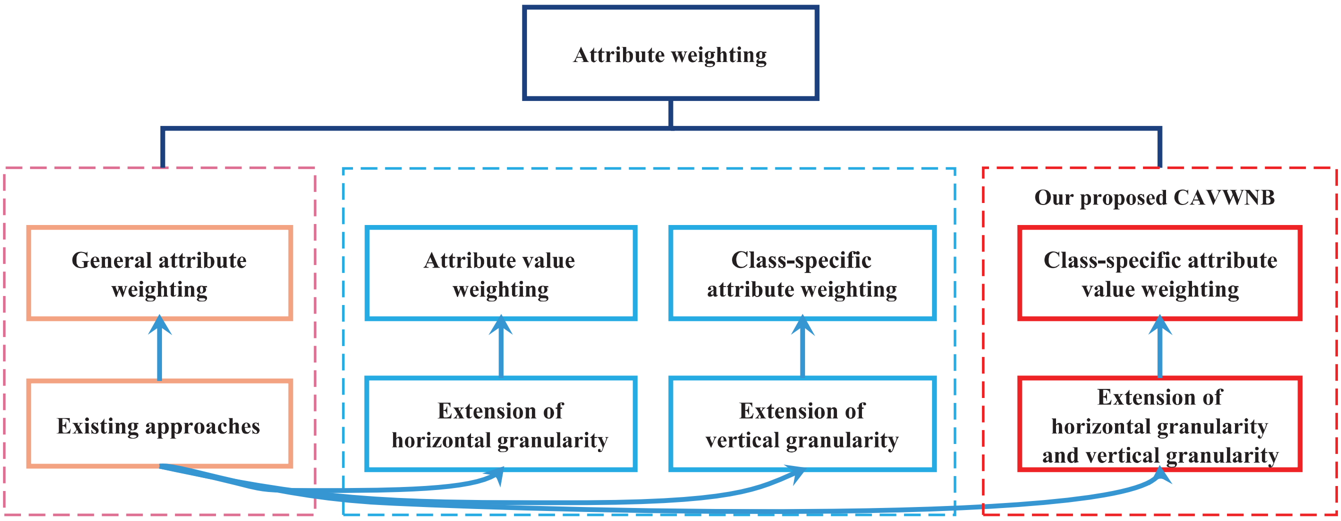

Fig. 1. Different attribute weighting paradigms for Naive Bayes.

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

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 $\delta(\cdot)$ is a binary function, which takes the value 1 if its two parameters are identical and 0 otherwise.

NB makes the conditional independence assumption that all attributes are fully independent given the class. However, this assumption is rarely true in reality, which would deteriorate its performance in the applications with sophisticated attribute dependencies. To alleviate NB's primary weakness, several enhancements have been proposed [14,26], which can be broadly divided into six main categories: structure extension [11,25,26], instance selection [10,22,38], instance weighting [20,23,44], attribute selection [3,21,37], attribute weighting [24,27,45], and fine tuning [2,6,19]. Among all methods, attribute weighting is a flexible method to relax the attribute conditional independence assumption, which assigns a continuous weight to each attribute between 0 and 1.

However, to the best of our knowledge most existing attribute weighting approaches assign each attribute the same weight for all classes. In this study, we refer to such approaches as general attribute weighting [33,47,48]. Recently, a few experts have suggested that fine-grained attribute weighting approaches may be more powerful for the NB classifier [28,31,32,46]. In Fig. 1, we summarize the proposed fine-grained approaches (blue rectangles) into two categories. One category is called attribute value weighting, which assigns each attribute value a different weight. Compared with general attribute weighting, methods in this category consider the horizontal granularity of attribute values. The other category is called class-specific attribute weighting, which assigns each class a different weight. Compared with general attribute weighting, approaches in this category consider the vertical granularity of class labels. Extensive empirical studies on the proposed fine-grained approaches have proven that both attribute value weighting and class-specific attribute weighting can obtain better performances than general attribute weighting.

However, despite the high performance of fine-grained attribute weighted NB, it has received less attention than it warrants. Motivated by the success of attribute value weighted NB and class-specific attribute weighted NB, we propose a novel fine-grained attribute weighting approach that can exploit attribute dependencies more accurately by considering both the vertical and horizontal granularities. As shown in Fig. 1 (red rectangles), we refer to the resulting model as class-specific attribute value weighted NB (CAVWNB). In CAVWNB, each attribute value is assigned a specific weight for each class in a discriminative manner. Compared with attribute value weighting, this approach considers the influences among different class labels. Compared with class-specific attribute weighting, CAVWNB can accurately estimate the influences of different attribute values. We believe that class-specific attribute value weighting could provide a more fine-grained attribute weighting approach for NB.

To learn the class-specific attribute value weight matrix of CAVWNB, two objective functions are designed, which either maximize the conditional log-likelihood (CLL) or minimize the mean squared error (MSE). Thus, two versions are proposed, which are simply denoted by $\text{CAVWNB}^{\text{CLL}}$ and $\text{CAVWNB}^{\text{MSE}}$, respectively. Two groups of extensive experiments are conducted in comparison with many state-of-the-art attribute weighting approaches, on a collection of 60 benchmark classification datasets from the University of California at Irvine (UCI) repository [8]. The comparison results show that both $\text{CAVWNB}^{\text{CLL}}$ and $\text{CAVWNB}^{\text{MSE}}$ can achieve better classification performances than their competitors.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive survey of various attribute weighting approaches for NB. Section 3 describes our proposed CAVWNB approach. Next, Section 4 describes the experi-

Table 1
Matrix representations of different attribute weighting methods.

(a) Matrix representation of general attribute weighting.

	A_1	A_2	...	A_{m-2}	A_{m-1}	A_m
c_1	w_1	w_2	...	w_{m-2}	w_{m-1}	w_m
c_2	w_1	w_2	...	w_{m-2}	w_{m-1}	w_m
...	...	...	...	...	...	...
c_{q-1}	w_1	w_2	...	w_{m-2}	w_{m-1}	w_m
c_q	w_1	w_2	...	w_{m-2}	w_{m-1}	w_m

(b) Matrix representation of class-specific attribute weighting.

	A_1	A_2	...	A_{m-2}	A_{m-1}	A_m
c_1	$w_{1,1}$	$w_{1,2}$	...	$w_{1,m-2}$	$w_{1,m-1}$	$w_{1,m}$
c_2	$w_{2,1}$	$w_{2,2}$	...	$w_{2,m-2}$	$w_{2,m-1}$	$w_{2,m}$
...	...	...	...	...	...	...
c_{q-1}	$w_{q-1,1}$	$w_{q-1,2}$	...	$w_{q-1,m-2}$	$w_{q-1,m-1}$	$w_{q-1,m}$
c_q	$w_{q,1}$	$w_{q,2}$	...	$w_{q,m-2}$	$w_{q,m-1}$	$w_{q,m}$

(c) Matrix representation of attribute value weighting.

	A_1		A_2		...	A_m	
	a_{11}	a_{12}	a_{21}	a_{22}	...	a_{m1}	a_{m2}
c_1	w_{11}	w_{12}	w_{21}	w_{22}	...	w_{m1}	w_{m2}
c_2	w_{11}	w_{12}	w_{21}	w_{22}	...	w_{m1}	w_{m2}
...	...	...	...	...	...	...	...
c_{q-1}	w_{11}	w_{12}	w_{21}	w_{22}	...	w_{m1}	w_{m2}
c_q	w_{11}	w_{12}	w_{21}	w_{22}	...	w_{m1}	w_{m2}

(d) Matrix representation of class-specific attribute value weighting.

	A_1		A_2		...	A_m	
	a_{11}	a_{12}	a_{21}	a_{22}	...	a_{m1}	a_{m2}
c_1	$w_{1,11}$	$w_{1,12}$	$w_{1,21}$	$w_{1,22}$	...	$w_{1,m1}$	$w_{1,m2}$
c_2	$w_{2,11}$	$w_{2,12}$	$w_{2,21}$	$w_{2,22}$	...	$w_{2,m1}$	$w_{2,m2}$
...	...	...	...	...	...	...	...
c_{q-1}	$w_{q-1,11}$	$w_{q-1,12}$	$w_{q-1,21}$	$w_{q-1,22}$	...	$w_{q-1,m1}$	$w_{q-1,m2}$
c_q	$w_{q,11}$	$w_{q,12}$	$w_{q,21}$	$w_{q,22}$	...	$w_{q,m1}$	$w_{q,m2}$

mental setup and presents the results. Finally, Section 5 concludes the study and outlines the main directions for future work.

2. Related work

2.1. General attribute weighted naive Bayes

Of the many approaches adopted to improve NB by relaxing its conditional independence assumption, general attribute weighting is known as a flexible model that discriminatively assigns a weight to each attribute. Attribute weights are incorporated into the NB formula to represent the significances of attributes as follows:

$$\hat{c}(\mathbf{x}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m} \theta_{A_j=a_j|c}^{w_j}, \tag{4}$$

where $w_j \in [0, 1]$ is the weight of the jth attribute A_j.

The weight matrix of general attribute weighted NB is shown in Table 1(a). Let us consider an instance that contains m attributes. General attribute weighting methods can then discriminatively learn m different weights for these. However, different attribute values in the same attribute share the same weight. In addition, the weight is class-independent (i.e., each attribute has the same importance for all class labels).

Existing general attribute weighting approaches fall into two broad categories: filters and wrappers. Filters utilize general data characteristics to obtain attribute weights, whereas attribute weights are computed directly based on heuristics before the classifier is run, to ensure that the final model places greater emphasis on more highly predictive attributes. The study [48] proposed a gain ratio-based attribute weighted NB (GRAWNB) model, which assumes that an attribute with a higher gain ratio deserves a larger weight, and vice versa. Instead of a gain ratio, [33] proposed a Kullback–Leibler (KL) measure-based attribute weighted NB (KLAWNB) model, which assumes that the weight of an attribute can be estimated based on the total information that the attribute gives to the class variable, calculated by the KL measure.

Although filters are considerably faster than wrappers in assigning attribute weights, wrappers can generally achieve a better classification accuracy, because the weights are determined based on the performance feedback from the classifier itself. Wrappers optimize attribute weights by minimizing the misclassification rate. The authors of [47] proposed a weighted NB algorithm called WANBIA to optimize attribute weights using gradient descent searches, by maximizing the CLL or minimizing the MSE.

2.2. Attribute value weighted Naive Bayes

Although general attribute weighting represents a flexible approach to relaxing the attribute conditional independence assumption of NB, methods of this type cannot estimate the influences of different attribute values of the same attribute. To test whether a fine-grained attribute weighting model can achieve a better performance, some experts have focused on a more fine-grained attribute weighting method, called attribute value weighting, for which the classification formula is

$$\hat{c}(\mathbf{x}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m} \theta_{A_j=a_j|c}^{w_{jk}}, \tag{5}$$

where $w_{jk} \in [0, 1]$ is the weight of the kth value of the jth attribute A_j.

The weight matrix of attribute value weighted NB is presented in Table 1(c). For simplicity, we assume that each attribute has two attribute values (i.e., the attribute A_j has the two attribute values a_{j1} and a_{j2}). In attribute value weighting methods, a_{j1} and a_{j2} can have different weights, whereas in general attribute weighting they can only share the same weight. In terms of attribute values, this model is more fine-grained than general attribute weighting, because each attribute value learns a different weight.

To date, only a few attribute value weighting approaches have been studied. We divide these into two broad categories: filters and wrappers. For filters, Lee [32] proposed a static filter based on the KL measure to estimate attribute value weights in NB learning. This attribute value weighting method could significantly improve the NB. However, this method only considers the attribute value–class relevance, and thus does not consider the attribute value–attribute value redundancy. The authors of [46] proposed a new filter, called the correlation-based attribute value weighted (CAVW) NB, by computing the difference between the attribute value–class relevance and average attribute value–attribute value redundancy. In CAVW, two attribute value weighting measures, the mutual information (MI) and KL measures, are employed. Thus, two versions are proposed, which we denote by CAVWMI and CAVWKL, respectively. Although filter approaches can be accelerated, the classification accuracy is no higher than that of wrappers. For wrappers, Lee [31] proposed a gradient approach to iteratively calculate the weight of each attribute value. Both filters and wrappers can obtain more satisfactory experimental results than the NB classifier and other existing attribute weighting approaches [31,32,46] (i.e., attribute value weighting is more fine-grained and effective than general attribute weighting).

2.3. Class-specific attribute weighted Naive Bayes

The strong performance of attribute value weighting has demonstrated the effectiveness of fine-grained attribute weighting methods. However, both general attribute weighting and attribute value weighting are class-independent, assigning a global weight for all class labels (i.e., they cannot reflect the dependencies between each attribute and each class label). Unfortunately, only one study on fine-grained attribute weighting in NB has focused on the vertical granularity of class labels. The authors of [28] proposed a new paradigm for attribute weighting, called class-specific attribute weighting, and proposed the class-specific attribute weighted NB (CAWNB) model, which discriminatively assigns a specific weight to each attribute for each class. The model assumes that the attribute importance should be different for each class. The classification formula of CAWNB is as follows:

$$\hat{c}(\mathbf{x}) = \arg\max_{c \in C} \pi_c \prod_{j=1}^{m} \theta_{A_j=a_j|c}^{w_{c,j}}, \tag{6}$$

where $w_{c,\,j} \in [0, 1]$ is the weight of the jth attribute A_j for the specific class c.

The weight matrix of class-specific attribute weighted NB is shown in Table 1(b). In class-specific attribute weighting, each attribute can have q different weights for q different class labels. For example, the jth attribute A_j can have q different weights for $c_1, c_2, \ldots, c_{q-1}$, and c_q, which are denoted by $w_{1,\,j}, w_{2,\,j}, \ldots, w_{q-1,\,j}$, and $w_{q,\,j}$, respectively. In terms of class labels, class-specific attribute weighting is more fine-grained than general attribute weighting, because each class label learns a different weight.

Although class-specific attribute weighting for NB has received less attention than it warrants, a few studies have been conducted that address other classifiers and algorithms. For example, Marchiori [35] proposed a decomposition of RELIEF into class-specific weight vectors, where each vector describes the relevance of attributes conditioned to one specific class. Both Jiang et al. [28] and Marchiori [35] validated the effectiveness of class-specific attribute weighting, thus providing a solid theoretical foundation for the feasibility of fine-grained attribute weighting.

3. Class-specific attribute value weighted Naive Bayes

3.1. Expansion of weight matrix

Existing fine-grained attribute weighting approaches either assign a different weight to each attribute value regardless of the class label or assign a different weight to each class for each attribute. However, in attribute value weighting, a combination function is needed to measure the importance of each attribute value for all classes, such as summation, maximization, or a weighted average. However, one of the main drawbacks of using a combination function is that this may bias the importance for different class labels. In addition, no reliable theory exists to instruct us on how to choose the best combination function. Therefore, the best combination operation must be determined through extensive experiments on classification applications [9]. For example, to analyze the importance of gender attributes on pregnancy-related disease classification problems, we must assign one weight to the male gender and another to the female gender to represent their respective importance. To our best knowledge, when the gender value is given, the gender value may have different effects on different class labels, and this cannot be reflected in attribute value weighting. However, class-specific attribute weighting is also unsatisfactory when the classification problem involves a strong dependency between the class labels and attribute values. The evident disadvantage is that existing class-specific approaches may bias the influences of different attribute values. In addition, if we consider the previously mentioned example, in class-specific attribute weighting we assign weights to class labels "yes" and "no" to the gender attributes. However, in this real-life problem it is obvious that the male gender attribute value has no effect on the class label, whereas the female gender attribute value has a considerable effect on the class label.

To overcome this drawback, in this study we propose a more fine-grained approach in the CAVWNB model, called class-specific attribute value weighting, where CAVWNB considers the effect of each attribute value and each class label. Table 1(d) shows the detailed weight matrix. In CAVWNB, each attribute value learns different weights with different class labels. We argue that different attribute values should have different weights for the same class label, and different class labels should also learn different weights for the same attribute value. The extension of the weight matrix may appear subtle, but this is in fact significant, because many real-world classification applications exist in which the complex dependencies cannot be estimated accurately unless the weight matrix is extended in terms of both horizontal and vertical granularities simultaneously.

3.2. Regularization

Because class-specific attribute value weighting is more fine-grained, it increases the number of parameters compared to other attribute weighting approaches. As an example, let us consider a classification problem with m attributes and q classes. We assume that r is the average number of values for an attribute. Therefore, the number of parameters to be estimated in the standard NB method is $qmr + q = q(mr + 1)$.

When we employ general attribute weighting for NB, m additional parameters are adopted for attribute weights. Therefore, the number of parameters in general attribute weighted NB is $qmr + q + m = q(mr + 1) + m$. Moreover, the number of parameters in attribute value weighted NB becomes $qmr + q + mr = q(mr + 1) + mr$. Furthermore, the number of parameters in class-specific attribute weighted NB becomes $qmr + q + qm = q(mr + 1 + m)$. Finally, the number of parameters in our class-specific attribute value weighted NB method becomes $qmr + q + qmr = q(2mr + 1)$. A greater number of parameters means that the model may be more adaptable to the characteristics of the training data. As a result, if the training instances contain considerable noise, then the complex model may also be more sensitive to noise than standard NB. Overfitting is a serious problem, which may occur particularly when the training data is sparse and has a high number of dimensions.

To solve this potential problem, we modify the error function by adding a penalty term that penalizes weights whose values are too different from the initial value. In this study, the original state of the weight matrix in CAVWNB is initialized to 1.0. We apply l_2 regularization, and thus add a $\lambda||\mathbf{w} - \mathbf{w_{one}}||^2$ penalty term in the error function. To preserve the simplicity of the final model (so that it is not remote from NB), we set all elements of $\mathbf{w_{one}}$ equal to 1.0, and the size of the matrix $\mathbf{w_{one}}$ equal to that of $\mathbf{w}$. Here, λ is a hyperparameter, which was also set to 1.0 in this study.

3.3. Objective function

The final issue to address is that of how to learn the weight matrix. As previously mentioned, Zaidi et al. [47] proposed an attribute weighted NB algorithm called WANBIA to adjust the weight matrix, using objective functions either maximizing the CLL or minimizing the MSE. The same objective functions are also adopted in CAWNB [28]. Both methods demonstrate the effectiveness of these objective functions. In this study, we follow WANBIA and CAWNB by optimizing the weight matrix of CAVWNB as a whole. Unlike WANBIA and CAWNB, whose weight matrix sizes are $1{*}m$ and $q{*}m$, respectively, the size of the weight matrix for our proposed CAVWNB is $q * \sum_{j=1}^{m} n_j$. Note that q is the number of classes, m is the number of attributes, and n_j is the number of values for the jth attribute A_j.

Given a test instance $\mathbf{x}$, our proposed CAVWNB uses Eqs. (7) and (8) to estimate its class membership probabilities and predict its class labels, respectively.

$$\hat{P}(c|\mathbf{x},\mathbf{w}) = \frac{\pi_c \prod_{j=1}^{m} \theta_{A_j=a_j|c}^{w_{c,jk}}}{\sum_{c'} \pi_{c'} \prod_{j=1}^{m} \theta_{A_j=a_j|c'}^{w_{c',jk}}}, \tag{7}$$

$$\hat{c}(\mathbf{x}) = \arg\max_{c\in C} \hat{P}(c|\mathbf{x},\mathbf{w}), \tag{8}$$

where $\mathbf{w}$ is the weight matrix to be optimized and $w_{c,\,jk} \in [0,\,1]$ is the weight of the kth attribute value of the jth attribute for the specific class c.

Following WANBIA [47] and CAWNB [28], we also employ the L-BFGS-M algorithm [49] to optimize the weight matrix. In CAVWNB, all of the empirical parameters utilized in the optimization procedure are consistent with WANBIA and CAWNB. The entire learning algorithm for our CAVWNB approach can be partitioned into training (CAVWNB-training) and classification (CAVWNB-classification) algorithms. These are briefly depicted in Algorithms 1 and 2, respectively.

Algorithm 1 CAVWNB-training (D, f).

Input: D-a training dataset; f-an objective function to be optimized
Output: $\mathbf{w}$-the optimized weight matrix
1: Estimate the prior probability π_c of each class c by Eq. (2)
2: Estimate the conditional probability $\theta_{A_j=a_j|c}$ of each attribute value a_j given the class c by Eq. (3)
3: Initialize all class-specific attribute value weights in the weight matrix to 1.0
4: Invoke the optimization procedure L-BFGS-M to optimize the given objective function f and obtain the optimized weight matrix $\mathbf{w}$
5: Return $\mathbf{w}$

Algorithm 2 CAVWNB-classification ($\mathbf{w}$, $\mathbf{x}$).

Input: $\mathbf{w}$-the optimized weight matrix; $\mathbf{x}$-a test instance
Output: $\hat{c}(\mathbf{x})$-The predicted class label of $\mathbf{x}$
1: Estimate the class membership probability $\hat{P}(c|\mathbf{x},\mathbf{w})$ of each class c by Eq. (7)
2: Predict the class label $\hat{c}(\mathbf{x})$ of $\mathbf{x}$ by Eq. (8)
3: Return $\hat{c}(\mathbf{x})$

The definition of a suitable objective function f is crucial. Inspired by WANBIA and CAWNB, we redefine the two objective functions for the CLL and MSE. Based on these two objective functions, two gradient-based versions are created, which we denote as CAVWNB$^{\text{CLL}}$ and CAVWNB$^{\text{MSE}}$, where these maximize the CLL and minimize the MSE, respectively.

The redefined CLL function can be defined as

$$\begin{aligned} CLL(\mathbf{w}) &= \log \hat{P}(\mathcal{L}|\mathcal{D},\mathbf{w}) + \lambda||\mathbf{w}-\mathbf{w_{one}}||^2 \\ &= \sum_{i=1}^{|\mathcal{D}|} \log \hat{P}(c_i|\mathbf{x_i},\mathbf{w}) + \lambda||\mathbf{w}-\mathbf{w_{one}}||^2 \\ &= \sum_{i=1}^{|\mathcal{D}|} \log \frac{\gamma_{c\mathbf{x_i}}(\mathbf{w})}{\sum_{c'} \gamma_{c'\mathbf{x_i}}(\mathbf{w})} + \lambda||\mathbf{w}-\mathbf{w_{one}}||^2, \end{aligned} \tag{9}$$

where

$$\mathbf{w_{one}} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}, \tag{10}$$

$$\gamma_{c\mathbf{x_i}}(\mathbf{w}) = \pi_c \prod_j \theta_{a_j|c}^{w_{c,jk}}, \tag{11}$$

and a_j is the attribute value of the jth attribute of $\mathbf{x_i}$.

To maximize CLL($\mathbf{w}$), the gradient of $\gamma_{c\mathbf{x_i}}(\mathbf{w})$ is represented as follows:

$$\begin{aligned}\frac{\partial}{\partial w_{c,jk}}\gamma_{c\mathbf{x_i}}(\mathbf{w}) &= \left(\pi_c \prod_{j'\neq j}\theta_{a_{j'}|c}^{w_{c,j'k'}}\right)\frac{\partial}{\partial w_{c,jk}}\theta_{a_j|c}^{w_{c,jk}} \\ &= \left(\pi_c \prod_{j'\neq j}\theta_{a_{j'}|c}^{w_{c,j'k'}}\right)\theta_{a_j|c}^{w_{c,jk}}\log(\theta_{a_j|c}) \\ &= \gamma_{c\mathbf{x_i}}(\mathbf{w})\log(\theta_{a_j|c}),\end{aligned} \tag{12}$$

where $w_{c,j'k'}$ is the weight of the k'th attribute value of the j'th attribute for the specific class c.

The gradient of CLL($\mathbf{w}$) is represented as follows:

$$\begin{aligned}\frac{\partial}{\partial w_{c,jk}}CLL(\mathbf{w}) &= \frac{\partial}{\partial w_{c,jk}}\sum_{\mathbf{x_i}\in\mathcal{D}}\left[\log(\gamma_{c\mathbf{x_i}}(\mathbf{w})) - \log\left(\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})\right)\right] + 2\lambda(w_{c,jk}-1) \\ &= \sum_{\mathbf{x_i}\in\mathcal{D}}\left[\delta(c_i,c)\frac{\gamma_{c\mathbf{x_i}}(\mathbf{w})\log(\theta_{a_j|c})}{\gamma_{c\mathbf{x_i}}(\mathbf{w})} - \frac{\gamma_{c\mathbf{x_i}}(\mathbf{w})\log(\theta_{a_j|c})}{\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})}\right] + 2\lambda(w_{c,jk}-1) \\ &= \sum_{\mathbf{x_i}\in\mathcal{D}}[\delta(c_i,c)\log(\theta_{a_j|c}) - \hat{P}(c|\mathbf{x_i})\log(\theta_{a_j|c})] + 2\lambda(w_{c,jk}-1).\end{aligned} \tag{13}$$

The MSE function can be defined as

$$MSE(\mathbf{w}) = \frac{1}{2}\sum_{\mathbf{x_i}\in\mathcal{D}}\sum_{c}(P(c|\mathbf{x_i}) - \hat{P}(c|\mathbf{x_i}))^2 + \lambda||\mathbf{w}-\mathbf{w_{one}}||^2, \tag{14}$$

where

$$P(c|\mathbf{x_i}) = \begin{cases}1, & \text{if } c = c_i \\ 0, & \text{if } otherwise\end{cases}. \tag{15}$$

The gradient of MSE($\mathbf{w}$) is

$$\frac{\partial MSE(\mathbf{w})}{\partial w_{c,jk}} = -\sum_{\mathbf{x_i}\in\mathcal{D}}(P(c|\mathbf{x_i}) - \hat{P}(c|\mathbf{x_i}))\frac{\partial\hat{P}(c|\mathbf{x_i})}{\partial w_{c,jk}} + 2\lambda(w_{c,jk}-1). \tag{16}$$

To minimize MSE($\mathbf{w}$), the gradient of $\hat{P}(c|\mathbf{x_i})$ is represented as

$$\begin{aligned}\frac{\partial\hat{P}(c|\mathbf{x_i})}{\partial w_{c,jk}} &= \frac{\frac{\partial}{\partial w_{c,jk}}\gamma_{c\mathbf{x_i}}(\mathbf{w})}{\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})} - \frac{\gamma_{c\mathbf{x_i}}(\mathbf{w})\frac{\partial}{\partial w_{c,jk}}\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})}{(\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w}))^2} \\ &= \frac{1}{\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})}\left[\frac{\partial\gamma_{c\mathbf{x_i}}(\mathbf{w})}{\partial w_{c,jk}} - \hat{P}(c|\mathbf{x_i})\frac{\partial\gamma_{c\mathbf{x_i}}(\mathbf{w})}{\partial w_{c,jk}}\right] \\ &= \frac{1}{\sum_{c'}\gamma_{c'\mathbf{x_i}}(\mathbf{w})}(1-\hat{P}(c|\mathbf{x_i}))\gamma_{c\mathbf{x_i}}(\mathbf{w})\log(\theta_{a_j|c}) \\ &= \hat{P}(c|\mathbf{x_i})(1-\hat{P}(c|\mathbf{x_i}))\log(\theta_{a_j|c}).\end{aligned} \tag{17}$$

Therefore, the gradient of MSE($\mathbf{w}$) is

$$\begin{aligned}\frac{\partial MSE(\mathbf{w})}{\partial w_{c,jk}} &= -\sum_{\mathbf{x_i}\in\mathcal{D}}(P(c|\mathbf{x_i}) - \hat{P}(c|\mathbf{x_i}))\hat{P}(c|\mathbf{x_i})(1-\hat{P}(c|\mathbf{x_i}))\log(\theta_{a_j|c}) + 2\lambda(w_{c,jk}-1) \\ &= -\sum_{\mathbf{x_i}\in\mathcal{D}}(\hat{P}(c|\mathbf{x_i})(\delta(c_i,c) - \hat{P}(c|\mathbf{x_i}))(1-\hat{P}(c|\mathbf{x_i}))\log(\theta_{a_j|c}) + 2\lambda(w_{c,jk}-1).\end{aligned} \tag{18}$$

4. Experiments and results

4.1. Experimental settings and benchmark data

We attempted to validate the performance of our proposed CAVWNB by running a series of experiments comparing our proposed CAVWNB^{CLL} and CAVWNB^{MSE} methods to standard NB and some existing state-of-the-art attribute weighting and attribute value weighting algorithms. These competitors and their abbreviations are listed as follows.

- CAWNB^{CLL}: NB with CLL-based class-specific attribute weighting [28].
- CAWNB^{MSE}: NB with MSE-based class-specific attribute weighting [28].

- WANBIA^{CLL}: NB with CLL-based attribute weighting [47].
- WANBIA^{MSE}: NB with MSE-based attribute weighting [47].
- CAVWMI: NB with correlation-based attribute value weighting measured by MI [46].
- GRAWNB: NB with gain ratio-based attribute weighting [48].
- KLAWNB: NB with Kullback–Leibler measure-based attribute weighting [33].
- NB: standard NB [30].

Experiments were conducted on a collection of 60 benchmark classification datasets from the University of California at Irvine (UCI) repository [8,34,42], which represent a wide range of domains and data characteristics. Table 2 lists the properties of these datasets. Because all the compared algorithms could only deal with nominal attributes without missing values, in our experiments, we first replaced all missing attribute values using the unsupervised attribute filter *ReplaceMissingValues* in the Waikato Environment for Knowledge Analysis (WEKA) platform [39]. We then applied the unsupervised filter *Discretize* in the WEKA platform to discretize the numeric attributes into nominal attributes. These data preprocessing steps were also utilized in the related studies [40,41]. In addition, we manually deleted four useless attributes in advance: "Hospital Number" in "colic.ORIG," "name" in "parkinsons," "instance name" in "splice," and "animal" in "zoo." Apparently, if the number of values of an attribute was equal to the number of instances in a dataset, then this it was a useless attribute.

4.2. Results and analysis

The experimental results were obtained by averaging the results from 10 separate runs of stratified 10-fold cross-validation. Tables 3 and 5 present detailed comparison results in terms of the classification accuracy. For each row, a field marked with • and ∘ signifies that the classification accuracy of CAVWNB was statistically and significantly better (upgradation) or worse (degradation) compared to the algorithm shown in the corresponding column, respectively, when paired two-tailed t-tests with a $p = 0.05$ significance level were conducted [36]. The averages and win/tie/lose (W/T/L) values are summarized at the bottoms of the tables. In addition to other statistics, each average (arithmetic mean) across all datasets provides a gross indication of the relative performance. Each entry's W/T/L in a table represents that, compared to their competitors, CAVWNB^{CLL} and CAVWNB^{MSE} won on *W* datasets, tied on *T* datasets, and lost on *L* datasets, respectively.

Simultaneously, based on the classification accuracy results presented in Tables 3 and 5, we employed the Knowledge Extraction based on Evolutionary Learning data-mining software tool [1] to conduct the Wilcoxon signed-ranks test [5,12], to obtain a thorough comparison between each pair of algorithms. The Wilcoxon signed-ranks test is a non-parametric statistical test, which ranks the differences in the performances of pairs of algorithms for each dataset while ignoring the signs, and compares the ranks for positive and negative differences. Tables 4 and 6 summarize the detailed comparison results. In Tables 4 and 6, ∘ indicates that the algorithm listed in the column outperformed the one in the corresponding row, and • indicates that the algorithm in the row performed better than that in the corresponding column. The lower-diagonal level of significance was $\alpha = 0.05$, and the upper-diagonal level was $\alpha = 0.1$.

These comparison results clearly show that our proposed CAVWNB method significantly outperformed not only the standard NB, but also all of the other attribute weighting algorithms, including KLAWNB, GRAWNB, WANBIA, and CAWNB. It also outperformed the attribute value weighting algorithm CAVWMI. For simplicity, we summarize these findings as follows:

- The averaged classification accuracies of CAVWNB^{CLL} and CAVWNB^{MSE} on 60 datasets were 83.56% and 83.40%, which were considerably higher than those of the standard NB (80.68%) and their state-of-the-art competitors: CAWNB^{CLL} (82.64%), WANBIA^{CLL} (82.11%), CAWNB^{MSE} (82.27%), WANBIA^{MSE} (82.30%), CAVWMI (80.61%), GRAWNB (80.23%), and KLAWNB (79.12%).
- Based on two-tailed t-tests, CAVWNB^{CLL} and CAVWNB^{MSE} performed the best overall. CAVWNB^{CLL} was better than CAWNB^{CLL} (14 wins and 2 losses), WANBIA^{CLL} (18 wins and 1 loss), CAVWMI (27 wins and 3 losses), GRAWNB (30 wins and 1 loss), KLAWNB (33 wins and 1 loss), and NB (28 wins and 2 losses). CAVWNB^{MSE} was better than CAWNB^{MSE} (12 wins and 3 losses), WANBIA^{MSE} (18 wins and 2 losses), CAVWMI (28 wins and 1 loss), GRAWNB (31 wins and 0 losses), KLAWNB (33 wins and 0 losses), and NB (29 wins and 0 losses).
- Based on the Wilcoxon signed-ranks test results, we can conclude that when the level significance was $\alpha = 0.1$, our proposed CAVWNB significantly outperformed NB and all other five existing state-of-the-art approaches (KLAWNB, GRAWNB, CAVWMI, WANBIA, and CAWNB). This fully verifies the universal applicability of our CAVWNB method for a wide range of domains and data characteristics.
- It is particularly noteworthy that our CAVWNB method can achieve significant improvements on certain datasets, such as "artificial-characters," "texture," and "vowel." From Tables 3 and 5, we can observe that on the dataset "vowel," the classification accuracy of CAVWNB^{CLL} increased by 10.89% compared to that of CAWNB^{CLL}, and even by 14.40% compared to NB. Similarly, the classification accuracy of CAVWNB^{MSE} increased by 11.82% compared to that of CAWNB^{MSE}, and was 13.76% higher than that of NB. In other words, our proposed CAVWNB method could learn the complex dependencies more accurately, and thus the performance of CAVWNB was significantly better than those of the other state-of-the-art approaches.

Table 2
Descriptions of 60 UCI datasets used in the experiments.

Dataset name	Instance number	Attribute number	Class number	Missing values	Numeric values
anneal	898	39	6	Y	Y
anneal.ORIG	898	39	6	Y	Y
artificial-characters	10218	8	10	N	Y
audiology	226	70	24	Y	N
autos	205	26	7	Y	Y
balance-scale	625	5	3	N	Y
breast-cancer	286	10	2	Y	N
breast-w	699	10	2	Y	N
car	1728	7	4	N	N
cardiotocography	2126	22	3	N	Y
climate-simulation-craches	540	21	2	N	Y
colic	368	23	2	Y	Y
colic.ORIG	368	28	2	Y	Y
connectionist-vowel	528	11	11	N	Y
credit-a	690	16	2	Y	Y
credit-g	1000	21	2	N	Y
cylinder-bands	540	41	2	Y	Y
diabetes	768	9	2	N	Y
ecoli	336	8	8	N	Y
energy-efficiency-y1	768	9	37	N	Y
energy-efficiency-y2	768	9	38	N	Y
glass	214	10	7	N	Y
hayes-roth	160	5	3	N	Y
heart-c	303	14	5	Y	Y
heart-h	294	14	5	Y	Y
heart-statlog	270	14	2	N	Y
hepatitis	155	20	2	Y	Y
hypothyroid	3772	30	4	Y	Y
ionosphere	351	35	2	N	Y
iris	150	5	3	N	Y
kr-vs-kp	3196	37	2	N	N
labor	57	17	2	Y	Y
letter	20000	17	26	N	Y
libras	360	91	15	N	Y
lymph	148	19	4	N	Y
mfeat-f	2000	77	10	N	Y
monks	556	7	2	N	Y
mushroom	8124	23	2	Y	N
newthyroid	215	6	3	N	Y
optdigits	5620	63	10	N	Y
page-blocks	5473	11	5	N	Y
parkinsons	195	23	2	N	Y
pendigits	10992	17	10	N	Y
primary-tumor	339	18	21	Y	N
qar-biodegradation	1055	42	2	N	Y
robot-24	5456	25	4	N	Y
segment	2310	20	7	N	Y
sick	3772	30	2	Y	Y
sonar	208	61	2	N	Y
soybean	683	36	19	Y	N
spectrometer	531	102	48	N	Y
splice	3190	62	3	N	N
steel-plates-faults	1941	34	2	N	Y
texture	5500	41	11	N	Y
thyroid-disease	7200	22	3	N	Y
vehicle	846	19	4	N	Y
vote	435	17	2	Y	N
vowel	990	14	11	N	Y
waveform-5000	1000	41	3	N	Y
zoo	101	18	7	N	Y

4.3. Case study

Moreover, to further reveal the advantages and applications of our CAVWNB method, we observed its performance on another typical dataset from the UCI repository. This dataset is related to the Connect-4 game [4].[1] Connect-4 is a game

[1] http://archive.ics.uci.edu/ml/datasets/Connect-4.

Table 3
Classification accuracy comparison for CAVWNB^{CLL} versus CAWNB^{CLL}, WANBIA^{CLL}, CAVWMI, GRAWNB, KLAWNB, and NB.

Dataset	CAVWNB^{CLL}	CAWNB^{CLL}	WANBIA^{CLL}	CAVWMI	GRAWNB	KLAWNB	NB
anneal	99.23 ± 0.82	98.60 ± 1.08	98.00 ± 1.40•	97.62 ± 1.48•	97.29 ± 1.66•	90.53 ± 3.74•	96.34 ± 1.80•
anneal.ORIG	91.76 ± 2.61	91.06 ± 2.69	90.89 ± 2.79	89.84 ± 3.19•	81.38 ± 4.16•	79.66 ± 3.93•	88.17 ± 3.12•
artificial-characters	46.52 ± 1.41	38.71 ± 1.26•	36.08 ± 1.37•	35.84 ± 1.34•	35.87 ± 1.43•	36.16 ± 1.44•	36.52 ± 1.34•
audiology	77.02 ± 6.62	82.10 ± 6.86○	78.08 ± 6.93	75.78 ± 8.13	75.95 ± 6.87	72.90 ± 8.54	75.74 ± 6.58
autos	75.94 ± 9.97	75.08 ± 9.68	74.98 ± 9.16	68.38 ± 10.24•	68.33 ± 10.53•	67.57 ± 11.43•	66.12 ± 11.12•
balance-scale	91.94 ± 0.74	91.44 ± 1.29	91.44 ± 1.29	87.97 ± 2.21•	90.26 ± 1.89•	90.26 ± 1.89•	91.44 ± 1.29
breast-cancer	68.57 ± 7.38	69.53 ± 7.37	71.00 ± 7.41	72.14 ± 7.49	71.54 ± 7.84	69.75 ± 8.72	72.32 ± 7.91○
breast-w	96.07 ± 1.97	96.20 ± 2.18	96.88 ± 2.04	97.28 ± 1.74○	97.31 ± 1.76○	97.48 ± 1.70○	97.31 ± 1.73○
car	90.12 ± 2.10	86.69 ± 2.55•	85.69 ± 2.51•	70.79 ± 0.74•	80.21 ± 3.06•	80.58 ± 3.16•	85.69 ± 2.51•
cardiotocography	91.96 ± 1.51	91.22 ± 1.92	90.79 ± 1.85	86.40 ± 2.12•	85.59 ± 2.36•	84.39 ± 2.34•	84.17 ± 2.22•
climate-simulation-craches	86.93 ± 3.11	88.89 ± 2.25	88.72 ± 2.31	90.48 ± 1.51○	76.74 ± 6.55•	76.63 ± 6.58•	87.20 ± 3.17
colic	79.90 ± 6.14	81.39 ± 5.99	82.69 ± 5.77	82.18 ± 5.71	82.28 ± 5.97	84.00 ± 5.51	78.65 ± 6.08
colic.ORIG	75.95 ± 6.48	76.77 ± 5.82	74.26 ± 6.26	74.40 ± 6.40	73.83 ± 6.72	71.11 ± 6.29•	73.67 ± 6.63
connectionist-vowel	82.75 ± 5.29	79.35 ± 5.69•	77.40 ± 5.83•	77.18 ± 5.84•	73.29 ± 6.06•	74.58 ± 5.86•	76.89 ± 6.26•
credit-a	84.14 ± 4.27	85.28 ± 3.80	85.29 ± 3.90	86.01 ± 3.68	85.51 ± 3.96	85.51 ± 3.96	84.33 ± 3.85
credit-g	74.94 ± 3.63	75.48 ± 3.75	76.13 ± 3.91	75.53 ± 3.43	70.10 ± 4.41•	70.15 ± 4.61•	75.75 ± 3.96
cylinder-bands	81.28 ± 5.01	77.81 ± 5.50•	78.89 ± 5.21•	81.09 ± 5.17	80.72 ± 5.08	77.56 ± 5.24•	81.46 ± 5.25
diabetes	75.14 ± 4.04	75.88 ± 4.90	76.15 ± 4.75	75.32 ± 4.36	75.51 ± 5.65	75.39 ± 5.57	75.12 ± 4.93
ecoli	83.60 ± 5.54	83.93 ± 6.01	83.75 ± 6.24	82.26 ± 5.50	82.83 ± 5.26	82.12 ± 6.01	84.78 ± 5.79
energy-efficiency-y1	55.35 ± 4.40	55.43 ± 4.54	51.98 ± 4.66•	49.35 ± 4.70•	47.88 ± 5.25•	44.11 ± 4.95•	50.37 ± 4.88•
energy-efficiency-y2	52.09 ± 4.60	51.98 ± 4.68	51.03 ± 4.51	49.32 ± 4.80•	50.49 ± 4.59	45.86 ± 4.40•	49.14 ± 4.84•
glass	59.41 ± 8.68	59.06 ± 9.06	59.87 ± 9.96	58.70 ± 9.13	61.31 ± 9.54	57.41 ± 10.57	60.00 ± 9.81
hayes-roth	84.50 ± 7.30	85.00 ± 7.85	84.37 ± 7.77	82.56 ± 8.53	85.25 ± 8.02	85.06 ± 7.99	83.88 ± 8.25
heart-c	80.97 ± 7.36	81.29 ± 6.72	82.18 ± 6.70	81.23 ± 6.95	83.74 ± 6.22	82.71 ± 6.99	82.65 ± 6.33
heart-h	82.15 ± 5.89	83.34 ± 5.84	84.22 ± 5.62	82.79 ± 6.25	83.38 ± 6.80	81.20 ± 6.66	83.71 ± 6.18
heart-statlog	81.78 ± 6.04	82.26 ± 6.10	82.96 ± 6.07	82.30 ± 5.95	83.52 ± 6.26	82.96 ± 6.05	83.30 ± 5.58
hepatitis	83.09 ± 8.88	84.95 ± 8.59	84.35 ± 8.85	85.86 ± 8.82	83.16 ± 10.14	84.89 ± 9.73	85.86 ± 9.12
hypothyroid	93.50 ± 0.58	93.53 ± 0.59	93.58 ± 0.55	93.53 ± 0.51	89.44 ± 1.31•	77.07 ± 2.07•	92.68 ± 0.75•
ionosphere	91.23 ± 4.30	91.83 ± 4.30	91.82 ± 4.10	91.09 ± 4.33	92.17 ± 4.17	91.86 ± 4.20	90.94 ± 4.21
iris	95.33 ± 5.88	96.47 ± 4.49	96.60 ± 4.78	93.67 ± 6.52	95.93 ± 5.27	95.93 ± 5.27	94.20 ± 6.47
kr-vs-kp	95.08 ± 1.16	94.31 ± 1.17•	93.43 ± 1.30•	90.21 ± 1.79•	89.67 ± 1.67•	90.83 ± 1.71•	87.81 ± 1.90•
labor	94.60 ± 9.41	94.07 ± 9.73	93.80 ± 10.82	93.33 ± 11.38	93.57 ± 9.42	91.47 ± 11.06	94.93 ± 9.00
letter	77.64 ± 1.67	71.25 ± 1.98•	68.42 ± 2.06•	67.89 ± 2.02•	68.91 ± 2.19•	69.23 ± 2.12•	67.14 ± 1.97•
libras	71.92 ± 7.08	68.44 ± 7.37	68.31 ± 7.84	70.36 ± 6.94	70.42 ± 6.52	70.36 ± 6.56	69.47 ± 7.10
lymph	84.05 ± 8.81	82.37 ± 9.64	84.09 ± 9.50	83.67 ± 8.71	81.42 ± 8.82	81.37 ± 8.97	85.09 ± 9.04
mfeat-f	77.96 ± 2.66	78.20 ± 2.73	78.67 ± 2.68	77.47 ± 2.59	79.01 ± 2.43	79.17 ± 2.46	76.97 ± 2.61
monks	74.64 ± 4.26	74.64 ± 4.26	74.64 ± 4.26	73.99 ± 4.72	74.64 ± 4.26	74.64 ± 4.26	74.64 ± 4.26
mushroom	99.82 ± 0.31	99.80 ± 0.34	99.69 ± 0.41	97.07 ± 1.50•	98.33 ± 0.98•	98.67 ± 0.89•	95.99 ± 1.58•
newthyroid	95.16 ± 4.44	95.25 ± 4.26	93.39 ± 4.14	92.92 ± 4.98	93.15 ± 4.94	94.18 ± 5.19	93.20 ± 5.01
optdigits	95.08 ± 0.85	95.62 ± 0.86	93.94 ± 1.00•	92.48 ± 1.09•	92.26 ± 1.09•	92.46 ± 1.11•	92.39 ± 1.08•
page-blocks	93.87 ± 0.73	93.16 ± 0.81•	92.77 ± 0.79•	92.32 ± 0.77•	91.95 ± 0.69•	86.82 ± 1.30•	92.59 ± 0.92•
parkinsons	91.33 ± 5.41	89.42 ± 6.80	90.39 ± 5.90	82.38 ± 8.75•	81.17 ± 8.98•	83.83 ± 8.73•	80.84 ± 8.52•
pendigits	97.19 ± 0.45	93.47 ± 0.73•	88.55 ± 1.00•	87.54 ± 1.02•	86.52 ± 1.02•	86.41 ± 1.01•	87.49 ± 1.00•
primary-tumor	48.26 ± 6.00	46.11 ± 5.94	47.52 ± 6.04	47.29 ± 5.55	45.56 ± 6.80	46.14 ± 7.14	47.11 ± 5.65
qar-biodegradation	83.94 ± 3.58	85.15 ± 3.33	85.10 ± 3.63	80.99 ± 4.08•	79.13 ± 4.16•	77.49 ± 4.55•	79.76 ± 4.14•
robot-24	92.56 ± 1.05	89.38 ± 1.26•	86.59 ± 1.41•	82.99 ± 1.58•	83.76 ± 1.50•	84.35 ± 1.42•	81.01 ± 1.61•
segment	93.99 ± 1.58	92.75 ± 1.76•	92.48 ± 1.74•	90.25 ± 1.60•	88.31 ± 1.74•	87.90 ± 1.73•	90.07 ± 1.65•
sick	97.71 ± 0.75	97.70 ± 0.76	97.38 ± 0.77	97.47 ± 0.76	96.47 ± 0.93•	96.47 ± 0.93•	96.94 ± 0.84•
sonar	77.63 ± 9.43	76.66 ± 8.77	75.56 ± 10.14	75.33 ± 10.33	74.85 ± 9.87	75.90 ± 9.47	75.86 ± 9.87
soybean	93.96 ± 2.58	94.45 ± 2.23	93.92 ± 2.35	93.68 ± 2.81	92.30 ± 3.05	92.83 ± 2.89	93.53 ± 2.79
spectrometer	48.36 ± 5.48	46.38 ± 5.65	47.72 ± 5.99	49.55 ± 6.35	49.23 ± 6.09	49.63 ± 6.14	48.93 ± 6.16
splice	95.03 ± 1.06	96.20 ± 1.04○	96.05 ± 1.06○	96.03 ± 1.04○	94.15 ± 1.26•	94.61 ± 1.23	95.58 ± 1.12
steel-plates-faults	99.66 ± 0.42	99.88 ± 0.23	98.32 ± 0.82•	94.87 ± 1.53•	99.37 ± 0.57	95.83 ± 1.58•	96.68 ± 1.38•
texture	96.44 ± 0.85	88.80 ± 1.15•	84.00 ± 1.18•	80.01 ± 1.62•	80.08 ± 1.54•	79.91 ± 1.53•	79.87 ± 1.62•
thyroid-disease	93.92 ± 0.34	93.90 ± 0.33	93.92 ± 0.36	93.38 ± 0.48•	93.09 ± 0.54•	79.65 ± 1.39•	93.51 ± 0.48•
vehicle	70.96 ± 4.21	64.65 ± 3.91•	64.43 ± 3.86•	61.27 ± 3.40•	60.41 ± 3.33•	59.02 ± 3.34•	61.07 ± 3.41•
vote	95.84 ± 2.86	95.81 ± 2.76	95.56 ± 2.89	90.67 ± 3.86•	93.03 ± 3.57•	93.01 ± 3.54•	90.30 ± 3.89•
vowel	82.19 ± 4.09	71.30 ± 4.60•	70.34 ± 4.46•	68.35 ± 4.74•	67.32 ± 4.40•	67.07 ± 4.50•	67.79 ± 4.67•
waveform-5000	85.29 ± 3.11	83.30 ± 3.36•	81.39 ± 3.28•	79.75 ± 2.94•	78.58 ± 2.99•	78.66 ± 2.94•	80.03 ± 2.87•
zoo	96.44 ± 4.99	95.35 ± 5.32	96.03 ± 5.66	96.25 ± 5.18	96.24 ± 5.04	93.97 ± 6.43	96.04 ± 5.65
Average	83.56	82.64	82.11	80.61	80.23	79.12	80.68
W/T/L	-	14/44/2	18/41/1	27/30/3	30/29/1	33/26/1	28/30/2

for two players, X and O, who take turns marking spaces in a 6×7 grid. The player who succeeds in placing four of their marks in a horizontal, vertical, or diagonal row wins the game. This database encodes the complete set of possible board configurations at the end of a Connect-4 game, where X is assumed to have played first. This database contains all legal eight-ply positions in the game of Connect-4 in which neither player has yet won, and in which the next move is not forced. The attribute set consists of 42 nominal primitives corresponding to the board locations, which are labeled as $a1$ to $g6$. Each

Table 4
Summary of the Wilcoxon test with respect to CAVWNB^{CLL}.

Algorithm	CAVWNB^{CLL}	CAWNB^{CLL}	WANBIA^{CLL}	CAVWMI	GRAWNB	KLAWNB	NB
CAVWNB^{CLL}	–	•	•	•	•	•	•
CAWNB^{CLL}		–	•	•	•	•	•
WANBIA^{CLL}	○	○	–	•	•	•	•
CAVWMI	○	○	○	–		•	
GRAWNB	○	○	○		–	•	
KLAWNB	○	○	○	○	○	–	○
NB	○	○	○			•	–

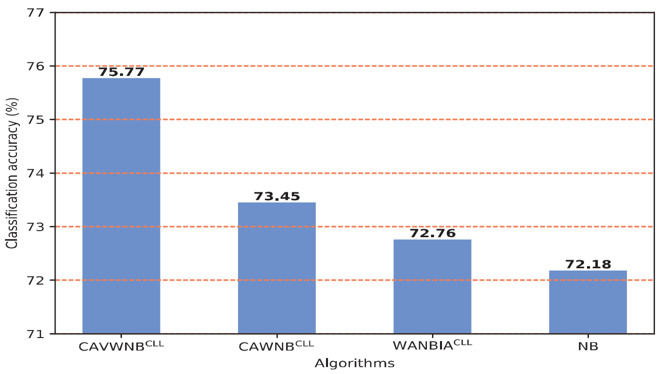

Fig. 2. Classification accuracy comparison for CAVWNB^{CLL} versus CAWNB^{CLL}, WANBIA^{CLL}, and NB on the Connect-4 dataset.

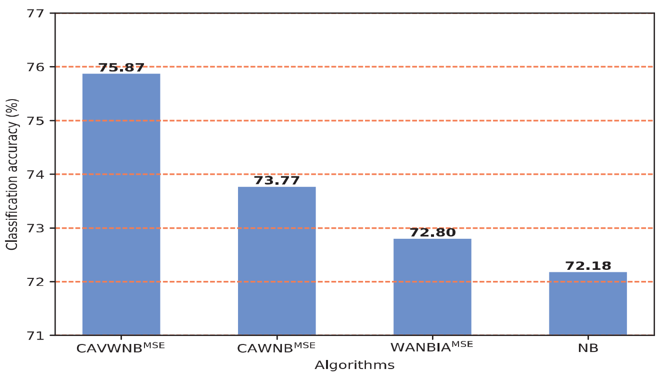

Fig. 3. Classification accuracy comparison for CAVWNB^{MSE} versus CAWNB^{MSE}, WANBIA^{MSE}, and NB on the Connect-4 dataset.

attribute has three attribute values: "x," "o," and "b." An outcome class of "win," "loss," or "draw" is the game-theoretical value for the first player.

Based on the domain knowledge,[2] the content in each grid is strongly related to the final winner in Connect-4. However, in general attribute value weighting, the attribute value's impact on different class labels is ignored. Similarly, in class-specific attribute weighting, only the attribute's influence on the class label is considered, rather than the attribute values. More specifically, this implies that the same position in the grid has the same effect on the final class label. Consequently, neither of these can accurately reflect the dependencies among attribute values and class labels.

We conducted experiments in the WEKA platform through a graphical user interface called *Explorer*. All experimental result estimates were obtained by averaging the results in a stratified 10-fold cross-validation. As with previous experiments, we present a comparison of the classification accuracies in Figs. 2 and 3. It can be observed that CAVWNB achieved considerably higher classification accuracies than its competitors, thus verifying our premises. In addition, Tables 7 and 8 present detailed comparisons of the weight matrices learned by CAVWNB^{CLL}, CAWNB^{CLL}, WANBIA^{CLL}, CAVWNB^{MSE}, CAWNB^{MSE}, and WANBIA^{MSE}. Owing to space limitations in this paper, we only show the weights of the attributes $b1$, $b2$, $b4$, $b5$, $f1$, $f2$, $f4$, and $f5$. These comparison results reveal that in CAVWNB, different attribute values for the same attribute discriminatively exhibit unique weights for different class labels. This case study also comprehensively verifies the superiority of our proposed CAVWNB method.

[2] https://en.wikipedia.org/wiki/Connect_Four .

Table 5
Classification accuracy comparison for CAVWNB^{MSE} versus CAWNB^{MSE}, WANBIA^{MSE}, CAVWMI, GRAWNB, KLAWNB, and NB.

Dataset	CAVWNB^{MSE}	CAWNB^{MSE}	WANBIA^{MSE}	CAVWMI	GRAWNB	KLAWNB	NB
anneal	99.03 ± 0.97	99.18 ± 0.97	98.22 ± 1.25•	97.62 ± 1.48•	97.29 ± 1.66•	90.53 ± 3.74•	96.34 ± 1.80•
anneal.ORIG	91.73 ± 2.63	91.92 ± 2.68	90.91 ± 2.83	89.84 ± 3.19•	81.38 ± 4.16•	79.66 ± 3.93•	88.17 ± 3.12•
artificial-characters	47.30 ± 1.27	36.65 ± 1.43•	36.32 ± 1.41•	35.84 ± 1.34•	35.87 ± 1.43•	36.16 ± 1.44•	36.52 ± 1.34•
audiology	76.79 ± 6.44	80.96 ± 7.22	78.08 ± 6.95	75.78 ± 8.13	75.95 ± 6.87	72.90 ± 8.54	75.74 ± 6.58
autos	70.65 ± 10.64	75.22 ± 10.02	76.99 ± 9.32∘	68.38 ± 10.24	68.33 ± 10.53	67.57 ± 11.43	66.12 ± 11.12•
balance-scale	91.98 ± 0.73	91.44 ± 1.29	91.44 ± 1.29	87.97 ± 2.21•	90.26 ± 1.89•	90.26 ± 1.89•	91.44 ± 1.29
breast-cancer	70.88 ± 6.64	69.78 ± 7.45	71.35 ± 7.24	72.14 ± 7.49	71.54 ± 7.84	69.75 ± 8.72	72.32 ± 7.91
breast-w	97.28 ± 1.72	96.11 ± 2.01•	96.74 ± 2.02	97.28 ± 1.74	97.31 ± 1.76	97.48 ± 1.70	97.31 ± 1.73
car	90.58 ± 2.06	87.09 ± 2.74•	85.69 ± 2.51•	70.79 ± 0.74•	80.21 ± 3.06•	80.58 ± 3.16•	85.69 ± 2.51•
cardiotocography	91.88 ± 1.76	91.28 ± 1.90	90.69 ± 1.83•	86.40 ± 2.12•	85.59 ± 2.36•	84.39 ± 2.34•	84.17 ± 2.22•
climate-simulation-craches	88.22 ± 2.95	88.93 ± 2.35	88.96 ± 2.22	90.48 ± 1.51∘	76.74 ± 6.55•	76.63 ± 6.58•	87.20 ± 3.17
colic	82.18 ± 5.82	81.42 ± 6.00	82.80 ± 5.59	82.18 ± 5.71	82.28 ± 5.97	84.00 ± 5.51	78.65 ± 6.08•
colic.ORIG	76.90 ± 6.09	76.84 ± 5.75	74.70 ± 6.19	74.40 ± 6.40	73.83 ± 6.72	71.11 ± 6.29•	73.67 ± 6.63
connectionist-vowel	81.31 ± 5.65	79.70 ± 5.74	77.21 ± 5.72•	77.18 ± 5.84•	73.29 ± 6.06•	74.58 ± 5.86•	76.89 ± 6.26•
credit-a	85.32 ± 3.99	85.17 ± 3.95	85.58 ± 4.04	86.01 ± 3.68	85.51 ± 3.96	85.51 ± 3.96	84.33 ± 3.85
credit-g	75.61 ± 3.75	75.33 ± 3.72	76.02 ± 3.84	75.53 ± 3.43	70.10 ± 4.41•	70.15 ± 4.61•	75.75 ± 3.96
cylinder-bands	80.91 ± 5.23	77.17 ± 5.53•	77.43 ± 5.33•	81.09 ± 5.17	80.72 ± 5.08	77.56 ± 5.24•	81.46 ± 5.25
diabetes	75.55 ± 4.28	76.08 ± 4.73	76.16 ± 4.72	75.32 ± 4.36	75.51 ± 5.65	75.39 ± 5.57	75.12 ± 4.93
ecoli	84.87 ± 5.60	83.78 ± 5.70	84.10 ± 6.23	82.26 ± 5.50•	82.83 ± 5.26	82.12 ± 6.01•	84.78 ± 5.79
energy-efficiency-y1	57.91 ± 4.70	55.53 ± 4.67•	53.19 ± 4.74•	49.35 ± 4.70•	47.88 ± 5.25•	44.11 ± 4.95•	50.37 ± 4.88•
energy-efficiency-y2	52.60 ± 4.76	52.38 ± 4.71	51.56 ± 4.50	49.32 ± 4.80•	50.49 ± 4.59	45.86 ± 4.40•	49.14 ± 4.84•
glass	59.94 ± 8.86	59.44 ± 8.60	59.45 ± 10.38	58.70 ± 9.13	61.31 ± 9.54	57.41 ± 10.57	60.00 ± 9.81
hayes-roth	84.81 ± 7.71	84.94 ± 7.85	84.75 ± 7.55	82.56 ± 8.53	85.25 ± 8.02	85.06 ± 7.99	83.88 ± 8.25
heart-c	82.05 ± 7.06	81.25 ± 7.09	82.37 ± 6.86	81.23 ± 6.95	83.74 ± 6.22	82.71 ± 6.99	82.65 ± 6.33
heart-h	82.86 ± 5.77	82.32 ± 6.17	83.74 ± 5.78	82.79 ± 6.25	83.38 ± 6.80	81.20 ± 6.66	83.71 ± 6.18
heart-statlog	83.04 ± 5.30	81.33 ± 5.37	82.67 ± 6.20	82.30 ± 5.95	83.52 ± 6.26	82.96 ± 6.05	83.30 ± 5.58
hepatitis	84.57 ± 8.47	83.68 ± 8.71	84.53 ± 8.09	85.86 ± 8.82	83.16 ± 10.14	84.89 ± 9.73	85.86 ± 9.12
hypothyroid	93.53 ± 0.58	93.49 ± 0.59	93.57 ± 0.54	93.53 ± 0.51	89.44 ± 1.31•	77.07 ± 2.07•	92.68 ± 0.75•
ionosphere	90.77 ± 4.37	91.48 ± 4.90	91.52 ± 4.37	91.09 ± 4.33	92.17 ± 4.17	91.86 ± 4.20	90.94 ± 4.21
iris	95.93 ± 5.35	96.20 ± 4.67	96.27 ± 4.86	93.67 ± 6.52	95.93 ± 5.27	95.93 ± 5.27	94.20 ± 6.47
kr-vs-kp	95.54 ± 1.23	95.20 ± 1.10	93.92 ± 1.32•	90.21 ± 1.79•	89.67 ± 1.67•	90.83 ± 1.71•	87.81 ± 1.90•
labor	94.60 ± 9.41	91.93 ± 11.90	94.30 ± 10.44	93.33 ± 11.38	93.57 ± 9.42	91.47 ± 11.06	94.93 ± 9.00
letter	75.58 ± 1.83	67.46 ± 1.94•	69.22 ± 2.12•	67.89 ± 2.02•	68.91 ± 2.19•	69.23 ± 2.12•	67.14 ± 1.97•
libras	69.75 ± 7.16	70.31 ± 7.51	69.11 ± 7.96	70.36 ± 6.94	70.42 ± 6.52	70.36 ± 6.56	69.47 ± 7.10
lymph	85.02 ± 8.49	82.31 ± 9.15	83.53 ± 8.82	83.67 ± 8.71	81.42 ± 8.82•	81.37 ± 8.97	85.09 ± 9.04
mfeat-f	77.69 ± 2.70	77.95 ± 2.44	78.87 ± 2.50	77.47 ± 2.59	79.01 ± 2.43	79.17 ± 2.46	76.97 ± 2.61
monks	74.62 ± 4.28	74.64 ± 4.26	74.64 ± 4.26	73.99 ± 4.72	74.64 ± 4.26	74.64 ± 4.26	74.64 ± 4.26
mushroom	99.82 ± 0.31	99.72 ± 0.39	99.80 ± 0.36	97.07 ± 1.50•	98.33 ± 0.98•	98.67 ± 0.89•	95.99 ± 1.58•
newthyroid	94.31 ± 4.53	95.48 ± 4.13	93.62 ± 4.48	92.92 ± 4.98	93.15 ± 4.94	94.18 ± 5.19	93.20 ± 5.01
optdigits	94.05 ± 0.98	95.02 ± 0.93∘	93.94 ± 0.98	92.48 ± 1.09•	92.26 ± 1.09•	92.46 ± 1.11•	92.39 ± 1.08•
page-blocks	93.86 ± 0.75	93.32 ± 0.74•	92.85 ± 0.79•	92.32 ± 0.77•	91.95 ± 0.69•	86.82 ± 1.30•	92.59 ± 0.92•
parkinsons	88.30 ± 6.80	89.00 ± 6.62	90.60 ± 6.29	82.38 ± 8.75•	81.17 ± 8.98•	83.83 ± 8.73•	80.84 ± 8.52•
pendigits	96.46 ± 0.51	93.72 ± 0.75•	89.00 ± 0.98•	87.54 ± 1.02•	86.52 ± 1.02•	86.41 ± 1.01•	87.49 ± 1.00•
primary-tumor	49.09 ± 5.65	47.15 ± 5.81	48.53 ± 6.24	47.29 ± 5.55	45.56 ± 6.80•	46.14 ± 7.14	47.11 ± 5.65
qar-biodegradation	85.23 ± 3.17	85.33 ± 3.27	84.87 ± 3.50	80.99 ± 4.08•	79.13 ± 4.16•	77.49 ± 4.55•	79.76 ± 4.14•
robot-24	91.69 ± 1.21	86.32 ± 1.55•	87.75 ± 1.27•	82.99 ± 1.58•	83.76 ± 1.50•	84.35 ± 1.42•	81.01 ± 1.61•
segment	92.88 ± 1.57	92.16 ± 1.73	92.42 ± 1.71	90.25 ± 1.60•	88.31 ± 1.74•	87.90 ± 1.73•	90.07 ± 1.65•
sick	98.20 ± 0.71	98.12 ± 0.75	97.40 ± 0.76•	97.47 ± 0.76•	96.47 ± 0.93•	96.47 ± 0.93•	96.94 ± 0.84•
sonar	77.81 ± 9.82	75.47 ± 9.54	75.90 ± 10.40	75.33 ± 10.33	74.85 ± 9.87	75.90 ± 9.47	75.86 ± 9.87
soybean	93.65 ± 2.58	94.31 ± 2.35	93.75 ± 2.45	93.68 ± 2.81	92.30 ± 3.05•	92.83 ± 2.89	93.53 ± 2.79
spectrometer	49.29 ± 6.36	49.17 ± 6.05	49.61 ± 6.15	49.55 ± 6.35	49.23 ± 6.09	49.63 ± 6.14	48.93 ± 6.16
splice	96.09 ± 0.98	95.81 ± 1.06	96.28 ± 0.98	96.03 ± 1.04	94.15 ± 1.26•	94.61 ± 1.23•	95.58 ± 1.12
steel-plates-faults	99.20 ± 0.62	99.88 ± 0.26∘	98.58 ± 0.82•	94.87 ± 1.53•	99.37 ± 0.57	95.83 ± 1.58•	96.68 ± 1.38•
texture	87.71 ± 1.45	80.09 ± 1.65•	85.08 ± 1.39•	80.01 ± 1.62•	80.08 ± 1.54•	79.91 ± 1.53•	79.87 ± 1.62•
thyroid-disease	93.98 ± 0.37	93.95 ± 0.34	93.91 ± 0.37	93.38 ± 0.48•	93.09 ± 0.54•	79.65 ± 1.39•	93.51 ± 0.48•
vehicle	67.68 ± 4.11	62.46 ± 3.67•	64.56 ± 3.90•	61.27 ± 3.40•	60.41 ± 3.33•	59.02 ± 3.34•	61.07 ± 3.41•
vote	92.83 ± 3.69	95.77 ± 2.62∘	95.52 ± 2.78∘	90.67 ± 3.86•	93.03 ± 3.57	93.01 ± 3.54	90.30 ± 3.89•
vowel	80.83 ± 4.33	69.01 ± 4.49•	70.20 ± 4.44•	68.35 ± 4.74•	67.32 ± 4.40•	67.07 ± 4.50•	67.79 ± 4.67•
waveform-5000	84.72 ± 3.34	83.21 ± 3.59	81.35 ± 3.39•	79.75 ± 2.94•	78.58 ± 2.99•	78.66 ± 2.94•	80.03 ± 2.87•
zoo	96.24 ± 5.43	95.75 ± 5.83	95.83 ± 5.87	96.25 ± 5.18	96.24 ± 5.04	93.97 ± 6.43	96.04 ± 5.65
Average	83.40	82.27	82.30	80.61	80.23	79.12	80.68
W/T/L	-	12/45/3	18/40/2	28/31/1	31/29/0	33/27/0	29/31/0

5. Conclusion and future work

Attribute weighting represents a flexible approach to relaxing the attribute conditional independence assumption of NB. However, most existing attribute weighting approaches assign the same weight to each attribute for all classes. Only a few fine-grained attribute weighting approaches consider attribute values or class labels. In this study, we proposed our CAVWNB

Table 6
Summary of the Wilcoxon test with respect to CAVWNB^{MSE}.

Algorithm	CAVWNB^{MSE}	CAWNB^{MSE}	WANBIA^{MSE}	CAVWMI	GRAWNB	KLAWNB	NB
CAVWNB^{MSE}	–	•	•	•	•	•	•
CAWNB^{MSE}	∘	–		•	•	•	•
WANBIA^{MSE}	∘		–	•	•	•	•
CAVWMI	∘	∘	∘	–		•	
GRAWNB	∘	∘	∘		–	•	
KLAWNB	∘	∘	∘	∘	∘	–	∘
NB	∘	∘	∘			•	–

Table 7
Weights learned by CAVWNB^{CLL}, CAWNB^{CLL}, and WANBIA^{CLL}.

Attribute	...	b1			b2			...	b4			b5			...
Attribute value	...	x	o	b	x	o	b	...	x	o	b	x	o	b	...
CAVWNB^{CLL} (win)	...	1.00	0.86	0.82	0.73	0.94	1.00	...	0.84	0.95	0.92	0.97	0.90	0.90	...
CAVWNB^{CLL} (loss)	...	0.79	1.00	1.00	1.00	0.92	0.91	...	1.00	0.90	0.91	0.90	0.99	0.90	...
CAVWNB^{CLL} (draw)	...	0.82	0.88	0.90	0.88	1.00	0.83	...	0.88	0.93	0.96	0.91	0.96	0.99	...
CAWNB^{CLL} (win)	...	0.79	0.79	0.79	0.96	0.96	0.96	...	0.97	0.97	0.97	1.00	1.00	1.00	...
CAWNB^{CLL} (loss)	...	0.85	0.85	0.85	1.00	1.00	1.00	...	1.00	1.00	1.00	0.99	0.99	0.99	...
CAWNB^{CLL} (draw)	...	1.00	1.00	1.00	0.92	0.92	0.92	...	0.93	0.93	0.93	0.98	0.98	0.98	...
WANBIA^{CLL}	...	1.00	1.00	1.00	0.95	0.95	0.95	...	1.00	1.00	1.00	1.00	1.00	1.00	...

Attribute	...	f1			f2			...	f4			f5			...
Attribute value	...	x	o	b	x	o	b	...	x	o	b	x	o	b	...
CAVWNB^{CLL} (win)	...	1.00	0.82	1.00	0.90	0.96	0.96	...	0.90	0.96	0.90	1.00	0.94	0.90	...
CAVWNB^{CLL} (loss)	...	0.87	1.00	0.90	0.99	0.92	0.94	...	0.99	0.91	0.90	0.83	0.90	0.90	...
CAVWNB^{CLL} (draw)	...	0.89	0.86	0.86	0.90	0.93	0.90	...	0.90	0.91	0.99	0.90	0.95	1.00	...
CAWNB^{CLL} (win)	...	1.00	1.00	1.00	1.00	1.00	1.00	...	1.00	1.00	1.00	1.00	1.00	1.00	...
CAWNB^{CLL} (loss)	...	0.95	0.95	0.95	0.99	0.99	0.99	...	0.99	0.99	0.99	0.92	0.92	0.92	...
CAWNB^{CLL} (draw)	...	0.94	0.94	0.94	0.92	0.92	0.92	...	0.93	0.93	0.93	0.95	0.95	0.95	...
WANBIA^{CLL}	...	1.00	1.00	1.00	0.81	0.81	0.81	...	0.42	0.42	0.42	1.00	1.00	1.00	...

Table 8
Weights learned by CAVWNB^{MSE}, CAWNB^{MSE}, and WANBIA^{MSE}.

Attribute	...	b1			b2			...	b4			b5			...
Attribute value	...	x	o	b	x	o	b	...	x	o	b	x	o	b	...
CAVWNB^{MSE} (win)	...	1.00	0.78	0.86	0.89	0.75	1.00	...	0.90	0.82	0.96	0.93	0.86	0.99	...
CAVWNB^{MSE} (loss)	...	0.69	1.00	1.00	1.00	0.77	0.85	...	1.00	0.79	0.93	0.84	1.00	0.94	...
CAVWNB^{MSE} (draw)	...	0.72	0.87	0.86	1.00	0.82	0.81	...	0.98	0.82	0.95	0.83	0.97	0.98	...
CAWNB^{MSE} (win)	...	0.98	0.98	0.98	0.97	0.97	0.97	...	0.98	0.98	0.98	0.96	0.96	0.96	...
CAWNB^{MSE} (loss)	...	0.93	0.93	0.93	0.99	0.99	0.99	...	0.99	0.99	0.99	0.96	0.96	0.96	...
CAWNB^{MSE} (draw)	...	0.95	0.95	0.95	0.95	0.95	0.95	...	0.93	0.93	0.93	0.93	0.93	0.93	...
WANBIA^{MSE}	...	0.98	0.98	0.98	0.95	0.95	0.95	...	1.00	1.00	1.00	1.00	1.00	1.00	...

Attribute	...	f1			f2			...	f4			f5			...
Attribute value	...	x	o	b	x	o	b	...	x	o	b	x	o	b	...
CAVWNB^{MSE} (win)	...	0.93	0.80	1.00	0.99	0.81	0.98	...	0.94	0.91	0.96	0.99	0.94	1.00	...
CAVWNB^{MSE} (loss)	...	0.69	1.00	0.77	1.00	0.83	0.92	...	1.00	0.89	0.94	0.83	0.96	0.93	...
CAVWNB^{MSE} (draw)	...	0.78	0.90	0.79	1.00	0.83	0.88	...	0.96	0.89	0.98	0.89	0.96	0.99	...
CAWNB^{MSE} (win)	...	0.99	0.99	0.99	0.99	0.99	0.99	...	0.98	0.98	0.98	0.99	0.99	0.99	...
CAWNB^{MSE} (loss)	...	0.99	0.99	0.99	0.98	0.98	0.98	...	0.96	0.96	0.96	0.93	0.93	0.93	...
CAWNB^{MSE} (draw)	...	0.93	0.93	0.93	0.94	0.94	0.94	...	0.94	0.94	0.94	0.93	0.93	0.93	...
WANBIA^{MSE}	...	1.00	1.00	1.00	0.86	0.86	0.86	...	0.57	0.57	0.57	1.00	1.00	1.00	...

method by extending the weight matrix to consider the horizontal granularity of attribute values and vertical granularity of class labels simultaneously. Two objective functions were redefined, to learn the weight matrix by maximizing the CLL or minimizing the MSE, respectively. Extensive experiments demonstrated that both CAVWNB^{CLL} and CAVWNB^{MSE} achieved better performances than the standard NB, as well as other existing state-of-the-art attribute weighting and attribute value weighting approaches.

The method employed to learn the weight matrix is crucial. In this study, we utilized a gradient-based search approach to determine class-specific attribute value weights. We believe that employing more sophisticated optimizing search approaches, such as the differential-evolution-based search approach [15–17], could further improve the performance of CAVWNB and enhance its advantages. Moreover, although attribute weighted wrappers generally exhibit better performances, the time complexity may be considerably higher than for filters. Exploiting some filter approaches of class-specific attribute value weighting represents another direction for our future work. Finally, a method of applying our proposed CAVWNB when attribute values are continuous and the weight of a continuous attribute value can be represented by a distribution represents another interesting direction for future work.

Conflict of Interest

I confirm that there is no conflict-of-interest in the submission, and the manuscript has been approved by all authors for publication.

Acknowledgments

The work was partially supported by the National Natural Science Foundation of China (U1711267), Fundamental Research Funds for the Central Universities (CUG2018JM18 and CUGGC03), and Hubei Provincial Natural Science Foundation of China (2018CFC893).

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17(2–3) (2011) 255–287.
[2] A. Alhussan, K.E. Hindi, Selectively fine-tuning Bayesian network learning algorithm, Int. J. Pattern Recognit Artif Intell. 30(8) (2016) 1–36.
[3] P. Bermejo, J.A. Gámez, J.M. Puerta, Speeding up incremental wrapper feature subset selection with naive Bayes classifier, Knowl. Based Syst. 55 (2014) 140–147.
[4] A.N. Burton, P.H.J. Kelly, Performance prediction of paging workloads using lightweight tracing, in: 17th International Parallel and Distributed Processing Symposium, 2003, p. 278.
[5] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7(1) (2006) 1–30.
[6] D.M. Diab, K.E. Hindi, Using differential evolution for fine tuning Naïve Bayesian classifiers and its application for text classification, Appl Soft Comput 54 (2017) 183–199.
[7] P.M. Domingos, M.J. Pazzani, Beyond independence: Conditions for the optimality of the simple Bayesian classifier, in: Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 105–112.
[8] D. Dua, E. Karra Taniskidou, UCI machine learning repository, 2017,
[9] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3(Mar) (2003) 1289–1305.
[10] E. Frank, M.A. Hall, B. Pfahringer, Locally weighted naive Bayes, in: Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, 2003, pp. 249–256.
[11] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29(2–3) (1997) 131–163.
[12] S. Garcia, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9(Dec) (2008) 2677–2694.
[13] Z. Geng, Z. Li, Y. Han, A new deep belief network based on RBM with glial chains, Inf. Sci. 463–464 (2018) 294–306.
[14] Z. Geng, Q. Meng, J. Bai, J. Chen, Y. Han, Q. Wei, Z. Ouyang, A model-free Bayesian classifier, Inf. Sci. 482 (2019) 171–188.
[15] W. Gong, Z. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution, Sol. Energy 94 (2013) 209–220.
[16] W. Gong, Z. Cai, L. Jiang, Enhancing the performance of differential evolution using orthogonal design method, Appl. Math. Comput. 206(1) (2008) 56–69.
[17] W. Gong, Y. Wang, Z. Cai, L. Wang, Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution, IEEE Trans. Syst. Man Cybern. (2018) 1–15.
[18] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann, 2011.
[19] K.E. Hindi, Fine tuning the naïve Bayesian learning algorithm, AI Commun. 27(2) (2014) 133–141.
[20] L. Jiang, Z. Cai, D. Wang, H. Zhang, Bayesian citation-knn with distance weighting, Int. J. Mach. Learn. Cybern. 5 (2) (2014) 193–199.
[21] L. Jiang, Z. Cai, H. Zhang, D. Wang, Not so greedy: randomly selected naive bayes, Expert Syst. Appl. 39(12) (2012) 11022–11028.
[22] L. Jiang, C. Li, Z. Cai, Learning decision tree for ranking, Knowl. Inf. Syst. 20 (1) (2009) 123–135.
[23] L. Jiang, C. Li, S. Wang, Cost-sensitive Bayesian network classifiers, Pattern Recognit. Lett. 45 (2014) 211–216.
[24] L. Jiang, C. Li, S. Wang, L. Zhang, Deep feature weighting for naive Bayes and its application to text classification, Eng Appl Artif Intell 52 (2016) 26–39.
[25] L. Jiang, S. Wang, C. Li, L. Zhang, Structure extended multinomial Naive Bayes, Inf Sci (Ny) 329 (2016) 346–356.
[26] L. Jiang, H. Zhang, Z. Cai, A novel bayes model: hidden Naive Bayes, IEEE Trans. Knowl. Data Eng. 21(10) (2009) 1361–1371.
[27] L. Jiang, L. Zhang, C. Li, J. Wu, A correlation-based feature weighting filter for Naive Bayes, IEEE Trans. Knowl. Data Eng. 31 (2) (2019) 201–213.
[28] L. Jiang, L. Zhang, L. Yu, D. Wang, Class-specific attribute weighted Naive Bayes, Pattern Recognit. 88 (2019) 321–330.
[29] M. Korytkowski, L. Rutkowski, R. Scherer, Fast image classification by boosting fuzzy classifiers, Inf. Sci. 327 (2016) 175–182.
[30] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: Proceedings of the 10th National Conference on Artificial Intelligence, 1992, pp. 223–228.
[31] C. Lee, A gradient approach for value weighted classification learning in Naive Bayes, Knowl. Based Syst. 85 (2015) 71–79.
[32] C. Lee, An information-theoretic filter approach for value weighted classification learning in Naive Bayes, Data Knowl. Eng. 113 (2018) 116–128.
[33] C. Lee, F. Gutierrez, D. Dou, Calculating feature weights in Naive Bayes with Kullback–Leibler measure, in: 11th IEEE International Conference on Data Mining, 2011, pp. 1146–1151.
[34] C. Li, L. Jiang, H. Li, Local value difference metric, Pattern Recognit. Lett. 49 (2014) 62–68.
[35] E. Marchiori, Class dependent feature weighting and k-nearest neighbor classification, in: Pattern Recognition in Bioinformatics - 8th IAPR International Conference, 2013, pp. 69–78.
[36] C. Nadeau, Y. Bengio, Inference for the generalization error, Mach. Learn. 52(3) (2003) 239–281.
[37] B. Tang, S. Kay, H. He, Toward optimal feature selection in Naive Bayes for text categorization, IEEE Trans. Knowl. Data Eng. 28(9) (2016) 2508–2521.
[38] S. Wang, L. Jiang, C. Li, Adapting naive Bayes tree for text classification, Knowl. Inf. Syst. 44(1) (2015) 77–89.
[39] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann, Elsevier, 2011.
[40] J. Wu, Z. Cai, S. Zeng, X. Zhu, Artificial immune system for attribute weighted Naive Bayes classification, in: The 2013 International Joint Conference on Neural Networks, 2013, pp. 1–8.

[41] J. Wu, Z. Cai, X. Zhu, Self-adaptive probability estimation for Naive Bayes classification, in: The 2013 International Joint Conference on Neural Networks, 2013, pp. 1–8.
[42] J. Wu, S. Pan, X. Zhu, P. Zhang, C. Zhang, SODE: self-adaptive one-dependence estimators for classification, Pattern Recognit. 51 (2016) 358–377.
[43] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14(1) (2008) 1–37.
[44] W. Xu, L. Jiang, L. Yu, An attribute value frequency-based instance weighting filter for Naive Bayes, J. Exp. Theoret. Artif.Intell. 31(2) (2019) 225–236.
[45] L. Yu, L. Jiang, D. Wang, L. Zhang, Attribute value weighted average of one-dependence estimators, Entropy 19(9) (2017) 501.
[46] L. Yu, L. Jiang, D. Wang, L. Zhang, Toward Naive Bayes with attribute value weighting, Neural Comput. Appl. (2018) 1–15.
[47] N.A. Zaidi, J. Cerquides, M.J. Carman, G.I. Webb, Alleviating Naive Bayes attribute independence assumption by attribute weighting, J. Mach. Learn. Res. 14(1) (2013) 1947–1988.
[48] H. Zhang, S. Sheng, Learning weighted Naive Bayes with accurate ranking, in: Fourth IEEE International Conference on Data Mining, 2004, pp. 567–570.
[49] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-bfgs-b: fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23(4) (1997) 550–560.