IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 19, 2020, accepted December 10, 2020, date of publication December 15, 2020,
date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044946

A Regularized Attribute Weighting Framework for Naive Bayes

SHIHE WANG, JIANFENG REN, (Member, IEEE), AND RUIBIN BAI
School of Computer Science, University of Nottingham Ningbo China, Ningbo 315100, China

Corresponding author: Jianfeng Ren (jianfeng.ren@nottingham.edu.cn)

This work was supported in part by the Ningbo Municipal Bureau Science and Technology under Grant 2017D10034 and Grant 2019B10026, and in part by the National Natural Science Foundation of China under Grant 72071116.

ABSTRACT The Bayesian classification framework has been widely used in many fields, but the covariance matrix is usually difficult to estimate reliably. To alleviate the problem, many naive Bayes (NB) approaches with good performance have been developed. However, the assumption of conditional independence between attributes in NB rarely holds in reality. Various attribute-weighting schemes have been developed to address this problem. Among them, class-specific attribute weighted naive Bayes (CAWNB) has recently achieved good performance by using classification feedback to optimize the attribute weights of each class. However, the derived model may be over-fitted to the training dataset, especially when the dataset is insufficient to train a model with good generalization performance. This paper proposes a regularization technique to improve the generalization capability of CAWNB, which could well balance the trade-off between discrimination power and generalization capability. More specifically, by introducing the regularization term, the proposed method, namely regularized naive Bayes (RNB), could well capture the data characteristics when the dataset is large, and exhibit good generalization performance when the dataset is small. RNB is compared with the state-of-the-art naive Bayes methods. Experiments on 33 machine-learning benchmark datasets demonstrate that RNB outperforms the compared methods significantly.

INDEX TERMS Attribute weighting, classification, naive Bayes, regularization.

I. INTRODUCTION

The Bayesian classification framework is fundamental to statistical pattern recognition and widely deployed in many machine-learning tasks [1]–[6]. Bayesian decision rule with 0/1 loss function leads to the optimal classification in statistical pattern recognition [7]. However, the estimated covariance matrix in Bayesian classification often deviates from the data population due to the curse of dimensionality, which may reduce classification performance [7]. To tackle the problem, many naive Bayes (NB) approaches [8]–[11] have been developed, which regularize the covariance matrix to a diagonal matrix. In these methods, it is assumed that each feature dimension is conditionally independent, and then the posterior probability can be estimated separately for each feature dimension. NB classifiers are competitive with many latest classifiers as shown in [12], [13].

However, NB may be oversimplified as the assumption of strong independence is often invalid, resulting in a decrease in classification performance [14]. Many improved naive Bayes classifiers have been developed to alleviate the conditional independence assumption, which can be broadly divided into five categories: 1) Structure extension [15], [16]; 2) Instance selection [17], [18]; 3) Instance weighting [19]; 4) Feature selection [20], [21]; 5) Feature weighting [22]–[36]. Among these methods, attribute-weighting methods [22]–[36] relieve the independence assumption by assigning different weights to different attributes so that the discriminative features will have a larger weight.

Attribute-weighting methods can be further divided into filter-based methods [22]–[27] and wrapper-based methods [28]–[36]. The former determines the attribute weights in advance by using the general characteristics of the data, while the latter determines the attribute weights by using classification feedback to minimize the classification error. In most cases, the filter-based methods calculate weights faster than the wrapper-based ones, but the classification accuracy of the latter is higher than that of the former.

Attribute-weighting methods often assign the same weight to each attribute in different classes, e.g. Zaidi *et al.* weighed

The associate editor coordinating the review of this manuscript and approving it for publication was Xian Sun.

the attributes to alleviate naive Bayes' independence assumption (WANBIA) [34]. In class-specific attribute weighted naive Bayes (CAWNB) [35], attributes of different classes are weighted differently to enhance the discrimination power of the model. CAWNB better captures the characteristics of dataset and achieves significant performance improvements compared with other attribute-weighting methods. However, with more weights to be optimized, the model complexity increases and hence over-fitting may occur, especially if the dataset is small. To alleviate the problem, we propose to add a regularization term to the formulation of CAWNB to penalize the model complexity, which will tend to use simpler models to avoid over-fitting, similarly as in [7], [37], [38].

Naive Bayes can be regarded as a regularized form of the Bayesian classification framework by restricting the covariance matrix to be diagonal [7]. L1- or L2-regularization has been widely used in machine-learning tasks [39], [40]. L2-regularization [40] could be applied on the model parameters to encourage the attribute weights with poor effect to decay towards zero and assign higher weights to attributes with higher effect. Alternatively, L1-regularization could be applied to the model parameters of CAWNB, which is more robust to noise and outliers than L2-regularization. L1-regularization in general produces better results, but at a higher computational cost [39]. Sparse representation is an example of L1-regularization [39].

Both L1-regularization and L2-regularization will introduce a significant computational overhead. In this paper, a simple yet effective way is proposed to regularize CAWNB, i.e. add a simpler model to constrain CAWNB. Simpler models usually achieve better generalization performance [41]. WANBIA is simpler than CAWNB, as the number of weights estimated in WANBIA are fewer than that in CAWNB. Hence, it will improve the generalization capability of CAWNB by integrating with the simpler model WANBIA. Furthermore, it will not significantly increase the computational complexity by integrating these two models, as both share similar procedures to solve the optimization problem [34], [35]. The proposed approach is named as regularized naive Bayes (RNB).

In the proposed RNB, the target is to find the optimal model parameters $\boldsymbol{M} = \{\boldsymbol{W}, \boldsymbol{w}, \alpha\}$ to minimize the difference between the posterior derived from the ground-truth label and the posterior $P(\boldsymbol{M})$ estimated from the data, where

$$P(\boldsymbol{M}) = \alpha P_D(\boldsymbol{W}) + (1-\alpha)P_I(\boldsymbol{w}). \tag{1}$$

$P_D(\boldsymbol{W})$ is the posterior probability with attributes weighted on a per-class basis, and $\boldsymbol{W}$ is the matrix to weight the attributes differently for different classes. $P_I(\boldsymbol{w})$ is the posterior probability with attributes weighted the same for all classes, and $\boldsymbol{w}$ is the weight vector for the attributes. $P_D(\boldsymbol{W})$ is a more complex model than $P_I(\boldsymbol{w})$, as more weights need to be optimized in $\boldsymbol{W}$ than that in $\boldsymbol{w}$. Thus, $P_I(\boldsymbol{w})$ is a simpler model that can provide better generalization capabilities.

Now the challenge is how to jointly find the optimal model parameters including $\boldsymbol{W}$, $\boldsymbol{w}$, and α. To achieve this, a gradient-based optimization procedure is proposed, similar to L-BFGS-M [42] used in CAWNB and WANBIA. More specifically, the partial derivatives of $P(\boldsymbol{M})$ w.r.t. $\boldsymbol{W}$, $\boldsymbol{w}$ and α are derived, and a gradient-descent-based method is utilized to iteratively update $\boldsymbol{W}$, $\boldsymbol{w}$ and α respectively, towards the objective of minimizing the classification error. Compared with other regularization methods, the proposed method requires minimal modifications to the optimization problem of CAWNB, and it does not significantly increase the computational complexity.

In the proposed formulation, α is used to automatically adjust the trade-off between discrimination power and generalization capability. More specifically, when the dataset is small and hence a simpler model is preferred, α will be smaller and hence a larger weight will be assigned to $P_I(\boldsymbol{w})$, which will ensure better generalization capabilities. This is verified by the experiments shown in Section IV.

To validate the effectiveness of the proposed RNB, a series of empirical comparisons have been conducted with state-of-the-art naive Bayes on the collection of 33 benchmark classification datasets from the University of California at Irvine (UCI) repository [43]. Experimental results show that the performance of RNB is significantly better than all compared methods [8], [21]–[23], [33]–[36].

The contributions of this paper are summarized as follow: 1) The poor generalization capability of CAWNB is identified and RNB is proposed to address the problem. 2) An optimization procedure is designed to derive the optimal model of the proposed RNB. 3) The proposed RNB improves the generalization performance of previous methods and automatically balances the discrimination power and the generalization capability, so that better performance can be obtained regardless of the size of datasets.

The rest of the paper is organized as follows. Section II reviews related work. Then, the proposed regularized naive Bayes is introduced in section III. In section IV, experimental comparisons with state-of-the-art naive Bayes are conducted to demonstrate the effectiveness of the proposed method. Finally, this work is concluded in section V.

II. RELATED WORKS

Naive Bayes classifiers have been widely used in many applications [9]–[11]. As the strong assumption of feature independence in NB is often invalid, many improvements have been developed, which can be broadly divided into 5 categories. The first category is structure extension [15], [16], which extends the structure of naive Bayes to represent the feature dependencies. The second category is instance selection [17], [18], which employs the principle of local learning to build a set of local naive Bayes classifiers using a subset of the dataset. The third category is instance weighting [19], which weights the instances differently in order to maximize the discriminant power. The fourth category is feature selection [20], [21], which removes the strongly correlated or irrelevant features, as those features are harmful to reliable classification, and/or selects the most

discriminative feature subset. The fifth category is weighted naive Bayes, which tackles the problem by assigning different weights to attributes so that the discriminative features have a larger weight and hence the discriminative power will increase [22]–[36]. The attribute-weighting methods can be further categorized into filter-based methods [22]–[27] and wrapper-based methods [28]–[36].

Filter-based methods [22]–[27] utilize the characteristics of the data to determine attribute weights. Lee *et al.* determined the weights by using the Kullback-Leibler (KL) divergence between attributes and class labels [25]. In [24], Hall defined the weights by utilizing the minimum depth in a decision tree. In [22], the conditional probabilities of naive Bayes are estimated by deeply computing feature weighted frequencies. Recently, Jiang *et al.* developed a correlation-based attribute-weighting NB, which defines the weight of each attribute as a sigmoid transformation of the difference between mutual relevance and average mutual redundancy [23]. Filter-based approaches determine the weights in advance by measuring the relationship between features and classification variables, such as mutual information, KL divergence and correlation.

Wrapper-based methods [28]–[36] utilize the classification feedback to optimize attribute weights. Due to the iterative process, wrapper-based methods usually have higher time complexity and better classification performance than filter-based ones. In [28], Zhang and Sheng updated attribute weights based on a hill-climbing strategy to maximize the classification accuracy. Wu and Cai utilized a differential evolution algorithm to determine the weights [33]. In [36], Yu *et al.* developed a hybrid attribute-weighting method by initializing the weights through a correlation-based filter and then adjusting them through a wrapper. Zaidi *et al.* optimized attribute weights by minimizing the mean squared error between predicted and ground-truth labels [34]. Very recently, Jiang *et al.* developed CAWNB [35], which determines the optimal weight for each attribute of different classes to capture more characteristics of the dataset, instead of ignoring the class dependency as in [34]. Hence it achieves excellent classification performance on many benchmark datasets.

Unlike WANBIA [34] that assigns the same attribute weight for all classes, CAWNB [35] assigns different weights to different classes. Thus, the CAWNB model is more complicated and more prone to over-fitting, especially when the dataset is small. Some form of regularization to CAWNB is required to improve its generalization performance.

III. REGULARIZED ATTRIBUTE-WEIGHTED NAIVE BAYES

A. PROBLEM ANALYSIS OF PREVIOUS NAIVE BAYES METHODS

In the Bayesian classification framework, the posterior probability is defined as:

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)P(c)}{P(\boldsymbol{x})}, \tag{2}$$

where $\boldsymbol{x}$ is the feature vector and c is the classification variable. Because it is difficult to reliably estimate the likelihood $P(\boldsymbol{x}|c)$ due to the curse of dimensionality, in naive Bayes methods, the likelihood is estimated by assuming that the attributes are independent given the classification variable c, which results in the following formulation:

$$P(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c), \tag{3}$$

where x_j is the j-th dimension of the feature vector $\boldsymbol{x}$, and m is the feature dimensionality. Then, the posterior probability can be estimated by:

$$P(c|\boldsymbol{x}) = \frac{P(c)\prod_{j=1}^{m} P(x_j|c)}{\sum_{c'} P(c')\prod_{j=1}^{m} P(x_j|c')}. \tag{4}$$

Naive Bayes regularizes the Bayesian framework by assuming that each attribute is independent conditioned on the classification variable, but this assumption is often invalid. To alleviate the problem, weights are assigned to attributes in WANBIA [34], and the weights are optimized via minimizing the mean squared error between the estimated posteriors and the posteriors derived using ground-truth labels.

Jiang *et al.* showed that attribute weighting should be class-specific to enhance the discrimination power of naive Bayes [35]. Thus, different weights are assigned to the attributes for different classes in CAWNB [35]. CAWNB is more complicated than WANBIA considering the number of model parameters. Class-specific attribute weights provide CAWNB with greater discrimination. However, the model complexity is considerably increased, so the generalization capability may decrease. The problem will be severe when the dataset is small, so the training samples are not enough to derive a reliable naive Bayes model.

To improve the generalization capability of CAWNB, we propose to add a simpler model, WANBIA, to constrain CAWNB. Besides, CAWNB is an improved version of WANBIA, and both share the similar optimization procedure. It will not significantly increase the computational complexity by integrating WANBIA into CAWNB.

B. OVERVIEW OF PROPOSED REGULARIZED NAIVE BAYES

In the proposed method, the target is to use the classification feedback to optimize the attribute weights. More precisely, the target is to find the optimal attribute weights to minimize the difference between the estimated posteriors and the posteriors derived from the ground-truth labels. The mean squared error is often used to capture such differences:

$$f = \frac{1}{2}\sum_{\boldsymbol{x}_i \in D}\sum_{c}(P(c|\boldsymbol{x}_i) - \hat{P}(c|\boldsymbol{x}_i))^2, \tag{5}$$

where D represents the whole dataset, $\hat{P}(c|\boldsymbol{x}_i)$ is the estimated posterior of class c given $\boldsymbol{x_i}$, and the posteriors derived from the ground-truth labels are defined as:

$$P(c|\boldsymbol{x}_i) = \begin{cases} 1 & if\ c = c_i, \\ 0 & otherwise. \end{cases} \tag{6}$$

The posterior $\hat{P}(c|\boldsymbol{x}_i)$ consists of two parts. The first part that emphasizes the discriminative power of the model, whose attributes are weighted on a class-dependent basis, is defined as:

$$\hat{P}_D(c|\boldsymbol{x}) = \frac{\pi_c \prod_j \theta_{c,j}^{w_{c,j}}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_{c',j}}}, \tag{7}$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_l]$ are the prior probabilities, and π_c is the prior probability that sample $\boldsymbol{x}$ belongs to class c. The matrix $\boldsymbol{\Theta}$ of likelihood probabilities is defined as:

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,m} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{l,1} & \theta_{l,2} & \cdots & \theta_{l,m}, \end{bmatrix}$$

where $\theta_{c,j}$ is the likelihood of the j-th attribute of $\boldsymbol{x}$ given the class c. $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$ are estimated from training samples using (13) and (14) respectively, as shown in section III-C later on.

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{l,1} & w_{l,2} & \cdots & w_{l,m} \end{bmatrix}$$

is the attribute-weighting matrix on a per-class basis and $w_{c,j}$ is the weight of the j-th attribute for class c.

The other posterior probability $\hat{P}_I(c|\boldsymbol{x})$ that emphasizes the generalization capability of the model, whose attributes are weighted on a class-independent basis, is defined as:

$$\hat{P}_I(c|\boldsymbol{x}) = \frac{\pi_c \prod_j \theta_{c,j}^{w_j}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_j}}, \tag{8}$$

where $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]$ is the weight vector and w_j is the weight of the j-th attribute.

In the proposed RNB, the regularized posterior probability is defined as:

$$\hat{P}(c|\boldsymbol{x}) = \alpha \hat{P}_D(c|\boldsymbol{x}) + (1-\alpha)\hat{P}_I(c|\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{M} = \{\boldsymbol{W},\ \boldsymbol{w},\ \alpha\}$ consists of class-dependent attribute weights $\boldsymbol{W}$, class-independent attribute weights $\boldsymbol{w}$ and a hyper-parameter α. α is used to balance the trade-off between the discrimination power and the generalization capability.

The block diagram of the proposed regularized naive Bayes is shown in Fig. 1. In the training process, the elements in $\boldsymbol{W}$ and $\boldsymbol{w}$ are all initialized to 1 and α is initialized to 0.5, so that the initial model is the original naive Bayes. Then, $\hat{P}_D(c|\boldsymbol{x})$ and $\hat{P}_I(c|\boldsymbol{x})$ are estimated using training samples and these two posteriors are integrated as the regularized posterior $\hat{P}(c|\boldsymbol{x})$ with the weighting factor α, as shown in (9). Then, f is calculated as the sum of the squared differences between $P(c|\boldsymbol{x})$ and $\hat{P}(c|\boldsymbol{x})$, as shown in (5). The model parameters are optimized iteratively by using a gradient-descent-based method to minimize f until convergence. The detailed procedures to derive the optimal model parameters are given in Section III-D. The class-independent weights significantly improve the generalization capability of the model, as evidenced in Section IV.

In the testing process, the estimated prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$ and the optimal model parameters $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$ are used to compute the posterior probability $\hat{P}(c|\boldsymbol{t})$ for a given test instance $\boldsymbol{t}$ by using (9). Finally, the class label of $\boldsymbol{t}$ is estimated by using MAP estimation as follows:

$$\hat{c}(\boldsymbol{t}) = \arg\max_{c \in \boldsymbol{C}} \hat{P}(c|\boldsymbol{t}), \tag{10}$$

where $\boldsymbol{C}$ is the set of labels for all classes.

C. ESTIMATION OF PRIOR PROBABILITIES AND LIKELIHOOD PROBABILITIES

Firstly, prior probabilities $\boldsymbol{\pi}$ and likelihood probabilities $\boldsymbol{\Theta}$ are estimated based on training samples. Traditionally, the prior probability π_c for class c is estimated as follows:

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c)}{n}, \tag{11}$$

where n is the number of training samples, c_i is the class label of the i-th training instance, and $\delta(\bullet)$ is a binary function, which is 1 if its two parameters are identical and 0 otherwise. The likelihood function $\theta_{c,j}$ for the j-th attribute of class c is estimated as follows:

$$\theta_{c,j} = \frac{\sum_{i=1}^{n} \delta(x_{ij}, x_j)\delta(c_i, c)}{\sum_{i=1}^{n} \delta(c_i, c)}, \tag{12}$$

where x_{ij} is the j-th attribute value of the i-th training instance and x_j is the j-th attribute.

To make the estimation numerically stable, e.g. to avoid estimating π_c to 0 due to insufficient training samples, in the proposed method, the prior probability π_c and the likelihood $\theta_{c,j}$ are estimated by adding a regularization term as follow:

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c) + \frac{1}{l}}{n+1}, \tag{13}$$

$$\theta_{c,j} = \frac{\sum_{i=1}^{n} \delta(x_{ij}, x_j)\delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} \delta(c_i, c) + 1}, \tag{14}$$

where n_j is the number of discretized values for the j-th attribute.

The aforementioned procedures work for discrete features. Continuous features are transformed into the discrete features by using the Fayyad & Irani's MDL method [44]. Then, (13) and (14) are used to compute prior probabilities and

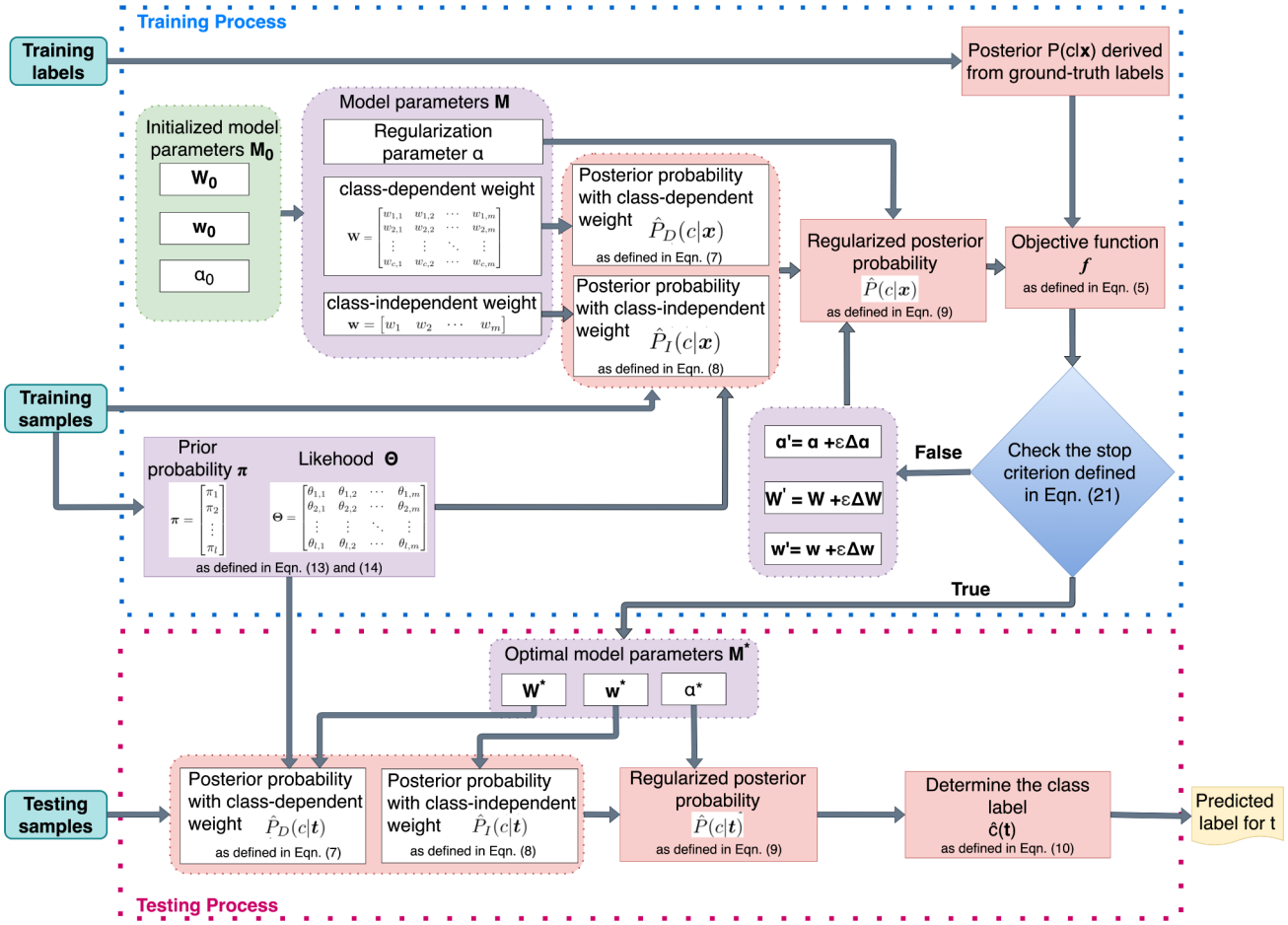

FIGURE 1. Proposed regularized attribute weighting framework for naive Bayes. In the training process, the model parameters are initialized and the posteriors $\hat{P}(c|x)$ are estimated from training samples, which consist of two parts: the posteriors with attributes weighted on a class-dependent basis, and the posteriors with attributes weighted on a class-independent basis. Then, the model parameters are optimized iteratively through a gradient-descent-based algorithm using the classification feedback. When the classifier error is small enough, the optimized model parameters will be then used in the testing process. Finally during testing, the posterior for each testing sample t will be estimated and the class label for t is derived by using MAP estimation.

likelihood probabilities of continuous features respectively in the same way as discrete ones.

D. SOLVING THE OPTIMIZATION PROBLEM

Now the challenge is how to jointly find the optimal model parameters $\boldsymbol{M}$ including $\boldsymbol{W}$, $\boldsymbol{w}$, and α. To achieve this, a gradient-descent-based optimization procedure is proposed, similar to L-BFGS-M [42] used in CAWNB and WANBIA. More specifically, the target is to find the gradient direction of the objective function w.r.t. the model parameters $\boldsymbol{W}$, $\boldsymbol{w}$, and α, respectively. Then, the model parameters are updated iteratively along the gradient direction to minimize the error function defined in (5).

The partial derivative of f w.r.t. each element of $\boldsymbol{W}$, $w_{c,j}$, is given as follows:

$$\frac{\partial f}{\partial w_{c,j}} = -\alpha \sum_{\boldsymbol{x}\in D} \Big(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\Big) \times \Big[\hat{P}_D(c|\boldsymbol{x})(1-\hat{P}_D(c|\boldsymbol{x}))\log(\theta_{c,j})\Big]. \quad (15)$$

Similarly, the partial derivative of f w.r.t. each element of $\boldsymbol{w}$, w_j is calculated as:

$$\frac{\partial f}{\partial w_j} = (\alpha-1)\sum_{\boldsymbol{x}\in D}\sum_{c}\Big(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\Big)\hat{P}_I(c|\boldsymbol{x}) \times \left(log(\theta_{a_j|c}) - \sum_{c'}\hat{P}_I(c'|\boldsymbol{x})log(\theta_{c',j})\right). \quad (16)$$

The detailed derivations are omitted here and a brief derivation is described in Appendix. Finally, the partial derivative of f w.r.t. α can be calculated as:

$$\frac{\partial f}{\partial \alpha} = \sum_{\boldsymbol{x}\in \boldsymbol{X}}\Big(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\Big)\Big(\hat{P}_D(c|\boldsymbol{x}) - \hat{P}_I(c|\boldsymbol{x})\Big). \quad (17)$$

After deriving the partial derivatives of the objective function f w.r.t. the model parameters, the model parameters $\boldsymbol{W}$, $\boldsymbol{w}$, and α are iteratively updated to minimize the classification error. After the i-th iteration of optimization, the model parameters $\boldsymbol{W}_i$, $\boldsymbol{w}_i$, α_i are updated using the following

equations:

$$\boldsymbol{W}_{i+1} = \boldsymbol{W}_i + \epsilon \nabla \boldsymbol{W}_i, \tag{18}$$

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \epsilon \nabla \boldsymbol{w}_i, \tag{19}$$

$$\alpha_{i+1} = \alpha_i + \epsilon \nabla \alpha_i, \tag{20}$$

where $\nabla \boldsymbol{W}_i$ is the gradient matrix whose elements are defined in (15), $\nabla \boldsymbol{w}_i$ is the gradient vector whose elements are defined in (16), $\nabla \alpha_i$ is the partial derivative defined in (17) and ϵ is the learning rate. The iteration will stop when:

$$\frac{f_i - f_{i+1}}{\max(|f_i|, |f_{i+1}|, 1)} < \eta, \tag{21}$$

where η is a predefined small constant. The optimal model is denoted as $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.

The learning algorithms for training and testing are summarized in **Algorithm 1** and **Algorithm 2**, respectively.

Algorithm 1 Training Algorithm

Input: $\boldsymbol{x}$: training samples, f: the objective function.

Output: the prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$, and the optimal model parameters $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.

1: Estimate the prior probability π_c using (13).
2: Estimate the likelihood probability $\theta_{c,j}$ using (14).
3: Derive the posterior probability $P(c|\boldsymbol{x})$ from the ground-truth labels using (6).
4: Initialize attribute weights of $\boldsymbol{W}$ and $\boldsymbol{w}$ to 1 and α to 0.5.
5: **while** stop condition (21) is NOT met **do**
6: Derive the class-dependent posterior $\hat{P}_D(c|\boldsymbol{x})$ by (7).
7: Derive the class-independent posterior $\hat{P}_I(c|\boldsymbol{x})$ by (8).
8: Derive the regularized posterior $\hat{P}(c|\boldsymbol{x})$ by (9).
9: Derive the objective function f using (5).
10: Derive the partial derivatives of f w.r.t. $\boldsymbol{W}$, $\boldsymbol{w}$, α using (15), (16) and (17), respectively.
11: Update $\boldsymbol{W}$, $\boldsymbol{w}$ and α using (18), (19) and (20), respectively.
12: **end while**
13: Return the prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$ and the optimal model parameters $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.

Algorithm 2 Testing Algorithm

Input: $\boldsymbol{t}$: a test instance, $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$: the set of the optimal model parameters, $\boldsymbol{\pi}$: the prior probabilities, $\boldsymbol{\Theta}$: the likelihood probabilities.

Output: the class label of the test instance $\boldsymbol{t}$.

1: Derive the class-dependent posterior $\hat{P}_D(c|\boldsymbol{t})$ using (7).
2: Derive the class-independent posterior $\hat{P}_I(c|\boldsymbol{t})$ using (8).
3: Derive the regularized posterior $\hat{P}(c|\boldsymbol{t})$ using (9).
4: Determine the class label $\hat{c}(\boldsymbol{t})$ of the test instance $\boldsymbol{t}$ using (10).
5: Return the predicted class label $\hat{c}(\boldsymbol{t})$.

TABLE 1. Description of competitors: original NB, Gaussian NB, one feature-selection-based method, two filter-based attribute-weighting methods and four wrapper-based attribute-weighting methods.

Algorithm	Description
NB [45]	Original naive Bayes method.
GNB [8]	Gassian naive Bayes method.
TCSFS-NB [21]	Test-cost-sensitive feature selection.
DAWNB [22]	Filter-based attribute weighting, with deep attribute weighting.
CFW [23]	Filter-based attribute weighting, with correlation-based attribute weighting.
DEAWNB [33]	Wrapper-based attribute weighting, with differential evolution-based attribute weighting.
WANBIA [34]	Wrapper-based attribute weighting, with attributes weighted in a class-independent manner.
CAWNB [35]	Wrapper-based attribute weighting, with attributes weighted in a class-specific manner.
CWANB [36]	Wrapper-based attribute weighting, with filter-based initialization and wrapper-based optimization for attribute weighting of each attribute.

α is initialized to 0.5 so that the initial model will not bias the discrimination power or the generalization capability. α is optimized to achieve the best trade-off between discrimination power and generalization capability. A small value of α means that a small weight is assigned to $\hat{P}_D(c|\boldsymbol{x})$, and a large weight is assigned to $\hat{P}_I(c|\boldsymbol{x})$. As a result, a better generalization capability is expected. Note that in the extreme case, the model is reduced to $\hat{P}_D(c|\boldsymbol{x})$ for $\alpha = 1$, or $\hat{P}_I(c|\boldsymbol{x})$ for $\alpha = 0$. All the weights of $\boldsymbol{W}$ and $\boldsymbol{w}$ are initialized to 1, which means that the model is initialized to naive Bayes at the beginning. In the proposed regularized naive Bayes, not only the prior probabilities and the likelihood probabilities are regularized to avoid numerical instability as shown in (13) and (14), but also the posterior is regularized to improve the generalization capability as shown in (9).

IV. EXPERIMENTAL RESULTS

The proposed approach is compared with original naive Bayes [45], Gaussian naive Bayes [8] and several state-of-the-art NB algorithms. TCSFS-NB improves the performance of naive Bayes through feature selection [21]. DAWNB [22] and CFW [23] are two recent filter-based attribute-weighting methods. The comparisons with them can illustrate the performance gain of the proposed RNB over filter-based approaches. DEAWNB [33], WANBIA [34], CAWNB [35] and CWANB [36] are four wrapper-based attribute-weighting methods in recent years. They can provide a comprehensive comparison to wrapper-based attribute-weighting methods. These competitors are summarized in Table 1.

A. EXPERIMENTAL SETTINGS

Comprehensive experiments are conducted on a collection of 33 benchmark datasets from the UCI repository,[1] which represent a wide range of domains and data characteristics [43]. Most datasets are from real-world problems e.g. diabetes, hepatitis and primary tumor, vehicle classification,

[1]These 33 datasets could be downloaded from "https://archive.ics.uci.edu/ml/index.php"

TABLE 2. Most datasets are collected from real-world problems. The number of instances is widely distributed in 57 and 20000 which can provide a comprehensive evaluation on datasets of different sizes. The number of attributes/classes of these datasets also varies significantly. There are both numeric and nominal data in the datasets. Some datasets contain missing values. These datasets are hence diverse and challenging.

Dataset	Instance	Attributes	Classes	Missing values	Numeric values
anneal	898	39	6	Y	Y
audiology	226	70	24	Y	N
balance-scale	625	5	3	N	Y
breast-cancer	286	10	2	Y	N
breast-w	699	10	2	Y	N
colic	368	23	2	Y	Y
credit-a	690	16	2	Y	Y
credit-g	1000	21	2	N	Y
diabetes	768	9	2	N	Y
glass	214	10	7	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
lymphography	148	19	4	N	Y
mushroom	8124	23	2	Y	N
primary-tumor	339	18	21	Y	N
segment	2310	20	7	N	Y
sick	3772	30	2	Y	Y
sonar	208	61	2	N	Y
soybean	683	36	19	Y	N
splice	3190	62	3	N	N
vehicle	846	19	4	N	Y
vote	435	17	2	Y	N
vowel	990	14	11	N	Y
waveform-5000	5000	41	3	N	Y
zoo	101	18	7	N	Y

letter recognition and so on. Besides, the characteristics of the datasets including the number of instances, attributes and classes are significantly different. The sizes of datasets are between 57 and 20000, enough to evaluate how the algorithms perform on datasets of different sizes. For example, smaller datasets such as breast-cancer, heart-c and iris will prefer methods with better generalization capabilities. Attribute weighting methods with good discrimination power will perform better on larger datasets such as sick, hypothyroid, waveform-5000 and mushroom. In addition, 17 out of 33 datasets have missing values, which simulates the difficulties in real life when collecting datasets, and imposes additional challenges for classifiers. Besides numeric values, the attributes of some datasets are nominal values, which imposes another challenge for classifier design. These 33 benchmark datasets provide a comprehensive evaluation of the effectiveness of the proposed RNB. The dataset descriptions are summarized in Table 2.

The missing values in the datasets are replaced with the average value of the numeric attributes or the mode of the nominal attributes in the available data. In CAWNB, they use Fayyad & Irani's MDL method [44] to discretize numeric attributes which may lead to information loss. Thus, in the experiments, the Fayyad & Irani's MDL method is fine-tuned to reduce the information loss. Besides, two irrelevant attributes are deleted, i.e. "instance name" in "splice" and "animal" in "zoo".

The results of NB, DAWNB, DEAWNB, WANBIA and CAWNB are obtained from [35]. The results of TCSFS-NB, DAWNB and CWANB are obtained from [21], [22] and [36], respectively. GNB is implemented using Weka and the proposed RNB is implemented in MATLAB. The classification accuracy of the proposed algorithm on each dataset is derived via 10-fold cross-validation. During optimization, η is set to 10^{-7} in the stop criterion defined in (21). The learning rate ϵ is determined using the linear search programs [46].

B. COMPARISON TO STATE OF THE ART

The comparisons to the state-of-the-art algorithms on the 33 datasets are shown in Table 3. The symbol $\bullet$ represents the statistically significant improvements achieved by the proposed regularized naive Bayes for paired one-side t-test with the $p = 0.05$ significance level. The average classification accuracy and the *Win*/*Tie*/*Loss* on the 33 datasets for all the algorithms are summarized at the bottom of Table 3. The average classification accuracy over all the datasets can provide a straightforward comparison for their performance. Each entry of $W/T/L$ in the table indicates that the competitor wins on W datasets, ties on T datasets and loses on L datasets compared to the proposed RNB.

From Table 3, it is obvious that the proposed RNB obtains the highest average classification accuracy. Compared with the original naive Bayes and Gaussian naive Bayes, the proposed RNB achieves 2.34% and 6.15% of improvement respectively on average. Compared with filter-based approach, DAWNB [22] and CFW [23], the proposed RNB achieves 2.26% and 1.82% of improvements on average, respectively. Compared with feature-selection-based approach, TCSFS-NB [21], RNB achieves 2.32% of improvement on average.

Compared with the previous best algorithm, CAWNB, the proposed RNB achieves more than 1% of improvement for the average classification accuracy over the 33 datasets. Among them, the improvements on some datasets are significant. For example, the classification accuracies of RNB on balance-scale, glass, sonar and vowel are more than 5% higher than the most recent attribute-weighting method, CAWNB. On relatively small datasets such as glass, iris and sonar, the proposed approach significantly outperforms CAWNB and the others because of the good generalization capability. On relatively large datasets such as segment and letter, the proposed RNB also shows statistically significant improvements. All these demonstrate that the proposed approach could well adapt to the datasets of different sizes, and automatically adjust the balance between the discrimination power and the generalization capability.

TABLE 3. Experimental results for RNB versus NB [45], DAWNB [22], DEAWNB [33], WANBIA [34], CAWNB [35], CWANB [36], GNB [8], TCSFS-NB [21] and CFW [23]. It is obvious that overall RNB achieves the best classification accuracy among all approaches. The average classification accuracy of RNB is more than 2% higher than NB's. Besides, RNB obtains more than 1% of improvement on average compared with the previous best attribute-weighting method, CAWNB. The classification accuracies of RNB on some datasets e.g. balance-scale, glass, sonar, and vowel achieve about 5% of improvement compared with CAWNB.

Dataset	RNB	GNB [8]	TCSFS-NB [21]	CFW [23]	CWANB [36]	CAWNB [35]	NB [45]	DAWNB [22]	DEAWNB [33]	WANBIA [34]
anneal	99.22	86.30 •	98.26 •	98.50 •	98.55	99.47	96.36 •	97.45 •	98.41 •	98.69
audiology	80.08	71.24 •	74.20	74.22	77.52	80.96	75.74	77.11	76.08	78.08
balance-scale	78.55	90.40	70.72 •	73.76 •	70.01 •	71.08 •	71.08 •	71.99 •	69.26 •	71.08 •
breast-cancer	70.25	72.03	71.10	72.46	71.28	69.78	72.32	71.50	70.46	71.35
breast-w	96.99	96.00	96.58	97.14	97.07	96.50	97.25	97.30	96.91	96.51
colic	83.42	77.45 •	84.13	83.34	82.83	83.07	81.20 •	82.93	82.55	83.72
credit-a	86.09	77.68 •	85.93	86.99	86.26	86.14	86.17	86.49	86.81	86.23
credit-g	**78.60**	75.40 •	74.11 •	75.70 •	75.47 •	76.04 •	75.40 •	74.27 •	75.08 •	75.59 •
diabetes	78.64	76.30 •	78.15	78.01	78.37	78.67	77.88	78.70	77.85	78.48
glass	**80.01**	48.60 •	74.40 •	73.37 •	74.72 •	73.69 •	74.20 •	72.00 •	75.32 •	73.82 •
heart-c	83.54	82.84	82.48	82.94	83.71	83.03	83.73	83.11	82.38	83.73
heart-h	82.32	82.99	80.73	83.82	82.66	83.41	84.43	84.05	81.61	84.39
heart-statlog	82.96	83.70	83.70	83.44	85.04	84.33	83.74	83.33	83.59	84.74
hepatitis	**89.83**	83.87 •	86.99	85.95	86.02	86.66	85.05	84.80	86.66	86.61
hypothyroid	99.52	95.23 •	99.07 •	98.56 •	99.47	99.60	98.74 •	98.15 •	99.31	99.37
ionosphere	91.80	82.62 •	91.57	91.82	92.77	92.74	91.37	91.79	91.71	92.73
iris	**97.33**	96.00	95.33 •	94.40 •	94.60 •	94.67 •	94.33 •	94.53 •	94.13 •	94.33 •
kr-vs-kp	93.08	87.89 •	94.09	93.58	94.38	95.20	87.81 •	91.86 •	94.11	93.92
labor	91.90	91.23	87.13 •	92.10	94.60	92.63	93.83	93.57	94.63	95.60
letter	**76.62**	64.12 •	74.61 •	75.22 •	75.25 •	75.42 •	74.67 •	75.33 •	75.21 •	75.55 •
lymphography	84.30	83.11	82.20	84.81	81.47	83.76	85.70	83.39	84.24	84.48
mushroom	99.96	95.83 •	99.70 •	99.19 •	99.84 •	99.96	98.03 •	99.02 •	99.89 •	99.90 •
primary-tumor	47.30	46.90	46.25	47.20	45.69	47.15	47.11	43.84	47.34	48.53
segment	**95.84**	80.22 •	93.97 •	93.47 •	95.27	94.68 •	92.91 •	93.84 •	95.09	95.24
sick	97.56	92.92 •	97.21	97.36	97.44	97.54	97.07	96.86 •	97.59	97.47
sonar	**91.90**	67.79 •	80.55 •	82.56 •	82.71 •	84.58 •	84.96 •	83.72 •	84.10 •	83.85 •
soybean	94.00	92.09 •	91.64 •	93.66	93.79	94.31	93.53	93.35	93.71	93.75
splice	**96.39**	95.30 •	95.09 •	96.19	96.19	95.81	95.58 •	96.05	95.84	96.28
vehicle	69.61	44.80 •	66.60 •	62.91 •	68.32	70.33	62.64 •	62.82 •	66.30 •	68.57
vote	95.87	90.11 •	96.30	92.11 •	95.15	95.77	90.30 •	92.62 •	95.35	95.52
vowel	**75.56**	63.74 •	68.11 •	68.84 •	70.45 •	69.07 •	66.00 •	67.45 •	68.19 •	68.19 •
waveform-5000	**85.84**	80.00 •	81.67 •	83.11 •	84.22 •	85.56	80.76 •	80.99 •	83.80 •	84.65 •
zoo	**98.09**	95.05 •	93.69 •	95.96	96.15	95.95	95.75 •	94.05 •	95.45 •	95.75 •
AVERAGE	**86.45**	**80.30**	**84.13**	**84.63**	**85.07**	**85.38**	**84.11**	**84.19**	**84.82**	**85.35**
$W/T/L$	-	1/9/23	0/17/16	0/14/19	0/24/9	0/25/8	0/15/18	0/16/17	0/21/12	0/23/10

[1] • indicates that statistically significant improvement is achieved by the proposed RNB.
[2] The bold value of classification accuracy means the proposed RNB performs best on the dataset.

C. ANALYSIS OF EXPERIMENTAL RESULTS

In the statistical significance tests shown in Table 3, the proposed approach significantly outperforms CAWNB [35], CWANB [36], WANBIA [34], DEAWNB [33], CFW [23], DAWNB [22], TCSFS-NB [21] and GNB [8] on 8, 9, 10, 12, 14, 17, 17 and 23 datasets, respectively. Compared with the original NB, on more than half of the datasets, the proposed RNB achieves statistically significant improvements. Compared with the previous best algorithm, CAWNB [35], the proposed RNB achieves statistically significant improvements on 8 datasets, which demonstrates the effectiveness of the proposed approach.

Table 4 summarizes the results for statistical significance tests. For each entry $u(v)$, u is the number of datasets on which the proposed RNB outperforms the corresponding competitor, and v is the number of datasets on which the performance gain is statistically significant with significance level $p = 0.05$. Table 4 shows that on average the classification accuracies on more than two-thirds of 33 datasets improves and half of them are statistically significant. It hence can be concluded that the proposed RNB outperforms all compared approaches.

From the experimental results, it can be seen that the proposed regularized naive Bayes achieves a remarkable performance improvement. The hyper-parameter α is optimized along with class-dependent attribute weights and class-independent attribute weights. The optimal value of α on each dataset is shown in Table 5, together with the number of instances and the number of instances per class. The values of α^* vary on different datasets. In general, larger the dataset, higher the α^* value.

To better see the trend, the average value of α^* across datasets and the performance gain of the proposed RNB against the second best algorithm, CAWNB [35], are summarized in Table 6. The 33 datasets are divided into small and large datasets according to the number of instances per class, e.g. if it is larger than 500, the dataset is considered large, and small otherwise. Table 6 shows that for small datasets, the average α^* value is significantly smaller than that for large datasets. This indicates that α^* could be automatically adjusted during optimization so that for small datasets, α^* will be small to favor the generalization capability, whereas for large datasets, α^* will be large to favor the discrimination power. It can also be seen that the proposed

TABLE 4. Summary of the results for statistical significance tests. For example, RNB outperforms CAWNB on 21 datasets, among which 8 are statistically significant.

Algorithm	GNB [8]	TCSFS-NB [21]	CFW [23]	CWANB [36]	CAWNB [35]	NB [45]	DAWNB [22]	DEAWNB [33]	WANBIA [34]
RNB	29(23)	28(17)	24(14)	24(9)	21(8)	25(18)	26(17)	26(12)	22(10)

TABLE 5. The number of instances, instances per class and the optimal value of α on 33 datasets.

Datasets	Instance	Instance/class	α^*
anneal	898	150	1.0000
audiology	226	9	0.9875
balance-scale	625	208	0.7293
breast-cancer	286	143	0.4591
breast-w	699	350	0.4312
colic	368	184	0.4991
credit-a	690	345	0.4486
credit-g	1000	500	0.5258
diabetes	768	384	0.2839
glass	214	31	0.9741
heart-c	303	61	0.3981
hear-h	294	59	0.7923
heat-statlog	270	135	0.5767
hepetitis	155	78	0.5815
hypothroid	3772	943	1.0000
ionophere	351	176	0.4925
iris	150	50	0.1935
kr-vs-kp	3196	1598	1.0000
labor	57	29	1.0000
letter	20000	769	0.5536
lymphoraphy	148	37	0.8762
mushroom	8124	4062	1.0000
primary-tumor	339	16	1.0000
segment	2310	330	0.0223
sick	3772	1886	0.9155
sonar	208	104	0.0000
soybean	683	36	0.0000
splice	3190	1063	0.1932
vehicle	846	212	0.4588
vote	435	218	0.0000
vowel	990	90	0.4530
wave-5000	5000	1667	0.8445
zoo	101	14	0.9916

TABLE 6. The average value of α^* and the performance gain of the proposed RNB against CAWNB [35] for small/large datasets.

	Small datasets	Large datasets
Average α^*	0.5460	0.7541
Performance gain(%)	1.3267	0.2978

RNB indeed demonstrates good generalization capabilities for small datasets by achieving a larger performance gain than that on large datasets.

V. CONCLUSION

In this paper, after a thorough literature review of the state-of-the-art attribute-weighting naive Bayes methods, we find that class-dependent attribute-weighting naive Bayes has poor generalization capabilities on relatively small datasets. Therefore, we propose to add a regularization term to alleviate the problem. The regularization term is extracted from a simpler naive Bayes which has better generalization capabilities. The proposed regularized naive Bayes is hence derived by integrating the regularization term into the CAWNB. A gradient-descent-based optimization procedure has been designed to derive the optimal model parameters including class-dependent weight matrix $\boldsymbol{W}$, class-independent weight vector $\boldsymbol{w}$ and the hyper-parameter α. Experimental results on the 33 datasets validate the effectiveness of the proposed RNB. The proposed method outperforms the previous best algorithm CAWNB on 21 datasets, of which 8 are statistically significant, and the average performance gain on the 33 datasets is more than 1%.

APPENDIX A

In this section, a brief derivation of the gradients of f w.r.t $\boldsymbol{W}$ and $\boldsymbol{w}$ is provided. Firstly, the partial derivative of f w.r.t. each element of $\boldsymbol{W}$, $w_{c,j}$, is calculated as:

$$\frac{\partial f}{\partial w_{c,j}} = -\alpha \sum_{\boldsymbol{x}\in D} \left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\right) \frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial w_{c,j}}. \quad (22)$$

Denote $\gamma_D(\boldsymbol{W}) = \pi_c \prod_j \theta_{c,j}^{w_{c,j}}$. Then, $\hat{P}_D(c|\boldsymbol{x})$ defined in (7) can be re-written as $\hat{P}_D(c|\boldsymbol{x}) = \frac{\gamma_D(\boldsymbol{W})}{\sum_{c'} \gamma_D(\boldsymbol{W})}$. It is easy to show that

$$\frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial \gamma_D(\boldsymbol{W})} = \frac{\sum_{c'\neq c} \gamma_D(\boldsymbol{W})}{(\sum_{c'} \gamma_D(\boldsymbol{W}))^2}, \quad (23)$$

$$\frac{\partial \gamma_D(\boldsymbol{W})}{\partial w_{c,j}} = \gamma_D(\boldsymbol{W}) \log(\theta_{c,j}). \quad (24)$$

Derive $\frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial w_{c,j}}$ using the chain rule by utilizing (23) and (24), and then plug it into (22) to obtain the partial derivative of f w.r.t. $w_{c,j}$ as defined in (15).

Secondly, the partial derivative of f w.r.t. w_j is derived as:

$$\frac{\partial f}{\partial w_j} = -(1-\alpha) \sum_{\boldsymbol{x}\in D} \sum_{c} \left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\right) \frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j}. \quad (25)$$

Denote $\gamma_I(\boldsymbol{w}) = \pi_c \prod_j \theta_{c,j}^{w_j}$. Similarly, $\hat{P}_I(c|\boldsymbol{x})$ defined in (8) can be re-written as $\hat{P}_I(c|\boldsymbol{x}) = \frac{\gamma_I(\boldsymbol{w})}{\sum_{c'} \gamma_I(\boldsymbol{w})}$. Note that every term in the summation of the denominator is a function of w_j. The partial derivative $\frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j}$ is calculated as:

$$\frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j} = \frac{1}{\sum_{c'} \gamma_I(\boldsymbol{w})} \left(\frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} - \hat{P}_I(c|\boldsymbol{x}) \sum_{c'} \frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} \right)$$

Similar to (24), it is easy to show that $\frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} = \gamma_I(\boldsymbol{w}) \log(\theta_{c,j})$. Plug it into (25), the partial derivative of f w.r.t. w_j shown in (16) can be obtained.

REFERENCES

[1] J. Ren, X. Jiang, and J. Yuan, "A complete and fully automated face verification system on mobile devices," *Pattern Recognit.*, vol. 46, no. 1, pp. 45–56, Jan. 2013.

[2] C. R. Ratto, K. D. Morton, Jr., L. M. Collins, and P. A. Torrione, "Bayesian context-dependent learning for anomaly classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 4, pp. 1969–1981, Apr. 2014.
[3] J. Ren, X. Jiang, and J. Yuan, "A chi-squared-transformed subspace of LBP histogram for visual recognition," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1893–1904, Jun. 2015.
[4] J. Ren, X. Jiang, J. Yuan, and N. Magnenat-Thalmann, "Sound-event classification using robust texture features for robot hearing," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 447–458, Mar. 2017.
[5] J. Ren and X. Jiang, "Regularized 2-D complex-log spectral analysis and subspace reliability analysis of micro-Doppler signature for UAV detection," *Pattern Recognit.*, vol. 69, pp. 225–237, Sep. 2017.
[6] X. Wang, X. Jiang, and J. Ren, "Blood vessel segmentation from fundus image by a cascade classification framework," *Pattern Recognit.*, vol. 88, pp. 331–341, Apr. 2019.
[7] X. Jiang, "Linear subspace learning-based dimensionality reduction," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 16–26, Mar. 2011.
[8] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, "Fast Gaussian Naïve Bayes for searchlight classification analysis," *NeuroImage*, vol. 163, pp. 471–479, Dec. 2017.
[9] T.-T. Wong, "A hybrid discretization method for Naïve Bayesian classifiers," *Pattern Recognit.*, vol. 45, no. 6, pp. 2321–2325, 2012.
[10] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naïve Bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, May 2018.
[11] C.-Z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naïve Bayes classifiers secure against the substitution-then-comparison attack," *Inf. Sci.*, vol. 444, pp. 72–88, May 2018.
[12] L. Dou, X. Li, H. Ding, L. Xu, and H. Xiang, "IRNA-m5C_NB: A novel predictor to identify RNA 5-Methylcytosine sites based on the Naïve Bayes classifier," *IEEE Access*, vol. 8, pp. 84906–84917, 2020.
[13] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, and R. Lara-Cabrera, "A collaborative filtering approach based on Naïve Bayes classifier," *IEEE Access*, vol. 7, pp. 108581–108592, 2019.
[14] Y. Zhang, J. Wu, C. Zhou, and Z. Cai, "Instance cloned extreme learning machine," *Pattern Recognit.*, vol. 68, pp. 52–65, Aug. 2017.
[15] J. Wu, S. Pan, X. Zhu, P. Zhang, and C. Zhang, "SODE: Self-adaptive one-dependence estimators for classification," *Pattern Recognit.*, vol. 51, pp. 358–377, Mar. 2016.
[16] L. Jiang, S. Wang, C. Li, and L. Zhang, "Structure extended multinomial Naïve bayes," *Inf. Sci.*, vol. 329, pp. 346–356, Feb. 2016.
[17] L. Jiang, Z. Cai, H. Zhang, and D. Wang, "Naïve Bayes text classifiers: A locally weighted learning approach," *J. Experim. Theor. Artif. Intell.*, vol. 25, no. 2, pp. 273–286, Jun. 2013.
[18] S. Wang, L. Jiang, and C. Li, "Adapting Naïve Bayes tree for text classification," *Knowl. Inf. Syst.*, vol. 44, no. 1, pp. 77–89, Jul. 2015.
[19] L. Jiang, D. Wang, and Z. Cai, "Discriminatively weighted Naïve Bayes and its application in text classification," *Int. J. Artif. Intell. Tools*, vol. 21, no. 1, Feb. 2012, Art. no. 1250007.
[20] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in Naïve Bayes for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2508–2521, Sep. 2016.
[21] L. Jiang, G. Kong, and C. Li, "Wrapper framework for test-cost-sensitive feature selection," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Mar. 26, 2019, doi: 10.1109/TSMC.2019.2904662.
[22] L. Jiang, C. Li, S. Wang, and L. Zhang, "Deep feature weighting for Naïve Bayes and its application to text classification," *Eng. Appl. Artif. Intell.*, vol. 52, pp. 26–39, Jun. 2016.
[23] L. Jiang, L. Zhang, C. Li, and J. Wu, "A correlation-based feature weighting filter for Naïve Bayes," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 201–213, Feb. 2019.
[24] M. Hall, "A decision tree-based attribute weighting filter for Naïve Bayes," in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.* Cham, Switzerland: Springer, 2006, pp. 59–70.
[25] C.-H. Lee, F. Gutierrez, and D. Dou, "Calculating feature weights in Naïve Bayes with kullback-leibler measure," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 1146–1151.
[26] C.-H. Lee, "An information-theoretic filter approach for value weighted classification learning in Naïve Bayes," *Data Knowl. Eng.*, vol. 113, pp. 116–128, Jan. 2018.
[27] H. Ma, W. Yan, Z. Yang, and H. Liu, "Real-time foot-ground contact detection for inertial motion capture based on an adaptive weighted Naïve Bayes model," *IEEE Access*, vol. 7, pp. 130312–130326, 2019.
[28] H. Zhang and S. Sheng, "Learning weighted Naïve Bayes with accurate ranking," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 567–570.
[29] S. Taheri, J. Yearwood, M. Mammadov, and S. Seifollahi, "Attribute weighted Naïve Bayes classifier using a local optimization," *Neural Comput. Appl.*, vol. 24, no. 5, pp. 995–1002, Apr. 2014.
[30] D. M. Diab and K. M. El Hindi, "Using differential evolution for fine tuning Naïve Bayesian classifiers and its application for text classification," *Appl. Soft Comput.*, vol. 54, pp. 183–199, May 2017.
[31] S. Ruan, H. Li, C. Li, and K. Song, "Class-specific deep feature weighting for Naïve Bayes text classifiers," *IEEE Access*, vol. 8, pp. 20151–20159, 2020.
[32] M. Li and K. Liu, "Causality-based attribute weighting via information flow and genetic algorithm for Naïve Bayes classifier," *IEEE Access*, vol. 7, pp. 150630–150641, 2019.
[33] J. Wu and Z. Cai, "Attribute weighting via differential evolution algorithm for attribute weighted Naïve Bayes (WNB)," *J. Comput. Inf. Syst.*, vol. 7, no. 5, pp. 1672–1679, Mar. 2011.
[34] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, "Alleviating Naïve Bayes attribute independence assumption by attribute weighting," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1947–1988, Jul. 2013.
[35] L. Jiang, L. Zhang, L. Yu, and D. Wang, "Class-specific attribute weighted Naïve Bayes," *Pattern Recognit.*, vol. 88, pp. 321–330, Apr. 2019.
[36] L. Yu, S. Gan, Y. Chen, and M. He, "Correlation-based weight adjusted Naïve Bayes," *IEEE Access*, vol. 8, pp. 51377–51387, 2020.
[37] X. Jiang, B. Mandal, and A. Kot, "Eigenfeature regularization and extraction in face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 383–394, Mar. 2008.
[38] X. Jiang and J. Lai, "Sparse and dense hybrid representation via dictionary decomposition for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1067–1079, May 2015.
[39] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.
[40] P. Cao, X. Liu, J. Yang, D. Zhao, M. Huang, and O. Zaiane, "ℓ_2, 1-ℓ_1 regularized nonlinear multi-task representation learning based cognitive performance prediction of Alzheimer's disease," *Pattern Recognit.*, vol. 79, pp. 195–215, Jul. 2018.
[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.
[42] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, no. 4, pp. 550–560, 1997.
[43] A. Asuncion and D. Newman, "UCI machine learning repository," Tech. Rep., 2007.
[44] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," Tech. Rep., 1993.
[45] A. R. Webb, *Statistical Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2003.
[46] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.*, vol. 20, no. 3, pp. 286–307, Sep. 1994.

SHIHE WANG received the B.Sc. degree from the University of Nottingham Ningbo China (UNNC), in 2019, where he is currently pursuing the Ph.D. degree in computer science with the scholarship cooperated with Microsoft Research Asia.

In 2018, he joined a summer research program in UNNC and learnt research methodology. His research interests include Bayesian classification framework, data transformation, pattern recognition, and machine learning.

JIANFENG REN (Member, IEEE) received the B.Eng. degree from the National University of Singapore, Singapore, in 2001, and the M.Sc. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 2009 and 2015, respectively. From 2003 to 2007, he worked in industrial sectors. In December 2007, he joined NTU as a Project Officer, responsible for the development of the face verification system on mobile devices. In September 2011, he joined the BeingThere Centre, Institute of Media Innovation, NTU, as a Research Associate. He worked as a Postdoctoral Research Fellow with the School of Electrical and Electronic Engineering, NTU, from 2015 to 2018. He has been working with the School of Computer Science, University of Nottingham Ningbo China, as an Assistant Professor, since 2018. He has authored 13 journal articles and ten conference papers, including two TIP papers, three SPL papers, and one TMM paper. His research interests include image/video processing, statistical pattern recognition, machine learning, and radar target recognition.

RUIBIN BAI received the bachelor's and master's degrees in mechanical engineering from Northwestern Polytechnic University, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the University of Nottingham, Nottingham, U.K., in 2005.

Before taking a Lectureship with the Division of Computer Science, University of Nottingham Ningbo, Ningbo, China, he spent two years with the School of Computer Science, University of Nottingham, as a Postdoctoral Research Fellow. His research interests include investigation and development of adaptive and intelligent decision support systems for scheduling and timetabling, logistics, transportation, layout, and space planning.

Dr. Bai is a member of the Automated Scheduling, Optimization and Planning Research Group. He has also refereed for a number of internationally recognized conferences, such as Genetic and Evolutionary Computation, in 2009, Evolutionary Computation in Combinatorial Optimization, from 2006 to 2009, Computational Intelligence in Scheduling, from 2007 to 2009, Intelligent Systems Design and Applications, in 2006/2007, Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, in 2007, International Conference on Tools with Artificial Intelligence, in 2006, and Artificial Intelligence and Pattern Recognition, in 2007. He has been working collaboratively with several internationally established researchers from different countries. He has served as a Referee for several leading journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Omega*, *Journal of Heuristics*, *Annals*, *Asia–Pacific Journal of Operational*, and *Journal of Retailing*.