

Received December 18, 2019, accepted January 19, 2020, date of publication January 23, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968984

# Class-Specific Deep Feature Weighting for Naïve Bayes Text Classifiers

SHUFEN RUAN<sup>ID</sup><sup>1,2</sup>, HONGWEI LI<sup>1</sup>, CHAOQUN LI<sup>1</sup>, AND KUNFANG SONG<sup>3</sup>

<sup>1</sup>School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, China

<sup>2</sup>Institute of Mechanical Engineering and Electronic Information, Wuhan University of Engineering Science, Wuhan 430200, China

<sup>3</sup>Institute of Mathematics and Computer Science, Wuhan Textile University, Wuhan 430073, China

Corresponding author: Hongwei Li (hwli@cug.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1711267, and in part by Fundamental Research Funds for Central Universities under Grant CUG2018JM18.

**ABSTRACT** Feature weighting is used to alleviate the conditional independence assumption of Naïve Bayes text classifiers and consequently improve their generalization performance. Most traditional feature weighting algorithms use general feature weighting, which assigns the same weight to each feature for all classes. We focus on class-specific feature weighting approaches, which discriminatively assign each feature a specific weight for each class. This paper uses a statistical feature weighting technique and proposes a new class-specific deep feature weighting method for Multinomial Naïve Bayes text classifiers. In this deep feature weighting method, feature weights are not only incorporated into the classification formulas but they are also incorporated into the conditional probability estimates of Multinomial Naïve Bayes text classifiers. Experimental results for a large number of text classification datasets validate the effectiveness and efficiency of our method.

**INDEX TERMS** Multinomial Naïve Bayes text classifiers, class-specific feature weighting, statistic, deep feature weighting.

## I. INTRODUCTION

With the explosive growth of text information on the Internet, automated processing of massive text data has become a challenge. Automatic text classification is used to automatically assign a textual document to a pre-specified set of classes, which can help people retrieve, query, and utilize information. Current common text classification algorithms include [8]: Naïve Bayes [2], K-nearest neighbors [3], decision trees [4], support vector machine (SVM) [5], and recent deep learning methods such as convolutional neural networks (CNNs) [6], recurrent neural networks (RNNs), and so on [7], [28].

Among these algorithms, the Naïve Bayes model is widely used in classification because it is simple, efficient, and easy to understand. The multinomial Naïve Bayes (MNB) model is a widely used text classification model. The MNB model assumes that each document is drawn from a multinomial distribution of words and that all features are conditionally independent with given values of the class variable [9]. This feature-independence assumption is rarely true in reality. To weaken this assumption, scholars have improved the

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Peng <sup>ID</sup>.

Naïve Bayes model from five main aspects: feature weighting [10]–[13], [20], feature selection [14]–[17], instance weighting [19], [23], instance selection [21], [22], and structure extension [18]. This study focuses on feature weighting approaches for MNB text classifiers.

Generally, feature weighting algorithms are mainly divided into two categories: general feature weighting and class-specific feature weighting. General feature weighting approaches assign the same weight to each feature for all classes. Class-specific feature weighting approaches discriminatively assign each feature a specific weight for each class [25]. Most traditional feature weighting algorithms use general feature weighting. However, the importance of features for different classes should be different; therefore, class-specific feature weighting is more reasonable than general feature weighting.

With regard to general feature weighting, there exist some approaches dramatically improved the Naïve Bayes text classifiers. Jiang *et al.* [11] and Wang *et al.* [13] proposed a CFS-based general feature weighting approach which firstly conducts a correlation-based feature selection(CFS) process to select a best feature subset and then assigns larger weights to the features in the best feature subset and smaller

weights to others. Zhang *et al.* [1] proposed a decision tree-based feature weighting approach for Naïve bayes text classifiers, in which the weight of a feature is set to  $\frac{5}{\sqrt{d}}$  if the minimum depth at which the feature is tested in the built tree is  $d$ , and 1 if the feature does not appear in the built tree. Kim *et al.* [8] proposed an approach to improve the Naïve Bayes text classifier with semantic tensor space model for document representation.

With regard to class-specific feature weighting, a few approaches have been investigated. Jiang *et al.* proposed a new class-specific feature weighting approach for Naïve Bayes classifiers; they showed that class-specific feature weighting is more fine-grained than general feature weighting for Naïve Bayes [25]. Tang *et al.* used a probabilistic density-function projection theorem to build a class-specific MNB classifier [24], [31]. Youn introduced a class-dependent feature weighting approach as a new feature-ranking method for Naïve Bayes [26]. Li proposed an enhanced Naïve Bayes classifier for text classification by weighting terms based on a variant  $\chi^2$  statistic, denoted by  $R_{wc}$  [12], [30].

Most of these existing approaches, however, do not incorporate the learned feature weights into the conditional probability estimates of Bayesian classifiers; this incorporation is included in deep feature weighting [11]. Wang *et al.* [22] and Zhang *et al.* [1] combined general feature weighting and deep feature weighting and showed that deep feature weighting can further improve the performance of Naïve Bayes text classifiers, but their approaches suffer from relative high execution time. To the best of our knowledge, there are no studies incorporating deep feature weighting into class-specific feature weighting for MNB classifiers.

Motivated by previous research achievements, in this study, we attempt to propose a new class-specific deep feature weighting method based on statistic metrics for MNB text classifiers. Our method not only assigns each feature a specific weight for each class but also estimates the conditional probabilities of text classifiers by deeply computing feature weighted frequencies from training data. Extensive experimental results show that our class-specific deep feature weighting approach outperforms other competitors.

The rest of this paper is organized as follows. In Section II, we describe the basic concepts of Naïve Bayes text classifiers and related feature weighting techniques for text categorization. In Section III, we propose our class-specific deep feature weighting approach. Section IV reports the experimental setup and results in detail. Conclusions and future works are presented in Section V.

## II. RELATED WORK

### A. MULTINOMIAL NAÏVE BAYES TEXT CLASSIFIER

Given a test document represented by a vector  $\langle a_1, a_2, \dots, a_m \rangle$ , according to the definition of multinomial distribution

and Bayes' rule, MNB classifies using (1).

$$c(d) = \arg \max_{c \in L} \left[ \log P(c) + \sum_{i=1}^m f_i \log p(a_i|c) \right] \quad (1)$$

where  $m$  is the number of features,  $a_i$  is the value of the  $i^{th}$  feature,  $L$  is the set of all class labels,  $c$  represents the value that the class variable can take, and  $f_i$  is the frequency count of the word  $a_i$  in a document  $d$ . The prior probability  $p(c)$  and the conditional probability  $p(a_i|c)$  are generally estimated by (2) and (3), respectively.

$$P(c) = \frac{\sum_{j=1}^n \delta(c_j, c) + 1}{n + n_c} \quad (2)$$

$$P(a_i|c) = \frac{\sum_{j=1}^n f_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^m \sum_{j=1}^n f_{ji} \delta(c_j, c) + m} \quad (3)$$

where  $n$  is the number of training documents,  $n_c$  is the number of classes,  $m$  is the number of different words in all of the documents,  $c_j$  is the class label of the  $j^{th}$  training document,  $f_{ji}$  is the  $i^{th}$  word's frequency count in the  $j^{th}$  training document, and  $\delta(c_j, c)$  is a binary function, which is defined as (5).

$$\delta(c_j, c) = \begin{cases} 1, & \text{if } c_j = c \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The MNB text classifier is based on the assumption that all features are conditionally independent, which is rarely true in reality. To relax the independence assumption, feature weighting approaches have been proposed considering two aspects: general feature weighting and class-specific feature weighting.

General feature weighting assigns the same weight value to each feature for all classes as follows (5).

$$c(d) = \arg \max_{c \in L} \left[ \log P(c) + \sum_{i=1}^m W_i f_i \log p(a_i|c) \right] \quad (5)$$

where  $W_i \in R^+$  represents the weight of the  $i^{th}$  feature (word)  $a_i$  for all classes.

Class-specific feature weighting assigns each feature a specific weight for each class as follows (6).

$$c(d) = \arg \max_{c \in L} \left[ \log P(c) + \sum_{i=1}^m W_{ic} f_i p(a_i|c) \right] \quad (6)$$

where  $W_{ic} \in R^+$  represents the weight of the  $i^{th}$  feature (word)  $a_i$  for a specific class  $c$ . In (5), we need to learn a  $m$ -dimensional feature weight vector, which is composed of  $W_i$ . In (6), a  $m \times n_c$  feature weight matrix is required to be learned, which is composed of  $W_{ic}$ . As we can see, (5) is the special case of (6) where for different classes,  $W_{ic} = W_i$  [25].

Thus, learning a feature weight matrix  $(W_{ic})_{m \times n_c}$  is crucial to improving Naïve Bayes text classifiers by class-specific

**TABLE 1.** Contingency table for class and term  $a_i$ .

	presence of $a_i$	absence of $a_i$
labeled as $c$	$A$	$B$
not labeled as $c$	$C$	$D$

feature weighting. To learn the weight matrix  $(W_{ic})_{m \times n_c}$ , statistical feature weighting approaches have been widely used. Kim et al. proposed a  $\chi^2$  statistics weighted approach based on a multivariate Poisson Naïve Bayesian model [20], [27]. Ng et al. proposed a variant of the  $\chi^2$  statistic metric. Li et al. proposed another modified  $\chi^2$  statistic feature weighting approach based on a MNB model [12]. In the next section, we provide a brief introduction of these statistical feature weighting approaches.

### B. STATISTICAL FEATURE WEIGHTING APPROACHES

The  $\chi^2$  statistic feature weighting approach measures the degree of dependency between a term and a specific class by measuring the difference between the observed co-occurrence frequencies and the expected frequencies according to an initial hypothesis (the hypothesis is that the term and class are independent of each other) [12], [29]. Then, it assigns the feature weight values based on the term-class dependency captured by the  $\chi^2$  statistic. To analyze the relationship between a term  $a_i$  and a class  $c$ , a two-way contingency table is created as shown in Table 1. The row variable (class variable) has two possible values  $\{c, \bar{c}\}$ . The column variable (term, i.e., feature variable) has two possible values  $\{a_i, \bar{a}_i\}$ .  $A$  is the number of documents that contain the term  $a_i$  and belong to the class  $c$ ,  $B$  is the number of documents that do not contain the term  $a_i$  and belong to the class  $c$ ,  $C$  is the number of documents that contain the term  $a_i$  and do not belong to the class  $c$ , and  $D$  is the number of documents that neither contain the term  $a_i$  nor belong to the class  $c$ . Let  $N$  be the total number of documents,  $E_{a_i,c}$  be the expected frequency, and  $N_{a_i,c}$  be the actual observation frequency.

When the term and class are independent of each other, the expected frequency  $E_{a_i,c}$  can be calculated as (7).

$$E_{a_i,c} = \frac{(A + C)(A + B)}{N} \quad (7)$$

The  $\chi^2$  statistic is defined as (8).

$$\chi_{a_i,c}^2 = \sum_{a_i} \sum_c \frac{(N_{a_i,c} - E_{a_i,c})^2}{E_{a_i,c}} \quad (8)$$

Equation (8) shows that the greater the difference between  $N_{a_i,c}$  and  $E_{a_i,c}$  is, the greater the  $\chi^2$  statistic is, and the more informative the term  $a_i$  is for the class  $c$ . Therefore,  $\chi_{a_i,c}^2$  can be used as the weight value of the feature  $a_i$  for the class  $c$ , i.e.,  $W_{i,c} = \chi_{a_i,c}^2$ .

Equation (8) can be interpreted with the probabilities as follows (9).

$$\chi_{a_i,c}^2 = \frac{N(p(a_i, c)p(\bar{a}_i, \bar{c}) - p(a_i, \bar{c})p(\bar{a}_i, c))^2}{p(a_i)p(\bar{a}_i)p(c)p(\bar{c})} \quad (9)$$

where  $p(a_i, c)$  represents the probability that the documents are in class  $c$  and contain the term  $a_i$ ,  $p(a_i)$  represents the probability that the documents contain the term  $a_i$ , and  $p(c)$  represents the probability that the documents are in the class  $c$ .

Ng et al. observed that the power of 2 at the numerator has the effect of equating the roles of the probabilities that indicate a positive correlation between  $a_i$  and  $c$  and those that indicate a negative correlation [32]. A variant of the  $\chi^2$  statistic called the *correlation coefficient* was proposed as follows [29].

$$CC_{a_i,c} = \frac{\sqrt{N}(p(a_i, c)p(\bar{a}_i, \bar{c}) - p(a_i, \bar{c})p(\bar{a}_i, c))}{\sqrt{p(a_i)p(\bar{a}_i)p(c)p(\bar{c})}} \quad (10)$$

where  $(CC_{a_i,c})^2 = \chi_{a_i,c}^2$ , and  $CC_{a_i,c}$  can be viewed as a ‘one-sided’  $\chi^2$  metric.

Li et al. thought that the  $\chi^2$  statistic and correlation coefficient have bias against the classes with small sizes when a term is uniformly distributed across multiple classes. He proposed another statistical metric named  $R_{a_i,c}$ , which can measure whether the dependency between a term and a class is positive or negative, then weights the terms based on the positive term-class dependency captured by  $R_{a_i,c}$  [12]. The  $R_{a_i,c}$  for a term  $a_i$  and a class  $c$  is defined as (11).

$$R_{a_i,c} = \frac{N_{a_i,c}}{E_{a_i,c}} = \frac{p(a_i, c)p(\bar{a}_i, \bar{c}) - p(a_i, \bar{c})p(\bar{a}_i, c)}{p(a_i)p(c)} \quad (11)$$

If  $R_{a_i,c} > 1$ , then there is a positive dependency between the term  $a_i$  and the class  $c$ , and  $W_{ic}$  is set to be  $R_{a_i,c}$ . Otherwise, if  $R_{a_i,c} \leq 1$ , there is a negative dependency between the term  $a_i$  and the class  $c$ , and  $W_{ic}$  is set to be 1. (Note that when  $R_{a_i,c} \leq 1$ ,  $W_{ic}$  is set to be 1 instead of  $R_{a_i,c}$ .)

Our experimental results show that among these approaches, the feature weighting approach based on  $R_{a_i,c}$  performs best. However, there still exist certain limitations in the  $R_{a_i,c}$  metric. In the next section, we use an example to indicate the related problem and introduce a new feature weighted measure denoted by  $CR_{a_i,c}$ .

### III. CLASS-SPECIFIC DEEP FEATURE WEIGHTED MULTINOMIAL NAÏVE BAYES TEXT CLASSIFIERS

*Example:* Let us consider a corpus with 63 labeled documents  $\{d_1, d_2, \dots, d_{63}\}$ , falling into three classes  $c_1, c_2, c_3$ . In total, there are three distinct features  $a_1, a_2, a_3$  in the corpus. The details are shown in Table 2.

According to (11), the values of the statistical metric  $R_{a_i,c}$  are listed in Table 3.

Equation (11) can be rewritten as (12).

$$R_{a_i,c} = \frac{p(a_i|c)}{p(a_i)} \quad (12)$$

**TABLE 2.** 63 documents in 3 classes with 3 terms.

	$c_1$	$c_2$	$c_3$	total
$a_1$	1	20	20	41
$a_2$	0	1	20	21
$a_3$	0	0	1	1
total	1	21	41	63

According to the corpus shown in Table 2,

$$R_{a_1,c_1} = \frac{p(a_1|c_1)}{p(a_1)} = \frac{100/100}{41/63} = 1.5366 \quad (13)$$

$$R_{a_1,c_2} = \frac{p(a_1|c_2)}{p(a_1)} = \frac{20/21}{41/63} = 1.4634 \quad (14)$$

Because  $R_{a_1,c_1} > R_{a_1,c_2}$ , we may obtain the conclusion that  $a_1$  is more relevant to the class  $c_1$  than to the class  $c_2$ . In fact, the distribution in Table 2 shows that although all of the documents in the class  $c_1$  contain  $a_1$ , only 1/41 of the documents containing  $a_1$  belong to  $c_1$ . 20/21 of the documents in the class  $c_2$  contain  $a_1$  and 20/41 of the documents containing  $a_1$  belong to  $c_2$ . Thus, we argue that  $a_1$  is less relevant to the class  $c_1$  than to the class  $c_2$ . The problem of  $R_{a_i,c}$  is that it overvalues the distribution of the term  $a_i$  in a specific class, but it does not consider the class distribution of documents containing the term  $a_i$ .

When we use statistical feature weighting approaches to measure the degree of dependency between a term  $a_i$  and a specific class  $c$ , we should consider the following information:

- (1) The distribution of the term  $a_i$  in the training documents.
- (2) The distribution of the term  $a_i$  in a specific class  $c$ .
- (3) The class distribution of the documents containing a term  $a_i$ .
- (4) The class frequency of the documents containing a term  $a_i$ , i.e., the number of classes of the documents containing a term  $a_i$ .

The statistical metric  $R_{a_i,c}$  only considers the first two aspects but ignores the last two. On the basis of the  $R_{a_i,c}$  statistic method, we propose a novel measure  $CR_{a_i,c}$  by introducing two new factors: the class distribution of the documents containing a term  $a_i$  and the class frequency factor. By introducing these two new factors, the new statistical metric  $CR_{a_i,c}$  can alleviate the problem of  $R_{a_i,c}$  and avoid the information loss caused by the negative correlation.

The new statistical metric  $CR_{a_i,c}$  for a term  $a_i$  and a class  $c$  is defined as (15)

$$CR_{a_i,c} = R_{a_i,c} \frac{p(a_i, c)}{p(a_i)} \ln(2 + \frac{n_c}{k_{a_i}}) \quad (15)$$

where  $p(a_i, c)$  represents the probability of the documents that are in the class  $c$  and contain the term  $a_i$ , and  $p(a_i)$  represents the probability of the documents that contain the term  $a_i$ .

**TABLE 3.** Statistical values  $R_{a_i,c}$  for the terms in classes.

	$c_1$	$c_2$	$c_3$
$a_1$	1.5366	1.4634	0.7496 (1)
$a_2$	0	0.1429 (1)	1.4634
$a_3$	0	0	1.5366

**TABLE 4.**  $CR_{a_i,c}$  statistical values for the terms in classes.

	$c_1$	$c_2$	$c_3$
$a_1$	0.0412	0.7842	0.4018
$a_2$	0	0.0085	1.7459
$a_3$	0	0	2.4730

$p(a_i, c)/p(a_i)$  is used to denote the class distribution of the documents containing a term  $a_i$ . The greater  $p(a_i, c)/p(a_i)$  is, the greater is the dependency between the term  $a_i$  and the class  $c$ .  $n_c$  is the number of classes, and  $k_{a_i}$  is the number of classes of documents that contain the term  $a_i$ . The greater  $k_{a_i}$  is, the smaller is the dependency between the term  $a_i$  and the class  $c$ .

According to (15), for the example in Table 2, the values of the statistic metric  $CR_{a_i,c}$  are calculated and listed in Table 4.

From Table 4, it is reasonable that the value of the statistic metric  $CR_{a_1,c_2}(0.7842)$  is larger than the value of the statistic data  $CR_{a_1,c_1}(0.0412)$ . In addition, Table 3 shows that both  $R_{a_1,c_3}(0.7496)$  and  $R_{a_2,c_2}(0.1429)$  are less than 1. This means that the dependencies between  $a_1$  and  $c_3$  and between  $a_2$  and  $c_2$  are considered negative; thus, both the weight value of  $a_1$  for the class  $c_3$  and the weight value of  $a_2$  for the class  $c_2$  are assigned as 1. Although  $R_{a_1,c_3}(0.7496)$  and  $R_{a_2,c_2}(0.1429)$  exhibit a large difference, their weight values are the same. This may lead to unreliable results and some useful information may also be lost. When our statistic metric  $CR_{a_i,c}$  is used to calculate  $CR_{a_1,c_3}(0.4018)$  and  $CR_{a_2,c_2}(0.0085)$ , the  $CR_{a_i,c}$  values are different. From the distribution given in Table 2, we know that the term  $a_3$  is a "rare word"; thus,  $CR_{a_3,c_3}(2.4730)$  is larger than  $CR_{a_2,c_3}(1.7459)$  and  $CR_{a_1,c_3}(0.4018)$ . This example shows that  $CR_{a_i,c}$  describes the term-class dependency more accurately than the  $R_{a_i,c}$  statistic metric.

Then, we set the weight value  $W_{a_i,c}$  to be  $CR_{a_i,c}$ . After obtaining the weight value of each feature  $a_i$  for the class  $c$  by employing (15), we apply the weight value  $W_{a_i,c}$  to (6) and (3) to improve the classification performance of MNB. That is, (6) and (3) are now modified as the following (16) and (17), respectively.

$$c(d) = \arg \max_{c \in L} \left[ \log P(c) + \sum_{i=1}^m CR_{a_i,c} f_i \log p(a_i|c) \right] \quad (16)$$

**Algorithm 1** CDFW-MNB ( $D, d$ )

**Require:** a training document set  $D$ , a test document  $d$   
**Ensure:** the class value  $c(d)$  of the test document  $d$

- 1: Estimate the prior probability  $p(c)$  of each class  $c$  by (2)
- 2: Estimate the class-specific feature weights  $W_{a_i, c}$  of each feature  $a_i$  for the specific class  $c$  by (15)
- 3: Estimate the conditional probability  $p(a_i|c)$  of each feature  $a_i$  given the class by (17)
- 4: For the test document  $d$ , predict its class value  $c(d)$  using (16)
- 5: Return the class value  $c(d)$  of  $d$ .

$$P(a_i|c) = \frac{\sum_{j=1}^n CR_{a_i c} f_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^m \sum_{j=1}^n CR_{a_i c} f_{ji} \delta(c_j, c) + m} \quad (17)$$

where  $a_i (i = 1, 2, \dots, m)$  is the  $i^{th}$  feature in the document  $d$ .

When we apply the feature weighting approach to MNB text classifiers, we call the resulting model a class-specific deep feature weighted MNB text classifier (CDFW-MNB). The detailed algorithm procedure is shown in Algorithm 1.

**IV. EXPERIMENTS AND RESULTS****A. EXPERIMENTAL SETUP AND BENCHMARK DATA**

The purpose of these experiments is to validate the classification performance of MNB text classifiers by employing our proposed class-specific deep feature weighting approach. We implemented our proposed CDFWMNB and other competitors MNB,  $\chi^2$ MNB, CCMNB,  $R_{wc}$ MNB, CFSMNB and DTWMNB on the WEKA platform [34]. We conducted our experiments on 19 widely used text classification benchmark datasets published on the main website of the WEKA platform. A detailed description of these 19 datasets is provided in Table 5. The algorithms compared and their abbreviations are as follows:

- MNB: Multinomial Naïve Bayes model [9].
- $\chi^2$ MNB: MNB model employing  $\chi^2$  statistic-based feature weighting [12].
- CCMNB: MNB model employing correlation coefficient statistic-based feature weighting [29].
- $R_{wc}$ MNB: MNB model employing the  $R_{wc}$  statistic-based feature weighting approach [12].
- CFSMNB: MNB model employing the CFS-based feature weighting approach [11], [13].
- DTWMNB: MNB model employing the decision tree weighting approach [1].
- CDFWMNB: MNB model employing the  $CR_{a_i c}$  class-specific deep feature weighting.

In our experiments, the classification accuracy of each algorithm on each dataset is obtained via 10 runs of 10-fold cross-validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same test

**TABLE 5.** Datasets used in our experiments.

Dataset	Document number	Word number	Class number
fbis	2463	2000	17
la1s	3204	13195	6
la2s	3075	12432	6
news	9558	26833	44
oh0	1003	3182	10
oh10	1050	3238	10
oh15	913	3100	10
oh5	918	3012	10
ohscal	11162	11466	10
re0	1657	3758	25
re1	1504	2886	13
tr11	414	6429	9
tr12	313	5804	8
tr21	336	7902	6
tr23	204	5832	6
tr31	927	10128	7
tr41	878	7454	10
tr45	690	8261	10
wap	1560	8460	20

sets. In particular, the cross-validation folds are the same for all the experiments on each dataset [13].

**B. EXPERIMENTAL RESULTS AND ANALYSIS**

The detailed experimental results are presented in Tables 6-9. Table 6 shows the classification accuracy of each algorithm on each dataset. The averages of the classification accuracy are listed at the bottom of the tables. These averages across all datasets provide a gross indication of the relative performance in addition to other statistics [1].

We then employed a Friedman test to compare multiple algorithms over multiple datasets [1], [28]. The Friedman test is a nonparametric equivalent of the repeated-measures ANOVA [33]. The average rankings of the algorithms obtained by applying the Friedman test are also summarized at the bottom of Table 6. With 7 algorithms and 19 datasets,  $F_F$  is distributed according to the  $F$  distribution with 6 and 114 degrees of freedom.  $F_F$  calculated from the average rankings is 14.837934, which is greater than the critical value of  $F(6, 114)$  for  $\alpha = 0.05$  (The table of critical values can be found in any statistical book). Therefore, we reject the null hypotheses.

**TABLE 6.** Classification accuracy (%) comparisons of MNB versus  $\chi^2$ MNB, CCMNB,  $R_{wc}$ MNB, CFSMNB, DTWMNB and CDFWMNB.

Dataset	MNB	$\chi^2$ MNB	CC MNB	$R_{wc}$ MNB	CFS MNB	DTW MNB	CDFW MNB
fbis.mat	77.11	71.54	75.77	79.87	77.02	79.37	80.57
la1.mat	88.41	81.12	82.70	87.88	84.28	85.67	86.57
la2.mat	89.88	82.11	84.53	88.72	86.34	86.53	87.78
new3.mat	79.28	71.59	76.07	80.66	59.85	60.03	80.13
oh0.mat	89.55	88.38	91.28	89.05	89.72	92.28	90.35
oh10.mat	80.60	78.94	79.97	80.41	80.82	82.59	80.88
oh15.mat	83.60	81.99	83.97	83.61	83.84	86.35	84.14
oh5.mat	86.63	86.41	89.44	86.46	86.89	90.99	89.36
ohscal.mat	74.70	70.26	74.02	74.18	81.02	81.06	75.10
re0.mat	80.02	80.07	74.16	77.07	80.30	81.18	80.33
re1.mat	83.31	79.73	85.92	82.72	83.81	86.10	86.92
tr11.mat	85.21	85.30	87.95	85.44	84.82	86.58	87.91
tr12.mat	80.99	82.70	78.57	84.76	81.63	84.89	85.62
tr21.mat	61.90	84.26	78.31	69.63	62.83	62.41	90.42
tr23.mat	71.15	80.20	82.54	73.82	71.54	78.56	88.03
tr31.mat	94.60	91.61	93.86	94.20	94.52	95.64	95.29
tr41.mat	94.65	91.37	94.42	93.05	94.73	95.25	94.73
tr45.mat	83.64	83.16	90.48	88.88	84.25	89.00	91.55
wap.mat	81.22	76.24	71.58	76.33	82.05	82.76	79.25
Average	82.44	81.42	82.92	82.99	81.60	83.54	86.05
Average Ranking	4.50	6.00	4.40	4.25	4.42	2.40	2.02

Then, we proceeded with a post-hoc Holm's test to further analyze the pairs of algorithms that are significantly different. Table 7 reports the obtained  $z$ -values and  $p$ -values and also indicates the pairs of algorithms that are significantly different.

Simultaneously, we take advantage of KEEL data mining software tool to complete Wilcoxon signed-ranks test for thoroughly comparing each pair of algorithms [28]. The Wilcoxon signed-ranks test is a non-parametric statistical test, which ranks the differences in performance of two algorithms for each dataset ignoring the signs, and compares the ranks for positive and negative differences. Table 8 reports the obtained results.

These experimental results show that our proposed class-specific deep feature weighting approach significantly outperforms its competitors. Our results can be summarized as follows:

(1) Table 6 shows that, in terms of the average classification accuracy, our class-specific deep feature weighting approach is clearly better than its competitors. The accuracy of our algorithm is 86.05%, that of MNB is 82.44%, that of  $\chi^2$ MNB is 81.42%, that of CCMNB is 82.92%, that of  $R_{wc}$ MNB is

**TABLE 7.** Classification accuracy post-hoc comparisons.

$i$	Algorithms	$z = (R_0 - R_i)/SE$	$p$	Holm
21	$\chi^2$ MNB vs. CDFWMNB	5.818804	0	0.004762
20	$\chi^2$ MNB vs. DTWMNB	5.26986	0	0.005
19	MNB vs. CDFWMNB	3.623029	0.000291	0.005263
18	CFSMNB vs. CDFWMNB	3.51324	0.000443	0.005556
17	CC MNB vs. CDFWMNB	3.476644	0.000508	0.005882
16	$R_{wc}$ MNB vs. CDFWMNB	3.257066	0.001126	0.00625
15	MNB vs. DTWMNB	3.074085	0.002111	0.006667
14	CFSMNB vs. DTWMNB	2.964296	0.003034	0.007143
13	CC MNB vs. DTWMNB	2.9277	0.003415	0.007692
12	$R_{wc}$ MNB vs. DTWMNB	2.708123	0.006767	0.008333
11	$\chi^2$ MNB vs. $R_{wc}$ MNB	2.561738	0.010415	0.009091
10	$\chi^2$ MNB vs. CC MNB	2.34216	0.019172	0.01
9	$\chi^2$ MNB vs. CFSMNB	2.305564	0.021135	0.011111
8	MNB vs. $\chi^2$ MNB	2.195775	0.028108	0.0125
7	DTWMNB vs. CDFWMNB	0.548944	0.583044	0.014286
6	MNB vs. $R_{wc}$ MNB	0.365963	0.714393	0.016667
5	$R_{wc}$ MNB vs. CFSMNB	0.256174	0.797817	0.02
4	CC MNB vs. $R_{wc}$ MNB	0.219578	0.8262	0.025
3	MNB vs. CC MNB	0.146385	0.883617	0.033333
2	MNB vs. CFSMNB	0.109789	0.912577	0.05
1	CC MNB vs. CFSMNB	0.036596	0.970807	0.1

Holm's procedure rejects those hypotheses that have an unadjusted  $p$ -value  $\leq 0.009091$ .

• $\chi^2$ MNB vs. CDFWMNB; • $\chi^2$ MNB vs. DTWMNB; •MNB vs. CDFWMNB; •CFSMNB vs. CDFWMNB; •CCMNB vs. CDFWMNB; • $R_{wc}$ MNB vs. CDFWMNB; •MNB vs. DTWMNB; •CFSMNB vs. DTWMNB; •CCMNB vs. DTWMNB; • $R_{wc}$ MNB vs. DTWMNB; • $\chi^2$ MNB vs.  $R_{wc}$ MNB.

82.99%, that of CFSMNB is 81.60%, and that of DTWMNB is 83.54%.

**TABLE 8.** The classification accuracy comparisons computed by the Wilcoxon test.

	MNB	$\chi^2$ MNB	CC MNB	RW MNB	CFS MNB	DTW MNB	CDFW MNB
MNB	-	•				o	o
$\chi^2$ MNB	o	-		o		o	o
CCMNB			-				o
$R_{wc}$ MNB		•		-		o	o
CFSMNB					-	o	o
DTWMNB	•	•			•	-	
CDFWMNB	•	•	•	•	•	-	-

Summary of the Wilcoxon test. •= the method in the row improves the method of the column. o= the method in the column improves the method of the row. Upper diagonal of level significance  $\alpha=0.1$ , lower diagonal level of significance  $\alpha=0.05$ .

**TABLE 9.** Elapsed training time(s) comparisons for MNB versus  $\chi^2$ MNB, CCMNB,  $R_{wc}$ MNB, CFSMNB, DTWMNB and CDFWMNB.

Dataset	MNB	$\chi^2$ MNB	CC MNB	$R_{wc}$ MNB	CFS MNB	DTW MNB	CDFW MNB
fbis.mat	0.00	0.00	0.00	0.00	20.76	27.77	0.00
la1.mat	0.01	0.01	0.01	0.01	389.01	245.01	0.01
la2.mat	0.00	0.01	0.01	0.00	221.73	83.91	0.01
new3.mat	0.04	0.06	0.06	0.05	216.15	119.22	0.08
oh0.mat	0.00	0.00	0.00	0.00	16.69	8.24	0.00
oh10.mat	0.00	0.00	0.00	0.00	6.01	12.14	0.00
oh15.mat	0.00	0.00	0.00	0.00	7.98	10.64	0.00
oh5.mat	0.00	0.00	0.00	0.00	4.88	5.54	0.00
ohscal.mat	0.01	0.01	0.01	0.01	485.90	173.86	0.01
re0.mat	0.00	0.00	0.00	0.00	13.83	27.33	0.00
re1.mat	0.00	0.00	0.00	0.00	13.12	30.31	0.01
tr11.mat	0.00	0.00	0.00	0.00	29.71	3.81	0.00
tr12.mat	0.00	0.00	0.00	0.00	13.16	1.34	0.00
tr21.mat	0.00	0.00	0.00	0.00	50.83	3.72	0.00
tr23.mat	0.00	0.00	0.00	0.00	7.32	0.50	0.00
tr31.mat	0.00	0.00	0.00	0.00	81.69	10.05	0.01
tr41.mat	0.00	0.00	0.00	0.00	46.18	10.88	0.01
tr45.mat	0.00	0.00	0.00	0.00	49.91	4.66	0.01
wap.mat	0.01	0.01	0.01	0.01	239.45	165.90	0.01
Average	0.01	0.01	0.01	0.01	100.75	49.73	0.01

(2) According to the Friedman test with the post-hoc Holm test based on the classification accuracy, the average

rankings of all approaches are respectively: CDFWMNB (2.02), MNB (4.5),  $R_{wc}$ MNB (6), CCMNB (4.4), and  $\chi^2$ MNB (4.25), CFSMNB (4.42) and DTWMNB (2.4). We can see that our feature weighting approach CDFWMNB is notably better than all of the other existing competitors.

(3) Both the classification accuracy post-hoc Holm comparisons in Table 7 and the Wilcoxon signed-ranks test results in Table 8 show that our class-specific deep feature weighting approach performs significantly better than its competitors: MNB,  $R_{wc}$ MNB, CCMNB, and  $\chi^2$ MNB and CFSMNB. This fully verifies the universal applicability of our feature weighting approach CDFWMNB for a wide range of domains and data characteristics.

In our another group of experiments below, we compare our approach to MNB,  $R_{wc}$ MNB, CCMNB,  $\chi^2$ MNB, CFSMNB and DTWMNB in terms of elapsed training time in seconds. Our experiments are performed on a desktop PC Quad core CPU @4.20 GHz and 16GB RAM. The detailed comparison results are shown in Table 9. From these comparison results, we can see that:

(4) In terms of the average elapsed training time, our feature weighting approach CDFWMNB runs as fast as its competitors: MNB,  $R_{wc}$ MNB, CCMNB,  $\chi^2$ MNB. But in terms of average classification accuracy, our approach CDFWMNB is notably better than MNB,  $R_{wc}$ MNB, CCMNB,  $\chi^2$ MNB.

(5) our feature weighting approach CDFWMNB runs significantly faster than the approaches CFSMNB and DTWMNB, especially for large datasets. The CFSMNB approach runs most slowly, because it uses a best first heuristic search to find a best feature subset from the whole feature space, which incurs an approximately quadratic time complexity.

In a word, in terms of average classification accuracy our CDFWMNB approach is obviously better than their competitors, and in terms of the average elapsed training time, our CDFWMNB approach runs much faster than the approaches CFSMNB and DTWMNB. Our class-specific deep feature weighting approach CDFWMNB keeps the best balance between classification accuracy and execution time.

## V. CONCLUSION AND FUTURE WORK

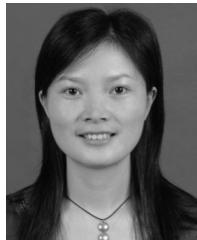
Most traditional feature weighting algorithms use general feature weighting in Naïve Bayes text classifiers. This study focuses on class-specific feature weighting approaches for Naïve Bayes text classifiers. In this study, we used the results of  $\chi^2$  statistic feature weighting algorithms to improve MNB text classifiers. We propose a new class-specific deep feature weighting method for MNB text classifiers, which not only assigns each feature a specific weight for each class but also estimates the conditional probabilities of the text classifier by deeply computing feature weighted frequencies from training data. Experimental results for a large number of text classification datasets validate the effectiveness and efficiency of our method.

In recent years, class-specific feature weighting has attracted increased attention from scholars. For the future

work, we will test whether our approach is effective for structured data. In recent years, deep learning has shown surprised performance in many fields, and we also will focus on more advanced text classifiers methods, such as CNN and RNN.

## REFERENCES

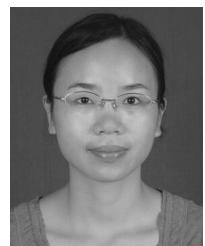
- [1] L. Zhang, L. Jiang, C. Li, and G. Kong, "Two feature weighting approaches for Naïve Bayes text classifiers," *Knowl.-Based Syst.*, vol. 100, pp. 137–144, May 2016.
- [2] J. Kramer and C. Gordon, "Improvement of a Naïve Bayes sentiment classifier using MRS-based features," in *Proc. 3rd Joint Conf. Lexical Comput. Semantics*, 2014, pp. 22–29.
- [3] S. Chen, "K-nearest neighbor algorithm optimization in text categorization," in *Proc. IOP Conf. Earth Environ. Sci.*, vol. 108, no. 5, 2018, pp. 52–74.
- [4] F. D. Comité, R. Gilleron, and M. Tommasi, "Learning multi-label alternating decision trees from texts and data," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.*, 2003, pp. 35–49.
- [5] Z. Wang and J. Liu, "PU Chinese text classifier based on support vector machine construction," *J. Nanjing Univ. Posts Telecommun. (Natural Sci. Ed.)*, vol. 35, no. 6, pp. 100–105, 2015.
- [6] A. Conneau, H. Schwenk, L. Barrau, and Y. Lecun, "Very deep convolutional networks for text classification," in *Proc. 15th Conf. Eur. Assoc. Comput. Linguistics: Long Papers*, 2016, pp. 1107–1116.
- [7] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, vol. 333, 2015, pp. 2267–2273.
- [8] H.-J. Kim, J. Kim, J. Kim, and P. Lim, "Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning," *Neurocomputing*, vol. 315, pp. 128–134, Nov. 2018.
- [9] A. McCallum and K. Nigam, "A comparison of event models for Naïve bayes text classification," in *Proc. AAAI Workshop Learn. Text Categorization*, vol. 752, 1998, pp. 41–48.
- [10] H. J. Escalante, M. A. García-Limón, A. Morales-Reyes, M. Graff, M. Montes-y-Gómez, E. F. Morales, and J. Martínez-Carranza, "Term-weighting learning via genetic programming for text classification," *Knowl.-Based Syst.*, vol. 83, no. 1, pp. 176–189, 2015.
- [11] 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.
- [12] Y. Li, C. Luo, and S. M. Chung, "Weighted Naïve Bayes for text classification using positive term-class dependency," *Int. J. Artif. Intell. Tools*, vol. 21, no. 1, 2012, Art. no. 250008.
- [13] S. Wang, L. Jiang, and C. Li, "A CFS-based feature weighting approach to Naïve Bayes text classifiers," in *Proc. 24th Int. Conf. Artif. Neural Netw.*, 2014, pp. 555–562.
- [14] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Speeding up incremental wrapper feature subset selection with Naïve Bayes classifier," *Knowl.-Based Syst.*, vol. 55, pp. 140–147, Jan. 2014.
- [15] K. Javed, S. Maruf, and H. A. Babri, "A two-stage Markov blanket based feature selection algorithm for text classification," *Neurocomputing*, vol. 157, pp. 91–104, Jun. 2015.
- [16] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory," *Inf. Fusion*, vol. 36, pp. 149–161, Jul. 2017.
- [17] L. Zhang, L. Jiang, and C. Li, "A new feature selection approach to Naïve Bayes text classifiers," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 2, 2016, Art. no. 1650003.
- [18] L. Jiang, S. Wang, C. Li, and L. Zhang, "Structure extended multinomial Naïve Bayes," *Inf. Sci.*, vol. 329, pp. 346–356, Feb. 2016.
- [19] J.-W. Bi, Y. Liu, Z. P. Fan, and E. Cambria, "Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model," *Int. J. Prod. Res.*, vol. 57, no. 22, pp. 7068–7088, 2019.
- [20] S.-B. Kim, H.-C. Rim, D. S. Yook, and H.-S. Lim, "Effective methods for improving Naïve Bayes text classifiers," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2002, pp. 414–423.
- [21] L. Jiang, Z. Cai, H. Zhang, and D. Wang, "Naïve Bayes text classifiers: A locally weighted learning approach," *J. Exp. Theor. Artif. Intell.*, vol. 25, no. 2, pp. 273–286, 2013.
- [22] S. Wang, L. Jiang, and C. Li, "Adapting Naïve Bayes tree for text classification," *Knowl. Inf. Syst.*, vol. 44, no. 1, pp. 77–89, 2015.
- [23] 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, 2012, Art. no. 1250007.
- [24] T. Bo, H. He, P. Baggenstoss, and S. Kay, "A Bayesian classification approach using class-specific features for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1602–1606, Jun. 2016.
- [25] L. Jiang, L. Zhang, L. Yu, and D. Wang, "Class-specific attribute weighted Naïve Bayes," *Pattern Recognit.*, vol. 88, pp. 321–330, Apr. 2019.
- [26] E. Youn and M. K. Jeong, "Class dependent feature scaling method using Naïve Bayes classifier for text datamining," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 477–485, 2009.
- [27] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for Naïve Bayes text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 11, pp. 1457–1466, Nov. 2006.
- [28] Y. Liu, J. W. Bi, and Z. P. Fan, "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms," *Expert Syst. Appl.*, vol. 80, pp. 323–339, Sep. 2017.
- [29] H. T. Ng, B. G. Wei, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," *AcM SIGIR Forum*, vol. 31, pp. 67–73, Jul. 2000.
- [30] Y. Li, C. Luo, and S. Chung, "Text clustering with feature selection by using statistical data," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 641–652, May 2008.
- [31] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive Bayes for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2508–2521, Sep. 2016.
- [32] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the use of feature selection and negative evidence in automated text categorization," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*, 2000, pp. 59–68.
- [33] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.
- [34] I. H. Witten, E. Frank, M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. San Mateo, CA, USA: Morgan Kaufmann, 2017.



**SHUFEN RUAN** received the M.S. degree in applied mathematics from China University of Geosciences (CUG), Wuhan, China, in 2007, where she is currently pursuing the Ph.D. degree. She is an Associate Professor with the School of Mechanical Engineering and Electronic Information, Wuhan University of Engineering Science, China. Her research interests include statistical signal processing, pattern recognition, and network information theory.



**HONGWEI LI** received the Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 1996. From July 1996 to July 1998, he was a Postdoctoral Fellow of the Institute of Information Science, Beijing Jiaotong University, Beijing, China. Since 1999, he has been a Professor with the School of Mathematics and Physics, China University of Geosciences, Wuhan, China. His research interests include pattern recognition, statistical signal processing, blind signal processing, multidimensional signal processing, and time series analysis.



**CHAOQUN LI** received the Ph.D. degree from the China University of Geosciences, Wuhan, China, in 2012. She is currently an Associate Professor with the Department of Mathematics, China University of Geosciences. She has been published over 30 refereed journal and conference papers, including articles in *Information Sciences*, *Knowledge and Information Systems*, *Engineering Applications of Artificial Intelligence*, *Knowledge-Based Systems*, *Expert Systems with Applications*, *Pattern Recognition Letters*, and *International Journal of Pattern Recognition and Artificial Intelligence*, and at ICANN, ICTAI, and PRICAI, since 2006. Her research interests include data mining and machine learning.



**KUNFANG SONG** received the B.S. and M.S. degrees in computer science and technology from Wuhan Textile University (WTU), in 2005 and 2011, respectively, and the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), in 2019. He is currently an Associate Professor with the Department of Mathematics and Computer Science, WTU. He has coauthored more than ten research articles. His research interests include computer architecture, parallel and distributed computing, and machine learning.

• • •