mathematics

MDPI

Article

Adapting Hidden Naive Bayes for Text Classification

Shengfeng Gan [1], Shiqi Shao [2], Long Chen [2], Liangjun Yu [1] and Liangxiao Jiang [2,*]

[1] College of Computer, Hubei University of Education, Wuhan 430205, China; sf_gan@hue.edu.cn (S.G.); yuliangjun@hue.edu.cn (L.Y.)
[2] School of Computer Science, China University of Geosciences, Wuhan 430074, China; sqshao@cug.edu.cn (S.S.); lchen@cug.edu.cn (L.C.)
* Correspondence: ljiang@cug.edu.cn; Tel.: +86-27-6788-3716

Abstract: Due to its simplicity, efficiency, and effectiveness, multinomial naive Bayes (MNB) has been widely used for text classification. As in naive Bayes (NB), its assumption of the conditional independence of features is often violated and, therefore, reduces its classification performance. Of the numerous approaches to alleviating its assumption of the conditional independence of features, structure extension has attracted less attention from researchers. To the best of our knowledge, only structure-extended MNB (SEMNB) has been proposed so far. SEMNB averages all weighted super-parent one-dependence multinomial estimators; therefore, it is an ensemble learning model. In this paper, we propose a single model called hidden MNB (HMNB) by adapting the well-known hidden NB (HNB). HMNB creates a hidden parent for each feature, which synthesizes all the other qualified features' influences. For HMNB to learn, we propose a simple but effective learning algorithm without incurring a high-computational-complexity structure-learning process. Our improved idea can also be used to improve complement NB (CNB) and the one-versus-all-but-one model (OVA), and the resulting models are simply denoted as HCNB and HOVA, respectively. The extensive experiments on eleven benchmark text classification datasets validate the effectiveness of HMNB, HCNB, and HOVA.

Keywords: text classification; multinomial naive Bayes; hidden multinomial naive Bayes; attribute conditional independence assumption; structure extension

Citation: Gan, S.; Shao, S.; Chen, L.; Yu, L.; Jiang, L. Adapting Hidden Naive Bayes for Text Classification. *Mathematics* **2021**, *9*, 2378. https://doi.org/10.3390/math9192378

Academic Editor: Fabio Caraffini

Received: 5 September 2021
Accepted: 22 September 2021
Published: 25 September 2021

1. Introduction

Due to its simplicity, efficiency, and effectiveness, naive Bayes (NB) has been widely used to analyze and solve many scientific and engineering problems, such as text classification [1,2], resistance of buildings [3], identification of areas susceptible to flooding [4], and urban flooding prediction [5]. Text classification is the task of assigning a text document to a pre-specified class, and it has been widely used in many real-world fields, such as spam filtering and short message service (SMS) filtering [2,6]. With the exponential growth of text data in various fields, text classification has attracted more and more attention from researchers in recent years. To address text classification tasks, text documents are generally featured by all of the words that occur in them. Because of the large numbers of documents, large numbers of words, and strong dependencies among these words, accurate and faster text classification presents unique challenges.

Beyond all questions, treating each word as a boolean variable is the simplest approach to applying machine learning for text classification. Based on this idea, multi-variate Bernoulli naive Bayes (BNB) [7] was proposed as the first statistical language model. BNB represents a document using a vector of binary feature variables, which indicates whether or not each word occurs in the document and, thus, ignores the frequency information of each word occurring in the document. To capture the frequency information of each occurring word, multinomial naive Bayes (MNB) [8] was proposed. Ref. [8] proved that MNB achieves, on average, a 27% reduction in the error rate compared to BNB at any vocabulary

size. However, when the number of training documents of one class is much greater than those of the others, MNB tends to select poor weights for the decision boundary. To balance the number of training documents and to address the problem of skewed training data, a complement variant of MNB called complement NB (CNB) was proposed [9]. As a combination of MNB and CNB, OVA [9] was proposed.

Given a test document d, which is generally represented by a word vector $< w_1, w_2, \cdots, w_m >$, MNB, CNB, and OVA classify it with Equations (1)–(3), respectively.

$$c(d) = \arg\max_{c \in C} \left(P(c) \prod_{i=1}^{m} P(w_i|c)^{f_i} \right) \tag{1}$$

$$c(d) = \arg\max_{c \in C} \left(-P(\overline{c}) \prod_{i=1}^{m} P(w_i|\overline{c})^{f_i} \right) \tag{2}$$

$$c(d) = \arg\max_{c \in C} \left(P(c) \prod_{i=1}^{m} P(w_i|c)^{f_i} - P(\overline{c}) \prod_{i=1}^{m} P(w_i|\overline{c})^{f_i} \right) \tag{3}$$

where c is each possible class label, C is the set of all classes, $\overline{c}$ is the complement classes of c, m is the number of different words in the text collection, w_i $(i = 1, 2, \cdots, m)$ is the ith word that occurs in d, and f_i is the frequency count of the word w_i in d. The prior probabilities $P(c)$ and $P(\overline{c})$ are computed in Equations (4) and (5), respectively, and the conditional probabilities $P(w_i|c)$ and $P(w_i|\overline{c})$ are computed in Equations (6) and (7), respectively.

$$P(c) = \frac{\sum_{j=1}^{n} \delta(c_j, c) + 1}{n + s} \tag{4}$$

$$P(\overline{c}) = \frac{\sum_{j=1}^{n} \delta(c_j, \overline{c}) + 1}{n + s} \tag{5}$$

$$P(w_i|c) = \frac{\sum_{j=1}^{n} f_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji} \delta(c_j, c) + m} \tag{6}$$

$$P(w_i|\overline{c}) = \frac{\sum_{j=1}^{n} f_{ji} \delta(c_j, \overline{c}) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji} \delta(c_j, \overline{c}) + m} \tag{7}$$

where s is the number of classes, n is the number of training documents, c_j is the class label of the jth training document, f_{ji} is the frequency count of the ith word in the jth training document, and $\delta(c_j, c)$ and $\delta(c_j, \overline{c})$ are two indicator functions defined by:

$$\delta(c_j, c) = \begin{cases} 1, & if\ c_j = c \\ 0, & otherwise \end{cases} \tag{8}$$

$$\delta(c_j, \overline{c}) = \begin{cases} 1, & if\ c_j \in \overline{c}, namely\ c_j \neq c \\ 0, & otherwise \end{cases} \tag{9}$$

Due to their simplicity, efficiency, and efficacy, MNB and its variants, including CNB and OVA, have been widely used for text classification. However, as in naive Bayes (NB), the assumption of the attributes' (i.e., features) conditional independence that they need is usually violated and, therefore, reduces their classification accuracy. To alleviate their assumption of features' conditional independence, many approaches have been proposed. These approaches can be divided into five categories [10,11]: (1) feature weighting; (2) feature selection; (3) instance weighting; (4) instance selection; (5) structure extension.

Among these approaches, structure extension has attracted far less attention from researchers. To the best of our knowledge, only structure-extended multinomial naive Bayes (SEMNB) [12] has been proposed so far. SEMNB averages all weighted super-parent one-dependence multinomial estimators and, therefore, is an ensemble learning

model. In this paper, we propose a single model called hidden multinomial naive Bayes (HMNB). HMNB creates a hidden parent for each feature, which synthesizes all the other qualified features' influences. To learn HMNB, we proposed a simple but effective learning algorithm without incurring a high-computational-complexity structure-learning process. Our improved idea can also be used to improve CNB and OVA, and the resulting models are simply denoted as HCNB and HOVA, respectively. The extensive experiments on eleven benchmark text classification datasets show that the proposed HMNB, HCNB, and HOVA significantly outperform their state-of-the-art competitors.

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

- We conducted a comprehensive survey on MNB extensions. Based on the survey, existing work can be divided into five categories: feature weighting, feature selection, instance weighting, instance selection, and structure extension.
- We found that structure extension has attracted much less attention from researchers, and only SEMNB was proposed so far. However, it is an ensemble learning model.
- We proposed a single model called hidden MNB (HMNB) by adapting the well-known hidden NB (HNB). HMNB creates a hidden parent for each feature, which synthesizes all of the other qualified features' influences. To learn HMNB, we proposed a simple but effective learning algorithm without incurring a high-computational-complexity structure-learning process. At the same time, we proposed HCNB and HOVA.
- The extensive experiments on eleven benchmark text classification datasets validate the effectiveness of HMNB, HCNB, and HOVA.

The remainder of this paper is organized as follows. Section 2 conducts a compact survey on five categories of existing approaches. Section 3 describes our proposed models in detail. Section 4 presents the experimental setup and results. Section 5 draws conclusions and outlines the main directions.

2. Related Work

2.1. Feature Weighting

The feature weighting approach assigns different weights W_i $(i = 1, 2, \cdots, m)$ to different features (i.e., attributes) in building MNB, CNB, and OVA. To learn W_i $(i = 1, 2, \cdots, m)$, Ref. [13] proposed χ^2 statistic-based feature weighting, which is denoted by $R_{w,c}$. When $R_{w,c}$ is used to improve MNB, CNB, and OVA, the resulting models are simply denoted by $R_{w,c}$MNB, $R_{w,c}$CNB, and $R_{w,c}$OVA, respectively. In addition, Ref. [14] proposed a deep feature weighting approach, simply denoted by DFW, which incorporates the learned weight W_i $(i = 1, 2, \cdots, m)$ into not only the classification of the formula, but also the conditional probability estimates. When DFW is applied to MNB, CNB, and OVA, the resulting models are simply denoted by DFWMNB, DFWCNB and DFWOVA, respectively.

Based on the idea of deep feature weighting, Ref. [15] adapted two other deep feature weighting approaches: gain-ratio-based feature weighting (GRW) and decision-tree-based feature weighting (DTW). GRW sets the weight of each feature to its gain ratio relative to the average gain ratio across all features. When GRW is applied to MNB, CNB, and OVA, the resulting models are denoted by GRWMNB, GRWCNB, and GRWOVA, respectively. DTW sets the weight of each feature to be inversely proportional to the minimum depth at which it is tested in the built tree. When DTW is applied to MNB, CNB, and OVA, the resulting models are denoted by DTWMNB, DTWCNB, and DTWOVA, respectively.

2.2. Feature Selection

The feature selection approach trains MNB, CNB, and OVA on only the selected features instead of all features. In the machine learning community, feature selection is not new. In this paper, we focus our attention on text classification problems. In text classification problems, the dimensionality of features is very high, which is a major characteristic and difficulty. Even a moderate-sized text collection may have many unique words. This is too high for many machine learning algorithms. Therefore, it is indeed

desirable to reduce the dimensionality without harming the classification accuracy. To execute feature selection, many approaches have been proposed. Ref. [16] conducted a comparative survey on five feature selection approaches. In addition, Ref. [17] proposed another feature selection approach based on a two-stage Markov blanket.

Generally, wrapper approaches have superior accuracy compared to filter approaches, but filter approaches always run faster than wrapper approaches. To integrate their advantages, Ref. [18] proposed gain-ratio-based feature selection (GRS). GRS takes advantage of base classifiers to evaluate the selected feature subsets like wrappers, but it does not need to repeatedly search feature subsets and train base classifiers. When GRS is applied to MNB, CNB, and OVA, the resulting models are simply denoted by GRSMNB, GRSCNB, and GRSOVA, respectively.

2.3. Instance Weighting

The instance weighting approach assigns different weights W_j $(j = 1, 2, \cdots, n)$ to different instances (i.e., documents) in building MNB, CNB and OVA. To learn W_j $(j = 1, 2, \cdots, n)$, the simplest way maybe boosting [19]. More specifically, the weights of the training instances misclassified by the base classifiers trained in the last iteration are increased, and then the base classifiers are trained from the re-weighted instances in the next iteration. After predefined rounds, this iteration process is stopped.

Different from boosting [19], Ref. [20] proposed a discriminative instance weighting approach, simply denoted by DW. In each iteration of DW, each different training instance is discriminatively assigned a different weight according to the computed conditional probability loss. This iteration process is repeated for predefined rounds. When DW is applied to MNB, CNB and OVA, the resulting models are simply denoted by DWMNB, DWCNB and DWOVA, respectively.

2.4. Instance Selection

The instance selection approach builds MNB, CNB, and OVA on the selected training instances rather than on all of the training instances. For conducting instance selection processes, the *k*-nearest neighbor algorithm (KNN) is the most well accepted. KNN selects training instances that drop into the neighborhood of a test instance, and it helps to alleviate the assumption of features' conditional independence required by MNB, CNB, and OVA. Therefore, combining KNN with MNB, CNB, and OVA is quite direct. When an instance is required for classification, a local MNB, CNB, or OVA is built on the *k*-nearest neighbors of the test instance, and then it is used to classify the test instance. Based on this improved idea, Ref. [21] proposed locally weighted MNB, CNB, and OVA; the resulting models are simply denoted by LWWMNB, LWCNB, and LWOVA, respectively.

Instead of the *k*-nearest neighbor algorithm, Ref. [22] applied the decision tree learning algorithm to find test instances' nearest neighbors, and then deployed MNB, CNB, or OVA on each leaf node of the built decision trees. The resulting models are simply denoted by MNBTree, CNBTree, and OVATree, respectively. MNBTree, CNBTree, and OVATree build binary trees, in which the split features' values are viewed as zero and nonzero. In addition, to reduce the time consumption, the information gain measure is used to build decision trees. Differently from LWWMNB, LWCNB, and LWOVA, which are lazy learning models, MNBTree, CNBTree, and OVATree are all eager learning models.

2.5. Structure Extension

The structure extension approach uses directed arcs to explicitly represent the dependencies among features. That is to say, we need to find an optimal feature parent set Π_{w_i} for each feature w_i. However, learning an optimal feature parent set Π_{w_i} for each w_i is almost an NP-hard problem [23]. In addition, when the training data are limited, the variance of a complex Bayesian network is high [24], and therefore, its probability estimations are poor. Thus, a multinomial Bayesian network without structure learning that can also represent feature dependencies is desirable.

Inspired by the weighted average of one-dependence estimators (WAODE) [25], Ref. [12] proposed structure-extended multinomial naive Bayes (SEMNB). SEMNB builds a one-dependence multinomial estimator for each present word, i.e., this word is all of the other present words' parent. Then, SEMNB averages all weighted super-parent one-dependence multinomial estimators, and therefore, it is an ensemble learning model. If we apply the structure extension approach to CNB and OVA, we can easily obtain their structure-extended versions. For the sake of convenience, we denote them as SECNB and SEOVA, respectively.

3. The Proposed Models

Structure extension is not new to the Bayesian learning community, and especially not to the semi-naive Bayesian learning community [26,27]. Researchers have proposed many state-of-the-art structure-extended naive Bayes models, such as tree-augmented naive Bayes (TAN) [24] and its variants [28,29]. However, When the structure extension approach comes to high-dimensional text classification data, a key issue that must be addressed is its high-computational-complexity structure learning process. This is the reason for why structure extension has attracted less attention from researchers. To the best of our knowledge, only structure-extended multinomial naive Bayes (SEMNB) [12] has been proposed so far. SEMNB averages all weighted super-parent one-dependence multinomial estimators and, thus, skillfully avoids high-computational-complexity structure-learning processes. The extensive experiments on a large number of text classification datasets validate its effectiveness. However, beyond all questions, SEMNB is an ensemble learning model. Therefore, a simple but effective single model that does not incur a high-computational-complexity structure-learning process is still desirable. This is our paper's main motivation.

To maintain NB's simplicity and efficiency while alleviating its assumption of attributes' conditional independence, hidden naive Bayes (HNB) [30] has achieved remarkable classification performance. Inspired by the success of HNB, in this paper, we expected to adapt it to text classification tasks. We call our adapted model hidden multinomial naive Bayes (HMNB). In HMNB, a hidden parent w_{hpi} is created for each present word w_i, which combines the influences from all of the other present qualified words w_t $(t = 1, 2, \cdots, m \wedge t \neq i)$. Now, given a test document d, HMNB classifies it by using Equation (10).

$$c(d) = \arg\max_{c \in C} \left(P(c) \prod_{i=1 \wedge f_i > 0}^{m} P(w_i | w_{hpi}, c)^{f_i} \right) \tag{10}$$

where $P(w_i | w_{hpi}, c)$ is computed by:

$$P(w_i | w_{hpi}, c) = \frac{\sum_{t=1 \wedge t \neq i \wedge f_t > 0 \wedge W_t \geq aveGR}^{m} W_t P(w_i | w_t, c)}{\sum_{t=1 \wedge t \neq i \wedge f_t > 0 \wedge W_t \geq aveGR}^{m} W_t} \tag{11}$$

where W_t $(t = 1, 2, \cdots, m \wedge t \neq i)$ indicates the importance of each possible parent word w_t in the hidden parent w_{hpi}. Therefore, for simplicity, we define it as the gain ratio $GainRatio(w_t)$ of the word w_t that splits the training data D. However, at the same time, we only select the word w_t whose gain ratio is above the average $aveGR$ of all words as the potential parent. The detailed calculation formulas are:

$$W_t = GainRatio(w_t) = \frac{\sum\limits_{f_t \in \{0, \bar{0}\}, c} P(f_t, c) \log \frac{P(f_t, c)}{P(f_t) P(c)}}{- \sum\limits_{f_i \in \{0, \bar{0}\}} P(f_t) \log P(f_t)} \tag{12}$$

$$aveGR = \frac{1}{m} \sum_{t=1}^{m} GainRatio(w_t) \tag{13}$$

where $f_t \in \{0, \overline{0}\}$. $f_t = 0$ indicates the absence of w_t, and $f_t = \overline{0}$ indicates the presence of w_t.

Now, the only thing left is the efficient calculation of $P(w_i|w_t, c)$; the conditional probability w_i appears given w_t and c. It is well known that the space complexity of estimating $P(w_i|w_t, c)$ directly from D is $O(sm^2)$. To our knowledge, for text classification tasks, m (the vocabulary size in the text collection) is often too large to save the tables of each joint pair of words and class frequencies from which the conditional probability $P(w_i|w_t, c)$ is estimated. At the same time, text data are usually in the form of a sparse matrix, and therefore, the number of different words present in a given document d—simply denoted by $|d|$—is much smaller than m. Therefore, as in SEMNB [12], we also transform a part of the training space consumption into classification time consumption. In more detail, we remove the step of computing $P(w_i|w_t, c)$ from the training stage to the classification stage. At the classification stage, when a test document d is predicted, $P(w_i|w_t, c)$ is computed according to D and d. More specifically, given a word w_t in d, we only select the documents in which w_t occurs to compute $P(w_i|w_t, c)$ by using Equation (14), which has the space complexity of $O(s|d|)$ only.

$$P(w_i|w_t, c) = \frac{\sum_{j=1 \wedge f_{jt}>0}^{n} f_{ji}\delta(c_j, c) + 1}{\sum_{i=1}^{m}\sum_{j=1 \wedge f_{jt}>0}^{n} f_{ji}\delta(c_j, c) + m} \tag{14}$$

In summary, the whole algorithm for learning HMNB is partitioned into a training algorithm (HMNB-Training) and a classification algorithm (HMNB-Classification). They are described by Algorithms 1 and 2, respectively. Algorithm 1 takes the time complexity of $O(nm + sm)$, and Algorithm 2 takes the time complexity of $O(n|d|^2 + s|d|^2 + s|d|)$, where n is the number of training documents, m is the number of different words in the text collection, s is the number of classes, and $|d|$ is the number of different words present in a given document d.

Algorithm 1: HMNB-Training (D).

Input: D—training data
Output: $P(c)$ and W_t ($t = 1, 2, \cdots, m$)
1: **for** each class c **do**
2: Use Equation (4) to compute $P(c)$ from D;
3: **end for**
4: **for** For each word w_t ($t = 1, 2, \cdots, m$) from D **do**
5: Compute W_t using Equation (12);
6: **end for**
7: Compute the averaged gain ratio $aveGR$ of all words using Equation (13);
8: **if** $GainRatio(w_t) \geq aveGR$ **then**
9: $W_t = GainRatio(w_t)$
10: **else**
11: $W_t = 0$
12: **end if**
13: Return $P(c)$ and W_t ($t = 1, 2, \cdots, m$)

Algorithm 2: HMNB-Classification (d, D, $P(c)$, W_t).

Input: d—a test document, D—training data, and the computed $P(c)$ and W_t
Output: $c(d)$
1: **for** For each word w_i ($i = 1, 2, \cdots, |d|$) in d **do**
2: **for** For each word w_t ($t = 1, 2, \cdots, |d| \wedge t \neq i$) in d **do**
3: Denote all training documents in which w_t occurs as D_{w_t};
4: **for** each class c **do**
5: Compute $P(w_i|w_t, c)$ from D_{w_t} using Equation (14);
6: **end for**
7: **end for**
8: **end for**
9: Use W_t and $P(w_i|w_t, c)$ to compute $P(w_i|w_{hpi}, c)$ with Equation (11);
10: Use $P(c)$ and $P(w_i|w_{hpi}, c)$ to predict the class label of d with Equation (10);
11: Return the predicted class label $c(d)$

Our improved idea can also be used to improve CNB and OVA. The resulting models are denoted as HCNB and HOVA, respectively. Given a test document d, HCNB and HOVA use Equations (15) and (16) to classify it.

$$c(d) = \arg\max_{c \in C} \left(-P(\overline{c}) \prod_{i=1 \wedge f_i > 0}^{m} P(w_i | w_{hpi}, \overline{c})^{f_i} \right) \tag{15}$$

$$c(d) = \arg\max_{c \in C} \left(P(c) \prod_{i=1 \wedge f_i > 0}^{m} P(w_i | w_{hpi}, c)^{f_i} - P(\overline{c}) \prod_{i=1 \wedge f_i > 0}^{m} P(w_i | w_{hpi}, \overline{c})^{f_i} \right) \tag{16}$$

where $P(w_i | w_{hpi}, \overline{c})$ is computed by:

$$P(w_i | w_{hpi}, \overline{c}) = \frac{\sum_{t=1 \wedge t \neq i \wedge f_t > 0 \wedge W_t \geq aveGR}^{m} W_t P(w_i | w_t, \overline{c})}{\sum_{t=1 \wedge t \neq i \wedge f_t > 0 \wedge W_t \geq aveGR}^{m} W_t} \tag{17}$$

where $P(w_i | w_t, \overline{c})$ is computed by:

$$P(w_i | w_t, \overline{c}) = \frac{\sum_{j=1 \wedge f_{jt} > 0}^{n} f_{ji} \delta(c_j, \overline{c}) + 1}{\sum_{i=1}^{m} \sum_{j=1 \wedge f_{jt} > 0}^{n} f_{ji} \delta(c_j, \overline{c}) + m} \tag{18}$$

Similarly to HMNB, the algorithms for learning HCNB and HOVA are also partitioned into training algorithms (HCNB-Training and HOVA-Training) and classification algorithms (HCNB-Classification and HOVA-Classification). They are described by Algorithms 3–6, respectively. From Algorithms 3–6, we can see that the time complexities of HCNB and HOVA are almost the same as that of HMNB.

Algorithm 3: HCNB-Training (D).

Input: D—training data
Output: $P(\overline{c})$ and W_t ($t = 1, 2, \cdots, m$)
for each class c **do**
  Use Equation (5) to compute $P(\overline{c})$ from D;
end for
for For each word w_t ($t = 1, 2, \cdots, m$) from D **do**
  Compute W_t using Equation (12);
end for
Compute the averaged gain ratio $aveGR$ of all words using Equation (13);
if $GainRatio(w_t) \geq aveGR$ **then**
  $W_t = GainRatio(w_t)$
else
  $W_t = 0$
end if
Return $P(\overline{c})$ and W_t ($t = 1, 2, \cdots, m$)

Algorithm 4: HCNB-Classification (d, D, $P(\overline{c})$, W_t).

Input: d—a test document, D—training data, and the computed $P(\overline{c})$ and W_t
Output: $c(d)$
for For each word w_i ($i = 1, 2, \cdots, |d|$) in d **do**
  for For each word w_t ($t = 1, 2, \cdots, |d| \wedge t \neq i$) in d **do**
    Denote all training documents in which w_t occurs as D_{w_t};
    for each class c **do**
      Compute $P(w_i | w_t, \overline{c})$ from D_{w_t} using Equation (18);
    end for
  end for
end for
Use W_t and $P(w_i | w_t, \overline{c})$ to compute $P(w_i | w_{hpi}, \overline{c})$ with Equation (17);
Use $P(\overline{c})$ and $P(w_i | w_{hpi}, \overline{c})$ to predict the class label of d with Equation (15);
Return the predicted class label $c(d)$

Algorithm 5: HOVA-Training (D).

Input: D—training data
Output: $P(c)$, $P(\overline{c})$, and W_t ($t = 1, 2, \cdots, m$)
for each class c **do**
  Use Equation (4) to compute $P(c)$ from D;
  Use Equation (5) to compute $P(\overline{c})$ from D;
end for
for For each word w_t ($t = 1, 2, \cdots, m$) from D **do**
  Compute W_t using Equation (12);
end for
Compute the averaged gain ratio $aveGR$ of all words using Equation (13);
if $GainRatio(w_t) \geq aveGR$ **then**
  $W_t = GainRatio(w_t)$
else
  $W_t = 0$
end if
Return $P(c)$, $P(\overline{c})$, and W_t ($t = 1, 2, \cdots, m$)

Algorithm 6: HOVA-Classification (d, D, $P(c)$, $P(\overline{c})$, W_t).

Input: d—a test document, D—training data, and the computed $P(c)$, $P(\overline{c})$, and W_t
Output: $c(d)$
for For each word w_i ($i = 1, 2, \cdots, |d|$) in d **do**
  for For each word w_t ($t = 1, 2, \cdots, |d| \wedge t \neq i$) in d **do**
    Denote all training documents in which w_t occurs as D_{w_t};
    for each class c **do**
      Compute $P(w_i|w_t, c)$ from D_{w_t} using Equation (14);
      Compute $P(w_i|w_t, \overline{c})$ from D_{w_t} using Equation (18);
    end for
  end for
end for
Use W_t and $P(w_i|w_t, c)$ to compute $P(w_i|w_{hpi}, c)$ with Equation (11);
Use W_t and $P(w_i|w_t, \overline{c})$ to compute $P(w_i|w_{hpi}, \overline{c})$ with Equation (17);
Use $P(c)$, $P(\overline{c})$, $P(w_i|w_{hpi}, c)$, and $P(w_i|w_{hpi}, \overline{c})$ to predict the class label of d with Equation (16);
Return the predicted class label $c(d)$

4. Experiments and Results

To validate the effectiveness of the proposed HMNB, HCNB, and HOVA, we designed and completed three groups of experiments. The first group of experiments compared HMNB with MNB, $R_{w,c}$MNB, GRSMNB, DWMNB, MNBTree, and SEMNB. The second group of experiments compared HCNB with CNB, $R_{w,c}$CNB, GRSCNB, DWCNB, CNBTree, and SECNB. The third group of experiments compared HOVA with OVA, $R_{w,c}$OVA, GRSOVA, DWOVA, OVATree, and SEOVA. We used the existing implementations of MNB and CNB in the platform of the Waikato environment for knowledge analysis (WEKA) [31] and implemented all of the other models by using the WEKA platform [31].

We conducted our three groups of experiments on eleven well-known text classification tasks published on the homepage of the WEKA platform [31], which cover a wide range of text classification characteristics. Table 1 lists the detailed data information of these eleven datasets. All of these eleven datasets were obtained from OHSUMED-233445, Reuters-21578, TREC, and the WebACE project. Ref. [32] originally converted them into term counts.

Tables 2–4 show the results of a comparison of the accuracy of each model on each dataset after averaging the classification accuracies from ten runs of 10-fold cross-validation, respectively. Then, we use two-tailed t-tests at 95% significance level [33] to compare the proposed HMNB, HCNB and HOVA to each of their competitors. In these tables, the symbols • and ○ denote statistically significant improvement or degradation with respect to the competitors, respectively. The averaged classification accuracies and the $Win/Tie/Lose$ ($W/T/L$) values are summarized at the bottom of the tables. The averaged classification accuracy of each model across all datasets provides a gross indicator of the relative classification performance in addition to the other statistics. Each $W/T/L$ value in

these tables indicates that, compared to their competitors, HMNB, HCNB, and HOVA won on W datasets, tied on T datasets, and lost on L datasets.

Table 1. Text classification datasets in our experiments.

Dataset	#Documents	#Words	#Classes	#Min Class	#Max Class	#Avg Class
fbis	2463	2000	17	38	506	144.9
la1s	3204	31,472	6	273	943	534.0
la2s	3075	31,472	6	248	905	512.5
oh0	1003	3182	10	51	194	100.3
oh10	1050	3238	10	52	165	105.0
oh15	913	3100	10	53	157	91.3
oh5	918	3012	10	59	149	91.8
ohscal	11,162	11,465	10	709	1621	1116.2
re0	1504	2886	13	11	608	115.7
re1	1657	3758	25	10	371	66.3
wap	1560	8460	20	5	341	78.0

Table 2. Comparisons of the classification accuracy for HMNB versus MNB, $R_{w,c}$MNB, GRSMNB, DWMNB, MNBTree, and SEMNB.

Dataset	HMNB	MNB	$R_{w,c}$MNB	GRSMNB	DWMNB	MNBTree	SEMNB
fbis	81.42	77.11 •	79.87 •	79.61 •	80.39	79.06 •	83.27 ○
la1s	89.20	88.41	87.88 •	88.40 •	88.85	87.22 •	89.15
la2s	90.73	89.88 •	88.72 •	89.33 •	90.14	87.34 •	91.01
oh0	91.70	89.55 •	89.05 •	90.18	89.64 •	88.93 •	88.87 •
oh10	83.87	80.60 •	80.41 •	81.10 •	80.64 •	83.25	80.66 •
oh15	86.51	83.60 •	83.61 •	84.38	83.29 •	79.01 •	83.36 •
oh5	90.00	86.63 •	86.46 •	89.72	86.87 •	88.74	87.55 •
ohscal	79.88	74.70 •	74.18 •	76.84 •	74.30 •	78.00 •	76.40 •
re0	83.29	80.02 •	77.07 •	80.56 •	81.81	77.30 •	82.73
re1	84.60	83.31	82.72 •	86.12 ○	83.13	84.26	82.22 •
wap	80.40	81.22	76.33 •	80.34	81.83 ○	75.42 •	80.53
Average	85.60	83.18	82.39	84.23	83.72	82.59	84.16
$W/T/L$	-	8/3/0	11/0/0	6/4/1	5/5/1	8/3/0	6/4/1

Table 3. Comparisons of the classification accuracy for HCNB vs. CNB, $R_{w,c}$CNB, GRSCNB, DWCNB, CNBTree, and SECNB.

Dataset	HCNB	CNB	$R_{w,c}$CNB	GRSCNB	DWCNB	CNBTree	SECNB
fbis	82.24	76.78 •	78.27 •	76.91 •	83.74 ○	79.32 •	81.42
la1s	88.12	86.30 •	87.33 •	85.99 •	88.48	87.21	87.82
la2s	89.86	88.26 •	88.94 •	87.69 •	89.61	88.08 •	89.47
oh0	92.73	92.31	92.49	91.41	92.36	90.76	89.82 •
oh10	84.88	81.76 •	82.20 •	80.13 •	82.36 •	85.16	81.24 •
oh15	88.19	84.38 •	85.32 •	85.36 •	84.27 •	81.74 •	83.81 •
oh5	91.34	90.58	90.96	89.96	90.51	89.99	88.18 •
ohscal	79.85	76.50 •	76.69 •	75.34 •	76.39 •	76.94 •	76.61 •
re0	84.71	82.37 •	80.74 •	81.48 •	85.35	79.62 •	83.79
re1	86.18	84.99	86.16	86.38	86.88	86.43	84.76 •
wap	79.74	77.53 •	78.10 •	76.31 •	79.32	76.69 •	80.13
Average	86.17	83.80	84.29	83.36	85.39	83.81	84.28
$W/T/L$	-	8/3/0	8/3/0	8/3/0	3/7/1	6/5/0	6/5/0

Table 4. Comparisons of the classification accuracy for HOVA versus OVA, $R_{w,c}$OVA, GRSOVA, DWOVA, OVATree, and SEOVA.

Dataset	HOVA	OVA	$R_{w,c}$OVA	GRSOVA	DWOVA	OVATree	SEOVA
fbis	82.21	80.94 •	80.80 •	80.95 •	82.68	81.72	80.80 •
la1s	88.91	88.52	88.11	88.36	88.83	87.69 •	86.94 •
la2s	90.22	90.23	89.32	89.90	90.36	87.94 •	88.56 •
oh0	91.70	91.49	90.12	91.09	91.53	90.05	91.51
oh10	84.25	81.86 •	81.51 •	81.39 •	81.94 •	84.20	84.04
oh15	86.86	84.39 •	84.50 •	85.51	84.07 •	80.35 •	85.95
oh5	89.65	89.44	88.31	90.16	89.75	89.46	90.03
ohscal	78.29	75.81 •	75.15 •	76.91 •	75.45 •	78.27	77.02 •
re0	83.08	81.54	78.81 •	81.18 •	83.41	78.11 •	81.35 •
re1	85.70	84.77	85.37	86.51	84.97	85.21	84.46 •
wap	78.93	80.65 ○	77.21 •	79.72	81.64 ○	75.90 •	74.71 •
Average	85.44	84.51	83.56	84.70	84.97	83.54	84.12
$W/T/L$	-	4/6/1	6/5/0	4/7/0	3/7/1	5/6/0	7/4/0

Based on the accuracy comparisons presented inTables 2–4, we then used the KEEL software [34] to complete the Wilcoxon signed-rank test [35,36] in order to thoroughly compare each pair of models. The Wilcoxon signed-rank test ranks the differences in the performance of two classification models for each dataset, ignoring the signs, and compares the ranks for the positive R^+ and the negative R^- differences [35,36]. According to the table of the exact critical values for the Wilcoxon test, for a confidence level of $\alpha = 0.05$ and $N = 11$ datasets, we speak of two classification models as being "significantly different" if the smaller of R^+ and R^- is equal to or less than 11, and thus, we reject the null hypothesis. Tables 5–7 summarize the related comparison results. In these tables, ○ denotes that the model in the column improves the model in the corresponding row, and • denotes that the model in the row improves the model in the corresponding column. In the lower diagonal, the significance level is $\alpha = 0.05$. In the upper diagonal, the significance level is $\alpha = 0.1$. From all of the above comparison results, we can draw the following highlights:

1. The average accuracy of HMNB on eleven datasets is 85.60%, which is notably higher than those of MNB (83.18%), $R_{w,c}$MNB (82.39%), GRSMNB (84.23%), DWMNB (83.72%), MNBTree (82.59%), and SEMNB (84.16%). HMNB substantially outperforms MNB (eight wins and zero losses), $R_{w,c}$MNB (11 wins and zero losses), GRSMNB (six wins and one loss), DWMNB (five wins and one loss), MNBTree (eight wins and zero losses), and SEMNB (six wins and one loss).
2. The average accuracy of HCNB on eleven datasets is 86.17%, which is notably higher than those of CNB (83.8%), $R_{w,c}$CNB (84.29%), GRSCNB (83.36%), DWCNB (85.39%), CNBTree (83.81%), and SECNB (84.28%). HCNB substantially outperforms CNB (eight wins and zero losses), $R_{w,c}$CNB (eight wins and zero losses), GRSCNB (eight wins and zero losses), DWCNB (three wins and one loss), CNBTree (six wins and zero losses), and SECNB (six wins and zero losses).
3. The average accuracy of HOVA on eleven datasets is 85.44%, which is notably higher than those of OVA (84.51%), $R_{w,c}$OVA (83.56%), GRSOVA (84.7%), DWOVA (84.97%), OVATree (83.54%), and SEOVA (84.12%). HOVA substantially outperformsOVA (four wins and one loss), $R_{w,c}$OVA (six wins and zero losses), GRSOVA (four wins and zero losses), DWOVA (three wins and one loss), OVATree (five wins and zero losses), and SEOVA (seven wins and zero losses).
4. In addition, according to the results of the Wilcoxon test, HMNB significantly outperforms MNB, $R_{w,c}$MNB, GRSMNB, DWMNB, MNBTree, and SEMNB. HCNB significantly outperforms CNB, $R_{w,c}$CNB, GRSCNB, CNBTree, and SECNB. HOVA significantly outperforms OVA, $R_{w,c}$OVA, OVATree, and SEOVA. All of these comparison results validate the effectiveness of the proposed HMNB, HCNB, and HOVA.

Table 5. Results of the Wilcoxon test with regard to HMNB.

Algorithm	HMNB	MNB	$R_{w,c}$MNB	GRSMNB	DWMNB	MNBTree	SEMNB
HMNB	-	•	•	•	•	•	•
MNB	○	-	•	○			
$R_{w,c}$MNB	○	○	-	○	○		○
GRSMNB	○		•	-		•	
DWMNB	○		•		-		
MNBTree	○					-	
SEMNB	○		•				-

Table 6. Results of the Wilcoxon test with regard to HCNB.

Algorithm	HCNB	CNB	$R_{w,c}$CNB	GRSCNB	DWCNB	MCBTree	SECNB
HCNB	-	•	•	•		•	•
CNB	○	-	○		○		
$R_{w,c}$CNB	○		-	•			
GRSCNB	○		○	-	○		
DWCNB		•		•	-	•	•
CNBTree	○					-	
SECNB	○				○		-

Table 7. Results of the Wilcoxon test with regard to HOVA.

Algorithm	HOVA	OVA	$R_{w,c}$OVA	GRSOVA	DWOVA	OVATree	SEOVA
HOVA	-	•	•	•		•	•
OVA	○	-	•				
$R_{w,c}$OVA	○	○	-	○	○		
GRSOVA			•	-			
DWOVA			•		-		
OVATree	○					-	
SEOVA	○						-

Finally, we conducted the Wilcoxon signed-rank test [35,36] to compare each pair of HMNB, HCNB, and HOVA. The detailed comparison results are shown in Table 8. From these, we can see that HMNB almost tied with HCNB and HOVA, and HCNB was notably better than HOVA. Considering the simplicity of the models, HMNB and HCNB could be appropriate choices.

Table 8. Results of the Wilcoxon test for HMNB, HCNB, and HOVA.

Algorithm	HMNB	HCNB	HOVA
HMNB	-		
HCNB		-	•
HOVA		○	-

5. Conclusions and Future Study

To alleviate MNB's assumption of features' conditional independence, this paper proposed a single model called hidden MNB (HMNB) by adapting the well-known hidden NB (HNB). HMNB creates a hidden parent for each feature that synthesizes all of the other qualified features' influences. For HMNB to learn, we proposed a simple but effective learning algorithm that does not incurring a high-computational-complexity structure-learning process. Our improved idea can also be used to improve CNB and OVA, and the resulting models are simply denoted as HCNB and HOVA, respectively. The extensive experiments show that the proposed HMNB, HCNB, and HOVA significantly outperform their state-of-the-art competitors.

In the proposed HMNB, HCNB, and HOVA, how the weight (importance) of each possible parent word is defined is crucial. Currently, we directly use the gain ratio of each possible parent word that splits the training data in order to define the weight, which is somewhat rough. We believe that using more sophisticated methods, such as the expectation-maximum (EM) algorithm, could further improve their classification performance and make their superiority stronger. This is a main topic for future study. In addition, to reduce the training space complexity, we transform a part of the training space consumption into classification time consumption, which leads to a relatively high classification time complexity. Therefore, the improvement of the efficiency of the proposed models is another interesting topic for future study.

Author Contributions: Conceptualization, S.G. and L.J.; methodology, S.G., S.S. and L.J.; software, L.C. and L.Y.; validation, L.C., L.Y. and S.G.; formal analysis, S.G. and L.J.; investigation, S.G. and L.J.; resources, S.G. and L.J.; data curation, S.S.; writing—original draft preparation, S.G. and L.J.; writing—review and editing, S.G. and L.J.; visualization, S.G.; supervision, L.J.; project administration, S.G. and L.Y.; funding acquisition, S.G. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by KLIGIP-2018A05, 2019AEE020, X201900, Q20203003, and 20RC07.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in our paper:

NB	Naive Bayes
BNB	Bernoulli NB
MNB	Multinomial NB
CNB	Complement NB
OVA	One-versus-all-but-one model
HMNB	Hidden MNB
HCNB	Hidden CNB
HOVA	Hidden OVA
SMS	Short message service
$R_{w,c}$	χ^2 statistic-based feature weighting
$R_{w,c}$MNB	MNB with $R_{w,c}$
$R_{w,c}$CNB	CNB with $R_{w,c}$
$R_{w,c}$OVA	OVA with $R_{w,c}$
DFW	Deep feature weighting
DFWMNB	MNB with DFW
DFWCNB	CNB with DFW
DFWOVA	OVA with DFW
GRW	Gain-ratio-based feature weighting
GRWMNB	MNB with GRW
GRWCNB	CNB with GRW
GRWOVA	OVA with GRW
DTW	Decision-tree-based feature weighting
DTWMNB	MNB with DTW
DTWCNB	CNB with DTW
DTWOVA	OVA with DTW

GRS	Gain ratio-based hybrid feature selection
GRSMNB	MNB with GRS
GRSCNB	CNB with GRS
GRSOVA	OVA with GRS
DW	Discriminative instance weighting
DWMNB	MNB with DW
DWCNB	CNB with DW
DWOVA	OVA with DW
LWWMNB	Locally weighted MNB
LWWCNB	Locally weighted CNB
LWWOVA	Locally weighted OVA
MNBTree	MNB tree
CNBTree	CNB tree
OVATree	OVA tree
SEMNB	Structure-extended MNB
SECNB	Structure-extended CNB
SEOVA	Structure-extended OVA
TAN	Tree-augmented NB
WAODE	Weighted average of one-dependence estimators
HNB	Hidden NB
WEKA	Waikato environment for knowledge analysis
KEEL	Knowledge extraction based on evolutionary learning

References

1. Chen, L.; Jiang, L.; Li, C. Modified DFS-based term weighting scheme for text classification. *Expert Syst. Appl.* **2021**, *168*, 114438. [CrossRef]
2. Chen, L.; Jiang, L.; Li, C. Using modified term frequency to improve term weighting for text classification. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104215. [CrossRef]
3. Rusek, J. The Point Nuisance Method as a Decision-Support System Based on Bayesian Inference Approach. *Arch. Min. Sci.* **2020**, *65*, 117–127.
4. Ali, S.A.; Parvin, F.; Pham, Q.B.; Vojtek, M.; Vojtekova, J.; Costache, R.; Linh, N.T.T.; Nguyen, H.Q.; Ahmad, A.; Ghorbani, M.A. GIS-based comparative assessment of flood susceptibility mapping using hybrid multi-criteria decision-making approach, naive Bayes tree, bivariate statistics and logistic regression: A case of Topla basin, Slovakia. *Ecol. Indic.* **2020**, *117*, 106620. [CrossRef]
5. Wang, H.; Wang, H.; Wu, Z.; Zhou, Y. Using Multi-Factor Analysis to Predict Urban Flood Depth Based on Naive Bayes. *Water* **2021**, *13*, 432. [CrossRef]
6. Deng, X.; Wang, Y.; Zhu, T.; Zhang, W.; Yin, Y.; Ye, L. Short Message Service (SMS) can Enhance Compliance and Reduce Cancellations in a Sedation Gastrointestinal Endoscopy Center: A Prospective Randomized Controlled Trial. *J. Med. Syst.* **2015**, *39*, 169. [CrossRef]
7. Ponte, J.M.; Croft, W.B. A language modeling approach to information retrieval. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998; pp. 275–281.
8. McCallum, A.; Nigam, K. A comparison of event models for naive Bayes text classification. In *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization*; AAAI Press: Palo Alto, CA, USA, 1998; pp. 41–48.
9. Rennie, J.D.; Shih, L.; Teevan, J.; Karger, D.R. Tackling the poor assumptions of Naive Bayes Text Classifiers. In Proceedings of the Twentieth International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003; pp. 616–623.
10. Zhang, H.; Jiang, L.; Yu, L. Class-specific attribute value weighting for Naive Bayes. *Inf. Sci.* **2020**, *508*, 260–274. [CrossRef]
11. Zhang, H.; Jiang, L.; Yu, L. Attribute and instance weighted naive Bayes. *Pattern Recognit.* **2021**, *111*, 107674. [CrossRef]
12. Jiang, L.; Wang, S.; Li, C.; Zhang, L. Structure extended multinomial naive Bayes. *Inf. Sci.* **2016**, *329*, 346–356. [CrossRef]
13. Li, Y.; Luo, C.; Chung, S.M. Weighted naive Bayes for Text Classification Using positive Term-Class Dependency. *Int. J. Artif. Intell. Tools* **2012**, *21*, 1250008. [CrossRef]
14. Jiang, L.; Li, C.; Wang, S.; Zhang, L. Deep feature weighting for naive Bayes and its application to text classification. *Eng. Appl. Artif. Intell.* **2016**, *52*, 26–39. [CrossRef]
15. Zhang, L.; Jiang, L.; Li, C.; Kong, G. Two feature weighting approaches for naive Bayes text classifiers. *Knowl. Based Syst.* **2016**, *100*, 137–144. [CrossRef]
16. Yang, Y.; Pedersen, J.O. A comparative study on feature selection in text categorization. In Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, USA, 8–12 July 1997; pp. 412–420.
17. Javed, K.; Maruf, S.; Babri, H.A. A two-stage Markov blanket based feature selection algorithm for text classification. *Neurocomputing* **2015**, *157*, 91–104. [CrossRef]

18. Zhang, L.; Jiang, L.; Li, C. A New Feature Selection Approach to Naive Bayes Text Classifiers. *Int. J. Pattern Recognit. Artif. Intell.* **2016**, *30*, 1650003:1–1650003:17. [CrossRef]
19. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
20. Jiang, L.; Wang, D.; Cai, Z. Discriminatively Weighted Naive Bayes and its Application in Text Classification. *Int. J. Artif. Intell. Tools* **2012**, *21*, 1250007. [CrossRef]
21. Jiang, L.; Cai, Z.; Zhang, H.; Wang, D. Naive Bayes text classifiers: A locally weighted learning approach. *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 273–286. [CrossRef]
22. Wang, S.; Jiang, L.; Li, C. Adapting naive Bayes tree for text classification. *Knowl. Inf. Syst.* **2015**, *44*, 77–89. [CrossRef]
23. Chickering, D.M. Learning Bayesian networks is NP-complete. In *Learning from Data*; Springer: New York, NY, USA, 1996; pp. 121–130.
24. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian network classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [CrossRef]
25. Jiang, L.; Zhang, H.; Cai, Z.; Wang, D. Weighted Average of One-Dependence Estimators. *J. Exp. Theor. Artif. Intell.* **2012**, *24*, 219–230. [CrossRef]
26. Sahami, M. Learning limited dependence Bayesian classifiers. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 335–338.
27. Abellán, J.; Cano, A.; Masegosa, A.R.; Moral, S. A memory efficient semi-Naive Bayes classifier with grouping of cases. *Intell. Data Anal.* **2011**, *15*, 299–318. [CrossRef]
28. Keogh, E.; Pazzani, M. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 3–6 January 1999; pp. 225–230.
29. Qiu, C.; Jiang, L.; Li, C. Not always simple classification: Learning SuperParent for Class Probability Estimation. *Expert Syst. Appl.* **2015**, *42*, 5433–5440. [CrossRef]
30. Jiang, L.; Zhang, H.; Cai, Z. A Novel Bayes Model: Hidden Naive Bayes. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1361–1371. [CrossRef]
31. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2011.
32. Han, E.; Karypis, G. Centroid-Based Document Classification: Analysis and Experimental Results. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2000, Lyon, France, 13–16 September 2000; pp. 424–431.
33. Nadeau, C.; Bengio, Y. Inference for the generalization error. *Mach. Learn.* **2003**, *52*, 239–281. [CrossRef]
34. Alcalá-Fdez, J.; Fernandez, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; Herrera, F. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *J. Mult.-Valued Log. Soft Comput.* **2011**, *17*, 255–287.
35. Garcia, S.; Herrera, F. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
36. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [CrossRef]