


Article

A Novel Hybrid Approach: Instance Weighted Hidden Naive Bayes

Liangjun Yu ^{1,2} , Shengfeng Gan ^{1,*}, Yu Chen ^{1,2} and Dechun Luo ^{3,4}

¹ College of Computer, Hubei University of Education, Wuhan 430205, China; yuliangjun@hue.edu.cn (L.Y.); chenyu@hue.edu.cn (Y.C.)

² Hubei Co-Innovation Center of Basic Education Information Technology Services, Hubei University of Education, Wuhan 430205, China

³ School of Management, Huazhong University of Science and Technology, Wuhan 430071, China; m202077097@hust.edu.cn

⁴ Wuhan Eight Dimension Space Information Technology Co., Ltd., Wuhan 430071, China

* Correspondence: sf_gan@hue.edu.cn

Abstract: Naive Bayes (NB) is easy to construct but surprisingly effective, and it is one of the top ten classification algorithms in data mining. The conditional independence assumption of NB ignores the dependency between attributes, so its probability estimates are often suboptimal. Hidden naive Bayes (HNB) adds a hidden parent to each attribute, which can reflect dependencies from all the other attributes. Compared with other Bayesian network algorithms, it offers significant improvements in classification performance and avoids structure learning. However, the assumption that HNB regards each instance equivalent in terms of probability estimation is not always true in real-world applications. In order to reflect different influences of different instances in HNB, the HNB model is modified into the improved HNB model. The novel hybrid approach called instance weighted hidden naive Bayes (IWHNB) is proposed in this paper. IWHNB combines instance weighting with the improved HNB model into one uniform framework. Instance weights are incorporated into the improved HNB model to calculate probability estimates in IWHNB. Extensive experimental results show that IWHNB obtains significant improvements in classification performance compared with NB, HNB and other state-of-the-art competitors. Meanwhile, IWHNB maintains the low time complexity that characterizes HNB.



Citation: Yu, L.; Gan, S.; Chen, Y.; Luo, D. A Novel Hybrid Approach: Instance Weighted Hidden Naive Bayes. *Mathematics* **2021**, *9*, 2982. <https://doi.org/10.3390/math9222982>

Academic Editor: María Purificación Galindo Villardón

Received: 13 October 2021

Accepted: 19 November 2021

Published: 22 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Bayesian network; hidden naive Bayes; instance weighting

1. Introduction

Bayesian network (BN) combines knowledge of network topology and probability. It is a classical method which can be used to predict a test instance [1]. The BN structure is a directed acyclic graph, and each edge in BN reflects the dependency between attributes. Unfortunately, it has been confirmed that finding the optimal BN from arbitrary BNs is an non-deterministic polynomial (NP)-hard problem [2,3]. Naive Bayes (NB) is one of the most classic and efficient models in BNs. It is easy to construct but surprisingly effective [4]. The NB model is shown in the Figure 1a. A_1, A_2, \dots, A_m denote m attributes. The class variable C is the parent node of each attribute. Each attribute A_i is independent from the others.

The classification performance of NB is comparable to well-known classifiers [5,6]. However, the conditional independence assumption of NB ignores the dependencies between attributes in real-world applications, so its probability estimates are often sub-optimal [7,8]. In order to reduce the primary weakness brought by the conditional independence assumption, a lot of improved approaches of NB have been proposed to alleviate the primary weakness in NB by manipulating attribute independence assertions [9,10]. These improved approaches can fall into five main categories: (1) Structure

extension by extending the NB's structure to overcome the attribute independence assertions [11–14]; (2) Instance weighting by constructing a NB classifier on an instance weighted dataset [15–18]; (3) Instance selection by constructing a NB classifier on a selected local instance subset [19–21]; (4) Attribute weighting by constructing a NB classifier on an attribute weighted dataset [22–26]; (5) Attribute selection by constructing a NB classifier on a selected attribute subset [27–30].

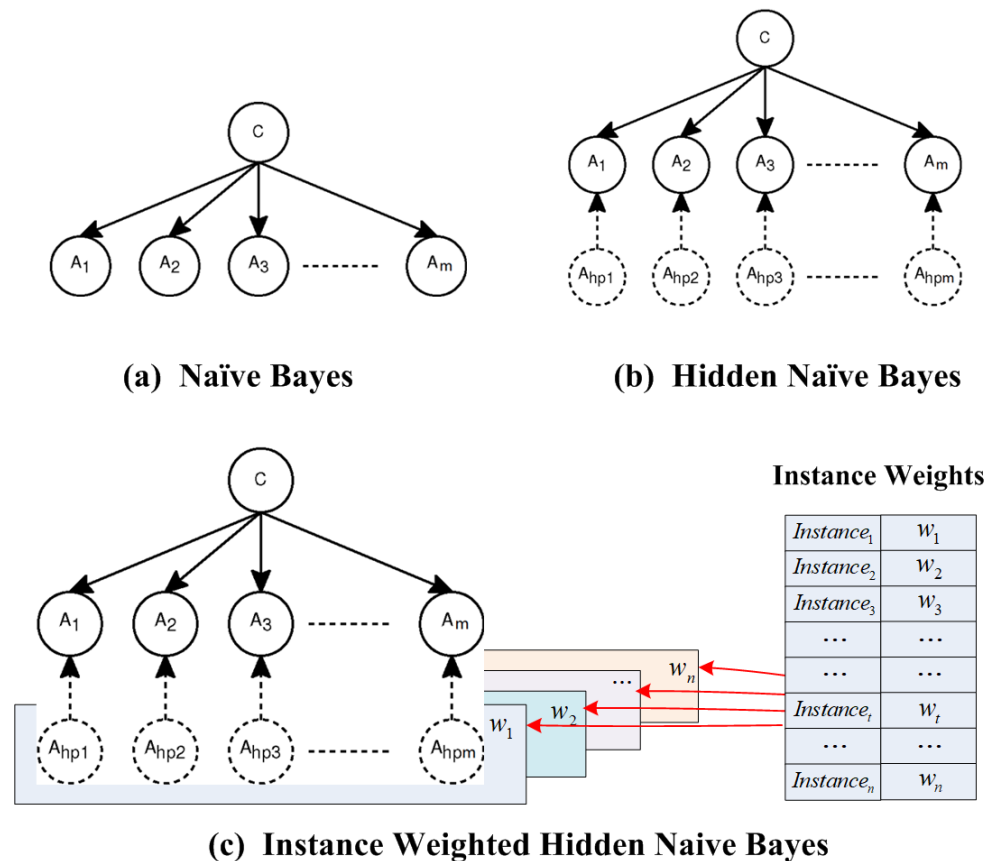


Figure 1. The different structures of related models.

Structure extension adds finite directed edges to reflect the dependencies between attributes [31]. It is efficient to overcome the conditional independence assumption of NB, since probabilistic relationships among attributes can be explicitly denoted by directed arcs [32]. Among various structure extension approaches, the hidden naive Bayes (HNB) is an improved model that essentially combines mixture dependencies of attributes [33]. It can display Bayesian network topology well and reflect the dependencies from all other attributes. However, HNB regards each instance as equally important when computing probability estimates. This assumption is not always true because different instances could have different contributions. In order to improve the classification performance of HNB, it will be interesting to study whether a better classification performance can be achieved by constructing an improved HNB model on the instance weighted dataset. The resulting model which combines instance weighting with the improved HNB model into one uniform framework inherits the effectiveness of HNB, and reflects different influences of different instances.

In this study, we propose the novel hybrid model which combines instance weighting with the improved HNB model into one uniform framework, referred to as instance weighted hidden naive Bayes (IWHNB). With the research of the existing HNB model, we propose an improved HNB model that can reflect different contributions of different instances. In contrast to the existing HNB model, the improved HNB model is built on the instance weighted dataset. Instance weights are incorporated into generating each

hidden parent to reflect mixture dependencies of both attributes and instances. In our IWHNB approach, the improved HNB model is proposed to approximate the ground-truth attribute dependencies. Meanwhile, instance weights are calculated by the attribute value frequency-based instance weighted filter. Each instance weight is incorporated into probability estimates and the classification formula in IWHNB.

We have completed experiments to compare IWHNB with NB, HNB, and other state-of-the-art competitors. Empirical studies show that IWHNB obtains more satisfactory classification performance than its competitors. Meanwhile, IWHNB maintains the low time complexity that characterizes HNB. The main contributions of the work presented in this paper can be briefly summarized as follows:

1. We reviewed the related work about structure extension and found that there is almost no method that focuses on the hybrid paradigm which combines structure extension with instance weighting.
2. We reviewed the related work about the existing instance weighting approaches and found that the Bayesian network in these researches is limited to NB.
3. The IWHNB approach is an improved approach which combines instance weighting with the improved HNB model into one uniform framework. It is a new paradigm to calculate discriminative instance weights for the structure extension model.
4. Although some training time is spent to calculate the weight of each instance, the experimental results show that our proposed IWHNB approach is still simple and efficient. Meanwhile, the classification performance of the IWHNB approach is more satisfactory than its competitors.

The paper is organized as follows. In Section 2, we review the related work with regard to this paper. In Section 3, we propose our IWHNB approach. In Section 4, we describe the experimental setup and results. In Section 5, we give our conclusions and outline suggestions for future research.

2. Related Work

2.1. Structure Extension

Structure extension adds finite directed edges to encode probabilistic relationships. The extended NB structure encodes attribute independence statements, where directed arcs can explicitly characterize the joint probability distribution. In the case of given its parents, the attribute is independent of its non descendants. Given a test instance x , represented by an attribute vector $\langle a_1, a_2, \dots, a_m \rangle$, Equation (1) is formalized to classify instance x in structure extended NB:

$$c(x) = \arg \max_{c \in C} P(c) \prod_{i=1}^m P(a_i | \Pi_{a_i}, c), \quad (1)$$

where m is the number of attributes, Π_{a_i} is the attribute value(s) of Π_{A_i} , and Π_{A_i} denotes the set of parent nodes of A_i except for the class node C . The prior probability $P(c)$ is defined by Equation (2) as follows:

$$P(c) = \frac{1 + \sum_{t=1}^n \delta(c_t, c)}{q + n}, \quad (2)$$

where n is the number of training instances, c_t is the class label of the t th training instance, q is the number of classes, and $\delta(\bullet)$ is a binary function, where the value is 1 when two variables are equal, and the value is 0 when unequal.

A number of structure extension approaches have been proposed to alleviate the primary weakness in NB [31]. The network structure of tree-augmented naive Bayesian (TAN) comprising all the attribute nodes is tree-like [14]. The class is the parent node of each attribute, and each attribute can have only one other attribute parent from other attributes. Aggregating one-dependence estimators (AODE) aggregate the joint probability distribution of all qualified classifiers [12]. This Bayesian model directly makes each

attribute the parent node of all other attributes, and does not need to learn the topological structure between attributes. Weighted average of one-dependence estimators (WAODE) is proposed to assign different weights to different one-dependence estimators [34]. Each attribute is set as the root attribute once.

Among various structure extension approaches, the hidden naive Bayes (HNB) [33] is an improved model that essentially combines mixture dependencies of attributes. In this study, our proposed IWHNB approach is based on the HNB model, so the HNB model is introduced in detail here. The existing HNB model is shown in the Figure 1b. C is the class label. Each hidden parent A_{hp_i} , $i = 1, 2, \dots, m$ is created for each attribute A_i . A dashed directed line which is from each hidden parent A_{hp_i} to attribute A_i distinguishes it from a regular arc. A test instance x , $x = \langle a_1, \dots, a_m \rangle$ classified by HNB is formalized as Equation (3):

$$c(x) = \arg \max_{c \in C} P(c) \prod_{i=1}^m P(a_i | a_{hp_i}, c), \quad (3)$$

where the prior probability $P(c)$ is also computed by Equation (2). A hidden parent A_{hp_i} is created for each attribute A_i . The probability $P(a_i | a_{hp_i}, c)$ is formalized as Equation (4):

$$P(a_i | a_{hp_i}, c) = \sum_{j=1, j \neq i}^m W_{ij} * P(a_i | a_j, c), \quad (4)$$

where W_{ij} ($i, j = 1, 2, \dots, m$ and $i \neq j$) are the weights calculated to reflect influences from other attributes. W_{ij} is calculated as Equation (5):

$$W_{ij} = \frac{I_P(A_i; A_j | C)}{\sum_{j=1, j \neq i}^m I_P(A_i; A_j | C)}, \quad (5)$$

where $I_P(A_i; A_j | C)$ is the conditional mutual information formalized as Equation (6):

$$I_P(A_i; A_j | C) = \sum_{a_i, a_j, c} P(a_i, a_j | c) \log \frac{P(a_i, a_j | c)}{P(a_i | c)P(a_j | c)}, \quad (6)$$

where $P(a_i | c)$, $P(a_j | c)$ and $P(a_i, a_j | c)$ are formalized as Equations (7)–(9), respectively:

$$P(a_i | c) = \frac{1 + \sum_{t=1}^n \delta(a_{ti}, a_i) \delta(c_t, c)}{n_i + \sum_{t=1}^n \delta(c_t, c)}, \quad (7)$$

$$P(a_j | c) = \frac{1 + \sum_{t=1}^n \delta(a_{tj}, a_j) \delta(c_t, c)}{n_j + \sum_{t=1}^n \delta(c_t, c)}, \quad (8)$$

$$P(a_i, a_j | c) = \frac{1 + \sum_{t=1}^n \delta(a_{ti}, a_i) \delta(a_{tj}, a_j) \delta(c_t, c)}{n_i + \sum_{t=1}^n \delta(a_{tj}, a_j) \delta(c_t, c)}, \quad (9)$$

where a_{ti} is the i th attribute value of the t th training instance, n_i is the number of values for the i th attribute.

In the HNB model, we can see that each hidden parent essentially combines mixture dependencies of all other attributes. The HNB model avoids structure learning with intractable computational complexity and reflects the dependencies from all other attributes, but it regards each instance equally important when computing probability estimates.

2.2. Instance Weighting

Naive Bayes (NB) is one of the most classic and efficient models in Bayesian networks. The classification performance of the NB classifier is comparable to state-of-the-art classifiers. The classifier of NB uses Equation (10) to classify a test instance x :

$$c(x) = \arg \max_{c \in C} P(c) \prod_{i=1}^m P(a_i|c), \quad (10)$$

where the prior probability $P(c)$ is also computed by Equation (2). In the meantime, the conditional probability $P(a_i|c)$ is defined by Equation (11):

$$P(a_i|c) = \frac{1 + \sum_{t=1}^n \delta(a_{ti}, a_i) \delta(c_t, c)}{n_i + \sum_{t=1}^n \delta(c_t, c)}, \quad (11)$$

where n_i is the number of values for the i th attribute.

Instance weighting is a practical way to improve NB by constructing a NB classifier on the instance weighted dataset [35]. It calculates the discriminative weight of each instance according to the distribution of the instance. Instance weighted NB still uses Equation (10) to classify a test instance x . The classification formula of instance weighted NB is the same as that for NB. However, different from the NB classifier, instance weighted NB calculates the discriminative weight of each instance, and incorporates discriminative instance weights into the prior probability and the conditional probability estimates. The prior probability $P(c)$ is redefined by Equation (12):

$$P(c) = \frac{1 + \sum_{t=1}^n w_t \delta(c_t, c)}{q + \sum_{t=1}^n w_t}, \quad (12)$$

where w_t is the weight of the t th training instance. In the meantime, the conditional probability $P(a_i|c)$ is redefined by Equation (13):

$$P(a_i|c) = \frac{1 + \sum_{t=1}^n w_t \delta(a_{ti}, a_i) \delta(c_t, c)}{n_i + \sum_{t=1}^n w_t \delta(c_t, c)}. \quad (13)$$

How to calculate the different weight of each instance to build an instance weighted NB classifier is crucial. Instance weighting approaches can broadly fall into two categories: eager learning and lazy learning. Eager learning uses general characteristics of instances to calculate instance weights during the training phase. Each instance weight is directly computed as a preprocessing step before the classification phase. Rather than calculating instance weights based on general characteristics of instances, lazy learning is more profitable to optimize instance weights at classification phase. Lazy learning first uses search algorithms to search for instance weights, and then optimize instance weights by building the target classifier on the instance weighted NB. Lazy learning spends more computational cost. So, eager learning normally is faster to calculate instance weights compared to lazy learning, but lazy learning has better classification performance than eager learning.

Discriminatively weighted NB uses the estimated conditional probability loss to calculate discriminative instance weights [15]. It is an eager learning approach, it achieves remarkable classification results in both classification accuracy and ranking. Attribute value frequency weighted NB is a simple and efficient eager learning approach [18]. This instance weighting filter focuses on the frequency of each attribute value to learn the weight of each instance. It calculates each instance weight according to its attribute value number and its attribute value frequency. Lazy NB clones each instance in the neighborhood [16]. It is a lazy learning approach. It calculates the similarity between the test instance and each training instance, and clones are then made based on the similarity. The improved algorithm called instance weighted NB finds the mode within training instances, and then calculates each weight according to the similarity between the mode and each instance [17].

Numerous instance weighting studies have revealed that the Bayesian networks in these existing instance weighting approaches are all limited to NB. It will be interesting to study whether a better classification performance can be obtained by exploiting instance weighting on the structure extended NB.

3. Instance Weighted Hidden Naive Bayes

The studies show that both structure extension and instance weighting can improve the classification performance. Structure extension extends the structure to overcome the unrealistic assumption, but regards each instance as equally important. Instance weighting weights each instance discriminatively to overcome the conditional independence assumption. Each instance weight is incorporated to calculate probability estimates, but the Bayesian network of existing instance weighting approaches is limited to NB. Based on the above analysis, we study whether more satisfying classification results can be obtained by exploiting instance weighting on the structure extended NB.

Following the reasons above, this paper focuses the research on the new hybrid paradigm which combines structure extension with instance weighting. The extended structure should be more accurate to reflect the dependency between attributes. Meanwhile, different instance weights can be incorporated into probability estimates and the classification formula to give more accurate results compared to traditional methods. Learned instance weights can reflect different contributions of different instances. Based on these, we propose a new hybrid approach which combines the improved hidden naive Bayes with instance weighting into one hybrid model. This improved hybrid approach is called instance weighted hidden naive Bayes (IWHNB). We modify the HNB model into the instance weighted hidden naive Bayes (IWHNB) model. In the following subsection, we describe the IWHNB model in detail.

3.1. The Instance Weighted Hidden Naive Bayes Model

Hidden naive Bayes (HNB) generates a hidden parent to each attribute to reflect dependencies from all other attributes [33]. Figure 1 effectively creates relationships among the models, as if they had evolved directly from one to the other. As Figure 1a shows, naive Bayes (NB) is one of the most classic and efficient models in BNs. As Figure 1b shows, the HNB model essentially adds a hidden parent to each attribute, but it regards each instance as equally important. The HNB model avoids structure learning with intractable computational complexity. It can be interpreted as the weight of each instance is set to 1 by default in HNB. However, in the training dataset, some instances contribute more to classification than others, so they should have more influence than less important instances. Different contributions for different instances can be a very important consideration.

Motivated by the work of HNB [33], we modify the HNB model into the instance weighted hidden naive Bayes model in our IWHNB approach. The instance weighted hidden naive Bayes model is shown in the Figure 1c. C is the class label, and is the parent node of each attribute. A hidden parent A_{hp_i} , $i = 1, 2, \dots, m$ is also created for each attribute A_i . n is the number of training instances. w_t is the weight of the t th training instance. In the improved HNB model, each instance weight w_t is integrated to generate the hidden parent to each attribute. A dashed directed line which is from each hidden parent A_{hp_i} to attribute A_i distinguishes it from a regular parent. Different from the existing HNB model, the improved HNB model not only essentially reflects dependencies from all other attributes but also can reflect different contributions of different instances.

In our IWHNB approach, the test instance $x = \langle a_1, \dots, a_m \rangle$ classified by IWHNB is formalized as Equation (14):

$$c(x) = \arg \max_{c \in C} P(c) \prod_{i=1}^m P(a_i | a_{hp_i}, c). \quad (14)$$

Although the classification formula of our IWHNB approach is the same as that for HNB, the calculations of the probabilities $P(c)$ and $P(a_i|a_{hp_i}, c)$ are different. We embed each instance weight w_t into the generation of each hidden parent. Instance weights are also incorporated into calculating probabilities. The detailed processes are described as follows. Firstly, we redefine the prior probability $P(c)$ as Equation (15):

$$P(c) = \frac{1 + \sum_{t=1}^n w_t \delta(c_t, c)}{q + \sum_{t=1}^n w_t}. \quad (15)$$

Secondly, the probability $P(a_i|a_{hp_i}, c)$ is formalized as Equation (16).

$$P(a_i|a_{hp_i}, c) = \sum_{j=1, j \neq i}^m W_{ij} * P(a_i|a_j, c), \quad (16)$$

where $P(a_i|a_j, c)$ and W_{ij} both are redefined in our IWHNB approach. We redefine the probability $P(a_i|a_j, c)$ as Equation (17):

$$P(a_i, a_j|c) = \frac{1 + \sum_{t=1}^n w_t \delta(a_{ti}, a_i) \delta(a_{tj}, a_j) \delta(c_t, c)}{n_i + \sum_{t=1}^n w_t \delta(a_{tj}, a_j) \delta(c_t, c)}, \quad (17)$$

where w_t is the weight of the t th training instance.

Thirdly, W_{ij} are weights which are measured by the conditional mutual information $I_P(A_i; A_j|C)$ to reflect influences from other attributes. W_{ij} is calculated as Equation (18):

$$W_{ij} = \frac{I_P(A_i; A_j|C)}{\sum_{j=1, j \neq i}^m I_P(A_i; A_j|C)}, \quad (18)$$

where $I_P(A_i; A_j|C)$ is defined as follows:

$$I_P(A_i; A_j|C) = \sum_{a_i, a_j, c} P(a_i, a_j|c) \log \frac{P(a_i, a_j|c)}{P(a_i|c)P(a_j|c)}. \quad (19)$$

In the process of computing $I_P(A_i; A_j|C)$ and W_{ij} , we incorporate instance weights to compute probability estimates. We redefine the probabilities $P(a_i, a_j|c)$, $P(a_i|c)$ and $P(a_j|c)$. The probability $P(a_i|a_j, c)$ is redefined as Equation (17). Meanwhile, $P(a_i|c)$ and $P(a_j|c)$ are respectively redefined as:

$$P(a_i|c) = \frac{1 + \sum_{t=1}^n w_t \delta(a_{ti}, a_i) \delta(c_t, c)}{n_i + \sum_{t=1}^n w_t \delta(c_t, c)}. \quad (20)$$

$$P(a_j|c) = \frac{1 + \sum_{t=1}^n w_t \delta(a_{tj}, a_j) \delta(c_t, c)}{n_j + \sum_{t=1}^n w_t \delta(c_t, c)}. \quad (21)$$

Finally, the probability $P(a_i|a_{hp_i}, c)$ is computed by Equation (16). The test instance is classified by Equation (14). Instance weights are incorporated into the process of calculating probability estimates and the classification formula.

In our IWHNB approach, the improved HNB model is modified to reflect the influences of both attributes and instances. Different contributions for different instances are considered when generating the improved HNB model. Different influences of different instance weights are embedded to generate a hidden parent of each attribute. Now, the only question is how to quantify different instance weights. To address this question, the next subsection will describe how to quantify the weight of each instance.

3.2. The Weight of Each Instance

In order to maintain the computational simplicity that characterizes HNB, we exploit eager learning, known as the attribute value frequency-based instance weighted filter, to calculate each single instance weight. The frequency of an attribute value means the ratio between the occurrence times of each attribute values and the instances' number. It can contain important information to define instance weights [18]. To quantify the frequency of an attribute value, f_{ti} is used to denote the frequency of attribute value a_{ti} (the i th attribute value of the t th instance). We define Equation (22) to denote the attribute value frequency:

$$f_{ti} = \frac{\sum_{r=1}^n \delta(a_{ri}, a_{ti})}{n}, \quad (22)$$

where a_{ri} is the i th attribute value of the r th instance. For the t th training instance, its attribute value frequency vector is denoted as $\langle f_{t1}, f_{t2}, \dots, f_{tm} \rangle$. The more frequently an attribute value appears, the more influence of an attribute value there is on the instance. The frequency of the occurrence of attribute values can well reflect the importance of the instance.

In our IWHNB approach, not only the attribute value frequency but also the number of values of different attributes is considered. $\langle n_1, n_2, \dots, n_m \rangle$ is used to denote values of each attribute's value number. It reflects the diversity of each attribute. Each instance weight has positive correlation with its attribute value frequency vector $\langle f_{t1}, f_{t2}, \dots, f_{tm} \rangle$ and the attribute value number vector $\langle n_1, n_2, \dots, n_m \rangle$.

Finally, we set the weight of each instance to be the dot product of the attribute value frequency vector and attribute value number vector. The weight of the t th instance w_t is formalized as the following Equation:

$$\begin{aligned} w_t &= \langle f_{t1}, f_{t2}, \dots, f_{tm} \rangle \bullet \langle n_1, n_2, \dots, n_m \rangle \\ &= \sum_{i=1}^m (f_{ti} * n_i). \end{aligned} \quad (23)$$

Based on the simple and efficient attribute value frequency-based instance weighted filter, a proper weight is assigned to each different instance. Discriminative instance weights are embedded to generate a hidden parent of each attribute to reflect the influences of both attributes and instances.

Now, the detailed learning algorithm for our instance weighted hidden naive Bayes (IWHNB for short) can be described as Algorithm 1. From Algorithm 1, the time complexity of computing instance weights is $O(3nm)$. n is the number of training instances. m is the number of attributes. IWHNB needs to compute the conditional mutual information for each pair of attributes. The time complexity is $O(qm^2v^2)$, v is the average number of values for an attribute, q is the number of class labels. The time complexity for computing each weight W_{ij} is $O(m^2)$. These formulas sum over n , thus, the training time complexity of IWHNB is $O(3nm + nm^2 + nqm^2v^2)$. The training procedure of the algorithm IWHNB is similar to that of HNB, except the additional procedure for calculating each instance weight. At classification time, Equation (14) is used to classify a test instance, and it takes $O(qn^2)$. The total time complexity of the IWHNB algorithm is $O(qn^2 + 3nm + nm^2 + nqm^2v^2)$, which shows that IWHNB is simple and efficient.

Algorithm 1 Instance Weighted Hidden Naive Bayes**Input:** TD -a training dataset; a test instance x **Output:** the predicted class label of x

- 1: Initialize all instance weights by the attribute value frequency-based instance weighted filter
- 2: **for** each training instance $t = 1$ to n **do**
- 3: **for** each training instance's attribute value, $i = 1$ to m **do**
- 4: Set new instance weight of t th instance to be the dot product of its attribute value frequency vector $\langle f_{t1}, f_{t2}, \dots, f_{tm} \rangle$ and the attribute value number vector $\langle n_1, n_2, \dots, n_m \rangle$
- 5: **end for**
- 6: **end for**
- 7: Discriminative instance weights are incorporated into the process of calculating probability estimates.
- 8: **for** each possible class label c that C takes **do**
- 9: Calculate $P(c)$ using Equation (15)
- 10: **for** each attribute A_i , $i = 1$ to m **do**
- 11: Calculate $P(a_i|c)$ using Equation (20)
- 12: **end for**
- 13: **for** each pair of attributes A_i and A_j ($i \neq j$) **do**
- 14: Calculate $P(a_i|a_j, c)$ as Equation (17)
- 15: Calculate $I_P(A_i; A_j|C)$ using Equation (19)
- 16: Calculate W_{ij} using Equation (18)
- 17: **end for**
- 18: Calculate $P(a_i|a_{hp_i}, c)$ using Equation (16)
- 19: **end for**
- 20: Predict the class label $c(x)$ of x by Equation (14)

4. Experiments and Results

In order to verify the performance of our proposed IWHNB, we completed experiments to compare IWHNB with NB, HNB and other state-of-the-art competitors. These state-of-the-art competitors and their abbreviations are listed as follows. HNB, AODE and TAN are state-of-the-art structure extension approaches. AVFWNB is an eager instance weighting approach. AIWNB is a new improved approach which combines instance weighting with attribute weighting.

- NB: Naive Bayes [36].
- HNB: Hidden naive Bayes [33].
- AVFWNB: Attribute value frequency weighted NB [18].
- AIWNB: Attribute and instance weighted NB [35].
- AODE: Aggregating one-dependence estimators [12].
- TAN: Tree-augmented NB [14].

We performed our study on the 36 University of California, Irvine (UCI) datasets [37]. These datasets are published on the WEKA platform [38]. They are from a wide range of fields and also have various data characteristics. In the process of preprocessing, we replace missing attribute values with the modes of the nominal attribute values or the means of the numerical attribute values. We also use the Fayyad & Irani's minimum description length (MDL) method [39] to discretize numerical attribute values. If the attribute's value number is the same instances' number, the attribute is redundant. So, we delete this type of attribute. There are three redundant attributes deleted: "Hospital Number" in the dataset "colic.ORIG", "instance name" in the dataset "splice", and "animal" in the dataset "zoo".

Table 1 shows the results of a comparison of the classification accuracy of each approach on each dataset after averaging the classification accuracies from ten runs of 10-fold cross-validation, respectively. Meanwhile, two-tailed t -test with the $p = 0.05$ significance level [40,41] is used to compare the proposed IWHNB with its competitors. We use the

symbol • to denote our proposed IWHNB is a significant improvement over its competitors, and use the symbol ◦ to denote it is a significant degradation over its competitors. The second-to-last line reveals the average accuracy of each algorithm, which can provide a gross indicator of its classification performance across all datasets. At the bottom of the Table 1, W/T/L reflects that our proposed IWHNB wins on *W* datasets, ties on *T* datasets and loses on *L* datasets over its competitors.

Then, the summary test results based on a corrected paired two-tailed *t*-test with the $p = 0.05$ significance level are shown in Table 2. For each entry $i(j)$, *i* is the number of datasets on which the algorithm in the column achieves higher classification accuracy than the algorithm in the corresponding row, and *j* is the number of datasets on which the algorithm in the column achieves significant wins with the $p = 0.05$ significance level with regard to the algorithm in the corresponding row. The ranking results are summarized in Table 3. The first column is the difference between the total number of wins and the total number of losses that the corresponding algorithm achieves compared with all the other algorithms, which is used to generate the ranking. The second column is the total number of winning datasets. The third column is the total number of losing datasets.

Table 1. Comparisons of the classification accuracy for IWHNB versus NB, HNB, AVFNB, AIWNB, AODE and TAN.

Dataset	IWHNB	NB	HNB	AVFNB	AIWNB	AODE	TAN
anneal	98.31 ± 1.29	96.13 ± 2.16 •	98.33 ± 1.22	98.62 ± 1.15	98.94 ± 1.05	98.01 ± 1.39	98.61 ± 1.02
anneal.ORIG	94.65 ± 2.24	92.66 ± 2.72 •	95.29 ± 2.04	93.32 ± 2.65 •	95.06 ± 2.23	93.35 ± 2.53	94.55 ± 2.10
audiology	78.17 ± 7.15	71.40 ± 6.37 •	69.04 ± 5.83 •	78.58 ± 8.44	83.93 ± 7.00 ◦	71.66 ± 6.42 •	65.35 ± 6.84 •
autos	85.56 ± 7.93	72.30 ± 10.31 •	82.17 ± 8.60	77.27 ± 9.43 •	78.04 ± 9.02 •	80.74 ± 8.68	80.85 ± 8.99 •
balance-scale	69.05 ± 3.74	71.08 ± 4.29 ◦	69.05 ± 3.75	71.10 ± 4.30 ◦	73.75 ± 4.22 ◦	69.34 ± 3.82	70.75 ± 3.99 ◦
breast-cancer	70.47 ± 6.29	72.94 ± 7.71	73.09 ± 6.11	71.41 ± 7.98	71.90 ± 7.55	72.53 ± 7.15	69.53 ± 7.13
breast-w	96.30 ± 1.94	97.25 ± 1.79	96.32 ± 2.01	97.48 ± 1.68 ◦	97.17 ± 1.68	96.97 ± 1.87	96.27 ± 2.08
colic	81.20 ± 6.00	81.39 ± 5.74	82.09 ± 5.86	81.47 ± 5.86	83.45 ± 5.45	82.64 ± 5.83	81.00 ± 5.86
colic.ORIG	74.23 ± 6.52	73.62 ± 6.83	74.06 ± 5.79	72.91 ± 6.34	73.87 ± 6.40	74.62 ± 6.51	68.31 ± 6.04 •
credit-a	85.23 ± 3.82	86.25 ± 4.01	85.91 ± 3.70	86.23 ± 3.85	87.03 ± 3.83	86.71 ± 3.82	85.39 ± 3.81
credit-g	75.85 ± 3.69	75.43 ± 3.84	76.12 ± 3.72	75.38 ± 3.90	75.81 ± 3.60	76.50 ± 3.89	73.54 ± 4.16 •
diabetes	76.75 ± 4.20	77.85 ± 4.67	76.81 ± 4.11	77.89 ± 4.66	77.87 ± 4.86	78.07 ± 4.56	78.70 ± 4.29 ◦
glass	77.70 ± 8.98	74.39 ± 7.95	77.80 ± 8.40	76.25 ± 8.07	74.02 ± 8.41	76.08 ± 8.07	76.23 ± 8.87
heart-c	81.52 ± 7.12	83.60 ± 6.42	82.31 ± 6.81	83.04 ± 6.68	82.71 ± 6.61	83.20 ± 6.20	81.62 ± 7.50
heart-h	84.56 ± 6.05	84.46 ± 5.92	84.87 ± 6.03	84.90 ± 5.68	84.29 ± 5.85	84.43 ± 5.92	84.05 ± 6.66
heart-statlog	82.33 ± 6.59	83.74 ± 6.25	82.33 ± 6.55	83.78 ± 6.29	83.22 ± 6.61	83.33 ± 6.61	82.44 ± 6.48
hepatitis	87.38 ± 8.43	84.22 ± 9.41	88.26 ± 7.28	85.38 ± 9.00	85.75 ± 8.97	84.98 ± 9.26	86.01 ± 8.25
hypothyroid	99.32 ± 0.40	98.48 ± 0.59 •	98.95 ± 0.48 •	98.98 ± 0.48 •	99.07 ± 0.48	98.76 ± 0.54 •	99.15 ± 0.44
ionosphere	93.96 ± 3.65	90.77 ± 4.76 •	91.82 ± 4.33 •	91.94 ± 4.09	92.40 ± 4.13	92.79 ± 4.26	92.25 ± 4.33
iris	93.27 ± 5.72	94.47 ± 5.61	93.80 ± 5.86	94.40 ± 5.50	94.40 ± 5.50	93.20 ± 5.76	94.20 ± 5.74
kr-vs-kp	92.70 ± 1.37	87.79 ± 1.91 •	92.36 ± 1.30 •	88.18 ± 1.86 •	93.73 ± 1.28 ◦	91.01 ± 1.67 •	92.88 ± 1.49
labor	95.90 ± 9.21	93.13 ± 10.56	94.87 ± 9.82	94.33 ± 10.13	94.33 ± 9.30	94.70 ± 9.15	92.47 ± 10.89
letter	90.17 ± 0.62	74.00 ± 0.88 •	88.20 ± 0.66 •	75.07 ± 0.84 •	75.56 ± 0.89 •	88.76 ± 0.70 •	85.49 ± 0.76 •
lymphography	85.89 ± 8.02	84.97 ± 8.30	85.84 ± 8.86	85.49 ± 7.83	84.68 ± 7.99	86.98 ± 8.32	85.30 ± 8.79
mushroom	99.96 ± 0.06	95.52 ± 0.78 •	99.94 ± 0.10	99.12 ± 0.31 •	99.53 ± 0.23 •	99.95 ± 0.07	99.99 ± 0.04
primary-tumor	46.14 ± 6.17	47.20 ± 6.02	47.66 ± 6.21	45.85 ± 6.53	47.76 ± 5.25	47.67 ± 6.30	44.77 ± 6.84
segment	96.87 ± 1.07	91.71 ± 1.68 •	95.88 ± 1.19 •	93.69 ± 1.41 •	94.16 ± 1.38 •	95.77 ± 1.23 •	95.58 ± 1.32 •
sick	97.52 ± 0.76	97.10 ± 0.84 •	97.56 ± 0.74	97.02 ± 0.86 •	97.33 ± 0.85	97.39 ± 0.79	97.40 ± 0.76
sonar	84.63 ± 7.72	85.16 ± 7.52	84.63 ± 7.34	84.49 ± 7.79	82.23 ± 8.65	86.60 ± 6.91	84.45 ± 8.31
soybean	94.61 ± 2.18	92.20 ± 3.23 •	93.88 ± 2.47	94.52 ± 2.36	94.74 ± 2.19	93.28 ± 2.84	94.98 ± 2.38
splice	96.24 ± 1.00	95.42 ± 1.14 •	95.84 ± 1.10 •	95.61 ± 1.11 •	96.21 ± 0.99	96.12 ± 1.00	94.95 ± 1.18 •
vehicle	73.70 ± 3.41	62.52 ± 3.81 •	72.37 ± 3.35 •	63.36 ± 3.87 •	63.59 ± 3.92 •	72.31 ± 3.62	73.39 ± 3.26
vote	94.39 ± 3.21	90.21 ± 3.95 •	94.43 ± 3.18	90.25 ± 3.95 •	92.18 ± 3.76 •	94.52 ± 3.19	94.43 ± 3.34
vowel	90.32 ± 2.71	65.23 ± 4.53 •	85.12 ± 3.65 •	67.46 ± 4.62 •	69.98 ± 4.11 •	80.87 ± 3.82 •	86.09 ± 3.91 •
waveform-5000	86.24 ± 1.45	80.72 ± 1.50 •	86.21 ± 1.44	80.65 ± 1.46 •	82.98 ± 1.37 •	86.03 ± 1.56	82.22 ± 1.71 •
zoo	98.33 ± 3.72	93.98 ± 7.14	97.73 ± 4.64	96.05 ± 5.60	96.05 ± 5.60	94.66 ± 6.38	95.15 ± 6.68
Average	86.37	83.31	85.86	84.21	84.94	85.68	84.95
W/T/L	-	17/18/1	9/27/0	13/21/2	8/25/3	6/30/0	9/25/2

Table 2. Summary test results on classification accuracy.

Algorithm	IWHNB	NB	HNB	AVFWNB	AIWNB	AODE	TAN
IWHNB	-	11 (1)	17 (0)	12 (2)	15 (3)	14 (0)	11 (2)
NB	25 (17)	-	27 (16)	26 (11)	27 (15)	29 (13)	20 (13)
HNB	18 (9)	9 (1)	-	13 (3)	16 (4)	18 (1)	12 (2)
AVFWNB	24 (13)	10 (0)	23 (11)	-	25 (10)	24 (10)	16 (8)
AIWNB	21 (8)	9 (0)	20 (7)	8 (0)	-	21 (8)	15 (6)
AODE	22 (6)	7 (1)	18 (3)	12 (3)	15 (7)	-	16 (4)
TAN	25 (9)	16 (2)	24 (5)	20 (2)	21 (5)	20 (6)	-

Table 3. Ranking test results on classification accuracy.

Algorithm	Wins-Losses	Wins	Losses
IWHNB	54	62	8
HNB	22	42	20
AIWNB	15	44	29
AODE	14	38	24
TAN	6	35	29
AVFWNB	−31	21	52
NB	−80	5	85

Based on comparison results, the conclusion is evident that our IWHNB approach obtains the best experimental results compared with its competitors. We summarize the conclusions briefly as follows:

1. According to results in Table 1, the averaged classification accuracy of IWHNB across all datasets is 86.37%. It is considerably higher than its competitors, such as NB (83.31%), HNB (85.86%), AVFWNB (84.21%), AIWNB (84.94%), AODE (85.68%) and TAN (84.95%). This suggests that our proposed IWHNB approach is effective.
2. IWHNB obtains the most satisfactory experimental results in accuracy. IWHNB outperforms NB (17 wins, 18 ties and 1 loss), HNB (9 wins, 27 ties and 0 losses), AVFWNB (13 wins, 21 ties and 2 losses), AIWNB (8 wins, 25 ties and 3 losses), AODE (6 wins, 30 ties and 0 losses) and TAN (9 wins, 25 ties and 2 losses).
3. The summary and ranking test results show that IWHNB is overall the best across all datasets (62 wins and 8 losses). The descending sort across all datasets is IWHNB, HNB, AIWNB, AODE, TAN, AVFWNB and NB.
4. Compared with HNB, IWHNB considerably improves the classification accuracy (nine wins and zero losses). This suggests that this improved hybrid approach which combines the improved HNB model with instance weighting improves the classification performance effectively.

Furthermore, we observe the performance of IWHNB in terms of the elapsed training time (in milliseconds). Our experiments were conducted on a Linux machine with 3.2 GHz processor and 8 GB of RAM. The elapsed training time comparison results are shown in Tables 4–6. Note that the meanings of the t-test results in these tables are opposite to those in Tables 1–3. For the elapsed training time, a small number which indicates lower time complexity is better than a large number. Thus, in Table 4, the symbols ◦ and • denote statistically significant improvement or degradation over its competitors, respectively. Each W/T /L implies that compared to its competitors, our proposed IWHNB wins on W datasets, ties on T datasets, and loses on L datasets. In Table 5, i of value $i(j)$ denotes the number of datasets that the algorithm corresponding to the column loses compared to the algorithm corresponding to the row. In Table 6, the second and third columns represent the total numbers of losses and wins, respectively. The first column is the difference between the second column of losses and third column of wins. We summarize the main highlights of these comparisons as follows:

1. According to results in Table 4, the averaged elapsed training time of IWHNB is 13.15 milliseconds, which is a little bigger than that of HNB (12.56 milliseconds). Therefore, our proposed IWHNB approach maintains the computational simplicity that characterizes HNB. It is a simple, efficient and effective approach.
2. Compared with TAN, IWHNB has the lower time complexity. The averaged elapsed training time of IWHNB is smaller than that of TAN (15.84 milliseconds). It reduces the elapsed training time on 8 datasets, and loses on 0 datasets.
3. According to Tables 4–6, IWHNB indeed has higher time complexity than NB, AVFWNB, AIWNB, AODE, HNB, but it still has low computational simplicity. Structure extension and instance weighting are both completed in our IWHNB approach.

Table 4. Comparisons of the elapsed training time for IWHNB versus NB, HNB, AVFWNB, AIWNB, AODE and TAN.

Dataset	IWHNB			NB			HNB			AVFWNB			AIWNB			AODE			TAN							
anneal	19.42	±	12.25	0.45	±	0.88	•	13.28	±	6.04	•	0.93	±	1.37	•	6.20	±	4.53	•	7.84	±	1.64	•	17.18	±	7.88
anneal.ORIG	16.80	±	3.95	0.25	±	0.44	•	12.32	±	1.79	•	0.36	±	0.50	•	4.20	±	0.65	•	6.95	±	1.00	•	14.19	±	1.10
audiology	63.53	±	18.79	0.13	±	0.34	•	57.39	±	4.66	•	0.24	±	0.43	•	6.53	±	1.38	•	7.12	±	1.20	•	95.49	±	9.07
autos	4.65	±	1.25	0.03	±	0.17	•	4.35	±	0.63	•	0.09	±	0.29	•	1.15	±	0.41	•	0.86	±	0.43	•	4.89	±	0.65
balance-scale	0.18	±	0.39	0.06	±	0.42	•	0.08	±	0.27	•	0.07	±	0.26	•	0.17	±	0.40	•	0.05	±	0.22	•	0.18	±	0.46
breast-cancer	0.52	±	0.70	0.02	±	0.14	•	0.40	±	0.49	•	0.01	±	0.10	•	0.27	±	0.45	•	0.16	±	0.37	•	0.38	±	0.55
breast-w	0.62	±	0.49	0.03	±	0.17	•	0.48	±	0.50	•	0.04	±	0.20	•	0.48	±	0.52	•	0.30	±	0.48	•	0.59	±	0.49
colic	1.95	±	0.50	0.09	±	0.29	•	1.77	±	0.49	•	0.13	±	0.34	•	0.83	±	0.40	•	0.98	±	0.38	•	2.34	±	0.52
colic.ORIG	3.77	±	1.20	0.09	±	0.29	•	3.28	±	0.53	•	0.13	±	0.34	•	1.54	±	0.54	•	1.44	±	0.62	•	4.05	±	0.87
credit-a	1.83	±	0.88	0.05	±	0.22	•	1.34	±	0.50	•	0.18	±	0.39	•	0.78	±	0.44	•	0.87	±	0.46	•	1.59	±	0.73
credit-g	3.00	±	0.59	0.15	±	0.36	•	3.12	±	0.59	•	0.26	±	0.44	•	1.66	±	0.81	•	2.17	±	0.45	•	3.42	±	0.81
diabetes	0.48	±	0.50	0.09	±	0.29	•	0.36	±	0.48	•	0.09	±	0.29	•	0.37	±	0.49	•	0.31	±	0.46	•	0.40	±	0.55
glass	0.42	±	0.50	0.00	±	0.00	•	0.38	±	0.49	•	0.02	±	0.14	•	0.13	±	0.34	•	0.07	±	0.26	•	0.44	±	0.52
heart-c	0.70	±	0.61	0.10	±	0.30	•	0.68	±	0.49	•	0.04	±	0.20	•	0.29	±	0.46	•	0.23	±	0.42	•	0.90	±	0.61
heart-h	0.63	±	0.49	0.07	±	0.26	•	0.68	±	0.49	•	0.03	±	0.17	•	0.31	±	0.46	•	0.27	±	0.47	•	0.71	±	0.56
heart-statlog	0.60	±	0.51	0.05	±	0.22	•	0.37	±	0.51	•	0.03	±	0.17	•	0.29	±	0.46	•	0.28	±	0.45	•	0.43	±	0.54
hepatitis	0.81	±	0.51	0.06	±	0.24	•	0.72	±	0.51	•	0.03	±	0.17	•	0.35	±	0.48	•	0.39	±	0.49	•	0.95	±	0.39
hypothyroid	19.24	±	1.92	1.16	±	0.60	•	18.46	±	1.46	•	1.90	±	0.70	•	10.27	±	1.63	•	17.96	±	3.36	•	21.08	±	2.44
ionosphere	5.20	±	0.77	0.06	±	0.24	•	5.26	±	0.63	•	0.09	±	0.29	•	2.36	±	0.50	•	2.78	±	1.38	•	7.79	±	1.23
iris	0.03	±	0.17	0.00	±	0.00	•	0.05	±	0.22	•	0.04	±	0.20	•	0.05	±	0.22	•	0.05	±	0.22	•	0.05	±	0.22
kr-vs-kp	27.11	±	5.29	1.33	±	0.64	•	23.27	±	1.22	•	1.71	±	0.71	•	23.76	±	23.79	•	23.15	±	4.53	•	30.78	±	5.71
labor	0.51	±	0.85	0.00	±	0.00	•	0.38	±	0.51	•	0.02	±	0.14	•	0.18	±	0.39	•	0.05	±	0.22	•	0.52	±	0.56
letter	81.00	±	21.55	4.51	±	0.86	•	72.44	±	8.74	•	9.73	±	1.05	•	74.51	±	36.78	•	66.65	±	23.94	•	79.07	±	13.71
lymphography	1.16	±	0.72	0.02	±	0.14	•	1.15	±	0.67	•	0.04	±	0.20	•	0.47	±	0.50	•	0.26	±	0.44	•	1.18	±	0.67
mushroom	24.98	±	3.41	1.85	±	1.37	•	24.47	±	1.47	•	4.26	±	0.79	•	25.80	±	4.66	•	25.55	±	3.54	•	27.06	±	3.80
primary-tumor	3.28	±	0.57	0.06	±	0.24	•	3.57	±	0.76	•	0.09	±	0.29	•	0.91	±	0.47	•	0.58	±	0.50	•	3.76	±	1.68
segment	11.00	±	1.36	0.42	±	0.52	•	12.32	±	1.52	•	0.78	±	0.50	•	7.52	±	1.49	•	6.87	±	1.19	•	13.28	±	1.78
sick	18.14	±	1.98	1.04	±	0.53	•	17.77	±	1.64	•	1.85	±	0.59	•	11.92	±	2.79	•	16.74	±	1.54	•	22.35	±	5.21
sonar	7.16	±	0.85	0.07	±	0.26	•	7.76	±	1.40	•	0.18	±	0.39	•	2.24	±	0.51	•	4.17	±	1.09	•	29.32	±	3.92
soybean	17.57	±	2.01	0.30	±	0.46	•	17.80	±	1.84	•	0.53	±	0.50	•	4.85	±	1.86	•	5.59	±	1.54	•	20.50	±	2.12
splice	81.68	±	7.16	1.79	±	0.67	•	89.51	±	4.19	•	3.14	±	0.75	•	63.97	±	8.34	•	80.66	±	11.04	•	101.74	±	11.40
vehicle	2.95	±	0.58	0.13	±	0.37	•	2.87	±	0.51	•	0.25	±	0.44	•	1.45	±	0.52	•	1.63	±	0.60	•	3.09	±	0.64
vote	0.92	±	0.34	0.07	±	0.26	•	0.90	±	0.46	•	0.11	±	0.31	•	0.52	±	0.54	•	0.64	±	0.54	•	1.00	±	0.45
vowel	2.86	±	0.59	0.15	±	0.36	•	3.04	±	0.63	•	0.23	±	0.42	•	1.14	±	0.40	•	1.11	±	0.31	•	2.93	±	0.57
waveform-5000	47.56	±	2.10	1.61	±	0.58	•	49.06	±	2.40	•	3.05	±	0.85	•	29.01	±	1.84	•	46.70	±	4.64	•	55.69	±	2.93
zoo	0.98	±	0.38	0.00	±	0.00	•	1.01	±	0.33	•	0.08	±	0.27	•	0.19	±	0.39	•	0.21	±	0.43	•	1.03	±	0.36
Average	13.15			0.45			12.56			0.85			7.96			9.21			15.84							
W/T/L	-			0/3/33			1/32/3			0/4/32			0/15/21			0/19/17			8/28/0							

Table 5. Summary test results on elapsed training time.

Algorithm	IWHNB	NB	HNB	AVFWNB	AIWNB	AODE	TAN
IWHNB	-	0 (0)	12 (1)	1 (0)	2 (0)	2 (0)	27 (8)
NB	36 (33)	-	36 (32)	30 (6)	36 (25)	35 (23)	36 (31)
HNB	24 (3)	0 (0)	-	0 (0)	5 (0)	1 (0)	33 (13)
AVFWNB	35 (32)	5 (0)	36 (31)	-	36 (24)	35 (23)	36 (32)
AIWNB	34 (21)	0 (0)	29 (21)	0 (0)	-	17 (6)	35 (24)
AODE	34 (17)	1 (0)	34 (18)	1 (0)	18 (0)	-	35 (25)
TAN	8 (0)	0 (0)	2 (0)	0 (0)	0 (0)	0 (0)	-

Table 6. Ranking test results on elapsed training time.

Algorithm	Losses-Wins	Losses	Wins
TAN	133	133	0
IWHNB	97	106	9
HNB	87	103	16
AODE	−8	52	60
AIWNB	−23	49	72
AVFWNB	−136	6	142
NB	−150	0	150

5. Conclusions and Future Work

Hidden naive Bayes (HNB) adds a hidden parent to each attribute to encode attribute dependencies. However, it regards each instance as equally important. In this paper, we propose an improved hybrid approach which combines the improved NB model with instance weighting into one hybrid model, called instance weighted hidden naive Bayes (IWHNB). In our IWHNB approach, different contributions for different instances are considered when generating the improved HNB model. Experiments are conducted to compare IWHNB with NB, HNB and other state-of-the-art competitors in terms of the classification accuracy and the elapsed training time. The classification accuracy comparison results show that our IWHNB approach obtains the best experimental results compared with its competitors. The elapsed training time comparison results show that IWHNB maintains the computational simplicity that characterizes HNB. IWHNB is a simple, efficient and effective approach.

How to calculate optimal instance weights to overcome the unrealistic assumption is crucial. We think that more sophisticated algorithms can be used to learn more optimal instance weights to optimize our current version.

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

Funding: The work was partially supported by Science and Technology Project of Hubei Province—Unveiling System (2019AEE020), Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP-2018A05), Scientific Research Foundation for Talent introduction (20RC07), Teaching Research Project of Hubei University of Education (X2019009).

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 this manuscript:

NB	Naive Bayes
HNB	Hidden NB
AVFWNB	Attribute Value Frequency Weighted NB
AIWNB	Attribute and Instance Weighted NB
AODE	Aggregating One-Dependence Estimators
TAN	Tree-augmented NB

IWHNB	Instance Weighted HNB
BN	Bayesian Network
WAODE	Weighted Average of One-Dependence Estimators
UCI	University of California, Irvine
WEKA	Waikato Environment for Knowledge Analysis
MDL	Minimum Description Length
NP	Non-deterministic Polynomial

References

- Zhang, Y.; Wu, J.; Zhou, C.; Cai, Z. Instance cloned extreme learning machine. *Pattern Recognit.* **2017**, *68*, 52–65. [\[CrossRef\]](#)
- Yu, L.; Jiang, L.; Wang, D.; Zhang, L. Attribute Value Weighted Average of One-Dependence Estimators. *Entropy* **2017**, *19*, 501. [\[CrossRef\]](#)
- Wu, J.; Cai, Z.; Zhu, X. Self-adaptive probability estimation for Naive Bayes classification. In Proceedings of the International Joint Conference on Neural Networks, Dallas, TX, USA, 4–9 August 2013.
- Qiu, C.; Jiang, L.; Li, C. Not always simple classification: Learning superparent for class probability estimation. *Expert Syst. Appl.* **2015**, *42*, 5433–5440. [\[CrossRef\]](#)
- Jiang, L.; Wang, S.; Li, C.; Zhang, L. Structure extended multinomial naive Bayes. *Inf. Sci.* **2016**, *329*, 346–356. [\[CrossRef\]](#)
- Yu, L.; Gan, S.; Chen, Y.; He, M. Correlation-Based Weight Adjusted Naive Bayes. *IEEE Access* **2020**, *8*, 51377–51387. [\[CrossRef\]](#)
- Jiang, L.; Li, C.; Cai, Z. Learning decision tree for ranking. *Knowl. Inf. Syst.* **2009**, *20*, 123–135. [\[CrossRef\]](#)
- Bai, Y.; Wang, H.; Wu, J.; Zhang, Y.; Jiang, J.; Long, G. Evolutionary lazy learning for Naive Bayes classification. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 3124–3129.
- Bermejo, P.; Gámez, J.A.; Puerta, J.M. Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowl.-Based Syst.* **2014**, *55*, 140–147. [\[CrossRef\]](#)
- Hall, M. A decision tree-based attribute weighting filter for naive Bayes. *Knowl.-Based Syst.* **2007**, *20*, 120–126. [\[CrossRef\]](#)
- Webb, G.I.; Boughton, J.R.; Fei, Z.; Ting, K.; Salem, H. Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification. *Mach. Learn.* **2012**, *86*, 233–272. [\[CrossRef\]](#)
- Webb, G.I.; Boughton, J.R.; Wang, Z. Not so naive Bayes: Aggregating one-dependence estimators. *Mach. Learn.* **2005**, *58*, 5–24. [\[CrossRef\]](#)
- Yang, Y.; Webb, G.; Korb, K.; Boughton, J.; Ting, K. To Select or To Weigh: A Comparative Study of Linear Combination Schemes for SuperParent-One-Dependence Estimators. *IEEE Trans. Knowl. Data Eng.* **2007**, *9*, 1652–1665. [\[CrossRef\]](#)
- Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian network classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [\[CrossRef\]](#)
- Jiang, L.; Wang, D.; Cai, Z. Discriminatively weighted naive bayes and its application in text classification. *Int. J. Artif. Intell. Tools* **2012**, *21*, 1250007. [\[CrossRef\]](#)
- Jiang, L.; Guo, Y. Learning Lazy Naive Bayesian Classifiers for Ranking. In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, Hong Kong, China, 14–16 November 2005; pp. 412–416.
- Jiang, L.; Cai, Z.; Wang, D. Improving Naive Bayes for Classification. *Int. J. Comput. Appl.* **2010**, *32*, 328–332. [\[CrossRef\]](#)
- Xu, W.; Jiang, L.; Yu, L. An attribute value frequency-based instance weighting filter for naive Bayes. *J. Exp. Theor. Artif. Intell.* **2019**, *31*, 225–236. [\[CrossRef\]](#)
- Kohavi, R. Scaling Up the Accuracy of Naive-Bayes Classifier: A Decision-Tree Hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*; AAAI Press: Portland, ON, USA, 1996; pp. 202–207.
- Xie, Z.; Hsu, W.; Liu, Z.; Lee, M. A Selective Neighborhood Based Naive Bayes for Lazy Learning. In *Proceedings of the Sixth Pacific Asia Conference on KDD*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 104–114.
- Frank, E.; Hall, M.; Pfahringer, B. Locally Weighted Naive Bayes. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2003; pp. 249–256.
- Lee, C.H.; Gutierrez, F.; Dou, D. Calculating feature weights in naive bayes with kullback-leibler measure. In Proceedings of the 11th International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011; pp. 1146–1151.
- Jiang, L.; Zhang, L.; Li, C.; Wu, J. A Correlation-Based Feature Weighting Filter for Naive Bayes. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 201–213. [\[CrossRef\]](#)
- Zhang, H.; Jiang, L.; Yu, L. Class-specific Attribute Value Weighting for Naive Bayes. *Inf. Sci.* **2020**, *508*, 260–274. [\[CrossRef\]](#)
- Zaidi, N.A.; Cerquides, J.; Carman, M.J.; Webb, G.I. Alleviating naive Bayes attribute independence assumption by attribute weighting. *J. Mach. Learn. Res.* **2013**, *14*, 1947–1988.
- 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. [\[CrossRef\]](#)
- Langley, P.; Sage, S. Induction of selective Bayesian classifiers. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*; Morgan Kaufmann: San Francisco, CA, USA, 1994; pp. 339–406.
- Jiang, L.; Cai, Z.; Zhang, H.; Wang, D. Not so greedy: Randomly Selected Naive Bayes. *Expert Syst. Appl.* **2012**, *39*, 11022–11028. [\[CrossRef\]](#)
- Jiang, L.; Zhang, H.; Cai, Z.; Su, J. Evolutional naive bayes. In Proceedings of the 2005 International Symposium on Intelligent Computation and Its Application, Wuhan, China, 22–24 October 2005; pp. 344–350.

-
30. Chen, S.; Webb, G.; Liu, L.; Ma, X. A novel selective naïve Bayes algorithm. *Knowl.-Based Syst.* **2020**, *192*, 105361. [[CrossRef](#)]
 31. Xiang, Z.; Kang, D. Attribute weighting for averaged one-dependence estimators. *Appl. Intell.* **2017**, *46*, 616–629. [[CrossRef](#)]
 32. Yu, L.; Jiang, L.; Wang, D.; Zhang, L. Toward naïve Bayes with attribute value weighting. *Neural Comput. Appl.* **2019**, *31*, 5699–5713. [[CrossRef](#)]
 33. Jiang, L.; Zhang, H.; Cai, Z. A novel Bayes model: Hidden naïve Bayes. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1361–1371. [[CrossRef](#)]
 34. Jiang, L.; Zhang, H.; Cai, Z.; Wang, D. Weighted average of one-dependence estimators. *J. Exp. Theor. Artif. Intell.* **2012**, *24*, 219–230. [[CrossRef](#)]
 35. Zhang, H.; Jiang, L.; Yu, L. Attribute and instance weighted naïve Bayes. *Pattern Recognit.* **2021**, *111*, 107674. [[CrossRef](#)]
 36. Langley, P.; Iba, W.; Thompson, K. An Analysis of Bayesian Classifiers. In Proceedings of the 10th National Conference on Artificial Intelligence, San Jose, CA, USA, 12–16 July 1992; pp. 223–228.
 37. Frank, A.; Asuncion, A. *UCI Machine Learning Repository*; University of California: Irvine, CA, USA, 2010.
 38. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed.; Morgan Kaufmann: San Francisco, CA, USA, 2011.
 39. Fayyad, U.M.; Irani, K.B. Multi-interval discretization of continuous-valued attributes for classification learning. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambéry, France, 28 August–3 September 1993; pp. 1022–1027.
 40. Nadeau, C.; Bengio, Y. Inference for the Generalization Error. *Mach. Learn.* **2003**, *52*, 239–281. [[CrossRef](#)]
 41. Jiang, L.; Zhang, L.; Yu, L.; Wang, D. Class-specific attribute weighted naïve Bayes. *Pattern Recognit.* **2019**, *88*, 321–330. [[CrossRef](#)]