Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 170324, 7 pages
http://dx.doi.org/10.1155/2015/170324

Research Article

Bayesian Prediction Model Based on Attribute Weighting and Kernel Density Estimations

Zhong-Liang Xiang,[1] Xiang-Ru Yu,[1] and Dae-Ki Kang[2]

[1] *Weifang University of Science & Technology, Shouguang, Shandong 262-700, China*
[2] *Department of Computer and Information Engineering, Dongseo University, 47, Churye-Ro, Sasang-Gu, Busan 617-716, Republic of Korea*

Correspondence should be addressed to Dae-Ki Kang; dkkang@dongseo.ac.kr

Received 26 March 2015; Accepted 29 July 2015

Academic Editor: Fons J. Verbeek

Although naïve Bayes learner has been proven to show reasonable performance in machine learning, it often suffers from a few problems with handling real world data. First problem is conditional independence; the second problem is the usage of frequency estimator. Therefore, we have proposed methods to solve these two problems revolving around naïve Bayes algorithms. By using an attribute weighting method, we have been able to handle conditional independence assumption issue, whereas, for the case of the frequency estimators, we have found a way to weaken the negative effects through our proposed smooth kernel method. In this paper, we have proposed a compact Bayes model, in which a smooth kernel augments weights on likelihood estimation. We have also chosen an attribute weighting method which employs mutual information metric to cooperate with the framework. Experiments have been conducted on UCI benchmark datasets and the accuracy of our proposed learner has been compared with that of standard naïve Bayes. The experimental results have demonstrated the effectiveness and efficiency of our proposed learning algorithm.

1. Introduction

Naïve Bayes classifier is a supervised learning method based on Bayes rule of probability theory, running on labeled training examples and driven by a strong assumption that all attributes in the training examples are independent from one another on the given training examples known as naïve Bayes assumption or naïve Bayes conditional independence assumption. Naïve Bayes classifier has high performance and rapid classification speed and has exhibited its effectiveness especially in huge training instances with plenty of attributes mainly because of its independence assumption [1].

In practice, classification performance is affected by the attribute independence assumption which is usually violated in real world. However, due to the attractive advantages of efficiency and simplicity, both stemming from the attribute independence assumption, many researchers have proposed effective methods to further improve the performance of naïve Bayes classifier by weakening the attribute independence without neglecting its advantages. We categorize some typical previous methods of relaxing naïve Bayes assumption and give brief reviews in Section 3. However, we have found out that attribute weighting method has drawn relatively little attention among those previous methods in improving naïve Bayes classifier, especially in the case when attribute weighting method is combined with kernel method in a reasonable way.

Although Chen and Wang [2] proposed attribute weighting method with the kernel, their weighting scheme generates a series of parameters from least squares cross-validation which is less meaningful in terms of interpretation than our proposed method. In contrast, we propose an attribute weighting algorithm based on attribute weighting framework with kernel method. Our method makes the weights embedded in kernel have relatively interpretable meaning; thus we can flexibly choose different metrics and methods to measure the weights based on our attribute weighting framework.

Contributions of this paper are threefold:

(i) We briefly make a survey of ways to improve naïve Bayes, especially focusing on those naïve Bayes weighting methods.

(ii) We propose a novel attribute weighting framework called Attribute Weighting with Smooth Kernel Density Estimation, simply AW-SKDE. The AW-SKDE framework employs a smooth kernel that makes the probabilistic estimation of likelihood to be dominated by the weights, which enables the combination of kernel methods and weighting methods. After setting up the kernel, we can generate a set of weights directly by using various methods cooperating with the kernel.

(iii) On the AW-SKDE framework, we propose a learner called AW-SKDE$^{\mathrm{MI}}$ in which we choose the mutual information criterion to measure the dependency between an attribute and its class label.

Our experimental results show that mutual information criterion based on AW-SKDE framework exhibits superior performance compared to standard naïve Bayes classifier.

The paper is organized as follows: we briefly make a survey of ways to improve naïve Bayes in Section 2. In Section 3, we introduce the background of our study. In Section 4, we first propose our attribute weighting framework based on kernel density estimation. After that, we propose a method employing the mutual information criterion for attribute weighting based on our proposed framework. In Section 5, we describe the experiment and results in detail. Lastly, we draw conclusions for our study and describe the future research in Section 6.

2. Related Work

A number of methods that weaken attribute independent assumption for naïve Bayes have been proposed in the recent years. Jiang et al. [3] made a survey about improving naïve Bayes method. Those methods are broadly divided into five main categories: structure extension, feature selection, data expansion, local learning, and attribute weighting. We make a brief review by following this categorization.

For data expansion, Kang and Sohn [4] have presented an algorithm called propositionalized attribute taxonomy learner, simply PAT-learner. In PAT-learner, the training data set is first disassembled into small pieces with attributes values; then, PAT-learner rebuilds a new data set called PAT-Table by using divergence between the distribution of the class labels associated with the corresponding attributes at the disassembled date set. Kang and Kim [5] also proposed a Bayes learner based on PAT-learner, called propositionalized attribute taxonomy guided naïve Bayes learner (PAT-NBL). They utilize propositionalized data set and PAT-Table that is generated from PAT-learner to build naïve Bayes classifiers.

Wong [6] has focused on the discretization method of attributes to improve naïve Bayes. Wong has proposed a hybrid method for continuous attributes and mentioned that discretizing continuous attributes in a data set using different methods can improve the performance of naïve Bayes learner. Also, Wong provides a nonparametric measure to evaluate the dependence level between a continuous attribute and the class.

In structure extension, Webb et al. [7] have proposed a method called aggregating one-dependence estimators, simply AODE. In AODE, the conditional probability of test instances given class is tuned by one attribute value which occurs in the test instance. After the training stage, AODE outputs an average one-dependence estimator. AODE is a lazy method of structure extension of Bayesian network. Jiang et al. [3] have proposed hidden naïve Bayes, simply HNB, which is also a kind of structure extension method.

As for attribute weighting methods, we have two ways to get attribute weights. The first one is to construct a function with the parameters of attribute weight and to let this function fit itself with the training data by estimating the weights. Zaidi et al. [8] have proposed a weighted naïve Bayes algorithm, called weighting to alleviate the naïve Bayes independence assumption, simply WANBIA. Based on WANBIA framework, the authors have described two methods to obtain the attribute weights: WANBIA$^{\mathrm{CLL}}$, which maximizes the conditional log likelihood function and WANBIA$^{\mathrm{MSE}}$, which minimizes mean squared error function.

Chen and Wang [2] have also proposed an algorithm to minimize mean squared error function in order to obtain the attribute weights. In another paper, Chen and Wang [9] have proposed a method called subspace weighting naïve Bayes (simply SWNB) that is a naïve Bayes weighting method to deal with high-dimensional data. Using the local feature-weighting technique, SWNB has the ability to describe different contributions of attributes in the training data set and outputs an optimal set of attribute weights fitting a Logit normal priori distribution.

There are many other methods that can be categorized into attribute weighting. Lee et al. [10] have calculated attributes weight via Kullback-Leibler divergence between the attribute and class label. Wu and Cai [11] have proposed decision tree-based attribute weighted AODE, simply DTWAODE. DTWAODE generates a set of attribute weights directly, and the weight value decreases according to attribute depth in the decision tree. Omura et al. [12] have proposed a weighting method, called confidence weight for naïve Bayes, and that confidence weight is derived from the probabilities of the majority class in the training data set.

3. Background

In this section, we explain the concepts of machine learning methods used in this paper, including naïve Bayes classifier, naïve Bayes attribute weighting, and kernel density estimation for naïve Bayes categorical attributes. The symbols used in this paper are summarized in Notations section.

3.1. Naïve Bayes Classifier. In supervised learning, consider a training data set $\mathscr{D} = \{\mathbf{x}^{(\mathbf{1})}, \ldots, \mathbf{x}^{(\mathbf{n})}\}$ composed of n instances, where each instance $\mathbf{x} = \langle x_1, \ldots, x_m \rangle \in \mathscr{D}$ (m-dimensional

vector) is labeled with class label $c \in C$. For the posterior probability of c given $\mathbf{x}$, we have

$$p(c \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c) \cdot p(c)}{p(\mathbf{x})} \propto p(\mathbf{x} \mid c). \quad (1)$$

But likelihood $p(\mathbf{x} \mid c)$ cannot be directly estimated from $\mathscr{D}$ because of insufficient data in practice. Naïve Bayes uses attributes independence assumption to alleviate this problem; from the assumption, $p(\mathbf{x} \mid c)$ is shown as follows:

$$p(\mathbf{x} \mid c) = \prod_{i=1}^{m} p(x_i \mid c). \quad (2)$$

In the training phase, only $p(x_i \mid c)$ and $p(c)$ need to be estimated for each class $c \in C$ and each attribute value $x_i \in A_i$. The estimation method uses the frequency of x_i given c and the frequency of c for $p(x_i \mid c)$ and $p(c)$, respectively.

In the classification phase, if we have a test instance $\mathbf{t} = \langle t_1, \ldots, t_m \rangle$ where t_m is an attribute value of the attribute m in the test instance, naïve Bayes classifier outputs a class label prediction of $\mathbf{t}$ based on the frequency estimation of $p(x_i \mid c)$ and $p(c)$ which have been generated in the training phase. The classifier of naïve Bayes is shown as follows:

$$C_{\mathrm{NB}}(\mathbf{t}) = \arg\max_{c \in C} \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(x_i \mid c). \quad (3)$$

As it was aforementioned, naïve Bayes assumption conflicts with most real world applications (note that it is rare that attributes in the same data set do not have any relationships between each other). Therefore, many researchers provide proposals to relax naïve Bayes assumption effectively, which have been reviewed in Section 2.

In this paper, we focus on attribute weighting methods combined with kernel density estimation technique which is applied to naïve Bayes learner in order to relax conditional independence assumption.

3.2. Naïve Bayes Attribute Weighting. Generally, naïve Bayes attribute weighting scheme can be formulated in several forms. Firstly, the weight to each attribute is defined as follows:

$$\widehat{p}(c \mid \mathbf{x}) = \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(x_i \mid c)^{w_i}. \quad (4)$$

If the weight depends on attribute and class, the corresponding formula is as follows:

$$\widehat{p}(c \mid \mathbf{x}) = \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(x_i \mid c)^{w_{ci}}. \quad (5)$$

The following formula is used for the case when the weight depends on attribute value:

$$\widehat{p}(c \mid \mathbf{x}) = \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(x_i \mid c)^{w_{i,x_i}}. \quad (6)$$

Referring back to (4), when $\forall w_i = w$, the formula is shown as follows:

$$\widehat{p}(c \mid \mathbf{x}) = \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(x_i \mid c)^{w}. \quad (7)$$

It is worthwhile to mention that (7) is considered as a special case of naïve Bayes classifier, where each attribute A_i has the same weight $\forall w_i = w = 1$. In other words, naïve Bayes classifier ignores the importance of attributes. From information theoretic perspective, naïve Bayes classifier abandons the chance of digging more information from $\mathscr{D}$ to reduce the entropy of class. This is one of the reasons why attribute weighting method provides more accuracy of classification result than naïve Bayes classifier.

In our approach, we follow (4) that assigns w_i which corresponds to the attribute A_i. But instead of using w_i as an exponential parameter, we incorporate w_i into $\widehat{p}(x_i \mid c)$ so that it works in a more generalized form. The weight in our paper works in the kernel, as is shown in (13), described in Section 4.1.

Based on information theoretic perspective, attribute weighting method tries to find out which attribute will give more information for classification than other attributes. If an attribute A_i in data set $\mathscr{D}$ provides more information to reduce the entropy of class label C than other attributes, then A_i will be assigned with a higher weight.

3.3. Kernel Density Estimation for Naïve Bayes Categorical Attributes. In naïve Bayes learner, which has been discussed in Section 3.1, the likelihood $p(a_i^{(j)} \mid c)$ is often estimated by $\overline{f}_c(a_i^{(j)})$, the frequency of $a_i^{(j)}$ given c; note that $a_i^{(j)}$ is the value of attribute i at the jth instance in a data set $\mathscr{D}$. From a statistical perspective, a nonsmooth estimator has the least sample bias, but it also has a large estimation variance [2, 13] at the same time. Aitchison and Aitken [14] have proposed a kernel function and Chen and Wang [2] have proposed a variant of smooth kernel function alternating the frequency. The definition of their kernel function in [2] is as follows.

Given a test instance $\mathbf{t} = \langle t_1, \ldots, t_m \rangle$ where t_m is an attribute value of the attribute m in the test instance,

$$\kappa\left(t_i, a_i^{(j)}, \lambda_{ci}\right) = \begin{cases} 1 - \dfrac{|A_i| - 1}{|A_i|}\lambda_{ci}: & t_i = a_i^{(j)} \\ \dfrac{1}{|A_i|}\lambda_{ci}: & t_i \neq a_i^{(j)}. \end{cases} \quad (8)$$

Note that $\kappa(t_i, a_i^{(j)}, \lambda_{ci})$ is a kernel function for A_i given c, which may become an indicator if $\lambda_{ci} = 0$. $\lambda_{ci} (= w_{ci} \cdot \lambda_c)$ is the bandwidth such that $\lambda_c = 1/\sqrt{n_c}$, $\lambda_{ci} \in [0, 1]$, and n_c is a number of instances in $\mathscr{D}$ given c.

In [2], they have used (8) to estimate $p(t_i \mid c)$ as follows:

$$\begin{aligned} \widehat{p}(t_i \mid c, \lambda_{ci}) &= \frac{1}{n_c} \sum_{j=1}^{n_c} \kappa\left(t_i, a_i^{(j)}, \lambda_{ci}\right) \\ &= \overline{f}_c(t_i) + \left(\frac{1}{|A_i|} - \overline{f}_c(t_i)\right)\lambda_{ci}, \end{aligned} \quad (9)$$

where we use $p(t_i \mid c, \lambda_{ci})$ instead of $p(t_i \mid c)$. (Note that $p(c)$ is still estimated by frequency.) They minimize the cost function to take out a series w_{ci} for each A_i in class c. The cost function is defined as follows:

$$J(w_c) = \sum_{i=1}^{m} \sum_{a_i}^{A_i} \left(\widehat{p}(a_i \mid c) - \widehat{p}(a_i \mid c, w_{ci})\right)^2. \quad (10)$$

Hence, the classifier is formulated as follows:

$$C(\mathbf{t}) = \arg\max_{c \in C} \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(t_i \mid c, \lambda_{ci}). \quad (11)$$

4. AW-SKDE Framework and AW-SKDE^{MI} Learner

As mentioned earlier, in this section, we propose an attribute weighting framework working on the categorical attribute called *Attribute Weighting with Smooth Kernel Density Estimations*, simply AW-SKDE. Based on the AW-SKDE framework, a learner named AW-SKDE^{MI} is proposed, in which mutual information attribute weighting is applied.

4.1. AW-SKDE Framework. In (8), we pose an assumption that if a certain attribute A_i has more importance for classification given class label, in other words, A_i can provide more information to reduce the indeterminacy of class c, then the value of $p(a_i^{(j)} \mid c)$ should be more close to $\overline{f}_c(a_i^{(j)})$; otherwise, if A_i is less meaningful for classification, then $p(a_i^{(j)} \mid c)$ should be more close to $1/|A_i|$. We let the bandwidth $\lambda_{ci} = (1-w_i)^2 \times \lambda_c$, where $w_i \in [0,1]$, $\lambda_c = 1/\sqrt{n_c}$, and n_c is the number of instances labeled $C = c$. The variation of (8) according to our proposal is as follows:

$$\kappa\left(t_i, a_i^{(j)}, w_i\right) = \begin{cases} 1 - \dfrac{|A_i| - 1}{|A_i|}(1 - w_i)^2 \lambda_c: & t_i = a_i^{(j)} \\ \dfrac{1}{|A_i|}(1 - w_i)^2 \lambda_c: & t_i \neq a_i^{(j)}. \end{cases} \quad (12)$$

The estimation $p(t_i \mid c, w_i)$ of probability of $p(t_i \mid c)$ is described as follows:

$$\widehat{p}(t_i \mid c, w_i) = \frac{1}{n_c} \sum_{j=1}^{n_c} \kappa\left(t_i, a_i^{(j)}, w_i\right) = \overline{f}_c(t_i) + \left(\frac{1}{|A_i|} - \overline{f}_c(t_i)\right) \frac{(1 - w_i)^2}{\sqrt{n_c}}. \quad (13)$$

Hence, AW-SKDE framework is defined as follows:

$$C_{\text{AW-SKDE}}(\mathbf{t}) = \arg\max_{c \in C} p(c) \prod_{i=1}^{m} \widehat{p}(t_i \mid c, w_i). \quad (14)$$

The AW-SKDE framework incorporates a smooth kernel to make the probabilistic estimation of likelihood dominated by the weights. This enables natural combination of kernel methods and weighting methods. After setting up the kernel, we can generate a set of weights estimated by various methods cooperating with the kernel.

Table 1: Time complexity (m: the number of attributes, n: the number of training examples, k: the number of classes, and v: the average number of values for an attribute).

Algorithm	Training time	Classification time
NB	$O(mn)$	$O(km)$
AW-SKDE^{MI}	$O(mnk + m^2 + mv)$	$O(km)$

4.2. AW-SKDE^{MI} Learner. Our approach generates a set of attribute weights $w_i \in [0,1]$ by employing mutual information between A_i and C. It makes sense that if one attribute has more mutual information with class label, the attribute will provide more classification ability than other attributes and therefore should be assigned a larger weight.

The average weight w_{i_avg} of each attribute A_i is defined as follows:

$$w_{i\_\text{avg}} = \frac{I(A_i; C)}{\sum_{i=1}^{m} I(A_i; C)}, \quad (15)$$

where the definition of $I(A_i; C)$ is as follows:

$$I(A_i; C) = \sum_{i,c} \widehat{p}(a_i \mid c)\, \widehat{p}(c) \log \frac{\widehat{p}(a_i \mid c)}{\widehat{p}(a_i)}. \quad (16)$$

We also incorporate split information used in C4.5 [15] with w_{i_split} into our weighting scheme to avoid choosing the attributes with lots of values. The split information for each A_i is defined as follows where $a_i^{(j)}$ is the value of attribute A_i at jth instance (as described in Notations section):

$$A_{i\_\text{split}} = -\sum_{a_i \in A_i} \widehat{p}(a_i) \log \widehat{p}(a_i). \quad (17)$$

Now, the weight of A_i is defined as follows:

$$w_i = \frac{w_{i\_\text{avg}} / A_{i\_\text{split}}}{\sum_{i=1}^{m} \left(w_{i\_\text{avg}} / A_{i\_\text{split}}\right)}. \quad (18)$$

We feed AW-SKDE^{MI} with a training data set $\mathcal{D}$. In the training stage, we generate w_{i_avg}, A_{i_split}, and w_i out for each A_i. In the classification phase, we give a test instance $\mathbf{t}$; then AW-SKDE^{MI} classifier is formed; a prediction of class is outputted finally. The learning algorithm of AW-SKDE^{MI} is described in Algorithm 1.

During the training phase, AW-SKDE^{MI} only needs to construct conditional probability tables (CPT), which are the tables that contain joint probabilities of attributes and a class label. In terms of time complexity, the calculation of $I(A_i; C)$, w_{i_avg}, A_{i_split}, and w_i requires $O(mnk)$, $O(m^2)$, $O(mv)$, and $O(m^2)$, respectively. Therefore, the total time complexity is $O(mnk + m^2 + mv)$ in the training phase. In the classification phase, the algorithm time complexity is $O(km)$. We summarize the time complexity of AW-SKDE^{MI} and naïve Bayes in Table 1.

Here, we also present a framework named *Attribute Weighting with Light Smooth Kernel Density Estimation*, simply AW-LSKDE, which does not consider the bandwidth.

AW-SKDE^{MI}:
Input: training data set $\mathscr{D}$ and a test instance **t**
Output: the class estimation of **t**
Training phase:
begin
(1) **for** each a_i and c **in** A_i and C:
estimate $p(a_i, c)$, $p(c)$, $p(a_i \mid c)$, $p(a_i)$ and $|A_i|$.
(2) **for** each A_i and C:

$$I(A_i; C) = \sum_{i,c} \widehat{p}(a_i \mid c)\, \widehat{p}(c) \log \frac{\widehat{p}(a_i \mid c)}{\widehat{p}(a_i)}$$

(3) **for** each A_i:

(a) $w_{i\_\text{avg}} = \dfrac{I(A_i; C)}{\sum_{i=1}^{m} I(A_i; C)}$

(b) $A_{i\_\text{split}} = -\sum_{a_i \in A_i} \widehat{p}(a_i) \log \widehat{p}(a_i)$

(c) $w_i = \dfrac{w_{i\_\text{avg}} / A_{i\_\text{split}}}{\sum_{i=1}^{m} \left(w_{i\_\text{avg}} / A_{i\_\text{split}} \right)}$

end.
Classification phase:
begin
(1) for each dimension of test instance **t** and C:

$$\widehat{p}(t_i \mid c, w_i) = \overline{f}_c(t_i) + \left(\frac{1}{|A_i|} - \overline{f}_c(t_i) \right) \frac{(1 - w_i)^2}{\sqrt{n_c}}$$

(2) Output the class value

$$C_{\text{AW-SKDE}^{\text{MI}}}(\mathbf{t}) = \arg\max_{c \in C} \widehat{p}(c) \prod_{i=1}^{m} \widehat{p}(t_i \mid c, w_i)$$

end.

Algorithm 1: Mutual information based Attribute Weighting with Smooth Kernel Density Estimation (AW-SKDE^{MI}) algorithm.

AW-LSKDE can be regarded as a simple version of AW-SKDE. According to (8), we directly let $\lambda_{ci} = 1 - w_i$ where $w_i \in [0, 1]$. Hence, the kernel $\kappa(t_i, a_i^{(j)}, \lambda_{ci})$ is changed to $\kappa(t_i, a_i^{(j)}, w_i)$ which is defined as follows:

$$\kappa\left(t_i, a_i^{(j)}, w_i\right) = \begin{cases} \dfrac{1}{|A_i|} + \dfrac{|A_i| - 1}{|A_i|} w_i : & t_i = a_i^{(j)} \\ \dfrac{1}{|A_i|}(1 - w_i) : & t_i \neq a_i^{(j)}. \end{cases} \tag{19}$$

The estimation $p(t_i \mid c, w_i)$ is described as follows:

$$\begin{aligned} \widehat{p}(t_i \mid c, w_i) &= \frac{1}{n_c} \sum_{j=1}^{n_c} \kappa\left(t_i, a_i^{(j)}, w_i\right) \\ &= \frac{1}{|A_i|} + w_i \left(\overline{f}_c(t_i) - \frac{1}{|A_i|} \right). \end{aligned} \tag{20}$$

We also build an attribute weighting naïve Bayes learner with mutual information metric based on this AW-LSKDE framework, called AW-LSKDE^{MI}. The method of obtaining the weight of A_i is the same as that of AW-SKDE^{MI} learner. Unfortunately, AW-LSKDE framework does not give us encouraging results. The experimental results of AW-LSKDE^{MI} learner can be found in Table 3 with analysis of the results.

Table 2: Description of data sets used in the experiments.

Data set	Instances	Attributes	Classes	Missing	Numeric
Anneal	898	39	6	Y	Y
Balance-scale	625	5	3	N	Y
Breast-cancer	286	10	2	Y	N
Breast-w	699	10	2	Y	N
Colic	368	23	2	Y	Y
Credit-a	690	16	2	Y	Y
Dermatology	366	35	6	Y	Y
Glass	214	10	7	N	Y
Heart-statlog	250	14	2	N	Y
Hepatitis	155	20	2	Y	Y
Ionosphere	351	35	3	N	Y
Lymph	148	19	4	N	Y
Primary-tumor	339	18	21	Y	N
Segment	2310	20	7	N	Y
Sick	3772	30	2	Y	Y
Vehicle	846	19	4	N	Y
Vote	435	17	2	Y	N

Table 3: Experimental results in terms of classifiers' accuracy. Note that accuracies are estimated using 10-fold cross-validation with 95% confidence interval.

Data set	Naïve Bayes	AW-SKDE^{MI}	AW-LSKDE^{MI}
Anneal	93.99 ± 1.55	**96.55 ± 1.19**	76.17 ± 2.79
Balance-scale	**91.36 ± 2.20**	**91.36 ± 2.20**	89.6 ± 2.39
Breast-cancer	71.68 ± 5.22	**72.38 ± 5.18**	70.28 ± 5.30
Breast-w	**97.28 ± 1.21**	96.85 ± 1.29	88.41 ± 2.37
Colic	**82.07 ± 3.92**	81.79 ± 3.94	79.62 ± 4.12
Credit-a	85.94 ± 2.59	**86.09 ± 2.58**	83.62 ± 2.76
Dermatology	**97.81 ± 1.50**	**97.81 ± 1.50**	75.14 ± 4.43
Glass	**77.10 ± 5.63**	76.64 ± 5.67	62.62 ± 6.48
Heart-statlog	**83.70 ± 4.58**	**83.70 ± 4.58**	77.78 ± 5.15
Hepatitis	**89.03 ± 4.92**	**89.03 ± 4.92**	79.35 ± 6.37
Ionosphere	**92.02 ± 2.83**	91.45 ± 2.93	86.61 ± 3.56
Lymph	**85.81 ± 5.62**	**85.81 ± 5.62**	76.35 ± 6.85
Primary-tumor	**50.15 ± 5.32**	49.85 ± 5.32	24.78 ± 4.60
Segment	**89.09 ± 1.27**	88.70 ± 1.29	75.28 ± 1.76
Sick	**97.48 ± 0.50**	97.03 ± 0.54	93.88 ± 0.76
Vehicle	66.67 ± 3.18	**66.90 ± 3.17**	61.82 ± 3.27
Vote	90.11 ± 2.81	89.89 ± 2.83	**91.49 ± 2.62**
Average	84.78 ± 3.23	**84.81 ± 3.22**	76.05 ± 3.86

5. Experimental Results

In order to compare AW-ESKD^{MI}, AW-LSKDE^{MI}, and naïve Bayes in terms of classification accuracy, we have conducted experiments on UCI Machine Learning Repository Benchmark Data Sets [16]. The UCI benchmark data sets that we have used are shown in Table 2. Note that we have conducted preprocessing to each data set: removing missing values and discretizing numerical attribute values.

In the implementation of our algorithm, all the probabilities including $\widehat{p}(C = c)$ and $\widehat{p}(A_i = a_i,\ C = c)$ are estimated via Laplacian smoothing which is shown as follows:

$$\widehat{p}\,(C = c) = \frac{\text{count}\,(c) + 1}{n + |C|}$$
$$\widehat{p}\,(A_i = a_i, C = c) = \frac{\text{count}\,(a_i, c) + 1}{n_i + |A_i| \times |C|}, \tag{21}$$

where n is the number of training examples for which the class value is known; n_i is the number of training examples for which both attribute i and the class are known. The count(•) is the count value of •. The quotient of $\widehat{p}(A_i = a_i,\ C = c)$ as the dividend and $\widehat{p}(C = c)$ as the divisor result in conditional probability $\widehat{p}(A_i = a_i \mid C = c)$.

To compare the performance of the algorithms, we have adapted t-test with 10-fold cross-validation. We have conducted the experiments by applying our algorithm and standard naïve Bayes on the same training data sets as well as the same test data sets. The performance of the algorithm is evaluated through classification accuracy.

Table 3 shows the comparison of accuracies among standard naïve Bayes, AW-SKDE$^{\text{MI}}$ learner, and AW-LSKDE$^{\text{MI}}$ learner.

It can be seen that AW-SKDE$^{\text{MI}}$ learner shows four better results, six even results, and seven worse results than naïve Bayes within seventeen UCI data sets. AW-LSKDE$^{\text{MI}}$ learner only has one better result. Note that accuracies are estimated using 10-fold cross-validation with 95% confidence interval. AW-SKDE$^{\text{MI}}$ has a significant performance in the anneal data set and the mean accuracy of the AW-SKDE$^{\text{MI}}$ learner is 84.81 which is better than that of naïve Bayes' 84.78. This experimental result can prove that our new attribute weighting model AW-SKDE$^{\text{MI}}$ is efficient and effective. AW-LSKDE$^{\text{MI}}$ learner has performed poorly due to the ignorance of bandwidth parameters in the kernel methods which results in a relatively larger bias.

6. Conclusions and Future Work

In this paper, a novel attribute weighting framework called *Attribute Weighting with Smooth Kernel Density Estimations*, simply AW-SKDE framework, has been proposed. The AW-SKDE framework enables the estimation of likelihood to be dominated by attribute weights. Based on AW-SKDE, AW-SKDE$^{\text{MI}}$ has been proposed to exploit mutual information. We have conducted experiments on seventeen UCI benchmark data sets and made a comparison of accuracy among the standard NB, AW-SKDE$^{\text{MI}}$, and AW-LSKDE$^{\text{MI}}$. The experimental result proves that our new learner, AW-SKDE$^{\text{MI}}$, is efficient and effective. Also, due to the relatively larger bias in the algorithm of AW-LSKDE$^{\text{MI}}$, it has underperformed.

Even though AW-SKDE$^{\text{MI}}$ shows comparable results, as shown in Table 3, it does not quite outperform naïve Bayes. In the future work, we plan to improve AW-SKDE framework and investigate more effective attribute weighting methods instead of the weight measurement method with mutual information between attributes and class label.

Notations

A_i:	The ith attribute in data set
$\lvert A_i\rvert$:	The cardinality of attribute i
$a_i^{(j)}$:	The value of A_i at jth instance
$\mathscr{D} = \{\mathbf{x}^{(\mathbf{1})}, \ldots, \mathbf{x}^{(\mathbf{n})}\}$:	Training data set consists of n instances
$\mathbf{x} = \langle x_1, \ldots, x_m\rangle$:	An instance, m-dimensional vector, $\mathbf{x} \in \mathscr{D}$
C:	Class label, $C = \{c_1, \ldots, c_{\lvert C\rvert}\}$
c:	An element of C, $c \in C$
$\mathbf{t} = \langle t_1, \ldots, t_m\rangle$:	A test instance, m-dimensional vector
$P(e)$:	The unconditioned probability of event e
$P(e \mid g)$:	The conditional probability of e given g
$\widehat{P}(\bullet)$:	An estimation of $P(\bullet)$
$\overline{f}_c(\cdot)$:	The frequency of $\cdot$ given c
$w_i \in [0, 1]$:	The weight-value of attribute A_i
$I(A_i; C)$:	The mutual information between A_i and C.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. NRF-2013R1A1A2013401).

References

[1] D. D. Lewis, "Naïve (Bayes) at forty: the independence assumption in information retrieval," in *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21-23, 1998 Proceedings*, vol. 1398 of *Lecture Notes in Computer Science*, pp. 4–15, Springer, Berlin, Germany, 1998.

[2] L. Chen and S. Wang, "Semi-naïve Bayesian classification by weighted kernel density estimation," in *Proceedings of the 8th International Conference on Advanced Data Mining and Applications (ADMA '12)*, pp. 260–270, Springer, Nanjing, China, December 2012.

[3] L. Jiang, H. Zhang, and Z. Cai, "A novel bayes model: hidden naive bayes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1361–1371, 2009.

[4] D.-K. Kang and K. Sohn, "Learning decision trees with taxonomy of propositionalized attributes," *Pattern Recognition*, vol. 42, no. 1, pp. 84–92, 2009.

[5] D.-K. Kang and M.-J. Kim, "Propositionalized attribute taxonomies from data for data-driven construction of concise classifiers," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12739–12746, 2011.

[6] T.-T. Wong, "A hybrid discretization method for naïve Bayesian classifiers," *Pattern Recognition*, vol. 45, no. 6, pp. 2321–2325, 2012.

[7] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive Bayes: aggregating one-dependence estimators," *Machine Learning*, vol. 58, no. 1, pp. 5–24, 2005.

[8] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, "Alleviating naïve Bayes attribute independence assumption by attribute weighting," *Journal of Machine Learning Research*, vol. 14, pp. 1947–1988, 2013.

[9] L. Chen and S. Wang, "Automated feature weighting in naïve Bayes for high-dimensional data classification," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, pp. 1243–1252, ACM, Maui, Hawaii, USA, October-November 2012.

[10] C.-H. Lee, F. Gutierrez, and D. Dou, "Calculating feature weights in naive Bayes with Kullback-Leibler measure," in *Proceedings of the IEEE 11th International Conference on Data Mining (ICDM '11)*, pp. 1146–1151, Washington, DC, USA, December 2011.

[11] J. Wu and Z. Cai, "Learning averaged one-dependence estimators by attribute weighting," *Journal of Information & Computational Science*, vol. 8, no. 7, pp. 1063–1073, 2011.

[12] K. Omura, M. Kudo, T. Endo, and T. Murai, "Weighted naïve Bayes classifier on categorical features," in *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA '12)*, pp. 865–870, IEEE, Kochi, India, November 2012.

[13] Q. Li and J. S. Racine, *Nonparametric Econometrics: Theory and Practice*, Princeton University Press, Princeton, NJ, USA, 2006.

[14] J. Aitchison and C. G. Aitken, "Multivariate binary discrimination by the kernel method," *Biometrika*, vol. 63, no. 3, pp. 413–420, 1976.

[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1993.

[16] K. Bache and M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml/.