Appl Intell (2016) 44:611–620
DOI 10.1007/s10489-015-0719-1

Experimental analysis of naïve Bayes classifier based on an attribute weighting framework with smooth kernel density estimations

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

Published online: 22 October 2015

Abstract Naïve Bayes learners are widely used, efficient, and effective supervised learning methods for labeled datasets in noisy environments. It has been shown that naïve Bayes learners produce reasonable performance compared with other machine learning algorithms. However, the conditional independence assumption of naïve Bayes learning imposes restrictions on the handling of real-world data. To relax the independence assumption, we propose a smooth kernel to augment weights for the likelihood estimation. We then select an attribute weighting method that uses the mutual information metric to cooperate with the proposed framework. A series of experiments are conducted on 17 UCI benchmark datasets to compare the accuracy of the proposed learner against that of other methods that employ a relaxed conditional independence assumption. The results demonstrate the effectiveness and efficiency of our proposed learning algorithm. The overall results also indicate the superiority of attribute-weighting methods over those that attempt to determine the structure of the network.

Keywords Attribute weighting · Mutual information · Smooth Kernel density estimation framework · Bayesian models

✉ Dae-Ki Kang
dkkang@dongseo.ac.kr

Zhong-Liang Xiang
ugoood@163.com

[1] Department of Computer Software Institute, Weifang University of Science & Technology, Shouguang 262-700, Shandong, China

[2] Department of Computer and Information Engineering, Dongseo University, 47, Churye-Ro, Sasang-Gu, Busan 47011, Republic of Korea

1 Introduction

Naïve Bayes classification is a supervised learning method based on Bayes rule of probability theory. The classification uses labeled training examples, and is driven by the strong assumption that all attributes in the training examples are independent of one another, given the class labels. This is known as the naïve Bayes assumption or naïve Bayes conditional independence assumption. Naïve Bayes classifiers exhibit high performance and rapid classification speed, and their effectiveness has been demonstrated using huge training instances with multiple attributes. This strong performance is mainly because of the independence assumption [11].

In practice, classification performance is affected by the attribute independence assumption, which is often violated in real-world data. However, the advantages of efficiency and simplicity, both stemming from the attribute independence assumption, have led many researchers to propose effective methods to further improve the performance of naïve Bayes classifiers by weakening the attribute independence without neglecting its advantages. We categorize and briefly review some typical methods of relaxing the naïve Bayes assumption in Section 3. However, attribute weighting methods have received relatively little attention among techniques to improve naïve Bayes classification, particularly when attribute weighting is combined with a kernel method in a reasonable manner.

Although [4] proposed an 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 framework with a kernel method, which enables the weights

embedded in the kernel to have relatively interpretable meaning. Thus, we can flexibly choose different metrics and methods to measure the weights based on our attribute weighting framework.

The contributions of this paper are threefold:

- We have briefly conducted a survey on ways to improve naïve Bayes classification, focusing on naïve Bayes weighting methods.
- We propose a novel attribute weighting framework called Attribute Weighting with Smooth Kernel Density Estimation (AW-SKDE). The AW-SKDE framework employs a smooth kernel whereby the weights dominate the probabilistic estimation of likelihood. This enables kernel methods to be combined with weighting methods. After setting up the kernel, we generate a set of weights directly via various methods that cooperate with the kernel.
- Under the AW-SKDE framework, we propose a learner called AW-SKDE^{MI}. This uses the mutual information criterion to measure the dependency between an attribute and its class label.

Our experimental results show that the mutual information criterion based on the AW-SKDE framework exhibits superior performance over standard naïve Bayes classifiers, and is comparable to other approaches that have included the relaxation of the conditional independence assumption.

The remainder of this paper is organized as follows: we first conduct a brief survey on methods to improve naïve Bayes classification in Section 2. In Section 3, we introduce the background to our study. Section 4 describes the proposed attribute weighting framework based on kernel density estimation. We then propose a method that uses the mutual information criterion for attribute weighting based on our proposed framework. In Section 5, we describe a series of experiments and discuss the results in detail. Finally, we draw conclusions from our study and describe avenues for future research in Section 6.

2 Related work

In recent years, a number of methods that weaken the attribute independence assumption of naïve Bayes learning have been proposed. [7] conducted a survey on improved naïve Bayes methods. Such methods can be divided into five main categories: data expansion, structure extension, attribute weighting, feature selection, and local learning. We now briefly review these categories.

For data expansion, [9] presented an algorithm called the propositionalized attribute taxonomy learner (PAT-learner). The PAT-learner first disassembles the training dataset into small pieces with attribute values, then rebuilds a new dataset called the PAT-Table using the divergence between the distribution of class labels associated with the corresponding attributes and the disassembled dataset. [8] also proposed a Bayes learner based on the PAT-learner, called propositionalized attribute taxonomy guided naïve Bayes learner (PAT-NBL). They used the propositionalized dataset and PAT-Table generated by the PAT-learner to build naïve Bayes classifiers.

[16] focused on the discretization of attributes to improve naïve Bayes classification. Wong proposed a hybrid method for continuous attributes, and mentioned that the discretization of continuous attributes in a dataset using different methods can improve the performance of naïve Bayes learning. Additionally, [16] provided a nonparametric measure to evaluate the level of dependence between a continuous attribute and the class.

In terms of structure extension, [15] proposed a system of aggregating one-dependence estimators (AODE). Under AODE, the conditional probability of the test instances given the class is tuned by one attribute value that occurs in the test instances. After the training stage, AODE outputs an average one-dependence estimator. AODE is considered a lazy method of extending the structure of a Bayesian network. [7] proposed a hidden naïve Bayes (HNB) learner, which is also a type of structure extension method.

Some approaches have attempted to discern the network structure. [5] proposed a Bayesian scoring metric and a heuristic search algorithm named K2. K2 learns the network structure using a greedy search by maximizing the tradeoff metric between network complexity and accuracy over the training data. [6] developed Tree Augmented Naïve Bayes (TAN) learning, in which each attribute has a single class variable and at most one other attribute as parents. With this approach, TAN augments the maximum weight spanning tree of the naïve Bayes learner.

There are two main approaches for determining the attribute weights. The first method constructs a function with the attribute weight parameters, and allows this function to fit itself to the training data by estimating the weights. [18] proposed a weighted naïve Bayes algorithm, called weighting to alleviate the naïve Bayes independence assumption (WANBIA). Based on the WANBIA framework, the authors described two methods to obtain the attribute weights: WANBIA^{CLL}, which maximizes the conditional log-likelihood function, and WANBIA^{MSE}, which minimizes the mean squared error function.

[4] also reported an algorithm to minimize the mean squared error function in order to obtain the attribute weights. In another paper, [3] developed a method called subspace weighting naïve Bayes (SWNB), which can deal with high-dimensional data. Using the local

feature-weighting technique, SWNB has the ability to describe different contributions of attributes in the training dataset, and outputs an optimal set of attribute weights fitting a Logit normal a priori distribution.

There are many other techniques for attribute weighting. For example, weights can be directly obtained by measuring the relationship among the attributes or the relationship between the attributes and class labels by some given metric, or measured by the Gain Ratio method [14]. [10] calculated the attribute weights via the Kullback–Leibler divergence between the attributes and class labels. [17] proposed the decision tree-based attribute weighted AODE (DTWAODE) method. DTWAODE generates a set of attribute weights directly, and the weights decrease according to the attribute depth in the decision tree. [13] developed the confidence weight for naïve Bayes method, whereby the confidence weight is derived from the probabilities of the majority class in the training dataset.

3 Background

In this section, we explain the concepts behind the machine learning methodologies used in this paper, including the 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 Table 1.

3.1 Naïve Bayes classifier

In a supervised learning scenario, consider a training dataset $\mathcal{D} = \left\{\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(n)}\right\}$ composed of n instances, where each instance $\mathrm{x} = \langle x_1, \ldots, x_m\rangle \in \mathcal{D}$ (m-dimensional vector) is labeled with some class label $c \in C$. For the posterior probability of c given x, we have

$$p(c|\mathrm{x}) = \frac{p(\mathrm{x}|c) \cdot p(c)}{p(\mathrm{x})} \propto p(\mathrm{x}|c) \tag{1}$$

However, in practice, the likelihood $p(\mathrm{x}|c)$ cannot be directly estimated from $\mathcal{D}$ because of insufficient data. Naïve Bayes learning uses the attribute independence assumption to alleviate this problem. From this assumption, $p(\mathrm{x}|c)$ is given as follows:

$$p(\mathrm{x}|c) = \prod_{i=1}^{m} p(x_i|c) \tag{2}$$

In the training phase, only $p(x_i|c)$ 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 $p(x_i|c)$ and $p(c)$ respectively.

In the classification phase, if there is 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, the naïve Bayes classifier will output a class label prediction of $\mathbf{t}$ based on the frequency estimations of $p(x_i|c)$ and $p(c)$ generated in the training phase. The naïve Bayes classifier is then characterized as follows:

$$C_{NB}(\mathbf{t}) = \arg\max_{c \in C} \hat{p}(c) \prod_{i=1}^{m} \hat{p}(x_i|c) \tag{3}$$

As mentioned above, the naïve Bayes assumption conflicts with most real-world applications (note that it is rare that attributes in the same dataset have no relationship between one another). Therefore, many researchers have

Table 1 Symbols and their descriptions

Notation	Description
A_i	the i^{th} attribute in dataset
$\lvert A_i\rvert$	the cardinality of attribute i
$a_i^{(j)}$	the value of A_i at j^{th} instance
$\mathcal{D} = \left\{\mathbf{x^{(1)}}, \ldots, \mathbf{x^{(n)}}\right\}$	training dataset consists n instances
$\mathbf{x} = \langle x_1, \ldots, x_m\rangle$	an instance, m-dimensional vector, $\mathbf{x} \in \mathcal{D}$
C	class label, $C = \left\{c_1, \ldots, c_{\lvert C\rvert}\right\}$
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
$P(e\|g)$	the conditional probability of e given g
$\hat{P}(\bullet)$	an estimation of $P(\bullet)$
$\bar{f}_c(a_i)$	the frequency of a_i 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

attempted to effectively relax the naïve Bayes assumption, as reviewed in Section 2.

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

3.2 Naïve Bayes attribute weighting

Generally, the naïve Bayes attribute weighting scheme can be formulated in several ways. First, the weight of each attribute is defined as follows:

$$\hat{p}(c|\mathbf{x}) = \hat{p}(c)\prod_{i=1}^{m}\hat{p}(x_i|c)^{w_i} \tag{4}$$

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

$$\hat{p}(c|\mathbf{x}) = \hat{p}(c)\prod_{i=1}^{m}\hat{p}(x_i|c)^{w_{ci}} \tag{5}$$

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

$$\hat{p}(c|\mathbf{x}) = \hat{p}(c)\prod_{i=1}^{m}\hat{p}(x_i|c)^{w_{i,x_i}} \tag{6}$$

When $\forall w_i = w$, (4) becomes:

$$\hat{p}(c|\mathbf{x}) = \hat{p}(c)\prod_{i=1}^{m}\hat{p}(x_i|c)^{w} \tag{7}$$

It is worth mentioning that (7) is considered to be a special case of the naïve Bayes classifier in which each attribute A_i has the same weight, $w_i = w = 1\forall i$. In other words, this naïve Bayes classifier ignores the importance of the attributes. From an information-theoretic perspective, naïve Bayes classifiers abandon the possibility of obtaining more information from $\mathcal{D}$ to reduce the entropy of each class. This is one of the reasons why attribute weighting methods provide more accurate classification results than naïve Bayes classifiers.

In our approach, we use (4), which assigns w_i according to the attribute A_i. However, instead of using w_i as an exponential parameter, we incorporate w_i into $\hat{p}(x_i|c)$ so that it works in a more generalized form. In our method, the weights are applied in the kernel, as shown in (13). This is described in Section 4.1.

From an information-theoretic perspective, attribute weighting attempts to determine which attributes provide more information for classification than other attributes. If an attribute A_i in dataset $\mathcal{D}$ provides more information to reduce the entropy of class label C than other attributes, then A_i will be assigned a higher weight.

3.3 Kernel density estimation for naïve Bayes categorical attributes

In the naïve Bayes learner discussed in Section 3.1, the likelihood $p(a_i^{(j)}|c)$ is often estimated as $\bar{f}_c(a_i^{(j)})$, the frequency of $a_i^{(j)}$ given c. Note that $a_i^{(j)}$ is the value of attribute i in the j^{th} instance of dataset $\mathcal{D}$. From a statistical perspective, a non-smooth estimator has the least sample bias, but has a large estimation variance [4, 12]. [1] proposed a kernel function, and [4] presented a variant of a smooth kernel function with an alternating frequency. The kernel function defined in [4] is as follows:

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

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

Note that $\kappa\left(t_i, a_i^{(j)}, \lambda_{ci}\right)$ 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 $\lambda_c = \frac{1}{\sqrt{n_c}}$, $\lambda_{ci} \in [0, 1]$, and n_c is the number of instances in $\mathcal{D}$ given c.

In [4], (8) was used to estimate $p(t_i|c)$ as follows:

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

where $p(t_i|c, \lambda_{ci})$ is used instead of $p(t_i|c)$. (Note that $p(c)$ is still estimated from the frequency.) They minimized a cost function to estimate a series w_{ci} for each A_i in class c. The cost function is defined as follows:

$$J(w_c) = \sum_{i=1}^{m}\sum_{a_i}^{A_i}\left(\hat{p}(a_i|c) - \hat{p}(a_i|c, w_{ci})\right)^2 \tag{10}$$

Hence, the classifier can be formulated as follows:

$$C(\mathbf{t}) = \arg\max_{c\in C}\ \hat{p}(c)\prod_{i=1}^{m}\hat{p}(t_i|c, \lambda_{ci}) \tag{11}$$

4 AW-SKDE framework and AW-SKDE$^{\text{MI}}$ learner

In this section, we describe the proposed attribute weighting framework for categorical attributes, which we call *Attribute Weighting with Smooth Kernel Density Estimations*. Based on the AW-SKDE framework, we propose a learner named AW-SKDE$^{\text{MI}}$, in which mutual information is used to determine the attribute weights.

Fig. 1 Mutual Information based Attribute Weighting with Smooth Kernel Density Estimation (AW-SKDE$^{\mathrm{MI}}$) algorithm

AW-SKDE$^{\mathrm{MI}}$:
Input: training dataset $\mathcal{D}$ and a test instance $\mathbf{t}$
Output: the estimated class of $\mathbf{t}$

Training phase:
begin

1. **for** each a_i and c **in** A_i and C:
   estimate $p(a_i, c)$, $p(c)$, $p(a_i|c)$, $p(a_i)$ and $|A_i|$.
2. **for** each A_i and C:
   $I(A_i; C) = \sum_{i,c} \hat{p}(a_i|c)\hat{p}(c) \log \frac{\hat{p}(a_i|c)}{\hat{p}(a_i)}$
3. **for** each A_i:
   (a) $w_{i\_avg} = \frac{I(A_i;C)}{\sum_{i=1}^{m} I(A_i;C)}$
   (b) $A_{i\_split} = -\sum_{a_i \in A_i} \hat{p}(a_i) \log \hat{p}(a_i)$
   (c) $w_i = \frac{\frac{w_{i\_avg}}{A_{i\_split}}}{\sum_{i=1}^{m} \frac{w_{i\_avg}}{A_{i\_split}}}$

end.

Classification phase:
begin

1. for each dimension of test instance $\mathbf{t}$ and C:
   $\hat{p}(t_i|c, w_i) = \bar{f}_c(t_i) + \left(\frac{1}{|A_i|} - \bar{f}_c(t_i)\right) \frac{(1-w_i)^2}{\sqrt{n_c}}$
2. Output the class value
   $C_{AW-SKDE^{MI}}(\mathbf{t}) = \arg\max_{c \in C} \hat{p}(c) \prod_{i=1}^{m} \hat{p}(t_i|c, w_i)$

end.

4.1 AW-SKDE framework

In (8), we made the assumption that, if a certain attribute A_i has more importance for classification given the class label (in other words, A_i provides more information to reduce the indeterminacy of class c), then the value of $p(a_i^{(j)}|c)$ should be closer to $\bar{f}_c(a_i^{(j)})$; otherwise, if A_i is less meaningful for classification, then $p(a_i^{(j)}|c)$ should be closer to $\frac{1}{|A_i|}$. We let the bandwidth $\lambda_{ci} = (1-w_i)^2 \times \lambda_c$, where $w_i \in [0, 1]$, $\lambda_c = \frac{1}{\sqrt{n_c}}$, and n_c is the number of instances labeled $C = c$. In the proposed method, (8) is modified as follows:

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

The estimate $p(t_i|c, w_i)$ of probability $p(t_i|c)$ is described as follows:

$$\begin{aligned} \hat{p}(t_i|c, w_i) &= \frac{1}{n_c} \sum_{j=1}^{n_c} \kappa\left(t_i, a_i^{(j)}, w_i\right) \\ &= \bar{f}_c(t_i) + \left(\frac{1}{|A_i|} - \bar{f}_c(t_i)\right) \frac{(1-w_i)^2}{\sqrt{n_c}} \end{aligned} \tag{13}$$

Hence, the AW-SKDE framework can be defined as:

$$C_{AW-SKDE}(\mathbf{t}) = \arg\max_{c \in C} p(c) \prod_{i=1}^{m} \hat{p}(t_i|c, w_i) \tag{14}$$

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

4.2 AW-SKDE$^{\mathrm{MI}}$ learner

The AW-SKDE$^{\mathrm{MI}}$ learner generates a set of attribute weights $w_i \in [0, 1]$ by calculating the mutual information between A_i and C. If one attribute shares more mutual information with the class label, that attribute will provide more classification ability than other attributes, and should therefore be assigned a higher weight.

Table 2 Time complexity (m: number of attributes; n: number of training examples; k: number of classes; v: average number of values for an attribute)

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

Table 3 Description of datasets used in the experiments

Dataset	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

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

$$w_{i\_avg} = \frac{I(A_i; C)}{\sum_{i=1}^{m} I(A_i; C)} \tag{15}$$

where:

$$I(A_i; C) = \sum_{i,c} \hat{p}(a_i|c)\hat{p}(c) \log \frac{\hat{p}(a_i|c)}{\hat{p}(a_i)} \tag{16}$$

We also incorporate the split information used in C4.5 [14] into our weighting scheme with w_{i_split} to avoid choosing attributes with lots of values. The split information for each A_i is defined as follows:

$$A_{i\_split} = -\sum_{a_i \in A_i} \hat{p}(a_i) \log \hat{p}(a_i) \tag{17}$$

where $a_i^{(j)}$ is the value of attribute A_i in instance j^{th} (as described in Table 1). Now, the weight of A_i is defined as follows:

$$w_i = \frac{\frac{w_{i\_avg}}{A_{i\_split}}}{\sum_{i=1}^{m} \frac{w_{i\_avg}}{A_{i\_split}}} \tag{18}$$

We supply AW-SKDE$^{\text{MI}}$ with a training dataset $\mathcal{D}$. In the training stage, we generate w_{i_avg}, A_{i_split}, and w_i for each A_i. In the classification phase, given a test instance $\mathbf{t}$, the AW-SKDE$^{\text{MI}}$ classifier predicts the class label. The learning algorithm of AW-SKDE$^{\text{MI}}$ is illustrated in Fig. 1.

During the training phase, AW-SKDE$^{\text{MI}}$ only needs to construct conditional probability tables , which contain the 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 takes $O(mnk)$, $O(m^2)$, $O(mv)$, and $O(m^2)$ time, respectively. Therefore, the total time complexity of the training phase is $O(mnk+m^2+mv)$. In the classification phase, the algorithm's time complexity is $O(km)$. The time complexity of AW-SKDE$^{\text{MI}}$ and naïve Bayes classification is summarized in Table 2.

We now describe a framework named *Attribute Weighting with Light Smooth Kernel Density Estimation* (AW-LSKDE), which does not consider the bandwidth. AW-LSKDE can be regarded as a simplified version of AW-SKDE. According to (8), we directly set $\lambda_{ci} = 1-w_i$, where $w_i \in [0, 1]$. Hence, the kernel $\kappa\left(t_i, a_i^{(j)}, \lambda_{ci}\right)$ becomes $\kappa\left(t_i, a_i^{(j)}, w_i\right)$, which is defined as follows:

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

The estimate $p(t_i|c, w_i)$ is then:

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

We can also construct an attribute weighting naïve Bayes learner with the mutual information metric based on this AW-LSKDE framework. This is referred to as AW-LSKDE$^{\text{MI}}$. The weights of attributes A_i are obtained in the same manner as for the AW-SKDE$^{\text{MI}}$ learner. Unfortunately, the AW-LSKDE framework does not produce encouraging results. The experimental results for the AW-LSKDE$^{\text{MI}}$ learner are presented in Table 4.

5 Experimental results

To compare AW-SKDE$^{\text{MI}}$, AW-LSKDE$^{\text{MI}}$, and naïve Bayes learning in terms of classification accuracy, we conducted experiments on UCI Machine Learning Repository Benchmark Datasets [2]. The UCI benchmark datasets used in the experiments are listed in Table 3. Note that we have discretized the numerical attribute values for each dataset.

In the implementation of the proposed algorithm, all probabilities (including $\hat{p}(C = c)$, $\hat{p}(A_i = a_i, C = c)$) were estimated via the following Laplacian smoothing:

$$\hat{p}(C = c) = \frac{count(c) + 1}{n + |C|} \tag{21}$$

$$\hat{p}(A_i = a_i, C = c) = \frac{count(a_i, c) + 1}{n_i + |A_i| \times |C|} \tag{22}$$

where n is the number of training examples for which the class value is known, and n_i is the number of training

Table 4 Experimental results in terms of classifier accuracies. Accuracies were estimated using 10-fold cross-validation with 95% confidence interval

Dataset	Naïve Bayes	AW-SKDE (MI)	AW-LSKDE (MI)	PAT-NBL CMDL	PAT-NBL CAIC	PAT-NBL CLL	AODE	HNB	K2 BIC	TAN BIC
Anneal	93.99±1.55	96.55±1.19	76.17±2.79	89.87±1.97	89.87±1.97	89.87±1.97	97.77±0.97	**98.89±0.69**	94.54±1.49	**98.89±0.69**
Balance-scale	**91.36±2.20**	**91.36±2.20**	89.6±2.39	75.20±3.39	75.20±3.39	75.20±3.39	89.60±2.39	90.08±2.34	**91.36±2.20**	86.56±2.67
Breast-cancer	71.68±5.22	72.38±5.18	70.28±5.30	**73.08±5.14**	72.73±5.16	72.73±5.16	71.68±5.22	N/A	72.03±5.20	69.58±5.33
Breast-w	**97.28±1.21**	96.85±1.29	88.41±2.37	**97.28±1.21**	**97.28±1.21**	**97.28±1.21**	97.00±1.27	N/A	**97.28±1.21**	95.14±1.59
Colic	82.07±3.92	81.79±3.94	79.62±4.12	77.45±4.27	78.26±4.21	75.27±4.41	**84.24±3.72**	N/A	82.34±3.90	82.34±3.90
Credit-a	85.94±2.59	86.09±2.58	83.62±2.76	86.23±2.57	83.77±2.75	84.35±2.71	**87.54±2.46**	N/A	85.65±2.62	87.25±2.49
Dermatology	97.81±1.50	97.81±1.50	75.14±4.43	98.09±1.40	**98.36±1.30**	98.09±1.40	98.09±1.40	97.27±1.67	97.81±1.50	96.72±1.82
Glass	77.10±5.63	76.64±5.67	62.62±6.48	71.96±6.02	70.56±6.11	70.56±6.11	78.97±5.46	**79.91±5.37**	78.50±5.50	78.50±5.50
Heart-statlog	83.70±4.58	83.70±4.58	77.78±5.15	84.07±4.36	84.07±4.36	84.07±4.36	84.07±4.36	**84.44±4.32**	**84.44±4.32**	82.22±4.56
Hepatitis	89.03±4.92	89.03±4.92	79.35±6.37	84.52±5.70	85.16±5.60	83.87±5.79	**89.68±4.79**	N/A	87.10±5.28	87.74±5.16
Ionosphere	92.02±2.83	91.45±2.93	86.61±3.56	89.46±3.21	92.31±2.79	92.02±2.83	93.45±2.59	93.45±2.59	92.31±2.79	**94.02±2.48**
Lymph	85.81±5.62	85.81±5.62	76.35±6.85	54.73±8.02	54.73±8.02	54.73±8.02	85.81±5.62	83.78±5.94	**86.49±5.51**	82.43±6.13
Primary-tumor	**50.15±5.32**	49.85±5.32	24.78±4.60	24.78±4.60	24.78±4.60	47.20±5.31	49.56±5.32	N/A	46.90±5.31	45.13±5.30
Segment	89.09±1.27	88.70±1.29	75.28±1.76	91.04±1.16	88.83±1.28	88.83±1.28	92.68±1.06	**94.33±0.94**	89.39±1.26	94.11±0.96
Sick	97.48±0.50	97.03±0.54	93.88±0.76	93.08±0.81	93.11±0.81	93.11±0.81	97.51±0.50	N/A	97.32±0.52	**97.64±0.48**
Vehicle	66.67±3.18	66.90±3.17	61.82±3.27	62.29±3.27	59.34±3.31	61.35±3.28	73.52±2.97	**76.95±2.84**	66.43±3.18	76.83±2.84
Vote	90.11±2.81	89.89±2.83	91.49±2.62	88.51±3.00	88.74±2.97	88.51±3.00	89.89±2.83	N/A	90.11±2.80	**94.48±2.15**
Average	84.78±3.23	84.81±3.22	76.05±3.86	78.92±3.54	78.65±3.52	79.83±3.59	85.94±3.11	**88.79±2.97**	84.71±3.21	85.27±3.18

examples for which both the attribute i and the class are known. The function $count(\bullet)$ is the count value of $\bullet$. Dividing $\hat{p}(A_i = a_i, C = c)$ by $\hat{p}(C = c)$ gives the conditional probability $\hat{p}(A_i = a_i|C = c)$.

To compare the performance of the algorithms, we used an adapted t-test with 10-fold cross-validation. Using the same training datasets and test datasets, we conducted experiments on the proposed algorithms, standard naïve Bayes, PAT-NBL [8], AODE [15], HNB [7], K2 [5], and TAN [6]. The algorithms performance was evaluated in terms of the classification accuracy.

Table 4 compares the accuracy of standard naïve Bayes, AW-SKDE$^{\mathrm{MI}}$, AW-LSKDE$^{\mathrm{MI}}$, and the previously developed algorithms that use a relaxed conditional independence assumption. Mutual information (MI) was used for AW-SKDE and AW-LSKDE. The conditional minimum description length (CMDL), conditional Akaike information criterion (CAIC), and conditional log-likelihood (CLL) were used in PAT-NBL, and the Bayesian information criteria (BIC) was used in K2 and TAN. Some results are missing for the HNB algorithm, because HNB cannot handle datasets with missing values. Instead of using traditional methods to fill the missing values or treating each missing value as a new attribute value, we have simply omitted HNB from the experiments using datasets with missing values for fair comparison.

It can be seen that the AW-SKDE$^{\mathrm{MI}}$ learner produced four better results, six comparative results, and seven worse results than the naïve Bayes classifier. However, the AW-LSKDE$^{\mathrm{MI}}$ learner only outperformed the naïve Bayes learner on one dataset. Note that the accuracies were estimated using 10-fold cross-validation with a 95% confidence interval. The win/tie/lose results are summarized in Table 5. Note that the overall win/tie/lose record between naïve Bayes and our AW-SKDE$^{\mathrm{MI}}$ was 8/5/4. Although the naïve Bayes classifier achieved more wins than AW-SKDE$^{\mathrm{MI}}$, the overall average accuracy of AW-SKDE$^{\mathrm{MI}}$ (84.81±3.22) was higher than that of the naïve Bayes method (84.78±3.23).

These experimental results prove that our new attribute weighting model AW-SKDE$^{\mathrm{MI}}$ achieves comparable and sometimes better performance than the classical naïve Bayes method. AW-SKDE$^{\mathrm{MI}}$ exhibited better performance than PAT-NBL, with a win/tie/lose record of 11/0/6.

Unfortunately, AW-SKDE$^{\mathrm{MI}}$ does not outperform AODE and HNB. We have examined the results, and found that the training error of AW-SKDE$^{\mathrm{MI}}$ is usually higher than that of AODE and HNB. This indicates that AW-SKDE$^{\mathrm{MI}}$ suffers from over-fitting. Overcoming this problem will be considered in future research.

Compared with K2 and TAN, AW-SKDE$^{\mathrm{MI}}$ exhibits comparable and sometimes better performance, with win/tie/lose records of 6/2/9 and 8/0/9, respectively.

Table 5 Win/tie/lose results of standard naïve Bayes, PAT-NBL, AODE, HNB, K2, and TAN

Win/tie/lose	AW-SKDE (MI)	AW-LSKDE (MI)	PAT-NBL CMDL	PAT-NBL CAIC	PAT-NBL CLL	AODE	HNB	K2 BIC	TAN BIC
Naïve Bayes	8/5/4	16/0/1	11/1/5	12/1/4	12/2/3	4/2/11	3/0/6	5/4/8	8/0/9
AW-SKDE$^{\mathrm{MI}}$		16/0/1	11/0/6	11/0/6	11/0/6	3/2/12	3/0/6	6/2/9	8/0/9
AW-LSKDE$^{\mathrm{MI}}$			5/1/11	6/1/10	6/0/11	1/1/15	0/0/9	1/0/16	2/0/15
PAT-NBL$^{\mathrm{CMDL}}$				5/6/6	7/7/3	2/2/13	1/0/8	4/1/12	4/0/13
PAT-NBL$^{\mathrm{CAIC}}$					5/9/3	3/1/13	1/0/8	2/2/13	4/0/13
PAT-NBL$^{\mathrm{CLL}}$						2/2/13	1/0/8	3/1/13	5/0/12
AODE							2/1/6	11/0/6	11/0/6
HNB								5/1/3	7/1/1
K2$^{\mathrm{BIC}}$									7/2/8

The AW-LSKDE^{MI} learner performed poorly because of its ignorance of bandwidth parameters in the kernel methods, which results in a relatively large bias.

In the experiments we performed, it is interesting to note that attribute weighting methods (AW-SKDE, AODE, and HNB) were generally superior to network structure elicitation methods (K2 and TAN). This indicates that stressing significant attributes may be more important than eliciting the dependence relationship between attributes.

6 Conclusions and future work

In this paper, a novel attribute weighting framework called *Attribute Weighting with Smooth Kernel Density Estimations* has been proposed. The AW-SKDE framework enables the estimation of likelihood to be dominated by attribute weights. Based on the AW-SKDE framework, mutual information was exploited to give the AW-SKDE^{MI} classifier. We conducted experiments on seventeen UCI benchmark datasets, and compared the accuracy of the standard naïve Bayes learner, AW-SKDE^{MI}, AW-LSKDE^{MI}, PAT-NBL, AODE, HNB, K2, and TAN. The experimental results demonstrated that our new learner, AW-SKDE^{MI}, is as efficient and effective as naïve Bayes, and has comparable performance to K2 and TAN. However, the relatively large bias in the AW-LSKDE^{MI} algorithm resulted in poor performance.

Even though AW-SKDE^{MI} produced comparable results, as shown in Table 4, it did not completely outperform naïve Bayes. In future work, we plan to improve the AW-SKDE framework and investigate more effective attribute weighting methods to reduce over-fitting. We will also investigate attribute weighting methods other than the weight measurement method with mutual information between attributes and class labels.

Acknowledgments This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2012.

References

1. Aitchison J, Aitken CGG (1976) Multivariate binary discrimination by the kernel method. Biometrika 63(3):413–420
2. Bache K, Lichman M (2013) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
3. Chen L, Wang S (2012a) 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. ACM, New York, NY, USA, pp 1243–1252. ISBN 978-1-4503-1156-4
4. Chen L, Wang S (2012b) Semi-naïve Bayesian classification by weighted kernel density estimation. In: Proceedings of the 8th international conference on advanced data mining and applications (adma 2012). Springer, Nanjing, China, pp 260–270
5. Cooper G, Herskovits E (1992) A bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347
6. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2-3):131–163
7. Jiang L, Zhang H, Cai Z (2009) A novel Bayes model: Hidden naive Bayes. IEEE Trans Knowl Data Eng 21(10):1361–1371
8. Kang D-K, Kim M-J (2011) Propositionalized attribute taxonomies from data for data-driven construction of concise classifiers. Expert Systems with Applications 38(10):12739–12746
9. Kang D-K, Sohn K (2009) Learning decision trees with taxonomy of propositionalized attributes. Pattern Recogn 42(1):84–92
10. Lee C-H, Gutierrez F, Dou D (2011) Calculating feature weights in naïve Bayes with Kullback-Leibler measure. In: Proceedings of the 2011 ieee 11th international conference on data mining. Icdm '11. IEEE Computer Society, Washington, pp 1146–1151. ISBN 978-0-7695-4408-3
11. Lewis DD (1998) Naïve (Bayes) at forty: The independence assumption in information retrieval. In: Proceedings of the 10th european conference on machine learning. Ecml '98. Springer, London, pp 4–15. ISBN 3-540-64417-2
12. Li Q, Racine JS (2006) Nonparametric econometrics: Theory and practice. Princeton University Press
13. Omura K, Kudo M, Endo T, Murai T (2012) Weighted naïve Bayes classifier on categorical features. In: Abraham A, Zomaya AY, Ventura S, Yager RR, Snásel V, Muda AK, Samuel P (eds) Proceedings of the 12th international conference on intelligent systems design and applications (isda 2012). IEEE, India, pp 865–870. ISBN 978-1-4673-5117-1
14. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc, San Francisco
15. Webb GI, Boughton JR, Wang Z (2005) Not so naïve Bayes: Aggregating one-dependence estimators. Mach Learn 58(1):5–24
16. Wong T-T (2012) A hybrid discretization method for naïve Bayesian classifiers. Pattern Recogn 45(6):2321–2325
17. Wu J, Cai Z (2011) Learning averaged one-dependence estimators by attribute weighting. Journal of Information & Computational Science 8(7):1063–1073
18. Zaidi NA, Cerquides J, Carman MJ, Webb GI (2013) Alleviating naïve Bayes attribute independence assumption by attribute weighting. J Mach Learn Res 14:1947–1988

Zhong-Liang Xiang earned a Ph.D. in computer science from Dongseo University in 2015. He received a science master degree in computer science at Ocean University of China in 2010 and a Bachelor of Science (BS) degree in computer science at Mudanjiang Normal University in 2003. His research interests include machine learning, especially in nonparametric Bayesian models.

Xiang-Ru Yu received a science master degree in computer science at Ocean University of China in 2010 and a Bachelor of Science (BS) degree in computer science at Mudanjiang Normal University in 2003. Currently, she is a lecturer at Weifang University of Science and Technology. Her research interests include data mining and machine learning.

Dae-Ki Kang is a professor at Dongseo University in South Korea. He was a senior member of engineering staff at the attached Institute of Electronics and Telecommunications Research Institute in South Korea. He earned a Ph.D. in computer science from Iowa State University in 2006. His research interests include intrusion detection, security informatics, ontology learning, and relational learning. Prior to joining Iowa State, he worked at a Bay-area startup company and at the Electronics and Telecommunication Research Institute in South Korea. He received a science master degree in computer science at Sogang University in 1994 and a bachelor of engineering (BE) degree in computer science and engineering at Hanyang University in 1992.