Non-Naive Bayesian Classifiers for Classification Problems With Continuous Attributes

Xi-Zhao Wang, *Fellow, IEEE,* Yu-Lin He, *Student Member, IEEE,* and Debby D. Wang, *Student Member, IEEE*

***Abstract*—An important way to improve the performance of naive Bayesian classifiers (NBCs) is to remove or relax the fundamental assumption of independence among the attributes, which usually results in an estimation of joint probability density function (p.d.f.) instead of the estimation of marginal p.d.f. in the NBC design. This paper proposes a non-naive Bayesian classifier (NNBC) in which the independence assumption is removed and the marginal p.d.f. estimation is replaced by the joint p.d.f. estimation. A new technique of estimating the class-conditional p.d.f. based on the optimal bandwidth selection, which is the crucial part of the joint p.d.f. estimation, is applied in our NNBC. Three well-known indexes for measuring the performance of Bayesian classifiers, which are classification accuracy, area under receiver operating characteristic curve, and probability mean square error, are adopted to conduct a comparison among the four Bayesian models, i.e., normal naive Bayesian, flexible naive Bayesian (FNB), the homologous model of FNB (FNB_{ROT}), and our proposed NNBC. The comparative results show that NNBC is statistically superior to the other three models regarding the three indexes. And, in the comparison with support vector machine and four boosting-based classification methods, NNBC achieves a relatively favorable classification accuracy while significantly reducing the training time.**

***Index Terms*—Joint probability density estimation, kernel function, naive Bayesian classifier (NBC), optimal bandwidth, probability mean square error.**

I. Introduction

NAIVE BAYESIAN classifier (NBC for short) is a simple and but efficient probabilistic model based on the Bayesian theory [17] in the supervised classification problems. NBC can achieve better performances for a number of practical applications such as a medical diagnosis [23], text categorization [39], email filtering [37] and information retrieval [25]. In many applications, NBC demonstrates favorable performances than other learning models such as decision trees [45], [46] and neural networks [6], [24], [32]. In addition, NBC is considered adequate to classify the datasets with large number of variables and instances due to its simplicity, low computational complexity, and less memory requirement [8].

It is well acknowledged that NBC can be implemented efficiently on classification problems with nominal attributes. For tasks with continuous attributes, we usually have two handling strategies. One is the discretization and the other is the density estimation. The former has been widely studied in NBC (e.g., [10], [49], and [51]), with the latest observation shows that the combination of various discretization methods can result in an improved classification accuracy [48]. During the process of discretization for NBC, one of the most difficult problems is the zero-counts problem [8], [33], which can be effectively solved the Laplace correction strategy [22].

The density estimation strategy intends to find an underlying distribution for the continuous attributes, instead of calculating the necessary probabilities by counting the frequency of values and combinations of values from a given dataset. A key step of this strategy is to estimate the class-conditional probability density function (p.d.f.) from a given set of training data with class information. The following three kinds of methodologies for estimating the class-conditional p.d.f. can be read in literatures.

1) The normal method (also named normal naive Bayesian, simply NNB) [2], [30]. NNB assumes that the continuous attributes are generated by a single Gaussian distribution, whose mean and standard deviation can be straightly calculated from the training dataset. NNB is a simple and common technique with the advantages of fast training/testing and little memory requirements, while it is usually criticized for the performance when continuous attributes do not follow the Gaussian distribution. More recently, as an extension of NNB, a nonparametric version of NBC [41] focusing on the application of diagnosis of breast cancer is proposed.
2) The flexible naive Bayesian (simply FNB) [19]. To cope with the case of non-Gaussian distribution, John and Langley [19] proposed the FNB in which the Parzen window [31] is used to estimate the underlying class-conditional p.d.f.. Specifically, the FNB uses the superposition of many p.d.f.s of the normal distribution to fit the true p.d.f. of each continuous attribute.
3) The homologous model of FNB (FNB_{ROT}) [28]. FNB_{ROT} is designed to classify the unknown instance according to the discriminant the same as FNB does. The dif-

Manuscript received June 24, 2012; revised January 31, 2013; accepted January 31, 2013. Date of publication February 26, 2013; date of current version December 12, 2013. This work was supported in part by the National Natural Science Foundations of China under Grant 60903088 and Grant 61170040, by the Natural Science Foundation of Hebei Province under Grant F2011201063 and Grant F2012201023, by the Project of Hebei Education Department under Grant ZD2010139, and by the Project of Hebei Science and Technology Department under Grant 12457662. This paper was recommended by Associate Editor D. Tao.

X.-Z. Wang and Y.-L. He are with the Machine Learning Center, College of Mathematics and Computer Science, Hebei University, Hebei 071002, China (e-mail: xizhaowang@ieee.org; csylhe@gmail.com).

D.-D. Wang is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon 999077, Hong Kong (e-mail: danwang6@student.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2245891

ference between FNB and FNB_{ROT} is the scheme of parameter selection. FNB assigns a most straightforward bandwidth for Parzen windows while FNB_{ROT} uses the rule of thumb scheme [15], [34], [38], [44] to determine the parameter.

Some comparative studies between the two strategies, i.e., the discretization and the density estimation, can be found in [2], [10], [49], [50], and [51]. The comparative results show difficulties on judging which one is universally better. The classification performance depends mainly on the problem domain and the experimental procedure setup (e.g., fivefold cross-validation [10], tenfold cross-validation [2], and selective tenfold cross-validation [2], etc.).

The aim of this paper is not to extend the comparative study between the discretization and the density estimation but to improve the classification performance of NBC by proposing a technique of estimating joint p.d.f. The three methods mentioned above, i.e., NNB, FNB and FNB_{ROT}, are based on such an assumption that all attributes are conditionally independent with each other. Nevertheless, in many real-world applications, this assumption does not always stand. Furthermore, the estimated p.d.f.s that are used in design of NNB, FNB, or FNB_{ROT} are far away from the true p.d.f. due to the inappropriate distribution assumption or parameter selection. Motivated by improving the classification performance via removing or relaxing the restriction of independence among attributes and obtaining a better estimation of the true p.d.f., in this paper we propose a non-naive Bayesian classifier (NNBC) where a model of joint p.d.f. is estimated by using Parzen windows [21] based on the multivariate kernel function. Specifically, the estimation is evaluated by seeking an optimal bandwidth for the Parzen window through minimizing the mean integrated squared error (MISE) between the true p.d.f. and the estimated p.d.f.. The choice of bandwidth is considered an essential issue for Parzen window based p.d.f. estimation [15], [34], [38], [44]. Simulations show that NNBC can indeed achieve a better p.d.f. estimation by selecting the optimal bandwidth.

The classification performances of NNBC are examined in terms of classification accuracy, ranking, and the quality of class-conditional probability estimation. The latter two indexes are measured by the area under ROC curve (AUC) [14], [40] and the probability mean square error (PMSE) [21] respectively. Our experimental results on 30 UCI datasets [43] demonstrate that NNBC outperforms NNB, FNB, and FNB_{ROT} with most of testing datasets.

The rest of this paper is organized as follows. In Section II, we summarize the basic NBC algorithm. In Section III, a non-naive Bayesian classification model based on the estimation of joint probability density is proposed. In Section IV, our experimental setup and results are given. We conclude this paper with some remarks in the last section.

II. NBC

This section will give a brief review on naive Bayesian classifiers. We firstly introduce a number of denotations.

Let X be a set of instances. Each instance is described by d condition attributes, which are used to depict the specific features of an instance, and one decision attribute indicating the class label of the instance. We assume that, all the condition attributes are continuous, and the decision attribute is discrete. Suppose that the decision attribute varies from $\{w_1, w_2, \ldots, w_c\}$, which implies that all instances are categorized into c classes. In this way, any instance in X will be denoted as a d-dimensional vector

$$\vec{\mathrm{x}}_i^{(k)} = \left\{x_{i1}^{(k)}, x_{i2}^{(k)}, \ldots, x_{id}^{(k)}\right\} (1 \le i \le n_k, 1 \le k \le c)$$

where c is the number of classes and n_k is the number of instances within the kth class. Let $\vec{\mathrm{x}} = (x_1, x_2, \ldots, x_d)$ indicate a new example whose value of decision attribute is unknown.

Bayesian classifier assigns the most likely class to the new example $\vec{\mathrm{x}} = (x_1, x_2, \ldots, x_d)$ by the Bayesian theorem. According to the prior probability and class-conditional probability of the new example, Bayesian classifier calculates the posterior probability and determines the value of decision attribute for the new example. The Bayesian classifier discriminates the class of the new sample $\vec{\mathrm{x}}$ in the following equation:

$$\begin{aligned} w &= \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{P\left(w_k \left| \vec{\mathrm{x}}\right.\right)\right\} \\ &= \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{\frac{P(w_k)\, P\left(\vec{\mathrm{x}} \left| w_k\right.\right)}{P\left(\vec{\mathrm{x}}\right)}\right\} \\ &= \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{P(w_k)\, P\left(\vec{\mathrm{x}} \left| w_k\right.\right)\right\} \end{aligned} \tag{1}$$

where $P(w_k)$ is the prior probability of the kth class, which can be estimated by the frequency of instances of the kth class, i.e., $P(w_k) = \frac{n_k}{N}$ in which $N = \sum_{k=1}^{c} n_k$ is the size of dataset X. $P\left(\vec{\mathrm{x}} \left| w_k\right.\right)$ denotes the class-conditional probability. The main objective of NBC is to estimate $P\left(\vec{\mathrm{x}} \left| w_k\right.\right)$ based on the training instances in the kth class.

NBC assumes that all condition attributes are independent given the decision attribute (i.e., conditional independence assumption). Hence, based on this assumption, the class-conditional probability can be expressed as

$$P\left(\vec{\mathrm{x}} \left| w_k\right.\right) = P\left(x_1, x_k, \ldots, x_d \left| w_k\right.\right) = \prod_{j=1}^{d} P\left(x_j \left| w_k\right.\right). \tag{2}$$

By replacing the class-conditional probability with (2), NBC obtains the following decision rule [in (3)] for determining the value of decision attribute of $\vec{\mathrm{x}}$

$$w = \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{\frac{n_k}{N} \prod_{j=1}^{d} P\left(x_j \left| w_k\right.\right)\right\}. \tag{3}$$

From (3), we extract that the calculation of $P\left(x_j \left| w_k\right.\right) (1 \le j \le d)$ is the key to apply the NBC to determining the class of the new instance. Based on the density estimation strategy, three handling-methodologies NNB [2], [30], FNB [19], and FNB_{ROT} [28] are popular ways to estimate the component $P\left(x_j \left| w_k\right.\right)$ for $\vec{\mathrm{x}}$.

A. NNB

NNB [2], [30] assumes that the n_k values of the jth condition attribute,i.e., $x_{1j}^{(k)}, x_{2j}^{(k)}, \ldots, x_{n_k,j}^{(k)}$, obey a single

Gaussian distribution. Then, $P\left(x_j \,|w_k\right)$ can be calculated by the following equation:

$$P\left(x_j \,|w_k\right) = \frac{1}{\sqrt{2\pi\sigma_j^{(k)}}} \exp\left[-\frac{\left(x_j-\mu_j^{(k)}\right)^2}{2\left(\sigma_j^{(k)}\right)^2}\right], \tag{4}$$

where $\mu_j^{(k)} = \frac{\sum_{i=1}^{n_k} x_{ij}^{(k)}}{n_k}$ and $\left(\sigma_j^{(k)}\right)^2 = \frac{\sum_{i=1}^{n_k}\left[x_{ij}^{(k)}-\mu_j^{(k)}\right]^2}{n_k}$ are the mean value and variance of $x_{1j}^{(k)}, x_{2j}^{(k)}, \ldots, x_{n_k,j}^{(k)}$, respectively.

B. FNB

The continuous attributes do not always follow the Gaussian distribution in many applications. To tackle the case of non-Gaussian distribution, John and Langley [19] proposed the FNB which estimates $P\left(x_j \,|w_k\right)$ through the following equation:

$$P\left(x_j \,|w_k\right) = \frac{1}{n_k h_j^{(k)}} \sum_{i=1}^{n_k}\left[K\left(\frac{x_j-x_{ij}^{(k)}}{h_j^{(k)}}\right)\right] \tag{5}$$

where $h_j^{(k)}$ is the bandwidth and $K(*)$ is the kernel function. In FNB, $h_j^{(k)} = \frac{1}{\sqrt{n_k}}$ and $K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$. This kernel is called the Gaussian kernel. The experimental study shows that the classification performance of FNB mainly depends on the selection of the bandwidth $h_j^{(k)}$.

C. FNB_{ROT}

In order to evaluate the impact of different parameter-selection methods on the classification performance, Liu *et al*. [28] apply the rule of thumb [38], [44] to the selection of bandwidth parameter of FNB. They replace the traditional bandwidth parameter in FNB $h_j^{(k)} = \frac{1}{\sqrt{n_k}}$ with the following equation:

$$h_j^{(k)} = \left(\frac{4}{3n_k}\right)^{\frac{1}{5}} \sigma_j^{(k)} \tag{6}$$

where $\left(\sigma_j^{(k)}\right)^2$ is the variance of the jth condition attribute values $x_{1j}^{(k)}, x_{2j}^{(k)}, \ldots, x_{n_k,j}^{(k)}$. In our study, we call these kind of Bayesian classifiers $\mathrm{FNB}_{\mathrm{ROT}}$. Besides the above-mentioned rule of thumb, we can access other parameter selection methods from references (e.g., [15], [34], [38], and [44]). As demonstrated in [28], sophisticated bandwidth selection schemes may not give favorable performance in the context of NBC classification, while some simple bandwidth selection schemes tend to achieve considerably better performances. Furthermore, in [28], the simple scheme, i.e., the rule of thumb scheme, is used for bandwidth selection in their experiments.

III. NNBC Based on Joint p.d.f. Estimation

As we mentioned in Section II, NBC assumes that all condition attributes are independent given the decision attribute. In this section, we will propose an improved Bayesian classification model, i.e., NNBC which eliminates the assumption of attribute-independence and is based on a technique of joint p.d.f. estimation. Firstly, the basic concept of joint p.d.f. estimation is introduced. Afterwards, the optimal parameter selection in the estimation of the joint p.d.f. is discussed. Finally, the NNBC model based on the joint p.d.f. with the optimal bandwidth is described in detail.

A. Joint p.d.f. Estimation

In probability theory and statistical inference, p.d.f. estimation [38], [44] refers to giving a specific function without unknown parameters such that the error between this function and the unobservable underlying p.d.f. can be small enough. Particularly, the estimation of p.d.f. for a continuous distribution from the representative samples has been considered as one of the major ingredients in machine learning and pattern recognition. The mostly used strategy to construct the underlying p.d.f. approximately is Parzen window method [31]. Based on the set of d-dimensional data $\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2, \ldots, \vec{\mathrm{x}}_N$ where $\vec{\mathrm{x}}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ $(1 \le i \le N)$, Parzen window method estimates the underlying joint p.d.f. through the following equation:

$$\begin{aligned}\hat{f}_h\left(\vec{\mathrm{x}}\right) &= \frac{1}{Nh^d}\sum_{i=1}^{N}\left[\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{x}}_i}{h}\right)\right] \\ &= \frac{1}{Nh^d}\sum_{i=1}^{N}\left[\mathrm{K}\left(\frac{x_1-x_{i1}}{h}, \frac{x_2-x_{i2}}{h}, \ldots, \frac{x_d-x_{id}}{h}\right)\right]\end{aligned} \tag{7}$$

where $\mathrm{K}(*)$ is a multivariate kernel function and h is a crucial parameter called bandwidth. The most common kernel is the multivariate Gaussian kernel as shown in

$$\mathrm{K}\left(\vec{\mathrm{x}}\right) = \frac{1}{\left(\sqrt{2\pi}\right)^d} \exp\left(-\frac{\vec{\mathrm{x}}\vec{\mathrm{x}}^{\mathrm{T}}}{2}\right) \tag{8}$$

where $\vec{\mathrm{x}}^{\mathrm{T}}$ is the transpose of vector $\vec{\mathrm{x}}$.

It is well acknowledged that the estimation performance of Parzen window method strongly relies on the selection of bandwidth h [15], [34], [38], [44]. Many researchers [29], [31], [36], [38], [44] have claimed that an appropriate selection of bandwidth can converge or minimize the estimated error between the true p.d.f. and estimated p.d.f.

B. Optimal Selection of Bandwidth

In order to find the optimal bandwidth for a joint p.d.f. estimation, in this section we adopt the MISE [3], [5], [21] as our error criterion to measure the difference between the true p.d.f. and the estimated p.d.f.. Let $f\left(\vec{\mathrm{x}}\right)$ be the true p.d.f. of the observed data $\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2, \ldots, \vec{\mathrm{x}}_N$; thus, MISE can be expressed as

$$\begin{aligned}\mathrm{MISE}(h) &= E\left[\int\left\{\hat{f}_h\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right\}^2 \mathrm{d}\vec{\mathrm{x}}\right] \\ &= \int \mathrm{var}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)\mathrm{d}\vec{\mathrm{x}} + \int \mathrm{bias}^2\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)\mathrm{d}\vec{\mathrm{x}}\end{aligned} \tag{9}$$

where

$$\mathrm{bias}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right) = E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right] - f\left(\vec{\mathrm{x}}\right)$$

and

$$\mathrm{var}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right) = E\left\{\hat{f}_h\left(\vec{\mathrm{x}}\right) - E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]\right\}^2.$$

In (9), $\int$ and $\mathrm{d}\vec{\mathrm{x}}$ are the abbreviations of $\int\int\ldots\int$ and $dx_1 dx_2 \ldots dx_d$, respectively. Next, we derive the expressions

for bias $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ and var $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$, respectively. Note the fundamental definitions of mathematical expectation

$$E\left(\vec{\mathrm{x}}\right)=\int \vec{\mathrm{x}} f\left(\vec{\mathrm{x}}\right) \mathrm{d}\vec{\mathrm{x}}, \text{ and } E\left[g\left(\vec{\mathrm{x}}\right)\right]=\int g\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) \mathrm{d}\vec{\mathrm{x}},$$

and the expressions of the estimated p.d.f. and the kernel functions are

$$\hat{f}_h\left(\vec{\mathrm{x}}\right)=\frac{\sum_{i=1}^{N}\left[\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{x}}_i}{h}\right)\right]}{Nh^d}, \text{ and } \mathrm{K}\left(\vec{\mathrm{x}}\right)=\frac{\exp\left(-\frac{\vec{\mathrm{x}}\vec{\mathrm{x}}^{\mathrm{T}}}{2}\right)}{\left(\sqrt{2\pi}\right)^d}$$

we can obtain the following equation:

$$E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]=\int\left[\frac{1}{h^d}\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}\right) f\left(\vec{\mathrm{y}}\right)\right]\mathrm{d}\vec{\mathrm{y}} \tag{10}$$

where $\vec{\mathrm{y}}$ is a random variable with the p.d.f. $f\left(\vec{\mathrm{y}}\right)$.

We now give the derivation of bias $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$. After replacing the component $E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]$ in bias $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ with (10), we gain the following equation:

$$\begin{aligned}
&\text{bias}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)\\
&=\int\left[\frac{1}{h^d}\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}\right) f\left(\vec{\mathrm{y}}\right)\right]\mathrm{d}\vec{\mathrm{y}}-f\left(\vec{\mathrm{x}}\right)\\
&=\int\left[\mathrm{K}\left(\vec{\mathrm{z}}\right) f\left(\vec{\mathrm{x}}-h\vec{\mathrm{z}}\right)\right]\mathrm{d}\vec{\mathrm{z}}-f\left(\vec{\mathrm{x}}\right)\\
&=\int\left\{\mathrm{K}\left(\vec{\mathrm{z}}\right)\left[f\left(\vec{\mathrm{x}}\right)-h\vec{\mathrm{z}}f'\left(\vec{\mathrm{x}}\right)+\frac{1}{2}h^2\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}f''\left(\vec{\mathrm{x}}\right)\right.\right.\\
&\quad\left.\left.+O\left(h^2\right)-f\left(\vec{\mathrm{x}}\right)\right]\right\}\mathrm{d}\vec{\mathrm{z}}\\
&=-hf'\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}+\frac{1}{2}h^2f''\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}\\
&\quad+O\left(h^2\right)\int f\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}
\end{aligned}\tag{11}$$

where $\vec{\mathrm{z}}=\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}$.

It is known that for the multivariate Gaussian kernel $\mathrm{K}\left(\vec{\mathrm{z}}\right)$, $\int\vec{\mathrm{z}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}=0$ and $\int\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}=1$ hold well. Substituting these two integrals in (11), we have the following equation:

$$\text{bias}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)=\tfrac{1}{2}h^2f''\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}+O\left(h^2\right). \tag{12}$$

Furthermore we give the derivation of var $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$. Noting that $E\left\{2\hat{f}_h\left(\vec{\mathrm{x}}\right)E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]\right\}=2\left\{E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]\right\}^2$, we can express var $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ as follows:

$$\begin{aligned}
&var\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)=E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]^2-\left\{E\left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]\right\}^2\\
&=\frac{1}{N}\int\left[\frac{1}{h^{2d}}\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}\right)^2 f\left(\vec{\mathrm{y}}\right)\right]\mathrm{d}\vec{\mathrm{y}}\\
&\quad-\frac{1}{N}\left\{\int\left[\frac{1}{h^{2d}}\mathrm{K}\left(\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}\right) f\left(\vec{\mathrm{y}}\right)\right]\mathrm{d}\vec{\mathrm{y}}\right\}^2\\
&=\frac{1}{Nh^d}\int\left[\mathrm{K}\left(\vec{\mathrm{z}}\right)^2 f\left(\vec{\mathrm{x}}-h\vec{\mathrm{z}}\right)\right]\mathrm{d}\vec{\mathrm{z}}\\
&\quad-\frac{1}{Nh^d}\left\{\int\left[\mathrm{K}\left(\vec{\mathrm{z}}\right) f\left(\vec{\mathrm{x}}-h\vec{\mathrm{z}}\right)\right]\mathrm{d}\vec{\mathrm{z}}\right\}^2\\
&=\frac{1}{Nh^d}\left\{f\left(\vec{\mathrm{x}}\right)\int\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}-hf'\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}\right.\\
&\quad\left.+\frac{1}{2}h^2f''\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}+O\left(h^2\right)\right\}+O\left(N^{-1}\right)
\end{aligned}\tag{13}$$

where $\vec{\mathrm{z}}=\frac{\vec{\mathrm{x}}-\vec{\mathrm{y}}}{h}$.

Because of

$$\frac{1}{Nh^d}\left[hf'\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\mathrm{K}\left(\vec{\mathrm{z}}\right)^2 d\vec{\mathrm{z}}\right]=O\left(N^{-1}\right)$$

and

$$\frac{1}{Nh^d}\left[\frac{1}{2}h^2f''\left(\vec{\mathrm{x}}\right)\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}\right]=O\left(N^{-1}\right)$$

the formula of var $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ can be denoted as

$$\text{var}\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)=\frac{1}{Nh^d}f\left(\vec{\mathrm{x}}\right)\int\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}+O\left(N^{-1}h^{-d}\right). \tag{14}$$

Through replacing bias $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ and var $\left(\hat{f}_h\left(\vec{\mathrm{x}}\right)\right)$ in (9) with the derived equations (12) and (14), respectively, we have the following equation:

$$\begin{aligned}
\text{MISE}\left(h\right)&=\frac{1}{Nh^d}\left[\int\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}\right]\left[\int f\left(\vec{\mathrm{x}}\right)\mathrm{d}\vec{\mathrm{x}}\right]\\
&\quad+\frac{1}{4}h^4\left[\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}\right]^2\left\{\int\left[f''\left(\vec{\mathrm{x}}\right)\right]^2\mathrm{d}\vec{\mathrm{x}}\right\}.
\end{aligned}\tag{15}$$

Let $R\left(\mathrm{K}\right)=\int\mathrm{K}\left(\vec{\mathrm{z}}\right)^2\mathrm{d}\vec{\mathrm{z}}$, $\mu_2\left(\mathrm{K}\right)=\int\vec{\mathrm{z}}\vec{\mathrm{z}}^{\mathrm{T}}\mathrm{K}\left(\vec{\mathrm{z}}\right)\mathrm{d}\vec{\mathrm{z}}$, and $R\left(f''\right)=\int\left[f''\left(\vec{\mathrm{x}}\right)\right]^2\mathrm{d}\vec{\mathrm{x}}$. Noting that $\int f\left(\vec{\mathrm{x}}\right)\mathrm{d}\vec{\mathrm{x}}=1$, we thus simplify the expression of MISE$\left(h\right)$ as follows:

$$\text{MISE}\left(h\right)=\frac{1}{Nh^d}\left[R\left(\mathrm{K}\right)\right]+\frac{1}{4}h^4\left[\mu_2\left(\mathrm{K}\right)\right]^2R\left(f''\right). \tag{16}$$

To find the optimal bandwidth that minimizes MISE$\left(h\right)$, we let the derivative of MISE$\left(h\right)$ (with respect to h) be 0, i.e., $\frac{d\text{MISE}(h)}{dh}=0$ which implies that the optimal h is achieved at

$$h_{\text{optimal}}^{(\text{MISE})}=\left[\frac{dR\left(\mathrm{K}\right)}{\left[\mu_2\left(\mathrm{K}\right)\right]^2R\left(f''\right)N}\right]^{\frac{1}{d+4}} \tag{17}$$

and the corresponding minimal MISE$\left(h\right)$ is given by

$$\begin{aligned}
&\inf_{h>0}\text{MISE}\left(h\right)\\
&=\frac{d+4}{4d}\left\{\left[\mu_2\left(\mathrm{K}\right)\right]^{2d}\left[dR\left(\mathrm{K}\right)\right]^4\left[R\left(f''\right)\right]^d N^{-4}\right\}^{\frac{1}{d+4}}.
\end{aligned}\tag{18}$$

In the following we point out how to compute the three components $R(\mathrm{K})$, $\mu_2(\mathrm{K})$, and $R\left(f''\right)$ in (17) and (18). For a multivariate Gaussian kernel, we adopt the following equations:

$$R(\mathrm{K}) = \frac{1}{\left(\sqrt{2\pi}\right)^{2d}} \prod_{j=1}^{d} \int \exp\left(-x_j^2\right) dx_j = (4\pi)^{-\frac{d}{2}}, \quad (19)$$

and

$$\mu_2(\mathrm{K}) = \frac{1}{\left(\sqrt{2\pi}\right)^{d}} \sum_{j=1}^{d} \left[\int x_j^2 \exp\left(-\frac{x_j^2}{2}\right) dx_j \right] = 1. \quad (20)$$

For the sake of a robust estimation, we consider $f\left(\vec{\mathrm{x}}\right)$ as a multivariate normal density function $\mathrm{N}(0, \Sigma)$ with the diagonal matrix $\Sigma = \mathrm{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2\right)$ where σ_j^2, $(1 \le j \le d)$ is the variance of $x_{1j}, x_{2j}, \ldots, x_{Nj}$. Now we envisage a specific case of $d = 2$ and give its derivation of $R\left(f''\right)$ where $f\left(\vec{\mathrm{x}}\right) = f(x_1, x_2) = \frac{1}{\left(\sqrt{2\pi}\right)^2 \sigma_1 \sigma_2} \exp\left[-\left(\frac{x_1^2}{2\sigma_1^2} + \frac{x_2^2}{2\sigma_2^2}\right)\right]$. The formula of $R\left[f''(x_1, x_2)\right]$ can be expressed as

$$\begin{aligned} & R\left[f''(x_1, x_2)\right] \\ &= \int\int \left[\frac{\partial^2 f(x_1, x_2)}{\partial^2 x_1} + \frac{\partial^2 f(x_1, x_2)}{\partial^2 x_2} \right]^2 dx_1 dx_2 \\ &= \frac{1}{4\left(\sqrt{2\pi}\right)^2 \sigma_1 \sigma_2} \left[2\left(\frac{1}{\sigma_1^4} + \frac{1}{\sigma_2^4}\right) + \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^2 \right]. \end{aligned} \quad (21)$$

Similarly, the derivation of $R\left[f''\left(\vec{\mathrm{x}}\right)\right]$ for the p.d.f. estimation with d variables $(d > 2)$ can be given by

$$R\left[f''\left(\vec{\mathrm{x}}\right)\right] = \frac{2\sum_{j=1}^{d} \frac{1}{\sigma_j^4} + \left(\sum_{j=1}^{d} \frac{1}{\sigma_j^2}\right)^2}{4\left(\sqrt{2\pi}\right)^d \prod_{j=1}^{d} \sigma_j} \quad (22)$$

i.e.,

$$R\left(f''\right) = \frac{(4\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \left\{2\mathrm{tr}\left(\Sigma^{-1}\Sigma^{-1}\right) + \mathrm{tr}^2\left(\Sigma^{-1}\right)\right\}}{4}. \quad (23)$$

After bringing (19), (20), (23) into (17) and (18), we acquire the optimal bandwidth in (24)

$$h_{\mathrm{optimal}}^{(\mathrm{MISE})} = \left(\frac{4d}{N |\Sigma|^{-\frac{1}{2}} \left\{2\mathrm{tr}\left(\Sigma^{-1}\Sigma^{-1}\right) + \mathrm{tr}^2\left(\Sigma^{-1}\right)\right\}} \right)^{\frac{1}{d+4}} \quad (24)$$

and the minimal MISE as follows:

$$\begin{aligned} \inf_{h>0} \mathrm{MISE}(h) = (4\pi)^{-\frac{d}{2}} & \left(\frac{d+4}{4d}\right) \left(\frac{d}{N}\right)^{\frac{4}{d+4}} \\ & \left(\frac{|\Sigma|^{-\frac{1}{2}} \left\{2\mathrm{tr}\left(\Sigma^{-1}\Sigma^{-1}\right) + \mathrm{tr}^2\left(\Sigma^{-1}\right)\right\}}{4} \right)^{\frac{d}{d+4}}. \end{aligned} \quad (25)$$

In our proposed NNBC, we use MISE to measure the error between the true p.d.f. and the estimated p.d.f.. Apart from MISE, another error criterion, i.e., integrated squared error (ISE), is commonly used in the kernel density estimation. The expression of ISE is given in

$$\mathrm{ISE}(h) = \int \left[\hat{f}_h\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}}. \quad (26)$$

Through extending (26), we can get the following equation:

$$\begin{aligned} \mathrm{ISE}(h) = & \int \left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}} - 2\int \hat{f}_h\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) \mathrm{d}\vec{\mathrm{x}} \\ & + \int \left[f\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}}. \end{aligned} \quad (27)$$

From (27), we can find that the third term $\int \left[f\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}}$ is not related to the unknown bandwidth h. Hence, the minimization of Eq. (27) equals to the minimization of the following equation:

$$\mathrm{ISE}^*(h) = \int \left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}} - 2\int \hat{f}_h\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) \mathrm{d}\vec{\mathrm{x}}. \quad (28)$$

Following the derivation in MISE mentioned above, we can obtain

$$\begin{aligned} \int \left[\hat{f}_h\left(\vec{\mathrm{x}}\right)\right]^2 \mathrm{d}\vec{\mathrm{x}} = & \frac{1}{\left(2\sqrt{\pi}\right)^d Nh^d} + \frac{1}{\left(2\sqrt{\pi}\right)^d N^2 h^d} \\ & \times \sum_{i=1}^{N} \sum_{j\neq i}^{N} \left\{ \exp\left[-\frac{1}{4} \sum_{k=1}^{d} \left(\frac{x_{ik} - x_{jk}}{h} \right)^2 \right] \right\} \end{aligned} \quad (29)$$

and

$$\begin{aligned} & \int \hat{f}_h\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) \mathrm{d}\vec{\mathrm{x}} \\ &= \frac{1}{\left(\sqrt{2\pi}\right)^d N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{\sqrt{\sigma_j^2 + h^2}} \exp\left[-\frac{x_{ij}^2}{2\left(\sigma_j^2 + h^2\right)} \right] \end{aligned} \quad (30)$$

where σ_j^2, $(1 \le j \le d)$ is the variance of $x_{1j}, x_{2j}, \cdots, x_{Nj}$.

Bring (29) and (30) into (28), we can get

$$\begin{aligned} \mathrm{ISE}^*(h) = & \frac{1}{\left(2\sqrt{\pi}\right)^d Nh^d} \\ & + \frac{1}{\left(2\sqrt{\pi}\right)^d N^2 h^d} \sum_{i=1}^{N} \sum_{j\neq i}^{N} \left\{ \exp\left[-\frac{1}{4} \sum_{k=1}^{d} \left(\frac{x_{ik} - x_{jk}}{h} \right)^2 \right] \right\} \\ & - \frac{2}{\left(\sqrt{2\pi}\right)^d N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{\sqrt{\sigma_j^2 + h^2}} \exp\left[-\frac{x_{ij}^2}{2\left(\sigma_j^2 + h^2\right)} \right]. \end{aligned} \quad (31)$$

From (31), we gain that in order to determine the optimal bandwidth h which minimizes $\mathrm{ISE}^*(h)$, an optimization scheme, e.g., brute-force or intelligent search algorithm, is required. It will lead to a significant increase of time complexity. However, by minimizing MISE, we can directly obtain the expression of optimal bandwidth h as shown in (24) and thus avoid the application of a time-consuming optimization scheme.

C. NNBC

As we discussed in the previous sections, NNB, FNB, and FNB_{ROT} have the following two restrictions: 1) they are based on such an assumption that all condition attributes are independent given the decision attribute, which obviously does not always stand in many real-world applications, and 2) in the process of estimating the marginal p.d.f. of each attribute, NNB assumes that each attribute follows a normal distribution; $\text{FNB}/\text{FNB}_{\text{ROT}}$ fixes the non-normal distribution problem to some extent, while they have not an appropriate strategy of the parameter selection. All these drawbacks seriously affect the precision of the p.d.f. estimation. Motivated by improving the classification performance via removing or relaxing the above-mentioned two restrictions, we propose the NNBC model in which the restraint of independence among the attributes is removed and the joint p.d.f. estimation replaces the marginal p.d.f. estimations. NNBC determines the class of a new sample $\vec{\mathrm{x}}$ as the following equation:

$$\begin{aligned} w &= \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{ \frac{n_k}{N} P\left(\vec{\mathrm{x}} \,|w_k\right) \right\} \\ &= \underset{w_k, k=1,2,\ldots,c}{\arg\max} \left\{ \frac{1}{N h_k^d} \sum_{i=1}^{n_k} \left[\mathrm{K}\left(\frac{x_1 - x_{i1}^{(k)}}{h_k}, \frac{x_2 - x_{i2}^{(k)}}{h_k}, \right.\right.\right. \\ &\qquad \left.\left.\left. \ldots, \frac{x_d - x_{id}^{(k)}}{h_k} \right) \right] \right\} \end{aligned} \tag{32}$$

where $\mathrm{K}\left(\vec{\mathrm{x}}\right) = \frac{1}{\left(\sqrt{2\pi}\right)^d} \exp\left(-\frac{\vec{\mathrm{x}}\vec{\mathrm{x}}^{\mathrm{T}}}{2}\right)$ is the multivariate Gaussian kernel as shown in (8), and $h_k\,(1 \le k \le c)$ is the optimal bandwidth which has been derived as in Section III-B.

Specifically, for a set of instances belonging to the kth class, the optimal bandwidth $h_k\,(1 \le k \le c)$ given in (24) can be simplified as

$$h_k = \left(\frac{4d}{n_k \left|\Sigma_k\right|^{-\frac{1}{2}} \left\{ 2\mathrm{tr}\left(\Sigma_k^{-1}\Sigma_k^{-1}\right) + \mathrm{tr}^2\left(\Sigma_k^{-1}\right) \right\}} \right)^{\frac{1}{d+4}},$$

where

$$\Sigma_k = \mathrm{diag}\left\{ \left[\sigma_1^{(k)}\right]^2, \left[\sigma_2^{(k)}\right]^2, \ldots, \left[\sigma_d^{(k)}\right]^2 \right\}.$$

The main differences among NNB, FNB, FNB_{ROT}, and our proposed NNBC are summarized as follows:

1) NNB, FNB, and FNB_{ROT} assume that all condition attributes are independent given the decision attribute. By calculating every component $P\left(x_j \,|w_k\right)$ $(1 \le j \le d, 1 \le k \le c)$ through the marginal p.d.f., NNB, FNB, and FNB_{ROT} obtain the class-conditional probability $P\left(\vec{\mathrm{x}} \,|w_k\right)$ for the new instance $\vec{\mathrm{x}} = \{x_1, x_2, \ldots, x_d\}$. Our proposed NNBC, which removes the independence assumption, establishes a model of joint p.d.f. in the estimation of $P\left(\vec{\mathrm{x}} \,|w_k\right)$ based on the multivariate kernel function.
2) Due to the inappropriate distribution assumption for NNB and the nonoptimal parameter selection in FNB and FNB_{ROT}, large estimated errors usually occur between the true p.d.f. and the estimated p.d.f.. The imprecise estimation of p.d.f. for NBC will lead to the dissatisfactory classification performance. By minimizing the MISE, our proposed NNBC gains the optimal bandwidth for the joint p.d.f. estimation. The optimal bandwidth enables the estimated error between the true p.d.f. and the estimated p.d.f. to reach the minimum.

Now, we analyze the time complexities of the above-mentioned four Bayesian classification algorithms,i.e., NNB, FNB, FNB_{ROT}, and NNBC. Let N be the number of training instances, M be the number of testing instances, and d be the number of condition attributes. Since NNB needs to calculate the means and variances for the d condition attributes, the training time complexity of NNB is $O(Nd)$ and the classification time complexity is $O(Md)$. FNB uses the superposition of N p.d.f.s of the normal distribution to fit the true p.d.f.; thus, the training and classification time complexities of FNB are $O(Nd)$ and $O(MNd)$ respectively. Compared with FNB, FNB_{ROT} applies the rule of thumb to obtain some increase in the training time, however the training and classification time complexities remain $O(Nd)$ and $O(MNd)$, respectively. Similar to FNB_{ROT}, our NNBC also needs the extra time to compute the optimal bandwidth in the training phase. However, the determination of the required parameter does not lead to additional increase of classification time complexity. Thus, the training and classification time complexities of NNBC are still $O(Nd)$ and $O(MNd)$ as well.

IV. Experimental Comparison of Classification Performances Among the Four Bayesian Models

In this section, we conduct an experimental comparison among the four Bayesian models (NNB, FNB, FNB_{ROT}, and NNBC) on 30 UCI benchmark datasets [43] with respect to the three indexes (i.e., the classification accuracy, the ranking performance (measured by AUC) [14], [40], and the quality of class-conditional probability density estimation (measured by PMSE) [21]). The experiment is configured on a PC having the OS of Windows 2000 with one Pentium 4 2.8 GHz processor and a 1024 MB RAM.

A. Data Preparation Guidelines and Experimental Procedure

In our experiment, 30 UCI benchmark datasets are used to test the classification performances of NNB, FNB, FNB_{ROT}, and NNBC. These datasets involve a wide range of real domains and data characteristics. The detailed description of 30 datasets can be referred to the information provided by UCI Machine Learning Repository [43].

In order to use these datasets more efficiently and specifically, we preprocess these datasets according to the following procedures.

1) Delete the nominal condition attributes. In our work, we aim to study the impacts of different density estimation methods on the classification performances of Bayesian models. The class-conditional probabilities of the nominal condition attributes can be calculated by counting the frequency of attribute-values and combinations of attribute values from a given dataset.

TABLE I
DETAILED EXPERIMENTAL RESULTS OF CLASSIFICATION ACCURACY AND STANDARD DEVIATION ON 30 UCI BENCHMARK DATASETS

	Datasets	**NNB**	**FNB**	**FNB$_{\mathrm{ROT}}$**	**NNBC**
1	Auto Mpg	0.673±0.004	0.662±0.006	0.675±0.006	0.727±0.008
2	Blood Transfusion	0.729±0.005	0.754±0.006	0.693±0.003	0.764±0.006
3	Breast Cancer	0.960±0.001	0.975±0.001	0.963±0.002	0.962±0.002
4	Breast Cancer W-D	0.932±0.003	0.944±0.002	0.946±0.002	0.947±0.004
5	Breast Cancer W-P	0.631±0.012	0.699±0.009	0.629±0.015	0.610±0.018
6	Contraceptive Method	0.472±0.003	0.498±0.005	0.462±0.004	0.501±0.004
7	Credit Approval	0.718±0.003	0.711±0.004	0.755±0.003	0.733±0.005
8	Cylinder Bands	0.632±0.011	0.711±0.006	0.686±0.011	0.698±0.011
9	Ecoli	0.845±0.006	0.852±0.004	0.848±0.007	0.852±0.007
10	Glass Identification	0.349±0.018	0.594±0.015	0.456±0.016	0.634±0.012
11	Haberman's Survival	0.753±0.005	0.738±0.006	0.727±0.007	0.700±0.011
12	Heart Disease	0.842±0.006	0.841±0.001	0.841±0.001	0.766±0.011
13	Image Segment	0.797±0.001	0.898±0.001	0.843±0.002	0.969±0.001
14	Ionosphere	0.809±0.005	0.905±0.003	0.903±0.004	0.937±0.004
15	Iris	0.953±0.004	0.959±0.005	0.959±0.004	0.963±0.005
16	Libras Movement	0.640±0.013	0.554±0.013	0.681±0.012	0.807±0.008
17	Magic Telescope	0.738±0.003	0.762±0.003	0.768±0.004	0.806±0.003
18	Musk Version 1	0.730±0.007	0.803±0.008	0.783±0.007	0.804±0.007
19	New Thyroid Gland	0.963±0.001	0.911±0.002	0.965±0.004	0.919±0.002
20	Page Blocks	0.861±0.008	0.868±0.001	0.914±0.006	0.904±0.004
21	Parkinsons	0.694±0.006	0.810±0.007	0.715±0.006	0.837±0.009
22	Pima Indian Diabetes	0.745±0.003	0.742±0.004	0.752±0.004	0.738±0.006
23	Sonar	0.676±0.009	0.768±0.011	0.741±0.011	0.818±0.010
24	SPECTF Heart	0.667±0.007	0.781±0.007	0.698±0.008	0.695±0.007
25	Vehicle Silhouettes	0.487±0.008	0.519±0.006	0.547±0.006	0.620±0.008
26	Vowel Recognition	0.655±0.009	0.575±0.013	0.781±0.010	0.961±0.004
27	Wine	0.971±0.005	0.959±0.004	0.979±0.003	0.988±0.003
28	Wine Quality-Red	0.371±0.005	0.580±0.004	0.517±0.004	0.607±0.006
29	Wine Quality-White	0.381±0.008	0.499±0.005	0.440±0.008	0.518±0.008
30	Yeast	0.500±0.004	0.571±0.003	0.520±0.005	0.559±0.005
	Average	**0.706±0.006**	**0.748±0.006**	**0.740±0.006**	**0.778±0.007**

2) Fill in the missing attribute values. We use the unsupervised filter named *ReplaceMissingValues* in WEKA [47] to fill in all the missing attribute values in each dataset. It replaces all missing values of continuous attributes with the means of the training data.
3) Reduce the large datasets. To compromise the running time, we adopt the unsupervised filter named *Resample* with the *sampleSizePercent* 10 in WEKA [47] to randomly reduce the sizes of three large datasets: Magic Telescope, Page Blocks and Wine Quality-White.

We evaluate the four Bayesian models in terms of classification (measured by accuracy), ranking performance (measured by AUC) [14], [40], and the quality of class-conditional probability density estimation (measured by PMSE) [21]. In our experiment the tenfold cross-validation procedure repeats 100 times. In every run, NNB, FNB, FNB$_{\mathrm{ROT}}$ and NNBC are trained on the same training sets and evaluated on the same testing set. When a partition of tenfolds on a dataset is given, the evaluations on classification accuracy, AUC, and PMSE are simultaneously carried out.

B. Comparisons for Classification Accuracy and Training Time

On the 30 UCI datasets the experimental results including the averaged classification accuracies and the standard deviations are summarized in Table I. Subsequently, a statistical analysis including the comparison of two classifiers on each single dataset and the comparison of multiple classifiers on all datasets are conducted based on the acquired classification results.

This statistical comparison is to assess whether the classification accuracies of two Bayesian models differ on a single dataset. The analysis is carried out based on the 100 runs of tenfold cross-validation described above. So as to compare the classification performances of the four models, i.e., NNB, FNB, FNB$_{\mathrm{ROT}}$, and NNBC on every dataset, the Wilcoxon signed-ranks test [7] is adopted here. Since the paired t-test suffers from several weaknesses [7], Wilcoxon signed-ranks test is often regarded as the alternative of the paired t-test. It ranks the absolute values of differences between the classification accuracies of two classifiers on every run of tenfold cross-validation and compares the ranks for positive and negative differences. In our experiment, all statistical comparisons are conducted under the significance level of 0.1.

According to the average classification accuracies of 100 runs of tenfold cross-validation in Table I and the statistical analysis with Wilcoxon signed-ranks test, we draw the following conclusions. Compared with NNB, NNBC obtains considerably better performances on 25 datasets. Compared with FNB, NNBC obtains considerably better performances on 20 datasets and has the equal performances on 2 datasets. Compared with FNB$_{\mathrm{ROT}}$, NNBC obtains considerably better performances on 21 datasets. The counts of wins, losses, and ties regarding the comparison among the four models are listed in Table VI.

For a given learning model, the number of wins obeys the normal distribution, i.e., $\mathrm{N}\left(\frac{m}{2}, \frac{\sqrt{m}}{2}\right)$ under the null-hypothesis in the sign test [7], where m is the number of datasets. If the number of wins is at least of $\frac{m}{2} + z_{\alpha/2} \times \frac{\sqrt{m}}{2}$, we conclude that the given learning model is considerably better than another one under the significance level α. In our study, 30 UCI

TABLE II
COMPARISON OF TRAINING TIME AMONG DIFFERENT CLASSIFICATION METHODS

Datasets	boostedNNB	boostedFNB	BoostedC4.5	BoostedSVM	SVM	NNB	FNB	FNB_{ROT}	NNBC
#1	0.0848	1.1943	0.1941	2.0969	0.4492	0.0006	0.0014	0.0018	0.0011
#2	0.1086	0.4180	0.1037	0.6169	0.0364	0.0010	0.0023	0.0024	0.0010
#3	0.2086	1.1479	0.5814	1.0349	0.0387	0.0023	0.0063	0.0069	0.0025
#4	0.5932	36.4797	0.7750	1.2959	0.0540	0.0041	0.0115	0.0151	0.0112
#5	0.2035	6.6344	0.3328	0.8882	0.0384	0.0021	0.0040	0.0056	0.0026
#6	0.0873	0.2026	0.1875	0.0813	0.3367	0.0020	0.0026	0.0028	0.0010
#7	0.0585	1.2339	0.1256	0.9031	0.0397	0.0025	0.0031	0.0037	0.0012
#8	0.4137	3.8174	1.4009	0.5973	0.0914	0.0036	0.0088	0.0098	0.0034
#9	0.0654	0.9572	0.1344	1.9106	0.7899	0.0005	0.0013	0.0029	0.0021
#10	0.0666	0.3849	0.1568	5.2616	1.2348	0.0006	0.0013	0.0015	0.0009
#11	0.0579	0.3062	0.0174	0.5975	0.0303	0.0005	0.0008	0.0008	0.0003
#12	0.0658	1.1686	0.1994	0.7733	0.0380	0.0010	0.0031	0.0038	0.0019
#13	1.3119	166.4212	3.1363	13.5033	3.1112	0.0123	0.0364	0.0408	0.0213
#14	0.3622	10.6366	0.7993	1.3526	0.0916	0.0049	0.0078	0.0095	0.0033
#15	0.0587	0.2035	0.0236	1.7529	0.1921	0.0003	0.0005	0.0005	0.0002
#16	3.2310	–	2.8991	–	6.4807	0.0077	0.0239	0.0274	0.0159
#17	0.6681	90.4503	2.4767	2.9130	0.2049	0.0055	0.0169	0.0226	0.0122
#18	2.2826	–	5.3123	10.2158	0.8144	0.0172	0.0522	0.0624	0.0317
#19	0.0624	0.5660	0.0489	2.0195	0.2285	0.0004	0.0008	0.0008	0.0004
#20	0.1909	4.8006	0.3772	3.6069	0.9426	0.0015	0.0048	0.0053	0.0024
#21	0.2454	3.7260	0.1561	0.8185	0.0280	0.0012	0.0028	0.0033	0.0017
#22	0.1453	5.9801	0.5107	0.4537	0.0363	0.0019	0.0045	0.0054	0.0028
#23	0.3642	14.0950	0.6664	1.0562	0.0685	0.0032	0.0079	0.0101	0.0044
#24	0.1357	3.7236	0.4293	0.8540	0.0374	0.0030	0.0085	0.0087	0.0027
#25	0.1456	6.1776	0.8431	2.4544	0.4731	0.0021	0.0060	0.0060	0.0023
#26	0.7223	21.3104	0.6153	11.9859	4.1316	0.0017	0.0044	0.0050	0.0024
#27	0.1376	1.7043	0.2256	1.5232	0.2035	0.0007	0.0015	0.0019	0.0010
#28	0.4545	9.7816	2.7406	4.3825	1.6616	0.0052	0.0143	0.0167	0.0071
#29	0.1170	3.0567	0.6449	2.6441	1.0338	0.0019	0.0048	0.0053	0.0020
#30	0.5960	11.0345	1.9844	10.5145	3.0285	0.0027	0.0104	0.0115	0.0057

datasets are used to test the classification performances of different learning models, that is, $m = 30$. Let $\alpha = 0.1$, then $\frac{m}{2} + z_{\alpha/2} \times \frac{\sqrt{m}}{2} = \frac{30}{2} + 1.645 \times \frac{\sqrt{30}}{2} \approx 20$. It indicates, compared with the other three learning models, NNBC will obtain considerably better classification performance if its number of wins on 30 UCI datasets reaches 20. This conclusion can be demonstrated from the results in Table VI. Through the statistical comparisons between two classifiers on each single dataset based on Wilcoxon signed-ranks test and sign test, we conclude that NNBC obtains better classification accuracies on the selected 30 UCI datasets compared with the other three models, i.e., NNB, FNB, and FNB_{ROT}. All in all, NNBC considerably outperforms the other three Bayesian models in classification accuracy on the selected 30 UCI datasets.

Apart from the classification accuracy, the training time of NNB, FNB, FNB_{ROT}, and NNBC are also compared based on the experimental process mentioned above. The comparative results of training time are listed in Table II. From the comparative results, we extract that the practical training time of NNBC is longer than NNB's but shorter than FNB and FNB_{ROT}'s. This is not contradictory to the theoretical analysis which indicates NNB, FNB, FNB_{ROT}, and NNBC have the same computational complexity of training time, i.e., $O(Nd)$, where N is the number of training instances and d is the number of condition attributes. The reason can be stated as, the computational complexity of an algorithm is a measure of required steps for the algorithm in the worst case for a specific-sized input, and the number of steps is measured as a function of that input size. Based on the experimental results, we give the following observations and explanations.

1) NNB obtains the least training time. By comparing the discriminants of NNB, FNB, FNB_{ROT}, and NNBC, we find that NNB avoids some operations of exponential components $\exp(u)$, while each of FNB, FNB_{ROT}, and NNBC cannot avoid them.
2) The training time of NNBC is shorter than FNB and FNB_{ROT}'s. The discriminant of FNB and FNB_{ROT} can be expressed as follows:

$$P_1\left(\vec{\mathrm{x}}\,|w_k\right) = \prod_{j=1}^{d} \frac{1}{n_k h_j^{(k)}} \sum_{i=1}^{n_k} \left[K\left(\frac{x_j - x_{ij}^{(k)}}{h_j^{(k)}} \right) \right]. \quad (33)$$

And, from (32), we know the discriminant of NNBC is

$$P_2\left(\vec{\mathrm{x}}\,|w_k\right) = \frac{1}{n_k h_k^d} \sum_{i=1}^{n_k} \prod_{j=1}^{d} K\left(\frac{x_j - x_{ij}^{(k)}}{h_k} \right). \quad (34)$$

By comparing these two discriminants, we know that the computational time of (33) is obviously longer than (34)'s, since the product-of-sums in (33) includes the sum-of-products in (34).

In this experiment, we also compare NNBC with four boosting-based classification methods [1] (i.e., boostedNNB, boostedFNB, boostedC4.5, and boostedSVM) and SVM [16] in terms of training time and classification accuracy . The boostedNNB, boostedFNB, boostedC4.5, boostedSVM, and SVM are the standard WEKA [47] source programs. The experimental results are the averages of 100 runs of tenfold cross-validation. For the 30 UCI datasets, the comparative training time and classification accuracies are summarized in Tables II and III.[1]

The Wilcoxon signed-ranks test under the significance level of 0.1 is used to conduct the statistical analysis on the classification results listed in Table VI. The statistical results for comparing NNBC with boostedNNB, boostedFNB, boostedC4.5,

[1]The digit in parenthesis is the number of principal components.

TABLE III
CLASSIFICATION ACCURACIES OF OTHER SOPHISTICATED AND COMPLEX CLASSIFICATION METHODS

	Datasets	boostedNNB	boostedFNB	BoostedC4.5	BoostedSVM	SVM	PCANNB	PCAFNB
1	Auto Mpg	0.686	0.747	0.872	0.701	0.691	0.679 (3)	0.704 (3)
2	Blood Transfusion	0.770	0.770	0.786	0.769	0.762	0.773 (3)	0.769 (3)
3	Breast Cancer	0.956	0.956	0.957	0.970	0.970	0.953 (8)	0.960 (8)
4	Breast Cancer W-D	0.958	0.965	0.956	0.974	0.975	0.930 (10)	0.930 (10)
5	Breast Cancer W-P	0.672	0.646	0.732	0.768	0.763	0.737 (13)	0.773 (13)
6	Contraceptive Method	0.505	0.509	0.508	0.474	0.474	0.470 (2)	0.520 (2)
7	Credit Approval	0.703	0.726	0.754	0.717	0.712	0.642 (6)	0.630 (6)
8	Cylinder Bands	0.680	0.685	0.774	0.650	0.652	0.607 (20)	0.628 (20)
9	Ecoli	0.878	0.878	0.853	0.859	0.838	0.872 (4)	0.872 (4)
10	Glass Identification	0.477	0.514	0.780	0.589	0.570	0.523 (6)	0.500 (6)
11	Haberman's Survival	0.748	0.745	0.725	0.761	0.735	0.722 (3)	0.716 (3)
12	Heart Disease	0.815	0.822	0.781	0.830	0.833	0.815 (12)	0.789 (12)
13	Image Segment	0.805	0.920	0.982	0.928	0.929	0.869 (10)	0.900 (10)
14	Ionosphere	0.903	0.934	0.917	0.858	0.838	0.906 (23)	0.926 (23)
15	Iris	0.933	0.960	0.933	0.980	0.960	0.893 (2)	0.887 (2)
16	Libras Movement	0.636	–	0.822	–	0.744	0.669 (9)	0.700 (9)
17	Magic Telescope	0.800	0.795	0.851	0.794	0.794	0.768 (7)	0.764 (7)
18	Musk Version 1	0.777	–	0.884	0.853	0.823	0.800 (35)	0.826 (35)
19	New Thyroid Gland	0.958	0.963	0.935	0.953	0.898	0.967 (4)	0.967 (4)
20	Page Blocks	0.881	0.914	0.937	0.920	0.916	0.872 (6)	0.901 (6)
21	Parkinsons	0.692	0.831	0.903	0.887	0.867	0.800 (8)	0.836 (8)
22	Pima Indian Diabetes	0.756	0.732	0.728	0.772	0.772	0.730 (8)	0.758 (8)
23	Sonar	0.822	0.846	0.779	0.769	0.760	0.673 (30)	0.726 (30)
24	SPECTF Heart	0.704	0.805	0.790	0.783	0.797	0.738 (25)	0.719 (25)
25	Vehicle Silhouettes	0.495	0.559	0.717	0.616	0.617	0.480 (5)	0.550 (5)
26	Vowel Recognition	0.813	0.884	0.926	0.631	0.633	0.542 (8)	0.661 (8)
27	Wine	0.955	0.961	0.966	0.978	0.983	0.983 (10)	0.978 (10)
28	Wine Quality-Red	0.547	0.596	0.662	0.582	0.582	0.544 (9)	0.557 (9)
29	Wine Quality-White	0.472	0.487	0.577	0.534	0.532	0.513 (9)	0.528 (9)
30	Yeast	0.582	0.593	0.575	0.571	0.571	0.550 (8)	0.570 (8)

boostedSVM, and SVM are 18/2/10, 15/2/13, 7/1/22, 14/0/16, and 15/3/12, respectively, where $w/t/l$ means w wins, t ties and l losses for NNBC in all 30 data sets. We observe that, compared with the sophisticated boostedNNB, boostedFNB, and SVM, our NNBC still achieves a slight improvement on classification accuracy. Besides, in comparison with boostedC4.5 and boostedSVM, NNBC obtains relatively inferior classification performances. However, through avoiding the time-consuming weight update in boosting and complicated parameter optimization in SVM, our NNBC obtains the shortest training time among all the compared methods. Consequently, we summarize that NNBC can achieve a relatively satisfactory classification accuracy with less time requirements in comparison with the sophisticated and complex classification methods, i.e., boosting and SVM.

C. Comparison for Ranking Performance (AUC)

AUC [14], [40] is based on the fact that the cost of classifying a sample into the wrong class is significantly lower than the reverse. AUC takes the misclassification cost into account. Let there be n_0 testing samples in class 0 and n_1 testing samples in class 1. $p_i^{0\to 0}$ $(i = 1, 2, \ldots, n_0)$ denotes the estimated probability that the i-th 0-class testing sample belongs to class 0; $p_j^{1\to 0}$ $(j = 1, 2, \ldots, n_1)$ denotes the estimated probability that the jth 1-class testing sample belongs to class 0. Next, rank all $p_i^{0\to 0}$ and $p_j^{1\to 0}$ $(i = 1, 2, \ldots, n_0; j = 1, 2, \ldots, n_1)$ in an increasing order. Let r_i denote the rank of the i-th 0-class testing sample; then, the AUC of these can be defined as the following formula:

$$\mathrm{AUC}(0, 1) = \frac{\sum_{i=1}^{n_0} r_i - \frac{n_0(n_0+1)}{2}}{n_0 n_1}. \tag{35}$$

Based on the AUC of two classes, the definition of AUC for k $(k > 2)$ classes can be given by

$$\mathrm{AUC} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \mathrm{AUC}(i, j). \tag{36}$$

From (35), we know its maximum is 1, which can be achieved when all $p_j^{1\to 0}$ are smaller than any $p_i^{0\to 0}$ $(i = 1, 2, \ldots, n_0)$. Then, the maximum of (36) is also 1 when all $\mathrm{AUC}(i, j)$ $(i < j)$ are 1.

Table IV gives the detailed results for average AUCs and standard deviations of 100-runs tenfold cross-validation. Furthermore, the statistical results based on the sign test are summarized in Table VI. According to the experimental comparisons and statistical analyses, we conclude that NNBC indeed obtains the better ranking performance on the selected 30 UCI datasets compared with the other three models.

D. Comparison for the Quality of Class-Conditional Probability Density Estimation (PMSE)

The PMSE [20] is defined as

$$\mathrm{PMSE}(D) = \frac{1}{N} \sum_{\vec{\mathrm{x}} \in D} \sum_{p=1}^{c} \left[P\left(\vec{\mathrm{x}} \,|\, w_p\right) - t_p\left(\vec{\mathrm{x}}\right) \right]^2 \tag{37}$$

where D is the training dataset on which the estimated error is computed, N is the size of D, $P\left(\vec{\mathrm{x}} \,|\, w_p\right)$ $\left(\vec{\mathrm{x}} \in D, p = 1, 2, \ldots, c\right)$ denotes the estimated posterior probability of instance $\vec{\mathrm{x}}$ belonging to the class w_p, and $\vec{\mathrm{t}}\left(\vec{\mathrm{x}}\right) = \left\{t_1\left(\vec{\mathrm{x}}\right), t_2\left(\vec{\mathrm{x}}\right), \ldots, t_c\left(\vec{\mathrm{x}}\right)\right\}$ is a c-dimensional vector in which the pth component is 1 and the other components are 0 only if the actual class of $\vec{\mathrm{x}}$ is w_p.

TABLE IV
DETAILED EXPERIMENTAL RESULTS OF RANKING PERFORMANCE AND STANDARD DEVIATION ON 30 UCI BENCHMARK DATASETS

	Datasets	NNB	FNB	FNB_{ROT}	NNBC
1	Auto Mpg	0.799±0.008	0.803±0.004	0.830±0.005	0.889±0.004
2	Blood Transfusion	0.707±0.002	0.695±0.004	0.715±0.002	0.725±0.006
3	Breast Cancer	0.975±0.001	0.993±0.002	0.981±0.025	0.978±0.001
4	Breast Cancer W-D	0.987±0.001	0.987±0.002	0.989±0.001	0.989±0.002
5	Breast Cancer W-P	0.666±0.015	0.618±0.008	0.647±0.018	0.648±0.017
6	Contraceptive Method	0.676±0.001	0.668±0.002	0.671±0.001	0.696±0.001
7	Credit Approval	0.790±0.005	0.790±0.003	0.833±0.003	0.783±0.004
8	Cylinder Bands	0.702±0.003	0.788±0.002	0.771±0.002	0.760±0.006
9	Ecoli	0.973±0.002	0.966±0.001	0.964±0.002	0.969±0.002
10	Glass Identification	0.772±0.012	0.725±0.009	0.787±0.012	0.843±0.009
11	Haberman's Survival	0.644±0.016	0.689±0.012	0.692±0.008	0.659±0.012
12	Heart Disease	0.899±0.007	0.903±0.006	0.908±0.007	0.834±0.006
13	Image Segment	0.974±0.002	0.986±0.001	0.987±0.002	0.994±0.001
14	Ionosphere	0.809±0.005	0.905±0.003	0.903±0.004	0.937±0.004
15	Iris	0.987±0.005	0.986±0.005	0.988±0.005	0.990±0.002
16	Libras Movement	0.960±0.005	0.954±0.004	0.964±0.005	0.986±0.002
17	Magic Telescope	0.756±0.001	0.830±0.001	0.808±0.002	0.889±0.001
18	Musk Version 1	0.816±0.003	0.894±0.004	0.901±0.003	0.948±0.003
19	New Thyroid Gland	0.995±0.002	0.995±0.002	0.994±0.002	0.993±0.003
20	Page Blocks	0.650±0.012	0.586±0.022	0.646±0.004	0.540±0.007
21	Parkinsons	0.851±0.006	0.902±0.009	0.860±0.006	0.974±0.008
22	Pima Indian Diabetes	0.814±0.003	0.824±0.003	0.828±0.004	0.795±0.002
23	Sonar	0.799±0.007	0.861±0.007	0.845±0.008	0.921±0.008
24	SPECTF Heart	0.845±0.009	0.777±0.010	0.849±0.008	0.797±0.010
25	Vehicle Silhouettes	0.770±0.002	0.783±0.002	0.798±0.003	0.859±0.004
26	Vowel Recognition	0.965±0.001	0.943±0.002	0.984±0.001	0.999±0.000
27	Wine	1.000±0.001	0.999±0.001	1.000±0.000	1.000±0.000
28	Wine Quality-Red	0.722±0.002	0.712±0.002	0.769±0.002	0.811±0.003
29	Wine Quality-White	0.489±0.004	0.481±0.002	0.492±0.004	0.510±0.004
30	Yeast	0.755±0.002	0.749±0.001	0.782±0.002	0.808±0.002
	Average	**0.818±0.005**	**0.826±0.005**	**0.840±0.005**	**0.851±0.004**

The experimental results of PMSE on 30 UCI datasets are summarized in Table V in detail. And, the statistical results based on the sign test are shown in Table VI.

According to the sign test, the numbers of wins of NNBC on these 30 UCI datasets are equal or greater than 20 in comparison with the other three learning models. It indicates that the PMSE of NNBC is the best in these 30 datasets, which implies that NNBC obtains the most accurate probability estimation. The reason why NNBC can acquire the best classification accuracy and ranking may reside here.

The following contents are used to depict how to use PMSE to evaluate the estimation qualities of different density estimation methods. Now we assume $\vec{\mathrm{x}}$ is an instance in a two-class problem, and the true class of $\vec{\mathrm{x}}$ is w_1, namely $\vec{\mathrm{t}}\left(\vec{\mathrm{x}}\right) = \{1, 0\}$. Let $\mathrm{M_A}$ and $\mathrm{M_B}$ be two different density estimation methods. $P_i\left(\vec{\mathrm{x}}\,|w_1\right)$ and $P_i\left(\vec{\mathrm{x}}\,|w_2\right)$ estimated by $\mathrm{M}_i(i = A \text{ or } B)$ represent the estimated posterior probabilities of $\vec{\mathrm{x}}$ belonging to the class w_1 and w_2 respectively. The PMSEs of $\mathrm{M_A}$ and $\mathrm{M_B}$ can be calculated in the following expressions:

$$\begin{cases} \mathrm{PMSE_A}\left(\vec{\mathrm{x}}\right) = \left[P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right]^2 + \left[P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right]^2 \\ \mathrm{PMSE_B}\left(\vec{\mathrm{x}}\right) = \left[P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right]^2 + \left[P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right]^2. \end{cases} \tag{38}$$

Because $P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_1\right) + P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_2\right) = 1$ and $P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_1\right) + P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_2\right) = 1$, (38) can be rewritten as follows:

$$\begin{cases} \mathrm{PMSE_A}\left(\vec{\mathrm{x}}\right) = 2\left[P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right]^2 \\ \mathrm{PMSE_B}\left(\vec{\mathrm{x}}\right) = 2\left[P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right]^2 \end{cases} \tag{39}$$

or

$$\begin{cases} \mathrm{PMSE_A}\left(\vec{\mathrm{x}}\right) = 2\left[P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right]^2 \\ \mathrm{PMSE_B}\left(\vec{\mathrm{x}}\right) = 2\left[P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right]^2. \end{cases} \tag{40}$$

If $\mathrm{PMSE_A}\left(\vec{\mathrm{x}}\right) < \mathrm{PMSE_B}\left(\vec{\mathrm{x}}\right)$, then, we can derive $\left|P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right| < \left|P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_1\right) - 1\right|$ or $\left|P_{\mathrm{A}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right| < \left|P_{\mathrm{B}}\left(\vec{\mathrm{x}}\,|w_2\right) - 0\right|$. It reveals that, compared with $\mathrm{M_B}$, $\mathrm{M_A}$ can obtain the better performance of p.d.f. estimation, i.e., $\mathrm{M_A}$ further enables the estimated posterior probability to reach the true probability. In other words, $\mathrm{M_A}$ derives the correct probability of $\vec{\mathrm{x}}$ belonging to the class w_1 to be closer to 1 or the wrong probability of $\vec{\mathrm{x}}$ belonging to the class w_2 to be closer to 0.

In order to give more explanations, we select 11 real samples in Vowel Recognition dataset to compare their PMSEs based on four Bayesian models (NNB, FNB, $\mathrm{FNB_{ROT}}$, and NNBC). Because there are 11 classes in Vowel Recognition dataset, we select one sample from each class. The 11 selected samples (other samples are used to train the classifiers) are listed in Table VII.

For the 11 instances, we calculate the 11 components of $\mathrm{PMSE}\left(\vec{\mathrm{x}}\right) = \sum_{p=1}^{11}\left[P\left(\vec{\mathrm{x}}\,|w_p\right) - t_p\left(\vec{\mathrm{x}}\right)\right]^2$ based on the four different estimation methods, respectively. The details of comparative results are summarized in Table VIII. From Table VIII, we extract that for each sample, when it belongs to the pth class, its pth component $\Delta\mathrm{E}_p = \left|P\left(\vec{\mathrm{x}}\,|w_p\right) - t_p\left(\vec{\mathrm{x}}\right)\right|$ $(p = 1, 2, \ldots, 11)$ computed by NNBC is the smallest compared with the other three results. For example, for the instance $\vec{\mathrm{x}}_6$ belonging to the sixth class in Table VII, the $\left|P\left(\vec{\mathrm{x}}\,|w_6\right) - t_6\left(\vec{\mathrm{x}}\right)\right|$ computed by these four estimation

TABLE V
DETAILED EXPERIMENTAL RESULTS OF ESTIMATION QUALITY AND STANDARD DEVIATION ON 30 UCI BENCHMARK DATASETS

	Datasets	**NNB**	**FNB**	**FNB$_{\mathrm{ROT}}$**	**NNBC**
1	Auto Mpg	0.478±0.003	0.449±0.002	0.438±0.003	0.367±0.002
2	Blood Transfusion	0.399±0.001	0.393±0.002	0.423±0.002	0.370±0.003
3	Breast Cancer	0.076±0.001	0.049±0.001	0.065±0.001	0.073±0.002
4	Breast Cancer W-D	0.126±0.004	0.105±0.002	0.097±0.003	0.087±0.003
5	Breast Cancer W-P	0.622±0.016	0.526±0.012	0.618±0.015	0.781±0.014
6	Contraceptive Method	0.628±0.001	0.618±0.001	0.616±0.001	0.597±0.002
7	Credit Approval	0.505±0.003	0.383±0.002	0.410±0.002	0.480±0.008
8	Cylinder Bands	0.521±0.011	0.410±0.007	0.440±0.014	0.496±0.014
9	Ecoli	0.221±0.003	0.222±0.001	0.216±0.004	0.219±0.007
10	Glass Identification	0.963±0.019	0.585±0.005	0.785±0.015	0.541±0.007
11	Haberman's Survival	0.419±0.003	0.396±0.004	0.402±0.005	0.426±0.003
12	Heart Disease	0.263±0.006	0.262±0.005	0.253±0.005	0.451±0.018
13	Image Segment	0.374±0.003	0.161±0.001	0.271±0.002	0.053±0.002
14	Ionosphere	0.341±0.005	0.171±0.004	0.176±0.005	0.117±0.008
15	Iris	0.074±0.003	0.069±0.004	0.069±0.003	0.056±0.006
16	Libras Movement	0.689±0.022	0.746±0.010	0.604±0.018	0.361±0.013
17	Magic Telescope	0.470±0.000	0.338±0.001	0.377±0.001	0.336±0.002
18	Musk Version 1	0.515±0.011	0.376±0.013	0.417±0.008	0.364±0.011
19	New Thyroid Gland	0.062±0.002	0.130±0.001	0.063±0.001	0.115±0.001
20	Page Blocks	0.241±0.016	0.218±0.001	0.155±0.007	0.172±0.006
21	Parkinsons	0.596±0.007	0.290±0.006	0.549±0.005	0.270±0.009
22	Pima Indian Diabetes	0.364±0.003	0.347±0.003	0.344±0.004	0.376±0.006
23	Sonar	0.571±0.013	0.351±0.013	0.421±0.008	0.322±0.013
24	SPECTF Heart	0.638±0.008	0.371±0.008	0.569±0.007	0.586±0.004
25	Vehicle Silhouettes	0.670±0.003	0.612±0.003	0.609±0.003	0.489±0.004
26	Vowel Recognition	0.456±0.007	0.601±0.004	0.307±0.006	0.058±0.003
27	Wine	0.042±0.007	0.056±0.002	0.032±0.004	0.020±0.003
I28	Wine Quality-Red	0.814±0.005	0.573±0.002	0.642±0.002	0.553±0.006
29	Wine Quality-White	0.889±0.006	0.648±0.002	0.793±0.006	0.676±0.006
30	Yeast	0.760±0.004	0.575±0.002	0.658±0.004	0.626±0.004
	Average	**0.460±0.007**	**0.368±0.004**	**0.394±0.006**	**0.348±0.006**

methods are $|0.842-1|$, $|0.786-1|$, $|0.797-1|$, and $|1.000-1|$ for NNB, ENB, FNB$_{\mathrm{ROT}}$, and NNBC, respectively. The correct posterior probability of $\vec{\mathrm{x}}_6$ belonging to the class w_6 is almost 1. Meanwhile, we also find that the performances of NNBC are the best for all other components of PMSE. It shows that NNBC can indeed obtain a more accurate estimation of joint p.d.f.

From our experiments, we can observe that the performance of NNBC, not only in classification accuracy and ranking performance but also in estimated quality, is overall the best among the models discussed in the paper. Now, we summarize some highlights briefly as follows.

1) NNBC statistically outperforms NNB in classification accuracy (25 wins and five losses), AUC (21 wins and eight losses), and PMSE (25 wins and five losses);
2) NNBC statistically outperforms FNB in classification accuracy (20 wins and eight losses), AUC (22 wins and eight losses), and PMSE (20 wins and ten losses);
3) NNBC statistically outperforms FNB$_{\mathrm{ROT}}$ in classification accuracy (21 wins and nine losses), PMSE (20 wins and ten losses), and slightly outperforms FNB$_{\mathrm{ROT}}$ in AUC (19 wins and nine losses).

In summary, this section experimentally confirms that, compared with NNB, FNB, and FNB$_{\mathrm{ROT}}$, NNBC presents the best performance in terms of the three indexes: classification accuracy, ranking, and estimation quality. When handling the classification tasks with continuous attributes, the main reasons that the proposed NNBC is more effective than other state-of-the-art p.d.f. estimation-based Bayesian classifiers can be listed as follows.

TABLE VI
DETAILED EXPERIMENTAL RESULTS OF ESTIMATION QUALITY AND STANDARD DEVIATION ON 30 UCI BENCHMARK DATASETS

	NNBC Versus NNB ($win/tie/loss$)	**NNBC Versus FNB** ($win/tie/loss$)	**NNBC Versus FNB$_{\mathrm{ROT}}$** ($win/tie/loss$)
Accuracy	25/0/5	20/2/8	21/0/9
AUC	21/1/8	22/0/8	19/2/9
PMSE	25/0/5	20/0/10	20/0/10

1) It relaxes the independence assumption among continuous attributes. It is universally acknowledged that the independence assumption is regarded as one of the main factors which limit the classification performance of naive Bayesian. Currently, the main developing schemes of relaxing this independence assumption include modifying the structure of traditional NBC [19], [28], projecting the attributes to other subspace [11], [18], and proving the inefficiency of independence assumption [4], [9]. Trying to solve this problem using the first scheme, NNBC removes the independence assumption by replacing the product of marginal p.d.f. with a joint p.d.f. when determining the class label for a new instance. Through expressing the class-conditional p.d.f.s as a joint p.d.f., the NNBC considers the dependence among continuous attributes in an appropriate and effective way.
2) It acquires a more accurate estimation of class-conditional p.d.f. by computing the optimal bandwidth for the joint p.d.f. Many references show that a more accurate density estimation can improve the classification performance of Bayesian classifiers. For example, the methods given in [19] by replacing Gaussian approxi-

TABLE VII

ELEVEN REPRESENTATIVE INSTANCES IN THE VOWEL RECOGNITION DATASET ($p = 1, 2, \ldots, 11$)

	x_{p1}	x_{p2}	x_{p3}	x_{p4}	x_{p5}	x_{p6}	x_{p7}	x_{p8}	x_{p9}	$x_{p,10}$	w_p
$\vec{x}_1$	–3.844	1.056	–0.19	1.685	0.617	1.245	–0.811	–0.506	–1.128	0.076	1
$\vec{x}_2$	–3.249	1.042	0.589	1.408	0.023	–0.821	–0.581	0.031	0.068	0.325	2
$\vec{x}_3$	–2.03	1.764	–0.386	–0.249	0.18	0.117	0.096	–0.121	0.067	–0.552	3
$\vec{x}_4$	–2.748	3.217	–0.976	–0.213	–0.792	0.771	–0.032	0.223	0.043	–0.825	4
$\vec{x}_5$	–2.497	1.607	–0.621	–0.446	–0.226	–0.152	1.16	0.122	–0.809	0.495	5
$\vec{x}_6$	–3.587	3.128	0.885	–0.188	–1.164	–0.215	0.051	1.334	0.641	–0.253	6
$\vec{x}_7$	–3.675	3.132	–0.241	1.587	–1.75	–0.222	0.039	1.052	0.545	0.233	7
$\vec{x}_8$	–4.079	2.663	–0.048	–0.315	0.234	0.861	0.335	0.435	–0.546	–0.928	8
$\vec{x}_9$	–4.188	2.637	0.502	0.552	0.735	0.395	–0.026	0.803	–0.874	–0.913	9
$\vec{x}_{10}$	–4.102	0.209	0.414	0.423	0.985	1.434	0.663	0.036	–0.784	–0.668	10
$\vec{x}_{11}$	–2.91	0.918	–0.138	–0.382	0.115	0.29	0.418	0.757	–0.898	–0.189	11

TABLE VIII

DETAILED DESCRIPTION TO PMSE OF 11 REPRESENTATIVE INSTANCES IN THE VOWEL RECOGNITION DATASET, WHERE $\Delta E_p = \left| P\left(\vec{x} \left| w_p \right.\right) - t_p\left(\vec{x}\right) \right|$ ($p = 1, 2, \ldots, 11$)

		ΔE_1	ΔE_2	ΔE_3	ΔE_4	ΔE_5	ΔE_6	ΔE_7	ΔE_8	ΔE_9	ΔE_{10}	ΔE_{11}
$\vec{x}_1$	NNB	$\lvert 0.439-1\rvert$	$\lvert 0.097-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.460-0\rvert$	$\lvert 0.002-0\rvert$
	FNB	$\lvert 0.836-1\rvert$	$\lvert 0.114-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.036-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.013-0\rvert$
	FNB_{ROT}	$\lvert 0.420-1\rvert$	$\lvert 0.409-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.006-0\rvert$	$\lvert 0.156-0\rvert$	$\lvert 0.009-0\rvert$
	NNBC	$\lvert 0.982-1\rvert$	$\lvert 0.017-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_2$	NNB	$\lvert 0.189-0\rvert$	$\lvert 0.788-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.019-0\rvert$	$\lvert 0.001-0\rvert$
	FNB	$\lvert 0.110-0\rvert$	$\lvert 0.890-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0.036-0\rvert$	$\lvert 0.949-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.013-0\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0.001-0\rvert$	$\lvert 0.999-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_3$	NNB	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.736-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.200-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.062-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.889-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.107-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.003-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.810-1\rvert$	$\lvert 0.003-0\rvert$	$\lvert 0.164-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.023-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.999-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_4$	NNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.874-1\rvert$	$\lvert 0.018-0\rvert$	$\lvert 0.071-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.036-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.862-1\rvert$	$\lvert 0.003-0\rvert$	$\lvert 0.134-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.846-1\rvert$	$\lvert 0.034-0\rvert$	$\lvert 0.108-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.012-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.998-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0.002-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_5$	NNB	$\lvert 0.013-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.003-0\rvert$	$\lvert 0.193-0\rvert$	$\lvert 0.608-1\rvert$	$\lvert 0.162-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.021-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.253-0\rvert$	$\lvert 0.738-1\rvert$	$\lvert 0.009-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.274-0\rvert$	$\lvert 0.658-1\rvert$	$\lvert 0.067-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.002-0\rvert$	$\lvert 0.998-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_6$	NNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.005-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.842-1\rvert$	$\lvert 0.132-0\rvert$	$\lvert 0.019-0\rvert$	$\lvert 0.002-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.016-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.786-1\rvert$	$\lvert 0.198-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.032-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.797-1\rvert$	$\lvert 0.164-0\rvert$	$\lvert 0.004-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.982-1\rvert$	$\lvert 0.018-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_7$	NNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.186-0\rvert$	$\lvert 0.801-1\rvert$	$\lvert 0.013-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.051-0\rvert$	$\lvert 0.948-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.097-0\rvert$	$\lvert 0.897-1\rvert$	$\lvert 0.004-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 1.000-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_8$	NNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.010-0\rvert$	$\lvert 0.526-1\rvert$	$\lvert 0.462-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.010-0\rvert$	$\lvert 0.868-1\rvert$	$\lvert 0.115-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.003-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.003-0\rvert$	$\lvert 0.012-0\rvert$	$\lvert 0.050-0\rvert$	$\lvert 0.676-1\rvert$	$\lvert 0.244-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.015-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.002-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.991-1\rvert$	$\lvert 0.007-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_9$	NNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.041-0\rvert$	$\lvert 0.408-0\rvert$	$\lvert 0.548-1\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$
	FNB	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.271-0\rvert$	$\lvert 0.729-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.023-0\rvert$	$\lvert 0.478-1\rvert$	$\lvert 0.495-1\rvert$	$\lvert 0.003-0\rvert$	$\lvert 0.001-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.044-1\rvert$	$\lvert 0.956-1\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$
$\vec{x}_{10}$	NNB	$\lvert 0.207-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0.039-0\rvert$	$\lvert 0.748-1\rvert$	$\lvert 0-0\rvert$
	FNB	$\lvert 0.077-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.923-1\rvert$	$\lvert 0-0\rvert$
	FNB_{ROT}	$\lvert 0.267-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.007-0\rvert$	$\lvert 0.024-0\rvert$	$\lvert 0.701-1\rvert$	$\lvert 0-0\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 1.000-1\rvert$	$\lvert 0-0\rvert$
$\vec{x}_{11}$	NNB	$\lvert 0.074-0\rvert$	$\lvert 0.084-0\rvert$	$\lvert 0.012-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.222-0\rvert$	$\lvert 0.012-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.021-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.574-1\rvert$
	FNB	$\lvert 0.002-0\rvert$	$\lvert 0.020-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.016-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.960-1\rvert$
	FNB_{ROT}	$\lvert 0.010-0\rvert$	$\lvert 0.052-0\rvert$	$\lvert 0.005-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.084-0\rvert$	$\lvert 0.007-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.008-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.833-1\rvert$
	NNBC	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.001-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0-0\rvert$	$\lvert 0.999-1\rvert$

mation with a kernel density estimation significantly improve the classification performances of naive Bayesian and, and the kernel density estimation-based Bayesian classifiers in [28] are usually superior to the traditional Bayesian classifiers which deal with the continuous attributes through discretization techniques. Our proposed NNBC applies the joint p.d.f. density estimation strategy and further determines the optimal bandwidth parameter in order to obtain a more accurate class-conditional p.d.f.

V. DISCUSSION

In this section, further discuss the dependence among the attributes. Firstly, a theoretical theorem demonstrated that, when the condition attributes are dependent, the joint p.d.f. estimation can find a better substitution for the underlying p.d.f. in comparison with the product of marginal p.d.f. estimations. The theorem shows that the joint p.d.f. estimation is optimal in the L_2 sense. Subsequently, we investigate the relationship between the classification accuracy and dependence among the attributes. The empirical results reflect that

TABLE IX
DETAILS OF DEPENDENCE BETWEEN ATTRIBUTES ON 20 UCI DATASETS

Datasets	**The bound of $\|R\|$**	**The pairs of dependent attributes** (A_i, A_j, R)
Auto Mpg	> 0.800	(1, 2, −0.805), (1, 4, −0.832), (2, 3, 0.897), (2, 4, 0.933), (3, 40.865)
Blood Transfusion	> 0.600	(2, 3, 1.000), (2, 4, 0.635), (3, 4, 0.635)
Breast Cancer W-D	> 0.970	(1, 3, 0.998), (1, 4, 0.987), (3, 4, 0.987), (3, 23, 0.970), (11, 13, 0.973), (21, 23, 0.994), (21, 24, 0.984), (23, 24, 0.978)
Contraceptive Method	> 0.500	(1, 2, 0.540)
Ecoli	> 0.800	(4, 5, 0.829)
Glass Identification	> 0.500	(1, 5, −0.542), (1, 7, 0.810)
Image Segment	> 0.900	(10, 11, 0.998), (10, 12, 0.996), (10, 13, 0.996), (10, 17, 0.997), (11, 12, 0.991), (11, 13, 0.994), (11, 17, 0.992), (12, 13, 0.985), (12, 17, 0.999), (13, 17, 0.990)
Ionosphere	> 0.680	(9, 15, 0.748), (11, 13, 0.826), (11, 19, 0.688), (13, 15, 0.699), (13, 19, 0.741), (15, 17, 0.685), (29, 31, 0.692)
Iris	> 0.800	(1, 3, 0.872), (1, 4, 0.818), (3, 4, 0.963)
Libras Movement	> 0.995	(1, 3, 1.000), (1, 5, 0.999), (1, 7, 0.996), (2, 4, 1.000), (2, 6, 0.999), (2, 8, 0.996), (3, 5, 1.000), (3, 7, 0.998), (4, 6, 1.000), (4, 8, 0.998), (5, 7, 0.999), (6, 8, 0.999), (7, 9, 0.997), (8, 10, 0.997), (10, 12, 0.996)
Magic Telescope	> 0.750	(1, 2, 0.787), (3, 4, −0.847), (3, 5, −0.801), (4, 5, 0.975)
Musk Version 1	> 0.980	(7, 82, 0.990), (7, 86, 0.988), (7, 119, 0.982), (9, 52, 0.983), (30, 121, 0.990), (30, 128, 0.989), (41, 71, 0.983), (41, 101, 0.984), (57, 100, 0.991), (57, 100, 0.991), (57, 119, 0.987), (100, 143, 0.988)
Page Blocks	> 0.800	(3, 8, 0.866), (3, 9, 0.945), (3, 10, 0.802), (8, 9, 0.932), (9, 10, 0.814)
Parkinsons	>0.980	(4, 6, 0.990), (4, 8, 0.990), (6, 8, 1.000), (9, 10, 0.987), (9, 11, 0.988), (9, 12, 0.983), (9, 14, 0.988), (11, 14, 1.000)
Sonar	> 0.875	(9, 10, 0.877), (15, 16, 0.913), (16, 17, 0.899), (17, 18, 0.926), (18, 19, 0.875), (20, 21, 0.905), (36, 37, 0.886), (48, 49, 0.895)
Vehicle Silhouettes	> 0.800	(1, 2, 0.950), (4, 8, −0.805), (7, 8, 0.894)
Vowel Recognition	> 0.800	(1, 3, −0.806), (1, 4, −0.848), (1, 8, 0.856), (3, 6, −0.828), (3, 8, −0.941), (4, 9, −0.807), (5, 8, −0.815), (6, 10, −0.821)
Wine	> 0.600	(1, 13, 0.644), (6, 7, 0.865), (6, 9, 0.612), (6, 12, 0.700), (7, 9, 0.653), (7, 12, 0.787)
Wine Quality-Red	> 0.650	(1, 3, 0.672), (1, 8, 0.668), (1, 9, −0.683), (6, 7, 0.668)
Wine Quality-White	> 0.800	(4, 8, 0.839), (8, 11, −0.826)

the Bayesian classifier (NNBC) that considers the dependence among the attributes can gain a better classification performance; conversely, the classifiers (NNB, FNB, and FNB_{ROT}) based on the independence assumption usually achieve inferior performances in classification accuracy.

A. Joint p.d.f. Estimation Is Optimal in L_2 Sense When the Attributes Are Dependent

As mentioned above, the joint p.d.f. estimation in NNBC removes the fundamental assumption of independence among the attributes, but also obtains a better estimation of p.d.f. when the attributes are dependent. In order to explain the advantages of joint p.d.f. estimation, we give the following theoretical theorem. For simplicity, we consider the case of 2-D normal p.d.f. estimation.

Theorem 1: Given the N observed data $\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2, \ldots, \vec{\mathrm{x}}_N$ obeying the joint p.d.f. $f\left(\vec{\mathrm{x}}\right)$, where

$$\vec{\mathrm{x}}_i = \{x_{i1}, x_{i2}\} \, (1 \le i \le N)$$

and

$$f\left(\vec{\mathrm{x}}\right) = f\left(x_1, x_2\right) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{x_1^2 + x_1^2 - 2\rho x_1 x_2}{2\left(1-\rho^2\right)}\right] (\rho \neq 0). \quad (41)$$

Then, as $N \to \infty$

$$\int\int \left[f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}} < \int\int \left[f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}} \quad (42)$$

where $f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right)$ and $f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right)$ are the estimated p.d.f.s through the joint and marginal p.d.f. estimations, respectively

$$f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{K}\left(\frac{x_1 - x_{i1}}{h}, \frac{x_2 - x_{i2}}{h}\right) \quad (43)$$

and

$$f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right) = \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{x_1 - x_{i1}}{h}\right)\frac{1}{N}\sum_{i=1}^{N} K\left(\frac{x_2 - x_{i2}}{h}\right). \quad (44)$$

Proof: Let

$$\Delta = \int\int \left[f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}} - \int\int \left[f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}} = \mathrm{A} - \mathrm{B} - 2\mathrm{C} + 2\mathrm{D} \quad (45)$$

where $\mathrm{A} = \int\int f_{\mathrm{J}}^2\left(\vec{\mathrm{x}}\right) d\vec{\mathrm{x}}$, $\mathrm{B} = \int\int f_{\mathrm{M}}^2\left(\vec{\mathrm{x}}\right) d\vec{\mathrm{x}}$, $\mathrm{C} = \int\int f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) d\vec{\mathrm{x}}$, and $\mathrm{D} = \int\int f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right) f\left(\vec{\mathrm{x}}\right) d\vec{\mathrm{x}}$. ■

Next, based on the given $f\left(\vec{\mathrm{x}}\right)$, $f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right)$, and $f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right)$ in (41), (43), and (44), we derive the expressions for A [in (46)], B [in (47)], C [in (48)], and D [in (49)], respectively

$$\mathrm{A} = \frac{1}{4\pi N h^2} + \frac{1}{4\pi N^2 h^2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left\{\exp\left[-\frac{1}{4}\sum_{k=1}^{2}\left(\frac{x_{ik} - x_{jk}}{h}\right)^2\right]\right\} = \frac{1}{4\pi N h^2} + o\left(\frac{1}{N}\right) \quad (46)$$

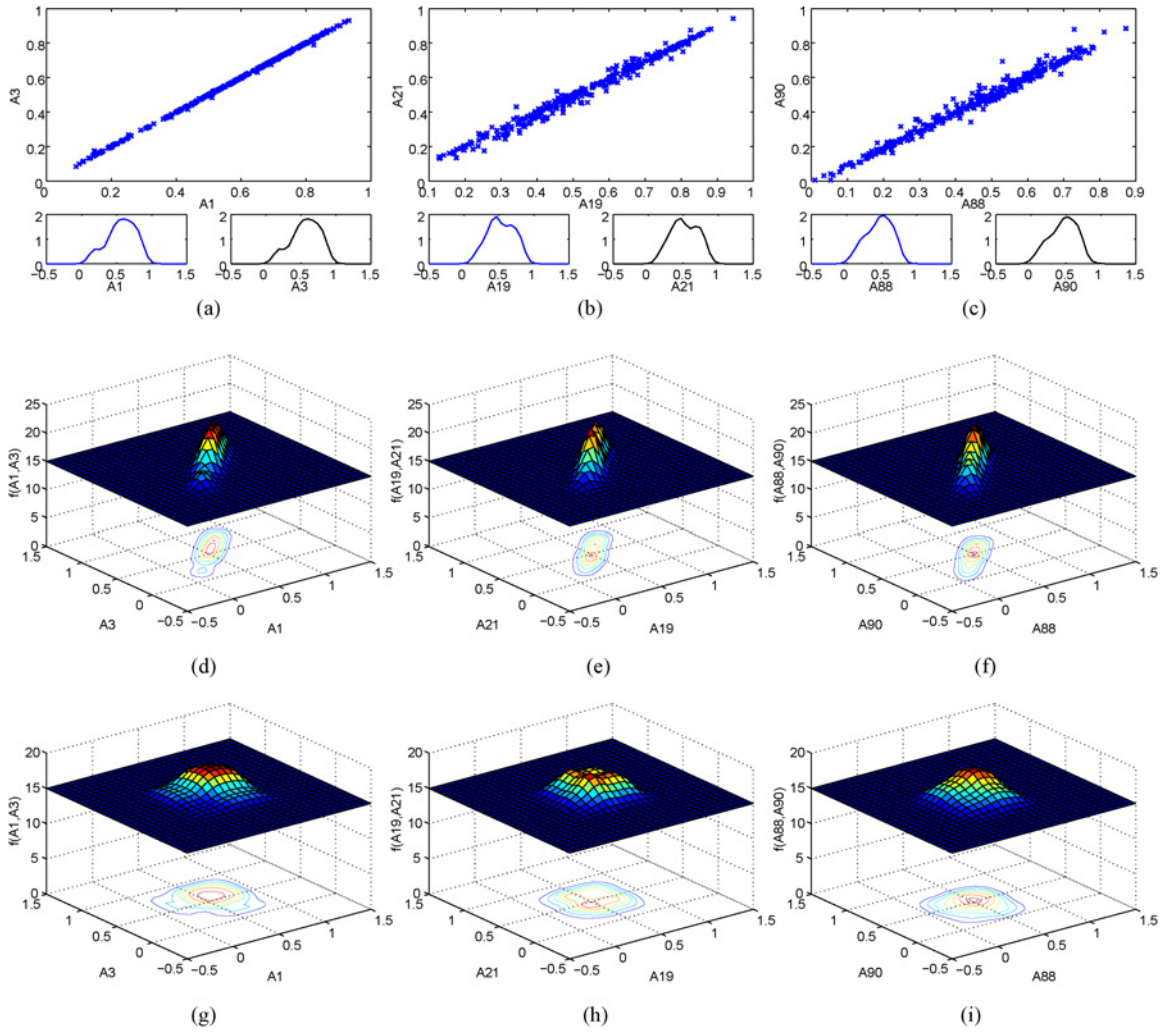


Fig. 1. Data distributions and joint p.d.f.s of the dependent attributes in Libras Movement dataset. (a) Data distribution of A_1 and A_3. (b) Data distribution of A_{19} and A_{21}. (c) Data distribution of A_{88} and A_{90}. (d) $f(A_1, A_3)$ with joint p.d.f. estimation. (e) $f(A_{19}, A_{21})$ with joint p.d.f. estimation. (f) $f(A_{88}, A_{90})$ with joint p.d.f. estimation. (g) $f(A_1, A_3)$ with marginal p.d.f. estimation. (h) $f(A_{19}, A_{21})$ with marginal p.d.f. estimation. (i) $f(A_{88}, A_{90})$ with marginal p.d.f. estimation.

$$\mathrm{B} = \frac{1}{4\pi N^2 h^2} + \frac{1}{4\pi N^4 h^2} \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N} \exp\left\{-\frac{1}{4}\left[\left(\frac{x_{i1}-x_{p1}}{h}\right)^2 + \left(\frac{x_{j2}-x_{q2}}{h}\right)^2\right]\right\} = o\left(\frac{1}{N}\right) \tag{47}$$

$$\mathrm{C} = \frac{\sqrt{1-\rho^2}}{2\pi N\left(1-\rho^2+h^2\right)} \sum_{i=1}^{N} \exp\left(-\frac{x_{i1}^2 + x_{i2}^2}{2\left(1-\rho^2+h^2\right)}\right) \tag{48}$$

$$\mathrm{D} = \frac{\sqrt{1-\rho^2}}{2\pi N^2\left(1-\rho^2+h^2\right)} \sum_{i=1}^{N}\sum_{j=1}^{N} \exp\left(-\frac{x_{i1}^2 + x_{j2}^2}{2\left(1-\rho^2+h^2\right)}\right) = o\left(\frac{1}{N}\right). \tag{49}$$

Hence, we gain

$$\Delta = \frac{1}{4\pi N h^2} - \frac{\sqrt{1-\rho^2}}{\pi N\left(1-\rho^2+h^2\right)} \sum_{i=1}^{N} \exp\left(-\frac{x_{i1}^2 + x_{i2}^2}{2\left(1-\rho^2+h^2\right)}\right) \tag{50}$$

provided $N \to \infty$.

Let

$$t = \min_{i=1,2,\cdots,N} \exp\left(-\frac{x_{i1}^2 + x_{i2}^2}{2\left(1-\rho^2+h^2\right)}\right) \tag{51}$$

then (50) can be rewritten as follows:

$$\Delta < \frac{1}{4\pi N h^2} - \frac{\sqrt{1-\rho^2}t}{\pi\left(1-\rho^2+h^2\right)}. \tag{52}$$

From (24), we know that $h = O\left[\left(\frac{1}{N}\right)^{\frac{1}{6}}\right]$. Thus, as $N \to \infty$

$$\frac{1}{4\pi N h^2} \to 0 \text{ and } \frac{\sqrt{1-\rho^2}\mathrm{t}}{\pi\left(1-\rho^2+\mathrm{h}^2\right)} \to \frac{\mathrm{t}}{\pi\sqrt{1-\rho^2}}.$$

In conclusion, when $N \to \infty$, we can get $\Delta < 0$, that is

$$\int\int \left[f_{\mathrm{J}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}} < \int\int \left[f_{\mathrm{M}}\left(\vec{\mathrm{x}}\right) - f\left(\vec{\mathrm{x}}\right)\right]^2 d\vec{\mathrm{x}}.$$

This completes the proof of the theorem. ∎

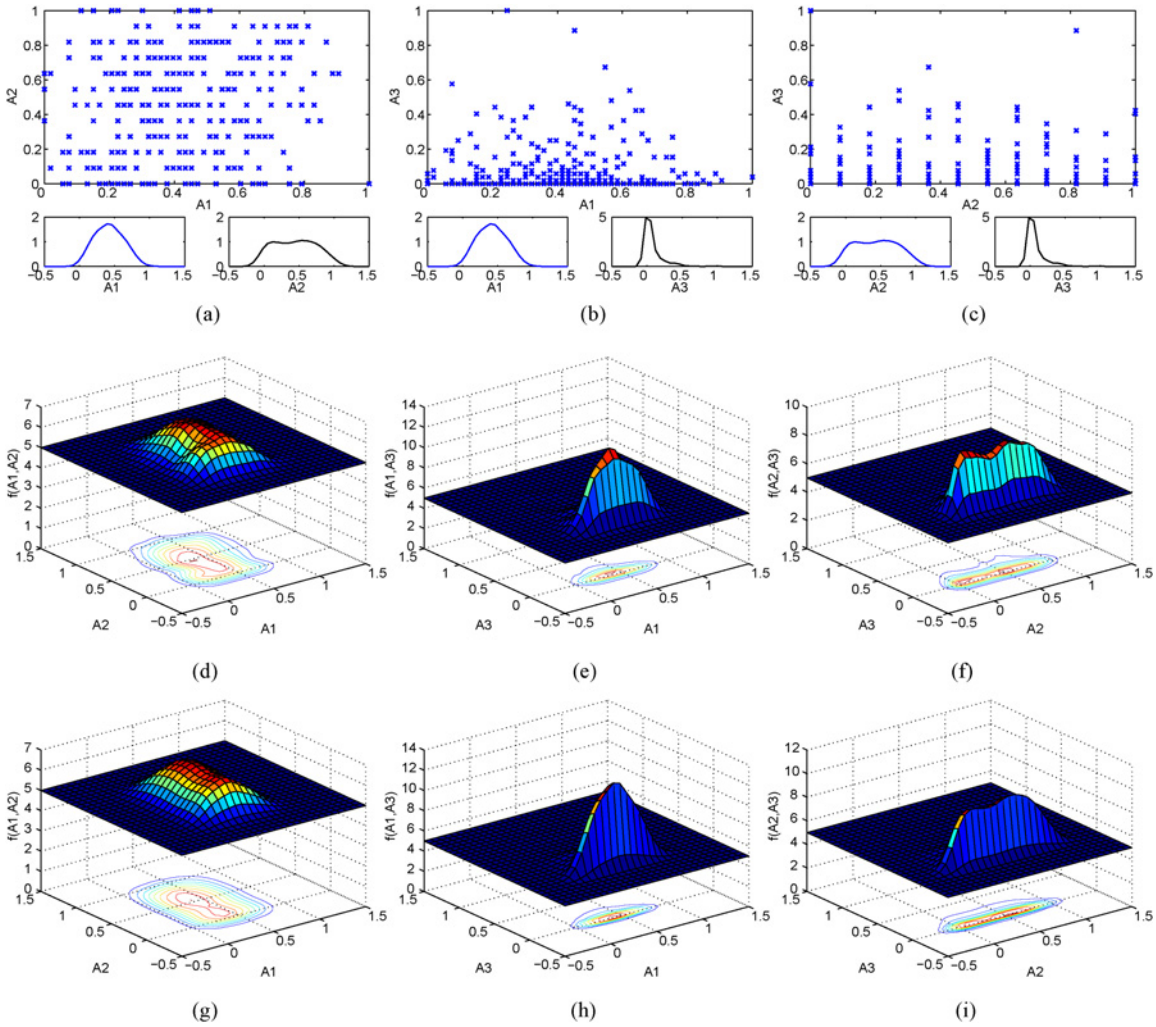


Fig. 2. Data distributions and joint p.d.f.s of the independent attributes in Haberman's Survival dataset. (a) Data distribution of A_1 and A_2. (b) Data distribution of A_1 and A_3. (c) Data distribution of A_2 and A_3. (d) $f(A_1, A_2)$ with joint p.d.f. estimation. (e) $f(A_1, A_3)$ with joint p.d.f. estimation. (f) $f(A_2, A_3)$ with joint p.d.f. estimation. (g) $f(A_1, A_2)$ with marginal p.d.f. estimation. (h) $f(A_1, A_3)$ with marginal p.d.f. estimation. (i) $f(A_2, A_3)$ with marginal p.d.f. estimation.

The theorem 1 reveals that when the dependence existed among the attributes, the joint p.d.f. estimation will make the estimated p.d.f. closer to the true p.d.f..

B. Relationship Between the Classification Accuracy and Dependence Among the Attributes

The previous experiments reflect that our proposed NNBC can obtain considerably better classification accuracy in comparison with the other Bayesian classifiers, i.e., NNB, FNB, and FNB_{ROT}. The main reason is that NNBC indeed relaxes the independence among attributes and acquires a more accurate estimation of the joint class-conditional p.d.f. Now, we investigate the relationship between the classification accuracy and dependence among the attributes. From the experimental results in Tables I and VI, we extract that on 20 UCI datasets NNBC obtains better classification accuracies than NNB, FNB, and FNB_{ROT}. In order to evaluate the impacts of dependence on the classification accuracy, we compute the dependent degrees R between any two attributes A_i and A_j. The details of dependence on these 20 UCI datasets are summarized in Table IX in which R is calculated according to the correlation coefficient [47].

From Table IX, we can see that all 20 UCI datasets contain pairs of attributes with strong dependence. For example, in Breast Cancer W-D, Image Segment, Libras Movement, Musk Version 1, and Parkinsons datasets, the numbers of pairs of attributes with strong dependence are 8 ($|R| > 0.970$), 10 ($|R| > 0.900$), 15 ($|R| > 0.995$), 11 ($|R| > 0.980$), and 8 ($|R| > 0.980$), respectively. We select three pairs of strongly dependent attributes in Libras Movement dataset, i.e., (A_1, A_3), (A_{19}, A_{21}), (A_{88}, A_{90}). The dependent degrees of these three pairs of attributes are 1.000, 0.992, and 0.991 respectively. Such dependence can be easily found from Fig. 1(a)–(c). Meanwhile, in order to further emphasize the optimality of joint p.d.f. estimation in Theorem 1, we give a comparison between the joint p.d.f. estimated by (43) and product of marginal p.d.f.s estimated by (44). We, respectively, estimate the joint p.d.f.s of dependent attributes pairs (A_1, A_3), (A_{19}, A_{22}), and (A_{88}, A_{90}) with the joint p.d.f. estimation [see Fig. 1(d)–(f)] and the product of marginal p.d.f. estimations [see Fig. 1(g)–(i)]. Although the true joint p.d.f.s of (A_1, A_3), (A_{19}, A_{22}), and (A_{88}, A_{90}) are unknown, we still get that the estimated results in Fig. 1(d)–(f) are better than the ones in Fig. 1(g)–(i). The characteristics of marginal p.d.f. can be

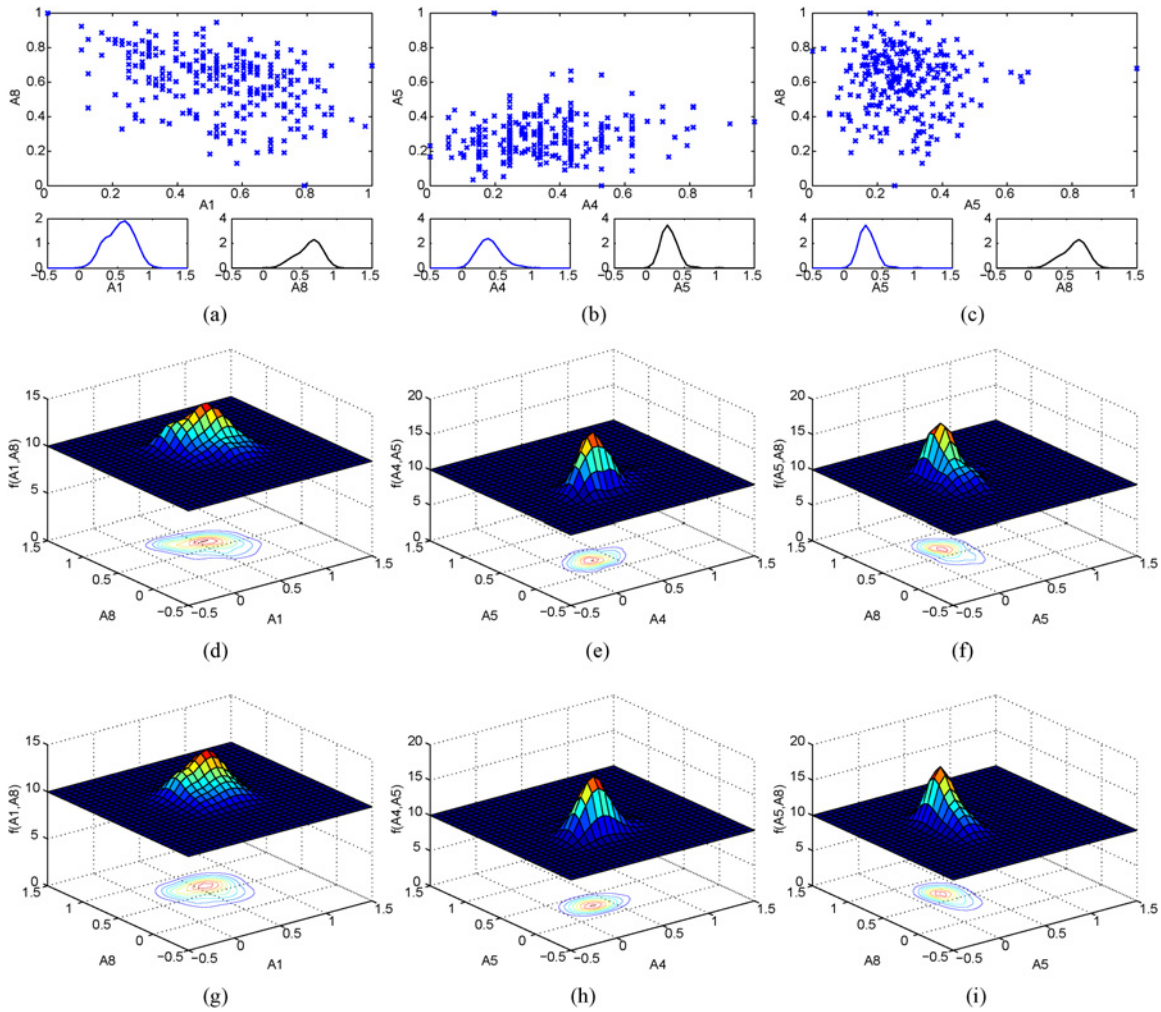


Fig. 3. Data distributions and joint p.d.f.s of the independent attributes in Heart Disease dataset. (a) Data distribution of A_1 and A_8. (b) Data distribution of A_4 and A_5. (c) Data distribution of A_5 and A_8. (d) $f(A_1, A_8)$ with joint p.d.f. estimation. (e) $f(A_4, A_5)$ with joint p.d.f. estimation. (f) $f(A_5, A_8)$ with joint p.d.f. estimation. (g) $f(A_1, A_8)$ with marginal p.d.f. estimation. (h) $f(A_4, A_5)$ with marginal p.d.f. estimation. (i) $f(A_5, A_8)$ with marginal p.d.f. estimation.

reflected in the joint p.d.f. estimated with (43) but the joint p.d.f. estimated with (44) cannot include these characteristics. For example, the marginal p.d.f. of attribute A_1 (or A_2) in Fig. 1(a) has two local maxima. Correspondingly, the joint p.d.f. of A_1 (or A_2) in Fig. 1(d) also has two local maxima which can be found in the contour of joint p.d.f.. However, the joint p.d.f. in Fig. 1(g) has only one local maximum. The same situation also exists in Fig. 1(b)–(h). This shows (43) can give a more accurate joint p.d.f. estimation for the dependent attributes of Libras Movement dataset in which NNBC obtains the better classification accuracy.

Figs. 2–4 also illustrate the estimated joint p.d.f.s for attribute pairs with weak dependence. Since the true joint p.d.f. is unknown, we compare the estimated performances of different methods by analyzing the characteristics of marginal p.d.f.s. For example, the two local maxima of marginal p.d.f. of attribute A_2 in Fig. 2(a) are not reflected in Fig. 2(d). Unsmooth contours in Fig. 3(e) and (f) are inconsistent with the marginal p.d.f.s without inflection points Fig. 3(b) and (c). Furthermore the marginal p.d.f. of attribute A_5 in Fig. 4(b) is closer to the projection of joint p.d.f. in Fig. 4(h). These experimental observations indicate that, in comparison with the estimation for product of marginal p.d.f.s, the joint p.d.f. estimation can not obtain a more accurate joint p.d.f. on Haberman's Survival, Heart Disease, and Pima Indian Diabetes.

From the above-mentioned analyses, we can extract that the dependence among attributes indeed affects the classification accuracy of NBC. When the conditional attributes are strongly dependent, NNBC will obtain much better classification accuracy by considering the dependence with the joint p.d.f. estimation.

VI. Conclusion and Future Works

In this paper, we proposed an NNBC which removes the independence assumption and estimates a more accurate class-conditional p.d.f. The main works of our paper can be concluded as follows: 1) the joint p.d.f. estimation was successfully applied to approximating the class-conditional p.d.f. in naive Bayesian, which effectively removed the traditional independence assumption and took the dependence among

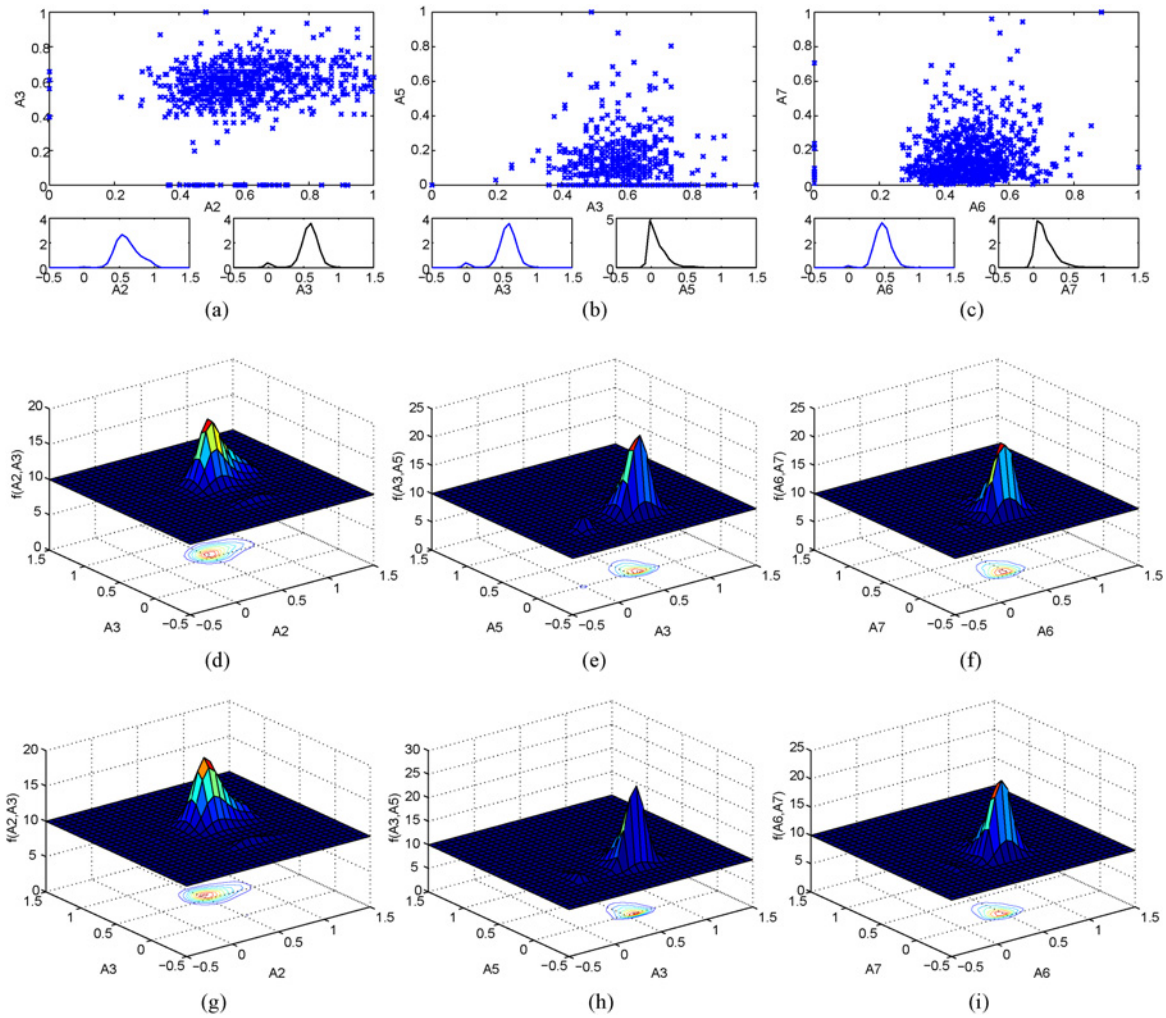


Fig. 4. Data distributions and joint p.d.f.s of the independent attributes in Pima Indian Diabetes dataset. (a) Data distribution of A_2 and A_3. (b) Data distribution of A_3 and A_5. (c) Data distribution of A_6 and A_7. (d) $f(A_2, A_3)$ with joint p.d.f. estimation. (e) $f(A_3, A_5)$ with joint p.d.f. estimation. (f) $f(A_6, A_7)$ with joint p.d.f. estimation. (g) $f(A_2, A_3)$ with marginal p.d.f. estimation. (h) $f(A_3, A_5)$ with marginal p.d.f. estimation. (i) $f(A_6, A_7)$ with marginal p.d.f. estimation.

continuous attributes into account; 2) the optimality of joint p.d.f. estimation in the L_2 sense was justified when the dependence existed among the continuous attributes; 3) a quick and simple bandwidth selection method for joint p.d.f. estimation based on the multidimensional Gaussian kernel function was derived, which can help the NNBC obtain higher classification accuracy without significantly increasing the time complexity; and 4) a detailed experimental comparison was conducted and the comparative results showed that NNBC obtained the statistically best classification performances among all density estimation-based Bayesian methods. Meanwhile, NNBC achieved a relatively favorable classification accuracy with the shortest time consumption in comparison with the other sophisticated and complex classification methods, i.e., boosting-based classification methods, PCA-based NBCs, and SVM.

Our scheduled further development in this research topic locates in the following four aspects. First, the comparative analysis between Bayesian network-based classifiers and NNBC will be conducted. Secondly, in order to verify whether the sophisticated bandwidth selection schemes, e.g., bootstrap, least-squares cross-validation, and biased cross-validation, can further lead to the improvement of classification accuracy of NNBC, we will observe the impacts of different bandwidth selection methods on the classification performance in multivariate domains. Thirdly, a generalization bound is drawn from the continuous-time Markov chains recently in [52] which deals with the empirical risk minimization-based learning processes when the assumption of independently and identically distributed samples is violated. Enlightened by the deviation inequality for time-dependent samples in a countable state space, our future work will include finding and formulating the locally and globally optimal conditions based on the empirical risk minimization for our proposed NNBC. Finally, several latest developments on the projection of subspace, e.g., exponential family PCA [26], diagonal covariance Bayesian PCA [27], geometric mean for subspace selection [42], manifold elastic net for sparse dimension reduction [53], double shrinkage for sparse learning [35], and NeNMF for non-negative matrix factorization [12], [13], etc., promise to effectively improve the performances of classification methods in different applications. Our preliminarily experimental results in Table III indicate that the projection strategy indeed

helps NNB and FNB improve classification accuracy. One of our future works will be exploring the feasibility of applying the subspace projection learning to the NNBC aiming to further improve the performance of NNBC.

REFERENCES

[1] D. Baumgartner and G. Serpen, "Performance of global-local hybrid ensemble versus boosting and bagging ensembles," *Int. J. Mach. Learn. Cybern.*, Apr. 2012. DOI: 10.1007/s13042-012-0094-8.

[2] R. R. Bouckaert, "Naive Bayes classifiers that perform well with continuous variables," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 3339. Berlin, Germany: Springer, Dec. 2004, pp. 85–116.

[3] R. Cao, A. Cuevas, and W. G. Manteiga, "A comparative study of several smoothing methods in density estimation," *Comput. Statist. Data Anal.*, vol. 17, no. 2, pp. 153–176, Feb. 1994.

[4] Z. Chen, "Why does naive Bayesian learning algorithm work," Harvard Univ., Cambrige, MA, Tech. Rep. 1999. [Available online]: http://www2.denizyuret.com/ref/chen/Why-does-NB-work.pdf.

[5] S. T. Chiu, "A comparative review of bandwidth selection for kernel density estimation," *Statist. Sinica*, vol. 6, no. 1, pp. 129–145, 1996.

[6] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, 1989.

[7] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.

[8] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple Bayesian classifier," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 105–112.

[9] P. Domingos and M, Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 103–130, 1997.

[10] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. Int. Conf. Mach. Learn.*, 1995, pp. 194–202.

[11] L. W. Fan and K. L. Poh, "A comparative study of PCA ICA and class-conditional ICA for naive Bayes classifier," in *Proc. 9th Int. Work Conf. Artif. Neural Netw.*, 2007, vol. 4507, pp. 16–22.

[12] N. Y. Guan, D. C. Tao, Z. G. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.

[13] N. Y. Guan, D. C. Tao, Z. G. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, Jul. 2012.

[14] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, 2001.

[15] B. E. Hansen. "Bandwidth selection for nonparametric distribution estimation," University of Wisconsin-Madison, Madison, Tech. Rep., 2004. [Available online]: http://www.ssc.wisc.edu/ bhansen/papers/smooth.pdf.

[16] Q. He and C. X. Wu, "Separating theorem of samples in banach space for support vector machine learning," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 1, pp. 49–54, Mar. 2011.

[17] C. Howson and P. Urbach, *Scientific Reasoning: The Bayesian Approach*. 3rd ed. Chicago, IL: Open Court, 2005.

[18] Z. Jin, F. Davoine, Z. Lou, J. Y. Yang, "A novel PCA-based Bayes classifier and face analysis," in *Advances in Biometrics* (Lecture Notes in Computer Science), vol. 3832. Berlin, Germany: Springer, 2005, pp. 144–150.

[19] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proc. Conf. Uncertain. Artif. Intell.*, 1995, pp. 338–345.

[20] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *J. Amer. Statist. Assoc.*, vol. 91, no. 433, pp. 401–407, Mar. 1996.

[21] M. Kobos, "Combination of independent kernel density estimators in classification," in *Proc. Int. Conf. Comput. Sci. Inf. Technol.*, 2009, vol. 4, pp. 57–63.

[22] R. Kohavi, B. Becker, and D. Sommerfield, "Improving simple bayes," in *Proc. Eur. Conf. Mach. Learn.*, 1997, pp. 78–87.

[23] I. Kononenko, "Inductive and Bayesian learning in medical diagnosis," *Appl. Artif. Intell.*, vol. 7, no. 4, pp. 317–337, 1993.

[24] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. Nat. Conf. Artif. Intell.*, 1992, pp. 223–228.

[25] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Proc. 10th Eur. Conf. Mach. Learn.*, 1998, vol. 1398, pp. 4–15.

[26] J. Li and D. C. Tao, "Simple exponential family PCA," *J. Mach. Learn. Res.*, vol. 9, pp. 453–460, Mar. 2010.

[27] J. Li and D. C. Tao, "On preserving original variables in Bayesian PCA with application to image analysis," *IEEE Trans. Signal Process.*, vol. 21, no. 12, pp. 4830–4843, Dec. 2012.

[28] B. Liu, Y. Yang, G. I. Webb, J. Boughton, "A comparative study of bandwidth choice in kernel density estimation for naive Bayesian classification," in *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science), vol. 5476. Berlin, Germany: Springer, 2009, pp. 302–313.

[29] J. S. Marron and W. Härdle, "Random approximations to some measures of accuracy in nonparametric curve estimation," *J. Multivariate Anal.*, vol. 20, no. 1, pp. 91–113, 1986.

[30] T. Mitchell. *Machine Learning*. New York: McGraw Hill, 1997.

[31] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[32] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill&Webert: Identifying interesting web sites," in *Proc. Nat. Conf. Artif. Intell.*, 1996, pp. 54–61.

[33] A. Pérez, P. Larrañaga, and I. Inza, "Bayesian classifiers based on kernel density estimation: Flexible classifiers," *Int. J. Approx. Reason.*, vol. 50, no. 2, pp. 341–362, Feb. 2009.

[34] A. Quintela-del-Rio and G. Estevez-Perez, "Nonparametric kernel distribution function estimation with kerdiest: An R package for bandwidth choice and applications," *J. Statist. Softw.*, vol. 50, no. 8, pp. 1–21, Aug. 2012.

[35] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, Mar. 2010.

[36] A. E. Rust and C. P. Tsokos, "On the convergence of kernel estimators of probability density functions," *Ann. Inst. Statist. Math.*, vol. 33, no. 1, pp. 233–246, 1981.

[37] K. M. Schneider, "A comparison of event models for naive Bayes anti-spam e-mail filtering," in *Proc. Conf. Eur. Assoc. Comput. Linguist.*, vol. 1, pp. 307–314, 2003.

[38] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: Wiley, 1992.

[39] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[40] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation. in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 4304. Berlin, Germany: Springer, 2006, pp. 1015–1021.

[41] D. Soria, J. M. Garibaldi, F. Ambrogi, E. M. Biganzoli, and I. O. Ellis, "A 'non-parametric' version of the naive Bayes classifier," *Knowl.-Based Syst.*, vol. 24, no. 6, pp. 775–784, Aug. 2011.

[42] D. C. Tao, X. L. Li, X. D. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[43] UCI Machine Learning Repository [Online]. Available: http://archive.ics.uci.edu/ml/

[44] M. P. Wand and M. C. Jones, *Kernel Smoothing*. London, U.K.: Chapman & Hall, 1995.

[45] X. Z. Wang and C. R. Dong, "Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 556–567, Jun. 2009.

[46] X. Z. Wang, L. C. Dong, and J. H. Yan, "Maximum ambiguity based sample selection in fuzzy decision tree induction," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1491–1505, Aug. 2012.

[47] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA: Morgan Kaufmann, 2005.

[48] T. T. Wong, "A hybrid discretization method for naive Bayesian classifiers," *Pattern Recognit.*, vol. 45, no. 6, pp. 2321–2325, Jun. 2012.

[49] Y. Yang and G. I. Webb, "On why discretization works for naive-Bayes classifiers," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 2903. Berlin, Germany: Spinger, 2003, pp. 440–452.

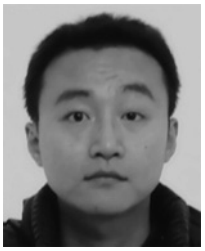
[50] Y. Yang and G. I. Webb, "A comparative study of discretization methods for naive-Bayes classifiers," in *Proc. Pac. Rim Knowl. Acquisit. Workshop*, 2002, pp. 159–173.

[51] Y. Yang and G. I. Webb, "Discretization for naive-Bayes learning: Managing discretization bias and variance," *Mach. Learn.*, vol. 74, no. 1, pp. 39–74, 2009.
[52] C. Zhang and D. C. Tao, "Generalization bounds of ERM-based learning processes for continuous-time Markov chains," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1872–1883, Dec. 2012.
[53] T. Y. Zhou, D. C. Tao, and X. D. Wu, "Manifold elastic net: A unified framework for sparse dimension reduction," *J. Data Min. Knowl. Discov.*, vol. 22, no. 3, pp. 340–371, 2010.

Xi-Zhao Wang (M'03–SM'04–F'12) received the Doctoral degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998.

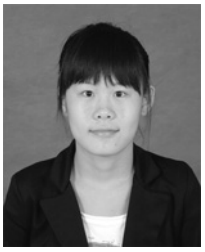
Since October 2001, he has been a Full Professor and the Dean of the College of Mathematics and Computer Science, Hebei University, Hebei, China. From September 1998 to September 2001, he was a Research Fellow in the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His main research interests include active learning, support vector machine, supervised learning, unsupervised learning, reinforcement learning, uncertainty, fuzzy sets and systems, fuzzy measures and integrals, extreme learning machine, manifold learning, unstructured learning, learning from big data, rough set, and transfer learning.

Dr. Wang is a member of the Board of Governors of the IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2005, 2007–2009, 2012–2014), the Chair of the Technical Committee on Computational Intelligence of the IEEE SMC, and a Distinguished Lecturer of the IEEE SMC. He was the Program Co-Chair of the IEEE SMC 2009 and 2010, and has been the recipient of many awards from the IEEE SMC Society. He is the Editor-in-Chief of the International Journal of Machine Learning and Cybernetics; an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics; an Associate Editor of the International Journal of Information Sciences; and an Associate Editor of the International Journal of Pattern Recognition and Artificial Intelligence.



Yu-Lin He (S'08) received the Bachelor's degree in applied mathematics and the Master's degree in computer science from Hebei University, Hebei, China, in June 2005 and June 2009, respectively, where he is currently working toward the Ph.D. degree in the College of Mathematics and Computer Science.

From February 2011 to January 2012, he was a Research Assistant in the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His research interests include computational intelligence in game, artificial neural networks, evolutionary optimization, approximate reasoning, and ontology-based knowledge representation.



Debby D. Wang (S'10) received the Bachelor's degree in information and computing sciences from Hebei University, Hebei, China, in June 2010. She is currently working toward the Ph.D. degree in the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong.

From May 2010 to July 2010, she was a student Research Assistant in the Department of Computer Science, Hong Kong University, Hong Kong. Her research interests include molecular dynamics, structural biology, machine learning algorithms, and their applications in bioinformatics.