Automated Feature Weighting in Naive Bayes for High-dimensional Data Classification

Lifei Chen
Department of Computer Science
University of Sherbrooke
Quebec, J1K 2R1, Canada
lifei.chen@usherbrooke.ca

Shengrui Wang
Department of Computer Science
University of Sherbrooke
Quebec, J1K 2R1, Canada
shengrui.wang@usherbrooke.ca

ABSTRACT

Naive Bayes (NB for short) is one of the popular methods for supervised classification in a knowledge management system. Currently, in many real-world applications, high-dimensional data pose a major challenge to conventional NB classifiers, due to noisy or redundant features and local relevance of these features to classes. In this paper, an automated feature weighting solution is proposed to result in a NB method effective in dealing with high-dimensional data. We first propose a locally weighted probability model, for Bayesian modeling in high-dimensional spaces, to implement a soft feature selection scheme. Then we propose an optimization algorithm to find the weights in linear time complexity, based on the Logitnormal priori distribution and the Maximum a Posteriori principle. Experimental studies show the effectiveness and suitability of the proposed model for high-dimensional data classification.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Statistical computing; I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models

General Terms

Algorithms, Performance, Verification

Keywords

Classification, Naive Bayes, high-dimensionality, soft feature weighting, Maximum a Posteriori

1. INTRODUCTION

Classification is a supervised learning technique aimed at assigning unlabeled samples to known classes, based on knowledge of labeled samples. The technique has been studied extensively in many domains, including text mining and bioinformatics. Well known classification methods include Support Vector Machine (SVM), Decision Trees (DT), Naive Bayes (NB) and many others [14, 24]. Currently, one of the challenging problems for a classification task is to handle high dimensional data (i.e., datasets with hundreds or thousands of features)[6]. Many types of real-world data consist of very high-dimensional features such as the documents represented in the Vector Space Model used in text mining and the micro-array gene expression data of bioinformatics. The large number of features presents an intrinsic challenge to the existing classification methods due to the curse-of-dimensionality [15].

We are interested in NB in this paper, because of its inherent simplicity, clear probabilistic semantics and its effectiveness reported in the literature [24, 27]. Roughly speaking, NB assigns a new sample to the most probable class by using the posterior probability of the response, based on the assumption that the predictive attributes are conditionally independent given the class attribute. Thanks to the independent assumption, NB yields incredible savings in number of model parameters, providing NB with the potential capacity to adapt to high dimensional data while mitigating the effect induced by the curse-of-dimensionality [14, 19].

However, the conventional NB cannot be directly applied for high-dimensional data classification, because it essentially assumes that all the features are equally important for classification, which hardly holds in high-dimensional spaces. One of the difficulties of high-dimensional classification is intrinsically caused by the existence of many noisy features that do not contribute to class prediction[6], and the redundant features that are not conditionally independent given the class [10]. Currently, a number of NB variants, alternatively known as the semi-naive Bayes methods [27], have been proposed to relax the independence assumption, either by structure extension or data reduction: examples include [9, 25, 26]. Another popular approach improving NB is to eliminate these features by combining feature subset selection techniques [17, 21, 22, 27]. In practice, the aforementioned methods suffer from high computational complexity when operating on real-world high-dimensional data. For the feature-selection-based methods, for example, it is in general not feasible to perform an exhaustive search to find the optimal feature subsets due to the huge number of admissible subsets which is exponential in the data dimensionality.

From the perspective of dimension reduction, *feature weighting*, which closely relates to feature subset selection while assigning a continuous value weight to each feature, can be used to address these problems. Though the importance

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

has been stressed in the literature [4, 7], few attempts have been made to combine feature weighting in NB classification. The major obstacle lies in the difficulties to estimate the feature weights while retaining the numerous strengths of NB. Actually, in the existing feature-weighting-based methods, including the SVM-based weighting NB [10], DT-based weighting NB [12] and the recently published FWNB [18], the weights are assigned to the features in a separated process. For these methods, feature weighting can only be regarded as a pre-processing step before NB classification, which is performed independently from the classification process.

High dimensionality also poses a challenge to the existing methods because of complex class structure. In real-world applications, the samples of a class may be correlated with one subset of features, and those of another class with a different subset of features. For example, in document classification, classes of documents are categorized by different subsets of keywords, since the keywords for one class may not occur in the documents of another class. Therefore, in high-dimensional spaces, it may only be possible to model classes in certain subspaces comprised of different combinations of features [2]. Existing methods such as those discussed above, global projection like SVM [1] and sparse Bayesian Learning ARD [5] are not able to respond to this challenge, because they model all the classes in a single subspace.

In this paper, a new NB method with automated feature weighting is proposed for high-dimensional data classification. In this method, the data are modeled using a weighted probability distribution, where the contributions of attributes to prediction of each class are described with a set of class-dependent feature weights. The weights are then used to characterize the subspace structures of classes contained in the high-dimensional data, and are jointly optimized in the Bayesian learning process. This local feature weighting scheme, in effect, equates to performing an adaptive (soft) feature selection on the data and projecting the classes into their individual subspaces in which the test samples are classified by Bayesian inference. The performance of the new model is evaluated on real-world datasets, and the experimental results show its effectiveness.

The remainder of this paper is organized as follows. Section 2 presents some preliminaries and related work. In Section 3 and Section 4, our weighted Naive Bayes model and the training algorithm, called *SWNB*, are described, respectively. Experimental results are presented in Section 5. Finally, Section 6 gives our conclusions and discusses directions for future work.

2. PRELIMINARIES AND RELATED WORK

In this section, we begin by introducing the notation used throughout the paper. In what follows, the training dataset is denoted by $DB=\{o_1, o_2, \ldots, o_N\}$, with $o_i=(\mathbf{x}_i, z_i)$, $i = 1, 2, \ldots, N$. Here, $\mathbf{x}_i=<x_{i1}, x_{i2}, \ldots, x_{iD}>$ is a D-dimensional input. Since the high-dimensional data of real-world applications such as documents and micro-array gene expression data are typically numeric, it is supposed that $\mathbf{x}_i \in \Re^D$. We use z_i to denote the pre-defined class of $\mathbf{x}_i$, $z_i \in \{1,2,\ldots,K\}$, where K stands for the number of classes contained in the training dataset. The kth class of DB, where $k = 1, 2, \ldots, K$, is denoted by $c_k = \{\mathbf{x}_i | (\mathbf{x}_i, k) \in DB\}$. The cardinality of c_k, i.e., the number of samples in the kth class, is denoted by $|c_k|$.

2.1 Naive Bayes

Let $\mathbf{x}_t$ be a test sample. $p(k|\mathbf{x}_t)$ stands for the probability that $\mathbf{x}_t$ be assigned to class k. In an NB model, the probability is computed by

$$p(k|\mathbf{x}_t) = \frac{p(k)\prod_{j=1}^{D} p(x_{tj}|k)}{p(\mathbf{x}_t)} \tag{1}$$

based on Bayes' rule and the independence assumption. Subsequently, the class of $\mathbf{x}_t$, say z_t, is predicted as

$$z_t = \arg\max_k p(k)\prod_{j=1}^{D} p(x_{tj}|k) \tag{2}$$

because the denominator of Eq. (1) dose not depend on k.

One common method for numeric inputs is to assume that the distribution of each attribute j is Gaussian for each class [14, 16]. Based on this assumption, $p(x_{tj}|k)$ can be estimated using a Gaussian probability density function:

$$G(x_{tj}; \mu_{kj}, \sigma_{kj}) = \frac{1}{\sqrt{2\pi}\sigma_{kj}} e^{-\frac{1}{2\sigma_{kj}^2}(x_{tj}-\mu_{kj})^2} \tag{3}$$

where μ_{kj} and σ_{kj} denote the mean and standard deviation of the Gaussian corresponding to the jth attribute, and are learned from the training data by Eqs. (4) and (5):

$$\mu_{kj} = \frac{1}{|c_k|}\sum\nolimits_{\mathbf{x}_i \in c_k} x_{ij} \tag{4}$$

$$\sigma_{kj}^2 = \frac{1}{|c_k|}\sum\nolimits_{\mathbf{x}_i \in c_k} (x_{ij} - \mu_{kj})^2. \tag{5}$$

Furthermore, the probability $p(k)$ can be estimated using a smoothing method such as the Laplace correction or, in the simplest form:

$$p(k) = \frac{|c_k|}{N},$$

if $|c_k| > 0$ for $k = 1, 2, \ldots, K$, which is the case considered in this paper.

It can be seen from Eq. (2) that the conventional NB identifies the classes in the entire data space, which is the same for all the classes. The probability model thus is not suitable for high-dimensional data, where classes are typically associated with different subspaces.

2.2 Semi-naive Bayes Methods

A number of semi-naive Bayes methods [27] have been proposed in recent years. They fall into two categories: structure-based and data-based methods. Those in the first category improve NB via structure extension, either by introducing latent variables to connect the correlated attributes, as in HNB [26], or by explicitly representing attribute dependencies. Interdependencies between attributes are allowed in the methods of the latter group and are often modeled as the so-called *z-dependence* [23], where each attribute is considered to depend upon the class and at most z other attributes. Most of the existing methods in this group establish 1-dependence classifiers; these include Tree Augmented Naive Bayes (TAN) [9] and its variants, as well as Averaged One-Dependence Estimators (AODE) [25] and its improvements. These methods typically have high space and time complexity and are thus inefficient for classifying high-dimensional data.

The methods in the second category aim at choosing the training data such that the dependencies within the chosen data are weaker than those in the whole dataset. The local learning methods, such as Lazy Bayesian Rules (LBR) and Locally Weighted Naive Bayes (LWNB) [7], accommodate violations of the independence assumption by choosing a desired set of the training samples on which NB is applied. This is built on the observation that the independence assumption may hold or approximately hold in a subset of the training set although violated in the whole set. The difficulties with applying such a method to high-dimensional data include reduced meaningfulness of dissimilarity measures such as Euclidean distance in searching for local neighbors, due to the fact that in high-dimensional spaces the data are inherently sparse, which tends to invalidate such measures [15].

Another group of methods in the data-based category apply NB on a subset of attributes. This is achieved by feature selection from the training data which removes irrelevant and redundant attributes. For example, the algorithm proposed by [17] identifies the attribute whose elimination best reduces the training error in a pre-processing step. The Automatic relevance determination (ARD) [5] addresses this issue by regularizing the solution space using a parameterized prior that effectively prunes away redundant features. However, feature selection performed in these methods is *global* with respect to individual classes; thus they are incapable of identifying classes embedded in the subspaces of high-dimensional data.

Recently, some attempts have been made to relax the independent assumption by feature weighting in a Bayesian model. Feature weighting assigns a continuous value weight to each feature; and thus is a more flexible method than feature selection which can be regarded as a special case of feature weighting with the weight value being restricted to either 0 or 1. In the methods proposed by [10, 18], the posteriori probability is estimated by

$$p(\mathbf{x}_t|k) = \prod_{j=1}^{D} p(x_{kj}|k)^{w(j)}$$

where $w(j)$ denotes the weight assigned to the jth attribute. The weights are typically assigned based on some heuristic measures developed for supervised dimensionality reduction, such as feature dependency, information gain and gain ratio used for constructing a decision tree [21]. In effect, using these methods, the features are weighted in the preprocessing step of a classification task. One might use a wrapper method [22], which aims at determining a desired subset of attributes for the certain algorithm to improve the classification quality, using some searching strategies such as the forward or backward method. However, in a high-dimensional space, the number of admissible subspaces is very large; thus such a method is time-consuming and becomes intractable in practice.

In this paper, we propose a new semi-naive Bayes classifier which combines local feature selection in a simple and efficient model. The model, described in the next section, is built on soft feature weighting techniques, such that variable selection becomes an integral part of it. Moreover, the feature weights can be optimized in linear time complexity. We will show that our method effectively reduces the average correlations between attributes, which consequently improves the classification quality.

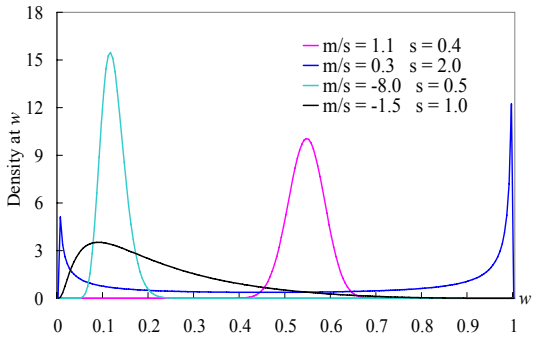

Figure 1: Changes in probability density with different standard deviation (s) and mean (m) of a Logitnormal distribution.

3. WEIGHTED NAIVE BAYES MODEL

3.1 Constrained Feature Weighting

As discussed previously, for many types of real-world data with high dimensionality, the classes are only correlated with a set of relevant attributes, while the attributes relevant to classes in the same dataset are generally different from each other. In order to formalize these characteristics, we employ a locally weighting method, where a weighting value w_{kj} is assigned to the jth attribute of class k, satisfying

$$0 < w_{kj} < 1, \quad k = 1, 2, ..., K; j = 1, 2, ..., D. \tag{6}$$

Here, the weight w_{kj} is defined to measure the contribution of the jth attribute to prediction of class k. The greater the contribution, the larger the weight should be. In NB, all the attributes have the same importance for classification and the classes are considered in the entire space, corresponding to the special case of $w_{k1} = w_{k2} = \ldots = w_{kD}$.

Typically, in many real-world applications, only a small subset of attributes significantly contributes to the class [2, 4]. By the soft weighting method, these attributes will be assigned with large weighting values, while numerous attributes in the remaining set have relatively small weights. To model such a distribution, we assume that the weights $w_{k1}, w_{k2}, \ldots, w_{kD}$ of each class c_k are taken from a Logitnormal density population. Formally, letting $\tilde{w}_k$ be a continuous random variable associated with the weights, the probability density function is

$$L(\tilde{w}_k; m_k, s) = \frac{1}{\sqrt{2\pi}s} \frac{1}{\tilde{w}_k(1-\tilde{w}_k)} e^{-\frac{1}{2s^2}(Logit(\tilde{w}_k)-m_k)^2} \tag{7}$$

where m_k and s denote the mean and standard deviation of the distribution, and $Logit(\tilde{w}_k)$ defines a *logit* function:

$$Logit(\tilde{w}_k) = \ln(\tilde{w}_k) - \ln(1-\tilde{w}_k).$$

Note that the definitional domain of Eq. (7) is $\tilde{w}_k \in (0, 1)$, which satisfies the constraints of Eq. (6).

The Logitnormal family of distributions are conventionally specified in terms of the *signal-to-noise* ratio (SNR for short) $\frac{m_k}{s}$ and the standard deviation s [8]. Curves of Eq. (7) are drawn in Figure 1, with different values of m_k and s. From the figure, a Logitnormal density is unimodal with a small value of s, and the skewness depends on m_k when s is fixed. In fact, the curve is symmetrical only if $m_k = \ln \frac{0.5}{1-0.5} = 0$. Since the density is used to model the

probabilities of feature weights where the mode typically has a small value (say, most of the weights are less than $\frac{1}{2}$ and thus $m_k \leq 0$), we confine Eq. (7) with

$$\forall k : \frac{m_k}{s} \leq -\tau \tag{8}$$

with $\tau \geq 0$ being a user-defined parameter defining the lower bound of SNR and s a fixed constant which is a small positive number to guarantee a unimodal distribution. For the work described here, we set $s = 0.5$.

The constraints defined in Eq. (8) differ from the ones commonly used in the literature [2, 4], where the constraint is the sum of weights to be one, i.e., $w_{k1}+w_{k2}+\ldots+w_{kD} = 1$. As discussed in [3], such a constraint easily yields a trivial solution of the feature weights: the attribute along which the samples exhibit the smallest variation is weighted one and the other attributes are assigned with zero weights. In our formulation, $0 < \prod_{j=1}^{D} \frac{w_{kj}}{1-w_{kj}} \leq e^{-D\tau s} \leq 1$; thus the trivial solution can be avoided. On the other hand, the inequality constraints allow the attributes to be automatically weighted according to their individual contributions to classification, by fitting a Logitnormal density with the desired m_k. Further discussions on τ will be presented in Theorem 1 and Theorem 2 in the following sections.

3.2 The Weighted Model

For a given c_k, the assignment of w_{k1},w_{k2},...,w_{kD} can be regarded as a soft feature selection procedure for the space in which c_k exists [4]. This is equivalent to projecting c_k onto a subspace, designated by the weighting matrix $diag[\sqrt{w_{k1}}, \sqrt{w_{k2}}, \ldots, \sqrt{w_{kD}}]$. According to this view, the projection of each sample $\mathbf{x}_i$ of c_k on jth dimension, denoted as y_{ij}, is computed as

$$y_{ij} = x_{ij} \times \sqrt{w_{kj}}. \tag{9}$$

We now suppose that the projections of the samples in c_k onto the jth dimension can be described using a 1D Gaussian function. The probability density function is $G(y_j; \bar{y}_{kj}, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(y_j - \bar{y}_{kj})^2}$, where $\bar{y}_{kj}$ and σ_k denote the mean and standard deviation of the Gaussian. Using Eq. (9) for both the samples and the means, the density function can be transformed into the original data space. The probability function thus becomes

$$G(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{w_{kj}}{2\sigma_k^2}(x_{ij} - \mu_{kj})^2}.$$

Followed by the conventional NB, it is assumed that the distribution of points on each of the dimensions spanning the projected subspace is independent of the others. Because $\int G(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k)\mathrm{d}x_{ij} = \frac{1}{\sqrt{w_{kj}}}$, we then suppose the samples in c_k are independently and identically distributed from the density population:

$$F(\mathbf{x}_i; \theta_k) = \prod_{j=1}^{D} P(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k)$$

where

$$P(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k) = \sqrt{w_{kj}} \times G(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k) \tag{10}$$

and $\theta_k = (\sigma_k, u_k, \omega_k)$ with $u_k = \{\mu_{kj} | 1 \leq j \leq D\}$ and $\omega_k = \{w_{kj} | 1 \leq j \leq D\}$. Intuitively, in the new model, the structures of the subspace classes involving in a high-dimensional space are characterized by two parameters: ω_k representing the "shape" of class c_k and the class-dependent variance σ_k representing to the "size" of c_k. The model then is used to improve the conventional NB, by replacing the probability density function with the one defined in Eq. (10). Here, the class of testing sample $\mathbf{x}_t$ is determined by the following rule:

$$z_t = \arg\max_k p(k) \prod_{j=1}^{D} P(x_{tj}; \mu_{kj}, w_{kj}, \sigma_k). \tag{11}$$

The improvement of this weighted Naive Bayes model compared with the conventional NB is two-fold. First, the introduction of the weighting values ω_k allows us describing and subsequently identifying the classes hidden in the low-dimensional subspaces. Second, examining the probability function defined in Eq. (10) in more detail, we can see that in effect the individual variance of c_k on the jth dimension, denoted by $\hat{\sigma}_{kj}^2$, is redefined as

$$\hat{\sigma}_{kj}^2 = \frac{\sigma_k^2}{w_{kj}} \tag{12}$$

in the projected space. By the transformation of Eq. (12), those attributes whose variances are equal to 0 will gain their non-zero projected variances, according to the attribute's proper contribution to class prediction. This differs from the existing models where the common approach to this case is to introduce an empirical constant or the simple Laplace estimation [14, 24].

3.3 On the Independence Assumption

As we can see from Eq. (11), the weighted Naive Bayes model is built on the conditional independence assumption made by the conventional NB, which would adversely affect the quality of classification results when it is violated. To mitigate the effect of violation of this assumption, an effective way is to reduce the correlations between attributes when applying a NB classifier [19, 24]. In the following, we shall show that our new model is able to automatically reduce the correlations between attributes by the feature weighting scheme, in terms of the Pearson product-moment correlation coefficient.

With the weighted Naive Bayes model, the samples are classified in the projected subspaces, designated by ω_k for c_k. For any two attributes j,$l \in [1, D]$ and $j \neq l$, according to Eq. (12), the Pearson correlation coefficient with respect to the projected space can be computed as

$$R(j,l) = \frac{\sum_{i=1}^{|c_k|} x_{ij}x_{il} - |c_k|\mu_{kj}\mu_{kl}}{(|c_k|-1)\sqrt{\frac{\sigma_k^2}{w_{kj}}}\sqrt{\frac{\sigma_k^2}{w_{kl}}}} = r(j,l)\frac{\sqrt{w_{kj}w_{kl}}X_{kj}X_{kl}}{\sigma_k^2}$$

with

$$X_{kj} = \sum_{i=1}^{|c_k|} (x_{ij} - \mu_{kj})^2$$

and $r(j,l)$ being the Pearson correlation coefficient in the original data space. Then we measure the change of attributes correlations after the projection by averaging $\frac{R(j,l)}{r(j,l)}$ over all the dimensions, i.e.,

$$\rho_k = \frac{1}{D(D-1)} \sum_{j=1}^{D} \sum_{l=1,\neq j}^{D} \frac{R(j,l)}{r(j,l)}.$$

Theorem 1 gives the relationship between τ and ρ_k.

THEOREM 1. *When $D > 1$, $\rho_k \leq 1$, in which the equality holds only if $\tau = 0$.*

PROOF. The proof is given in Appendix A. □

According to Theorem 1, the overall correlations between attributes can be reduced in their individual subspaces and subsequently improve the classification performance, using the feature weighting method if we set $\tau > 0$. In fact, when $\tau = 0$ the feature weights would easily reach another trivial solution: $w_{kj} \equiv \frac{1}{2}$, which results in degeneration of the weighted NB model into a conventional one (all the dimensions receive the same variance in this case).

4. MODEL LEARNING ALGORITHM

Given the training dataset $DB = \bigcup_{k=1}^{K} c_k$, the goal of this section is to learn the set of parameters θ_k for $k = 1, 2, \ldots, K$. Instead of using Maximum Likelihood Estimation (MLE), we have adopted Maximum A Posteriori (MAP) method [14], since we need to learn a desired set of feature weights for the weighted NB model.

4.1 Objective Function for MAP Learning

MAP aims to choose $\hat{\theta}_k$ so as to maximize the posterior $p(\theta_k|c_k)$, for the class $c_k (k = 1, 2, \ldots, K)$:

$$\hat{\theta}_k = \arg\max_{\theta_k} p(\theta_k|c_k).$$

Therefore, the optimized parameters should maximize the following objective:

$$\begin{aligned} J_1(\theta_k) &= p(\theta_k|c_k) \\ &\propto p(\theta_k)p(c_k|\theta_k) \end{aligned}$$

where $p(\theta_k)$ is called a *prior* before seeing any data of c_k.

Here, we consider the Logitnormal distribution of the feature weights, defined in Eq. (7), as a prior of c_k that needs to be maximized in the learning process at the same time. Taking a logarithmic transformation on $J_1(\theta_k)$, the objective function thus becomes:

$$\begin{aligned} &J_2(\theta_k) = \ln p(\theta_k) + \ln p(c_k|\theta_k) \\ &= \textstyle\sum_{\mathbf{x}_i \in c_k} \sum_{j=1}^{D} (\ln L(w_{kj}; m_k, s) + \ln P(x_{ij}; \mu_{kj}, w_{kj}, \sigma_k)) \end{aligned}$$

subject to the inequality constraint of Eq. (8). Theorem 2 provides an efficient manner for transforming Eq. (8) into an equality constraint such that it can be easily taken into account while optimizing $J_2(\theta_k)$.

THEOREM 2. *Letting* $d = \sum_{j=1}^{D} w_{kj} < \frac{D}{2}$, *then* $m_k \leq -\ln(\frac{D}{d} - 1)$.

PROOF. The proof is given in Appendix B. □

Based on Theorem 2 and Eq. (8), an equality constraint can be derived by setting $\tau s = \ln(D/d - 1)$:

$$\sum_{j=1}^{D} w_{kj} = \frac{D}{1 + e^{\tau s}}. \tag{13}$$

Since $\tau s > 0$, we always have $\sum_{j=1}^{D} w_{kj} < \frac{D}{2}$. Using Eqs. (7) and (10) to replace the two probability functions and letting λ_1 and λ_2 be the two Lagrange multipliers introducing the constraints, the resulting objective function is obtained as

$$\begin{aligned} J(\theta_k) = &-\textstyle\sum_{j=1}^{D} \ln(\sqrt{w_{kj}}(1 - w_{kj})) - \frac{1}{2|c_k|\sigma_k^2} \sum_{j=1}^{D} w_{kj} X_{kj} \\ &-D \ln \sigma_k + \lambda_1 (Ds - \textstyle\sum_{j=1}^{D} (Logit(w_{kj}) - m_k)^2) \\ &+\lambda_2 (\frac{D}{1+e^{\tau s}} - \textstyle\sum_{j=1}^{D} w_{kj}). \end{aligned} \tag{14}$$

4.2 Learning Algorithm

For the nonlinear optimization problem of Eq. (14), the optimal parameters can be learned by taking derivatives to the objective function. In detail, by setting the gradients of $J(\theta_k)$ with respect to μ_{kj} to zero for $\forall j$, the mean of each class is solved. The resulting expression is the same as Eq. (4). Then, we set $\frac{\partial J(\theta_k)}{\partial \sigma_k} = 0$ to solve the variance of each class as

$$\sigma_k^2 = \frac{1}{D|c_k|} \textstyle\sum_{j=1}^{D} w_{kj} X_{kj}. \tag{15}$$

Moreover, $\frac{\partial J(\theta_k)}{\partial w_{kj}} = 0$ for $\forall j$ and $\frac{\partial J(\theta_k)}{\partial m_k} = \frac{\partial J(\theta_k)}{\partial \lambda_1} = 0$ lead to

$$Logit(w_{kj}) = m_k + \frac{1}{4\lambda_1} \varphi(w_{kj}, \lambda_2, \sigma_k) \tag{16}$$

where

$$m_k = \frac{1}{D} \textstyle\sum_{j=1}^{D} Logit(w_{kj}) \tag{17}$$

and

$$(4\lambda_1)^2 = \frac{1}{Ds} \textstyle\sum_{j=1}^{D} \varphi^2(w_{kj}, \lambda_2, \sigma_k) \tag{18}$$

with

$$\varphi(w_{kj}, \lambda_2, \sigma_k) = (3w_{kj} - 1) - w_{kj}(1 - w_{kj})(\frac{X_{kj}}{|c_k|\sigma_k^2} + 2\lambda_2).$$

Because of the involvement of w_{kj} in the right side of Eq. (16), we do not have a closed-form solution to w_{kj}s. The *SWNB* (denotes Subspace Weighting Naive Bayes) algorithm, as outlined by Algorithm 1, utilizes an iterative process to learn the parameters. In each iterative step of the algorithm, we first re-estimate σ_k, m_k and λ_1 according to Eqs. (15), (17), (18); then they are used to resolve λ_2 by $\frac{\partial J(\theta_k)}{\partial \lambda_2} = 0$ which results in $\frac{D}{1+e^{\tau s}} = \sum_{j=1}^{D} w_{kj}$. Note that Eq. (16) can be rewritten as

$$w_{kj} = 1 - \frac{1}{1 + e^{m_k + \frac{1}{4\lambda_1} \varphi(w_{kj}, \lambda_2, \sigma_k)}}. \tag{19}$$

Thus the optimized value of λ_2 corresponds to the root of $\psi(\lambda_2)$:

$$\psi(\lambda_2) = \frac{De^{\tau s}}{1 + e^{\tau s}} - \sum_{j=1}^{D} \frac{1}{1 + e^{m_k + \frac{1}{4\lambda_1} \varphi(w_{kj}, \lambda_2, \sigma_k)}} = 0. \tag{20}$$

The above nonlinear equation can be solved using the common numerical solution, such as the well-known Newton-Raphson method. Next, the weights are re-estimated by the right side of Eq. (19). The new weights then are used to compute in the next step. The iterative process stops when the stop criterion is met, for example, the change of the weights is smaller than a termination criterion.

Actually, the algorithm can be regarded as an instance of the EM algorithm [14] that partially optimizes each parameter in a sequential structure. The computational complexity is $O(PD|c_k|)$, where P denotes the number of iterations. For a training dataset consisting of K classes, *SWNB* is called for K times, each for the kth class (k=1,2,...,K). Therefore, on the whole dataset, the time complexity of *SWNB* is $O(KND)$, which is linear with respect to the number of training samples and the number of dimensions.

Input: the training samples $\mathbf{x}_i (i = 1, 2, \ldots, |c_k|)$ in c_k;
Output: a set of model parameters θ_k;
begin
  Let $\hat{w}_{kj} = \frac{1}{1+e^{\tau s}}$, $\lambda_2 = 0$ and compute μ_{kj}s using Eq. (4), for $j = 1, 2, \ldots, D$;
  repeat
    Let $\omega_k = \hat{\omega}_k$, use Eqs. (15) and (17) to compute σ_k and m_k, respectively;
    Compute λ_1 using Eq. (18);
    Resolve Eq. (20) to determine λ_2;
    Update the weights using Eq. (19) and obtain $\hat{\omega}_k$;
  until *the stop criterion is met*;
end

Algorithm 1: Outline of the *SWNB* algorithm.

5. EXPERIMENTAL EVALUATION

In this section, we evaluate the performances of *SWNB* on real-world high-dimensional datasets. We also experimentally compare *SWNB* with mainstream classification methods.

5.1 A Case Study

This set of experiments aims at examining the performance of *SWNB*, by a case study on gene expression data. Gene expression data are generated by micro-array techniques, and are often presented as matrices of expression levels of genes under different conditions. On these data, the task of the experiments is to predict whether a gene belongs to a disease based on its expression levels, and to identify sets of genes that give rise to the particular disease.

The well-known dataset ALL-AML [11] was used, which was created for acute Leukemia diagnoses. The dataset contains expression levels of 7129 genes taken over 72 samples (47 ALL and 25 AML). It poses a challenge to the classification method due to its very high dimensionality compared to the number of samples. The dataset is available at Kent Ridge Bio-medical Data Set Repository (http://sdmc.lit.org.sg/GEDatasets/Datasets.html), and has been split into a standard training set (27 ALL and 11 AML) and a test set (20 ALL and 14 AML) that have been used as benchmarking data for high-dimensional data analysis. The dataset was normalized such that each value was ranged in [0,1]. This is because the feature weight generally depends on the dispersion of the values in each dimension; in order to obtain a meaningful set of weights, the values in all the dimensions need to be normalized into the same range.

Figure 2 shows two log-scale histograms of the feature weights learned by *SWNB* for the two classes in the training dataset. The y-axis of the figures are drawn using the values by logarithmic transformation on the frequency that the weights fall in the particular ranges. As expected, for each class, most of the attributes solely receive small weights, fitting well a unimodal Logitnormal distribution with a small mean (the resulting means of the two Logitnormal density functions are estimated as -4.253 and -4.246, respectively). The numbers of attributes whose weights are less than 0.1 are 7043 and 7052 for the two classes, which indicates that there are only 86 and 77 genes that mostly contribute to ALL and AML, respectively.

Figure 3 shows the weight distributions for the attributes of each class. The x-axis is the attribute index and the y-axis indicates the weight. This sort of weights distribution allows us to extract a reduced attribute subset for each class.

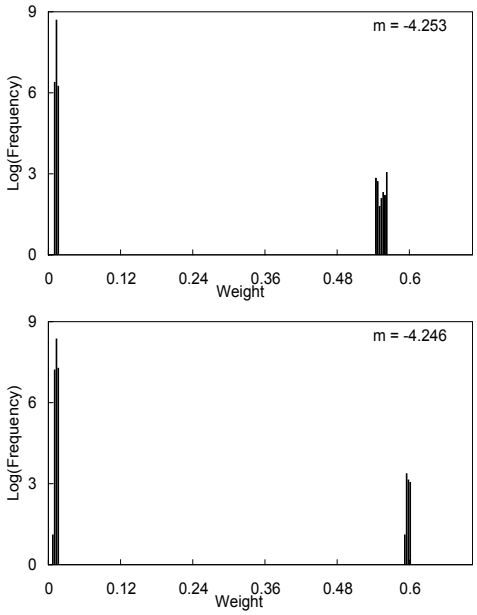

Figure 2: Log histograms of the feature weights obtained by *SWNB* for classes ALL and AML in the ALL-AML Leukemia training dataset.

Note that the two subsets associated with the two classes are quite different. This differs largely from the existing feature-weighting-based NB classifiers, such as [10] and [18], where a unique subset is chosen for all the classes in the same dataset. By merging the two subsets together, a reduced set consisting of 149 genes results with approximately 98% dimensionality reduction, for the ALL-AML Leukemia training dataset. From the reduced set, as shown in Figure 3, several important attributes that represent important genes for the acute Leukemia diagnoses can be identified, such as M96326 and U05259 that have been validated in the related work [11].

Table 1: Classification quality (in terms of F1-measure) of different classifiers on the ALL-AML Leukemia test dataset, with the whole attributes set and the reduced subset yielded by *SWNB*.

D	Class	*SWNB*	NB	LibSVM
7129	ALL	**0.9756**	0.9474	0.8889
	AML	**0.9630**	0.9333	0.7826
		$_{SWNB+}$*SWNB*	$_{SWNB+}$NB	$_{SWNB+}$LibSVM
149	ALL	0.9744	0.9474	**1.0000**
	AML	0.9655	0.9333	**1.0000**

In effect, the soft feature weighting technique used in *SWNB* equates to performing a dynamic feature selection for the respective class during the training process. Table 1 illustrates the classification quality of *SWNB*, in terms of F1-measure, with comparison to those of the highly efficient standard SVM algorithm LibSVM [1] and the conventional NB. Both the (original) test set with all the attributes and the test set with reduced attributes obtained by projection onto the subspaces yielded by *SWNB* also is tested. It can be seen that *SWNB* achieves high classification accuracy and outperforms NB on both the test

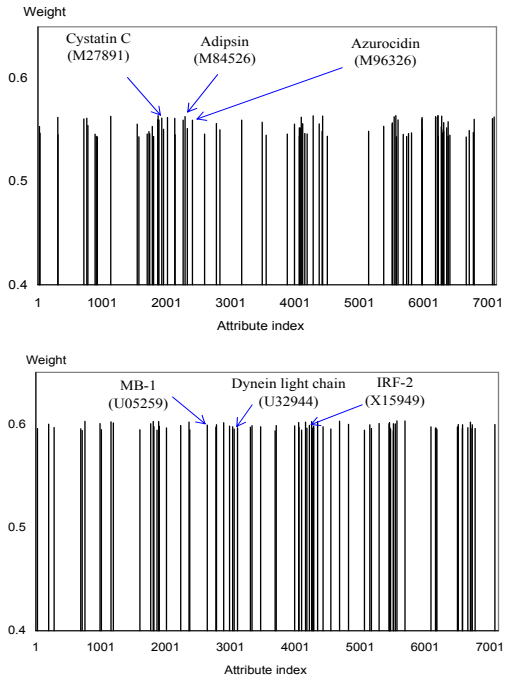

Figure 3: The weight distributions of genes in the ALL-AML Leukemia training dataset.

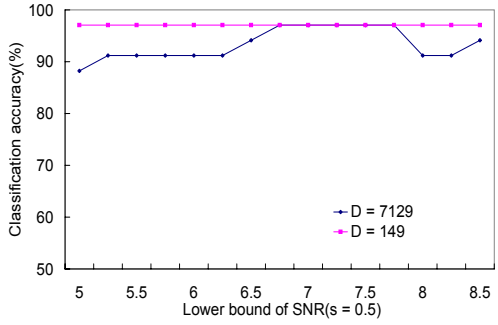

Figure 4: Change in classification accuracy with different τ (the lower bound of SNR) of *SWNB* on the ALL-AML Leukemia test dataset.

sets, whereas LibSVM perform poorly on the original data in very high dimensionality. However, on the test set with reduced attributes, LibSVM (shown as $_{SWNB+}$LibSVM in Table 1) yields 100% correct results. This is because the original dataset involves a large number of noisy attributes, which confuse the class boundaries. LibSVM benefits from our feature weighting method and achieves the best performance. Note that the feature selection is incorporated as a component of the weighted model in *SWNB*, and is jointly optimized along with learning the model. This differs from the one adopted in most of the existing methods [10, 18, 27], where the process is typically implemented by a separated pre-processing step.

Figure 4 shows the relationships between the classification accuracy (i.e., Micro-F1 $\times$ 100%) of *SWNB* and the user-defined parameter τ, as the lower bound of SNR for learning the weights. According to the *Rose criterion*, we know that an SNR of at least 5 is needed to be able to distinguish the important attributes from others at a high certainty. From the figure, *SWNB* is robust with respect to the parameter in the projected subspace, while the good performance can be obtained with $\tau \geq 7$ in the entire space. For the experiments presented in Table 1 (and in the next subsection), we set $\tau = 7.0$ based on the results.

5.2 Performance Comparison

The second set of experiments is designed to compare *SWNB* with other classifiers in terms of classification quality. Again, the experiments were conducted on the real-world high-dimensional datasets.

5.2.1 Real-world Datasets

Eight widely used high dimensional datasets were used. Table 2 lists the details of the datasets. We obtained four bio-medical datasets G1∼G4 from the Kent Ridge Bio-medical Data Set Repository as described in Section 5.1, each containing expression levels of genes (corresponding to the attributes) taken in different conditions (rows of the dataset). Standard training set and test set are available for all these four datasets. In the experiments, we learned the classification model on the training set and made the evaluation on the test set. All the attributes were normalized before training and testing.

Table 2: Details of the real-world datasets

ID	Datasets	D	#Samples	K
G1	MLL-Leukemia	12582	72	3
G2	ALL-Subtype-T-ALL	12558	248	2
G3	ALL-Subtype-Hyperdip50	12558	248	2
G4	ALL-AML	7129	72	2
T1	20NG-os-4-1	2000	2000	4
T2	20NG-os-5-1	2000	2498	5
T3	TREC-tr12	1000	304	7
T4	TREC-tr41	1000	869	9

The remaining four datasets were extracted from two popular documents corpora: 20-Newsgroups and TREC. The 20-Newsgroups corpus is a collection of newsgroup documents, partitioned across 20 different newsgroups. We have used the version of Machine Learning Group at UCD available at http://mlg.ucd.ie/files/datasets, and have chosen two datasets named os-4-1 and os-5-1 for the experiments. The second group of datasets was derived from the TREC collection. We obtained two TREC subsets, namely tr12 and tr41, from the CLUTO toolkit (available at http://www.users.cs.umn.edu/∼karypis/cluto). The topics 77 of tr12 and 356 of tr41 were removed because there are only 9 documents in the original datasets. These document datasets were chosen because they have been well pre-processed that can be used as standard benchmarking data.

All the documents have been represented in the simplest Vector Space Model (VSM), i.e., each element of the vector only indicates the frequency of the corresponding word in the document. The Max-Relevance feature selection algorithm [20] then was applied to select a set of terms with the highest relevance to the document category. In particular, we have chosen the top 2000 and 1000 attributes, respectively, creating the final 20-Newsgroups datasets T1,T2 and TREC datasets T3 and T4. To take different document lengths into consideration, each document vector in the resulting datasets has been scaled to unit length.

5.2.2 Experimental Setup

To compare the performance of *SWNB*, four classifiers: the conventional NB, locally weighted NB(LWNB) [7], feature weighted NB (FWNB)[18] and the flexible Bayes(FlexNB) [16] have been chosen in our experiments. A brief introduction of the four classifiers is given below:

NB: the conventional Naive Bayes that uses a Gaussian distribution modeling the continuous attribute. For the attributes having zero variances (for example, in the document datasets, the data are typically sparse and the dispersion of dimensions in a class often happens to be zero), we set $\sigma_{kj} = 10^{-5}$ for these attributes when training the NB model.

LWNB: a semi-naive Bayes method based on the locally weighted learning approach, which assigns the test sample a specified weight according to its neighborhood [7]. We adopted the implementations of LWNB from the WEKA system [13], and set k=50 for the document datasets, as this value is favorable to LWNB. For the four gene datasets, we set $k = 5$ because of their small data sizes.

FWNB: a recently published algorithm improving NB by heuristically feature weighting [18], where the weights are calculated by Kullback-Leibler measure. Because it is designed for categorical attributes, each continuous attribute in the gene datasets is discretized using the method suggested by the authors. For the document datasets, we used the binary presentation, i.e., each element of the document vector is either 1 or 0 representing presence and absence of the corresponding word in the document.

FlexNB: an extension of NB that eschews the single Gaussian assumption in favor of kernel density estimation [16]. FlexNB estimates the density of each continuous attribute j of c_k by $p(x_{tj}|k) = (|c_k|h)^{-1} \sum_{\mathbf{x}_i \in c_k} G(\frac{x_{tj}-x_{ij}}{h}; 0, 1)$ where $h = 1/\sqrt{|c_k|}$ is the bandwidth of the kernel.

FlexNB was chosen because it presents an alternative solution (rather than the popular Gaussian model) to the continuous attributes. We thus can evaluate the weighted Gaussian model of *SWNB* by making comparisons between *SWNB* and FlexNB on the same datasets. However, such kernel density estimation would run into problems on high dimensional data. To overcome the difficulties, an additional supervised feature selection was performed before applying FlexNB. First, the importance of each attribute was evaluated by measuring the gain ratio (GR for short) with respect to the class, using GainRatioAttributeEval of the WEKA system [13]. Then, approximate 2.5% and 25% attributes were retained for each gene dataset and document dataset, respectively. We will denote the algorithm by GR+FlexNB.

We also used the start-of-art SVM algorithm as the baseline classifier in the experiments. The SVM classifier implemented in LibSVM package [1] was employed. A linear function was adopted as the kernel and all remaining parameters were left as default values.

5.2.3 Experimental Results

The classification performances of the different classifiers were measured using the Micro-F1 (F1 over classes and samples) and Macro-F1 (average of within-class F1 values). Table 3 illustrates the classification results returned by each algorithm on the four gene datasets (recall that the training set and test set are available in these datasets). The two figures in each cell represent Micro-F1 and Macro-F1, respectively. The best results are marked in bold typeface.

Table 3: Comparison of classification quality on the gene datasets.

ID	*SWNB*	NB	LWNB	FWNB	GR+ FlexNB	LibSVM
G1	**1.0000**	**1.0000**	0.8667	0.8000	0.8000	**1.0000**
	1.0000	**1.0000**	0.8056	0.6889	0.6889	**1.0000**
G2	**1.0000**	-	0.9765	**1.0000**	**1.0000**	**1.0000**
	1.0000	-	0.9572	**1.0000**	**1.0000**	**1.0000**
G3	0.9765	-	0.9647	**0.9882**	0.9765	**0.9882**
	0.9693	-	0.9547	**0.9849**	0.9693	**0.9849**
G4	**0.9706**	0.9412	0.7059	0.9118	0.7941	0.8529
	0.9693	0.9404	0.6458	0.9079	0.7589	0.8357

Table 3 shows that *SWNB* is able to obtain high-quality overall results on datasets with very high dimensionality. LibSVM yields comparable results to that of *SWNB* but encounters difficulties on G4, whereas LWNB performs poorly on all the datasets. LWNB improves NB by neighbor searching, based on the similarity measure in the entire space. For these high-dimensional datasets, such a measurement would lead to close dissimilarity values between different samples [15], reducing its effectiveness. FWNB weights the features based on GR of each attribute computed like in C4.5 [18]. This method easily leads to the over-fitting problem especially on the datasets with few samples, such as G1 and G4 in the experiments, which only contain 72 samples (comparably, both G2 and G3 contain 248 samples. On these two datasets, FWNB archives obviously better performances than those on G1 and G4). In addition, NB fails in classifying G2 and G3, which indicates that the attributes might bias from the standard Gaussian. Interestingly, on these two datasets, the performances of *SWNB* are closed to those of GR+FlexNB, which is based on non-Gaussian modeling. This convinces us that the weighed Gaussian model of *SWNB* is suitable for high-dimensional classification.

Table 4: Comparison of classification quality on the document datasets.

ID	*SWNB*	NB	LWNB	FWNB	GR+ FlexNB	LibSVM
T1	**0.97±.01**	0.94±.02	0.89±.02	0.90±.02	0.90±.02	0.95±.01
	0.95±.02	0.91±.02	0.86±.03	0.88±.03	0.83±.03	0.94±.02
T2	**0.95±.01**	0.93±.01	0.82±.03	0.86±.02	0.87±.02	**0.95±.01**
	0.95±.01	0.93±.02	0.83±.03	0.86±.02	0.84±.02	**0.95±.02**
T3	**0.89±.05**	0.76±.06	0.80±.06	0.87±.06	0.49±.05	**0.89±.05**
	0.86±.07	0.69±.10	0.78±.07	0.86±.06	0.34±.09	**0.87±.06**
T4	0.94±.02	0.87±.04	0.95±.02	0.88±.03	0.63±.04	**0.96±.02**
	0.92±.03	0.79±.07	0.92±.04	0.82±.05	0.41±.05	**0.93±.04**

Due to the lack of standard training set and test set in the documents datasets, ten cross-validation were used for T1∼T4. Each dataset were classified by each algorithm for 10 executions and the average performances were reported, as illustrated in Table 4. The values associated with each dataset correspond to the average Micro-F1 and Macro-F1, in the format *average* ± 1 *standard deviation* in the table. Again, the best results are marked in bold typeface. We can see from the table that *SWNB* yields excellent performance approaching LibSVM, while exceeds the other competing algorithms on all the datasets. In these datasets, the document categories are typically featured by different attribute

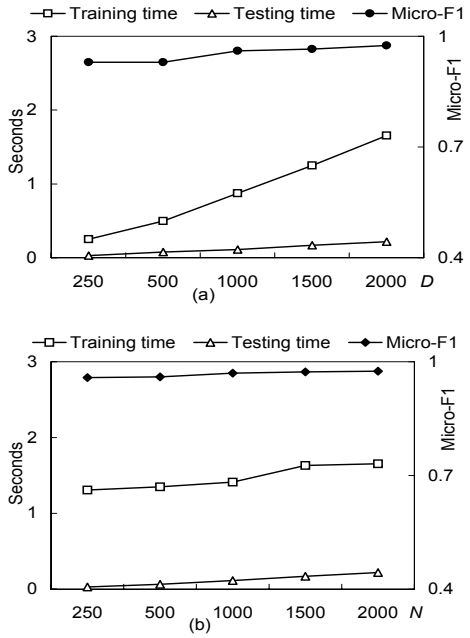

Figure 5: Relationships between training time, testing time, classification accuracy of *SWNB*, and different numbers of dimensions (in (a)) and samples (in (b)).

subsets. This results in poor performance of GR+FlexNB, as a global feature-selection method; and of FWNB which is based on the global feature-weighting technique.

5.3 Scalability

It may be interesting to compare the efficiency of *SWNB* with a feature-selection based Bayesian classifier, because the latter is able to reduce computation in classification due to the removal of irrelevant or redundant features. However, such a method is time-consuming for feature selection in high-dimensional data. In fact, we have attempted to conduct experiments using the Attribute Selective Bayesian Classifier [17] from the WEKA system [13], and have observed their failures in fulfilling the classification task within a reasonable time, for the datasets of Table 2. The efficiency of *SWNB* depends on the number of iterative steps used to compute the feature weights. In the experiments, we asserted the convergence of *SWNB* by examining whether the change of weights between two successive iterations is smaller than 10^{-5}. The numbers of iterations are 55(ALL), 59(MLL) and 67(AML) for G1, 21(T-ALL) and 71(Others) for G2, 25(Hyperdip>50) and 57(Others) for G3, and 64(ALL) and 55(AML) for G4. Because of the imbalance of the document datasets, in terms of the number of samples per class, we observed that the numbers of iterations differed largely over the classes. The number typically exceeded 100 for a large class; therefore we have added an additional criterion to the convergence of *SWNB* by limiting the maximal number of iterations to 100. In the following, we report the scalability of *SWNB* on the document datasets using the settings.

Figure 5 shows the average times used by *SWNB*, and the average Micro-F1 returned by *SWNB* on the os-4-1 dataset of the 20NewsGroup corpus described in Section 5.2.1. The scalability with respect to the dimensionality was tested on five datasets obtained by choosing 250, 500, 1000, 1500 and 2000 attributes from the original os-4-1 dataset using the Max-Relevance feature ranking method [20]. Figure 5(a) shows the results, from which we can see that both the training and testing times of *SWNB* increase linearly with respect to the data dimensionality. The classification accuracy is slightly affected by the number of dimensions, since the greater the number of relevant terms is, the more useful information can be used by *SWNB* to identify different contributions of terms for document category prediction. Figure 5(b) illustrates the relationships between the training time, the testing time and the data size. We have randomly chosen samples from the 4 classes, using certain sampling ratios, to create the five datasets with a fixed dimensionality of 2000. We can see that both the training time and testing time of *SWNB* increase linearly with respect to the number of samples, accompanied by high classification quality.

6. CONCLUSIONS

In this paper, we first discussed the problems faced by the Naive Bayes method in classifying high-dimensional data. This problem becomes difficult due to the high dimensionality, the high proportion of irrelevant and redundant attributes contained in the data and the fact that only a small number of the attributes may be considered in the classification process. Using the locally feature-weighting technique, we proposed a new NB model which has the ability to describe the different contributions of attributes in predicting different classes. We also proposed an efficient training algorithm, called *SWNB*, for learning an optimized set of feature weights in order to fit a Logitnormal priori distribution. The experiments were conducted on document corpora and gene micro-array datasets that are widely used in real-world applications, and the results show its effectiveness.

The source codes of *SWNB* are available at $http://info.usherbrooke.ca/Prospectus/Members/LChen/source-code-of-swnb/at\_download/file$. One future direction is to test *SWNB* on more extensive datasets, and to compare different methods using more evaluation measures such as the non-parametric Friedman tests [28]. Another avenue of further study is to extend the weighted Bayesian model to high-dimensional categorical data.

7. ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their invaluable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Discovery Accelerator Supplements grant No. 396097-2010, and partially by the National Natural Science Foundation of China under Grant No. 61175123.

8. REFERENCES

[1] C. Chang and C. Lin. Libsvm: A library for support vector machines. *http://www.csie.ntu.edu.tw/∼cjlin/libsvm/*, 2011.

[2] L. Chen, G. Guo, and K. Wang. Class-dependent projection based method for text categorization. *Pattern Recogn. Lett.*, 32:1493–1501, 2011.

[3] H. Cheng, K. Hua, and K. Vu. Constrained locally weighted clustering. *Proceedings of the VLDB Endowment*, 1(1):90–101, 2008.

[4] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. Locally adaptive metrics for clustering high dimensional data. *Data Min. Knowl. Disc.*, 14:63–97, 2005.

[5] D. Wipf, and S. Nagarajan. A new view of automatic relevance determination. In *Proc. of the NIPS*, pages 1625–1632, 2007.

[6] J. Fan and Y. Fan. High-dimensional classification using features annealed independence rules. *Ann. Stat.*, 36(6):2605–2637, 2008.

[7] E. Frank, M. Hall, and B. Pfahringer. Locally weighted naive bayes. In *Proc. of the UAI*, pages 249–256, 2003.

[8] P. Frederic and F. Lad. Two moments of the logit-normal distribution. *Commun. Stat.-Simul. Compu.*, 37(7):1263–1269, 2008.

[9] N. Friedman, D. Geiger, and M. Goldszmidt. Not so naive bayes: Aggregating one-dependence estimators. *Mach. Learn.*, 58(1):5–24, 2005.

[10] T. Gartner and P. Flach. Wbcsvm: Weighted bayesian classification based on support vector machines. In *Proc. of the ICML*, pages 154–161, 2001.

[11] T. Golub, D. K. Slonim, *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[12] M. Hall. A decision tree-based attribute weighting filter for naive bayes. *Knowl.-Based Syst.*, 20(2):120–126, 2007.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[15] A. Hinneburg, C. Aggarwal, and D. Keim. What is the nearest neighbor in high dimensional spaces. In *Proc. of the VLDB*, pages 506–515, 2000.

[16] G. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proc.of the UAI*, pages 338–345, 1995.

[17] P. Langley and S. Sage. Induction of selective bayesian classifiers. In *Proc.of the UAI*, pages 399–406, 1994.

[18] C. Lee, F. Gutierrez, and D. Dou. Calculating feature weights in naive bayes with kullback-leibler measure. In *Proc. of the IEEE ICDM*, pages 1146–1151, 2011.

[19] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proce. of the ECML*, pages 4–15, 1998.

[20] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1226–1238, 2005.

[21] C. Ratanamahatana and D. Gunopulos. Feature selection for the naive bayesian classifier using decision trees. *Appl. Artif. Intell.*, 17:475–487, 2003.

[22] R.Kohavi and G. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1):273–324, 1997.

[23] M. Sahami. Learning limited dependence bayesian classifiers. In *Proc. of the SIGKDD*, pages 334–338, 1996.

[24] M. Seeger. Bayesian modeling in machine learning: A tutorial review. *Tutorial, Saarland University*, http://lapmal.epfl.ch/papers/bayes-review.pdf, 2006.

[25] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29:131–163, 1997.

[26] H. Zhang, L. Jiang, and J. Su. Hidden naive bayes. In *Proc. of the AAAI*, pages 919–924, 2005.

[27] F. Zheng and G. Webb. A comparative study of semi-naive bayes methods in classification learning. In *Proce. of the Australalian Data Mining Workshop*, pages 141–156, 2005.

[28] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

APPENDIX

A. PROOF OF THEOREM 1

Proof. First, letting σ_k be the optimized variance that maximizes $J_2(\theta_k)$, we have $\sigma_k^2 = \frac{1}{D|c_k|}\sum_{j=1}^{D} w_{kj}X_{kj}$. Then, ρ_k can be rewritten as

$$\rho_k = \frac{1}{D(D-1)}\left(\sum_{j=1}^{D}\sqrt{\frac{w_{kj}X_{kj}}{|c_k|\sigma_k^2}}\right)^2 - \frac{1}{D(D-1)}\sum_{j=1}^{D}\frac{w_{kj}X_{kj}}{|c_k|\sigma_k^2}$$
$$\leq \frac{1}{D(D-1)} \times D\sum_{j=1}^{D}\frac{w_{kj}X_{kj}}{|c_k|\sigma_k^2} - \frac{1}{D(D-1)}\sum_{j=1}^{D}\frac{w_{kj}X_{kj}}{|c_k|\sigma_k^2} = 1$$

if $D > 1$. The penultimate step of the above derivations is based on the Chebyshev inequation in which the equality holds only if $\forall j : w_{kj}X_{kj} \equiv \mathcal{C}$(a constant). Next, we show that $w_{kj}X_{kj} \equiv \mathcal{C}$ only if $\tau = 0$. Suppose that m_k and w_{kj} for all j maximize $J_2(\theta_k)$ at the same time, we can obtain $m_k = \frac{1}{D}\sum_{j=1}^{D} Logit(w_{kj})$ and

$$3w_{kj} - 1 - \frac{X_{kj}}{|c_k|\sigma_k^2}w_{kj}(1-w_{kj}) = 2\frac{|c_k|}{s_k^2}(Logit(w_{kj}) - m_k).$$

If $w_{kj}X_{kj} \equiv \mathcal{C}$, which subsequently leads to $\sigma_k^2 = \frac{\mathcal{C}}{|c_k|}$, then $|c_k|\sum_{j=1}^{D}(1-2w_{kj}) = 0$ by adding up the above equations of w_{kj} for $j = 1, 2, \ldots, D$. Therefore, we obtain $\sum_{j=1}^{D} w_{kj} = \frac{D}{2}$ in the case where $w_{kj}X_{kj} \equiv \mathcal{C}$. According to Eq. (13), which asserts that $\sum_{j=1}^{D} w_{kj} = \frac{D}{2}$ only if $\tau = 0$, it can be concluded that $\rho_k \leq 1$ and the equality holds only if $\tau = 0$. □

B. PROOF OF THEOREM 2

Proof. We consider the constrained optimization problem:

$$\max f(\omega_k) = m_k = \frac{1}{D}\sum_{j=1}^{D}\ln\frac{w_{kj}}{1-w_{kj}}$$

subject to $\sum_{j=1}^{D} w_{kj} = d < \frac{D}{2}$. Using the Lagrangian multiplier technique, it can be transformed into an unconstrained optimization problem: $\max f'(\omega_k, \eta) = f(\omega_k) + \eta(d - \sum_{j=1}^{D} w_{kj})$, where η is the Lagrange multiplier corresponding to the constraints. If $(\hat{\omega}_k, \hat{\eta})$ minimizes $f'(\omega_k, \eta)$, where $\hat{\omega}_k = \{\hat{w}_{k1}, \hat{w}_{k2}, \ldots, \hat{w}_{kD}\}$, then it follows that $\forall j : \frac{\partial f'(\hat{\omega}_k, \hat{\eta})}{\partial \hat{w}_{kj}} = 0$ and $\frac{\partial f'(\hat{\omega}_k, \hat{\eta})}{\partial \hat{\eta}} = 0$, yielding $\forall j : \hat{w}_{kj} = \frac{d}{D} < \frac{1}{2}$. Due to the fact that the Hessian matrix $diag[\frac{2\hat{w}_{k1}-1}{D\hat{w}_{k1}^2(1-\hat{w}_{k1})^2}, \frac{2\hat{w}_{k2}-1}{D\hat{w}_{k2}^2(1-\hat{w}_{k2})^2}, \ldots, \frac{2\hat{w}_{kD}-1}{D\hat{w}_{kD}^2(1-\hat{w}_{kD})^2}]$ is negative definite, $f(\omega_k)$ reaches its maximum when $\omega_k = \hat{\omega}_k$, which is $-\ln(\frac{D}{d} - 1)$. Thus we obtain $m_k \leq -\ln(\frac{D}{d} - 1)$. □