Part I

High-Dimensional Classification

Chapter 1

High-Dimensional Classification*

Jianqing Fan†, Yingying Fan‡ and Yichao Wu§

Abstract

In this chapter, we give a comprehensive overview on high-dimensional classification, which is prominently featured in many contemporary statistical problems. Emphasis is given on the impact of dimensionality on implementation and statistical performance and on the feature selection to enhance statistical performance as well as scientific understanding between collected variables and the outcome. Penalized methods and independence learning are introduced for feature selection in ultrahigh dimensional feature space. Popular methods such as the Fisher linear discriminant, Bayes classifiers, independence rules, distance-based classifiers and loss-based classification rules are introduced and their merits are critically examined. Extensions to multi-class problems are also given.

Keywords: Bayes classifier, classification error rates, distanced-based classifier, feature selection, impact of dimensionality, independence learning, independence rule, loss-based classifier, penalized methods, variable screening.

1 Introduction

Classification is a supervised learning technique. It arises frequently from bioinformatics such as disease classifications using high throughput data like micorarrays or SNPs and machine learning such as document classification and image recognition. It tries to learn a function from training data consisting of pairs of input features and categorical output. This function will be used to predict a class label of any valid input feature. Well known classification methods include (multiple) logistic regression, Fisher discriminant analysis, k-th-nearest-neighbor classifier, support vector machines, and many others. When the dimensionality of the input

*The authors are partly supported by NSF grants DMS-0714554, DMS-0704337, DMS-0906784, and DMS-0905561 and NIH grants R01-GM072611 and R01-CA149569.

†Department of ORFE, Princeton University, Princeton, NJ 08544, USA, E-mail: jqfan@princeton.edu

‡Information and Operations Management Department, Marshall School of Business, University of Southern California, Los Angeles, CA 90089, USA, E-mail: fanyingy@marshall.usc.edu

§Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA, E-mail: wu@stat.ncsu.edu

feature space is large, things become complicated. In this chapter we will try to investigate how the dimensionality impacts classification performance. Then we propose new methods to alleviate the impact of high dimensionality and reduce dimensionality.

We present some background on classification in Section 2. Section 3 is devoted to study the impact of high dimensionality on classification. We discuss distance-based classification rules in Section 4 and feature selection by independence rule in Section 5. Another family of classification algorithms based on different loss functions is presented in Section 6. Section 7 extends the iterative sure independent screening scheme to these loss-based classification algorithms. We conclude with Section 8 which summarizes some loss-based multicategory classification methods.

2 Elements of classifications

Suppose we have some input space $\mathcal{X}$ and some output space $\mathcal{Y}$. Assume that there are independent training data $(\mathbf{X}_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$ coming from some unknown distribution P, where Y_i is the i-th observation of the response variable and $\mathbf{X}_i$ is its associated feature or covariate vector. In classification problems, the response variable Y_i is qualitative and the set $\mathcal{Y}$ has only finite values. For example, in the cancer classification using gene expression data, each feature vector $\mathbf{X}_i$ represents the gene expression level of a patient, and the response Y_i indicates whether this patient has cancer or not. Note that the response categories can be coded by using indicator variables. Without loss of generality, we assume that there are K categories and $\mathcal{Y} = \{1, 2, \ldots, K\}$. Given a new observation $\mathbf{X}$, classification aims at finding a classification function $g : \mathcal{X} \to \mathcal{Y}$, which can predict the unknown class label Y of this new observation using available training data as accurately as possible.

To access the accuracy of classification, a loss function is needed. A commonly used loss function for classification is the *zero-one loss*:

$$L(y, g(\mathbf{x})) = \begin{cases} 0, & g(\mathbf{x}) = y, \\ 1, & g(\mathbf{x}) \neq y. \end{cases} \tag{2.1}$$

This loss function assigns a single unit to all misclassifications. Thus the risk of a classification function g, which is the expected classification error for an new observation $\mathbf{X}$, takes the following form:

$$\begin{aligned} \overline{W}(g) &= E[L(Y, g(\mathbf{X}))] = E\left[\sum_{k=1}^{K} L(k, g(\mathbf{X}))P(Y = k|\mathbf{X})\right] \\ &= 1 - P(Y = g(\mathbf{x})|\mathbf{X} = \mathbf{x}), \end{aligned} \tag{2.2}$$

where Y is the class label of $\mathbf{X}$. Therefore, the optimal classifier in terms of minimizing the misclassification rate is

$$g^*(\mathbf{x}) = \arg\max_{k \in \mathcal{Y}} P(Y = k|\mathbf{X} = \mathbf{x}) \tag{2.3}$$

This classifier is known as the *Bayes classifier* in the literature. Intuitively, Bayes classifier assigns a new observation to the most possible class by using the posterior probability of the response. By definition, Bayes classifier achieves the minimum misclassification rate over all measurable functions:

$$\overline{W}(g^*) = \min_g \overline{W}(g). \tag{2.4}$$

This misclassification rate $\overline{W}(g^*)$ is called the Bayes risk. The Bayes risk is the minimum misclassification rate when distribution is known and is usually set as the benchmark when solving classification problems.

Let $f_k(\mathbf{x})$ be the conditional density of an observation $\mathbf{X}$ being in class k, and π_k be the prior probability of being in class k with $\sum_{i=1}^K \pi_i = 1$. Then by Bayes theorem it can be derived that the posterior probability of an observation $\mathbf{X}$ being in class k is

$$P(Y = k|\mathbf{X} = \mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_{i=1}^K f_i(\mathbf{x})\pi_i}. \tag{2.5}$$

Using the above notation, it is easy to see that the Bayes classifier becomes

$$g^*(\mathbf{x}) = \arg\max_{k\in\mathcal{Y}} f_k(\mathbf{x})\pi_k. \tag{2.6}$$

For the following of this chapter, if not specified we shall consider the classification between two classes, that is, $K = 2$. The extension of various classification methods to the case where $K > 2$ will be discussed in the last section.

The *Fisher linear discriminant analysis* approaches the classification problem by assuming that both class densities are multivariate Gaussian $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, respectively, where $\boldsymbol{\mu}_k$, $k = 1, 2$ are the class mean vectors, and $\boldsymbol{\Sigma}$ is the common positive definite covariance matrix. If an observation $\mathbf{X}$ belongs to class k, then its density is

$$f_k(\mathbf{x}) = (2\pi)^{-p/2}(\det(\boldsymbol{\Sigma}))^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right\}, \tag{2.7}$$

where p is the dimension of the feature vectors $\mathbf{X}_i$. Under this assumption, the Bayes classifier assigns $\mathbf{X}$ to class 1 if

$$\pi_1 f_1(\mathbf{X}) \geqslant \pi_2 f_2(\mathbf{X}), \tag{2.8}$$

which is equivalent to

$$\log\frac{\pi_1}{\pi_2} + (\mathbf{X} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geqslant 0, \tag{2.9}$$

where $\boldsymbol{\mu} = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$. In view of (2.6), it is easy to see that the classification rule defined in (2.8) is the same as the Bayes classifier. The function

$$\delta_F(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{2.10}$$

is called the *Fisher discriminant function*. It assigns $\mathbf{X}$ to class 1 if $\delta_F(\mathbf{X}) \geqslant \log \frac{\pi_2}{\pi_1}$; otherwise to class 2. It can be seen that the Fisher discriminant function is linear in $\mathbf{x}$. In general, a classifier is said to be linear if its discriminant function is a linear function of the feature vector. Knowing the discriminant function δ_F, the classification function of Fisher discriminant analysis can be written as $g_F(\mathbf{x}) = 2 - I(\delta_F(\mathbf{x}) \geqslant \log \frac{\pi_2}{\pi_1})$ with $I(\cdot)$ the indicator function. Thus the classification function is determined by the discriminant function. In the following, when we talk about a classification rule, it could be the classification function g or the corresponding discriminant function δ.

Denote by $\boldsymbol{\theta} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ the parameters of the two Gaussian distributions $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. Write $\overline{W}(\delta, \boldsymbol{\theta})$ as the misclassification rate of a classifier with discriminant function δ. Then the discriminant function δ_B of the Bayes classifier minimizes $\overline{W}(\delta, \boldsymbol{\theta})$. Let $\Phi(t)$ be the distribution function of a univariate standard normal distribution. If $\pi_1 = \pi_2 = \frac{1}{2}$, it can easily be calculated that the misclassification rate for Fisher discriminant function is

$$\overline{W}(\delta_F, \boldsymbol{\theta}) = \Phi\left(-\frac{d^2(\boldsymbol{\theta})}{2}\right), \tag{2.11}$$

where $d(\boldsymbol{\theta}) = \{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\}^{1/2}$ and is named as the *Mahalanobis distance* in the literature. It measures the distance between two classes and was introduced by Mahalanobis (1930). Since under the normality assumption the Fisher discriminant analysis is the Bayes classifier, the misclassification rate given in (2.11) is in fact the Bayes risk. It is easy to see from (2.11) that the Bayes risk is a decreasing function of the distance between two classes, which is consistent with our common sense.

Let Γ be some parameter space. With a slight abuse of the notation, we define the maximum misclassification rate of a discriminant function δ over Γ as

$$\overline{W}_\Gamma(\delta) = \sup_{\boldsymbol{\theta} \in \Gamma} \overline{W}(\delta, \boldsymbol{\theta}). \tag{2.12}$$

It measures the worst classification result of a classifier δ over the parameter space Γ. In some cases, we are also interested in the *minimax regret* of a classifier, which is the difference between the maximum misclassification rate and the minimax misclassification rate, that is,

$$R_\Gamma(\delta) = \overline{W}_\Gamma(\delta) - \sup_{\boldsymbol{\theta} \in \Gamma} \min_{\delta} \overline{W}(\delta, \boldsymbol{\theta}). \tag{2.13}$$

Since the Bayes classification rule δ_B minimizes the misclassification rate $\overline{W}(\delta, \boldsymbol{\theta})$, the minimax regret of δ can be rewritten as

$$R_\Gamma(\delta) = \overline{W}_\Gamma(\delta) - \sup_{\boldsymbol{\theta} \in \Gamma} \overline{W}(\delta_B, \boldsymbol{\theta}). \tag{2.14}$$

From (2.11) it is easy to see that for classification between two Gaussian distributions with common covariance matrix, the minimax regret of δ is

$$R_\Gamma(\delta) = \overline{W}_\Gamma(\delta) - \sup_{\boldsymbol{\theta} \in \Gamma} \Phi\left(-\frac{1}{2} d(\boldsymbol{\theta})\right). \tag{2.15}$$

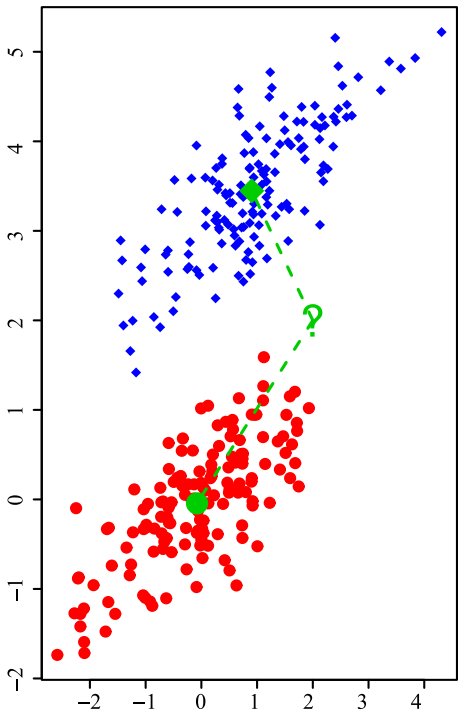

Figure 2.1 Illustration of distance-based classification. The centroid of each subsample in the training data is first computed by taking the sample mean or median. Then, for a future observation, indicated by query, it is classified according to its distances to the centroids.

The Fisher discriminant rule can be regarded as a specific method of distance-based classifiers, which have attracted much attention of researchers. Popularly used distance-based classifiers include support vector machine, naive Bayes classifier, and k-th-nearest-neighbor classifier. The distance-based classifier assigns a new observation $\mathbf{X}$ to class k if it is on average closer to the data in class k than to the data in any other classes. The "distance" and "average" are interpreted differently in different methods. Two widely used measures for distance are the Euclidean distance and the Mahalanobis distance. Assume that the center of class i distribution is $\boldsymbol{\mu}_i$ and the common convariance matrix is $\boldsymbol{\Sigma}$. Here "center" could be the mean or the median of a distribution. We use $\text{dist}(\mathbf{x}, \boldsymbol{\mu}_i)$ to denote the distance of a feature vector $\mathbf{x}$ to the centriod of class i. Then if the Euclidean distance is used,

$$\text{dist}_E(\mathbf{x}, \boldsymbol{\mu}_i) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_i)^T(\mathbf{x} - \boldsymbol{\mu}_i)}, \tag{2.16}$$

and the Mahalanobis distance between a feature vector $\mathbf{x}$ and class i is

$$\text{dist}_M(\mathbf{x}, \boldsymbol{\mu}_i) = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)}. \tag{2.17}$$

Thus the distance-based classifier places a new observation $\mathbf{X}$ to class k if

$$\arg\min_{i\in\mathcal{Y}} \text{dist}(\mathbf{X}, \boldsymbol{\mu}_i) = k. \tag{2.18}$$

Figure 2.1 illustrates the idea of distanced classifier classification.

When $\pi_1 = \pi_2 = 1/2$, the above defined Fisher discriminant analysis has the interpretation of distance-based classifier. To understand this, note that (2.9) is equivalent to

$$(\mathbf{X} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu}_1) \leqslant (\mathbf{X} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu}_2). \tag{2.19}$$

Thus δ_F assigns $\mathbf{X}$ to class 1 if its Mahalanobis distance to the center of class 1 is smaller than its Mahalanobis distance to the center of class 2. We will introduce in more details about distance-based classifiers in Section 4.

3 Impact of dimensionality on classification

A common feature of many contemporary classification problems is that the dimensionality p of the feature vector is much larger than the available training sample size n. Moreover, in most cases, only a fraction of these p features are important in classification. While the classical methods introduced in Section 2 are extremely useful, they no longer perform well or even break down in high dimensional setting. See Donoho (2000) and Fan and Li (2006) for challenges in high dimensional statistical inference. The impact of dimensionality is well understood for regression problems, but not as well understood for classification problems. In this section, we discuss the impact of high dimensionality on classification when the dimension p diverges with the sample size n. For illustration, we will consider discrimination between two Gaussian classes, and use the Fisher discriminant analysis and independence classification rule as examples. We assume in this section that $\pi_1 = \pi_2 = \frac{1}{2}$ and n_1 and n_2 are comparable.

3.1 Fisher discriminant analysis in high dimensions

Bickel and Levina (2004) theoretically studied the asymptotical performance of the sample version of Fisher discriminant analysis defined in (2.10), when both the dimensionality p and sample size n goes to infinity with p much larger than n. The parameter space considered in their paper is

$$\Gamma_1 = \{\boldsymbol{\theta} : d^2(\boldsymbol{\theta}) \geqslant c^2, c_1 \leqslant \lambda_{\min}(\boldsymbol{\Sigma}) \leqslant \ \lambda_{\max}(\boldsymbol{\Sigma}) \leqslant c_2, \boldsymbol{\mu}_k \in B, k = 1, 2\}, \tag{3.1}$$

where c, c_1 and c_2 are positive constants, $\lambda_{\min}(\boldsymbol{\Sigma})$ and $\lambda_{\max}(\boldsymbol{\Sigma})$ are the minimum and maximum eigenvalues of $\boldsymbol{\Sigma}$, respectively, and $B = B_{\mathbf{a},d} = \{\mathbf{u} : \sum_{j=1}^{\infty} a_j u_j^2 < d^2\}$ with d some constant, and $a_j \to \infty$ as $j \to \infty$. Here, the mean vectors $\boldsymbol{\mu}_k$, $k = 1, 2$ are viewed as points in l_2 by adding zeros at the end. The condition on eigenvalues ensures that $\frac{\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Sigma})} \leqslant \frac{c_2}{c_1} < \infty$, and thus both $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ are not ill-conditioned. The condition $d^2(\boldsymbol{\theta}) \geqslant c^2$ is to make sure that the Mahalanobis

distance between two classes is at least c. Thus the smaller the value of c, the harder the classification problem is.

Given independent training data $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$, the common covariance matrix can be estimated by using the sample covariance matrix

$$\widehat{\mathbf{\Sigma}} = \sum_{k=1}^{K} \sum_{Y_i = k} (\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_k)(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_k)^T / (n - K). \tag{3.2}$$

For the mean vectors, Bickel and Levina (2004) showed that there exist estimators $\widetilde{\boldsymbol{\mu}}_k$ of $\boldsymbol{\mu}_k$, $k = 1, 2$ such that

$$\max_{\Gamma_1} E_{\boldsymbol{\theta}} \|\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\|^2 = o(1). \tag{3.3}$$

Replacing the population parameters in the definition of δ_F by the above estimators $\widetilde{\boldsymbol{\mu}}_k$ and $\widehat{\mathbf{\Sigma}}$, we obtain the sample version of Fisher discriminant function $\hat{\delta}_F$.

It is well known that for fixed p, the worst case misclassification rate of $\hat{\delta}_F$ converges to the worst case Bayes risk over Γ_1, that is,

$$\overline{W}_{\Gamma_1}(\hat{\delta}_F) \to \overline{\Phi}(c/2), \text{ as } n \to \infty, \tag{3.4}$$

where $\overline{\Phi}(t) = 1 - \Phi(t)$ is the tail probability of the standard Gaussian distribution. Hence, $\hat{\delta}_F$ is asymptotically optimal for this low dimensional problem. However, in high dimensional setting, the result is very different.

Bickel and Levina (2004) studied the worst case misclassification rate of $\hat{\delta}_F$ when $n_1 = n_2$ in high dimensional setting. Specifically they showed that under some regularity conditions, if $p/n \to \infty$, then

$$\overline{W}_{\Gamma_1}(\hat{\delta}_F) \to \frac{1}{2}, \tag{3.5}$$

where the Moore-Penrose generalized inverse is used in the definition of $\hat{\delta}_F$. Note that $1/2$ is the misclassification rate of random guessing. Thus although Fisher discriminant analysis is asymptotically optimal and has Bayes risk when dimension p is fixed and sample size $n \to \infty$, it performs asymptotically no better than random guessing when the dimensionality p is much larger than the sample size n. This shows the difficulty of high dimensional classification. As have been demonstrated by Bickel and Levina (2004) and pointed out by Fan and Fan (2008), the bad performance of Fisher discriminant analysis is due to the diverging spectra (e.g., the condition number goes to infinity as dimensionality diverges) frequently encountered in the estimation of high-dimensional covariance matrices. In fact, even if the true covariance matrix is not ill conditioned, the singularity of the sample covariance matrix will make the Fisher discrimination rule inapplicable when the dimensionality is larger than the sample size.

3.2 Impact of dimensionality on independence rule

Fan and Fan (2008) studied the impact of high dimensionality on classification. They pointed out that the difficulty of high dimensional classification is intrinsically caused by the existence of many noise features that do not contribute to the

reduction of classification error. For example, for the Fisher discriminant analysis discussed before, one needs to estimate the class mean vectors and covariance matrix. Although individually each parameter can be estimated accurately, aggregated estimation error over many features can be very large and this could significantly increase the misclassification rate. This is another important reason that causes the bad performance of Fisher discriminant analysis in high dimensional setting. Greenshtein and Ritov (2004) and Greenshtein (2006) introduced and studied the concept of persistence, which places more emphasis on misclassification rates or expected loss rather than the accuracy of estimated parameters. In high dimensional classification, since we care much more about the misclassification rate instead of the accuracy of the estimated parameters, estimating the full covariance matrix and the class mean vectors will result in very high accumulation error and thus low classification accuracy.

To formally demonstrate the impact of high dimensionality on classification, Fan and Fan (2008) theoretically studied the *independence rule*. The discriminant function of independence rule is

$$\delta_I(\mathbf{x}) = (\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{D}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{3.6}$$

where $\boldsymbol{D} = \text{diag}\{\boldsymbol{\Sigma}\}$. It assigns a new observation $\mathbf{X}$ to class 1 if $\delta_I(\mathbf{X}) \geqslant 0$. Compared to the Fisher discriminant function, the independence rule pretends that features were independent and use the diagonal matrix $\boldsymbol{D}$ instead of the full covariance matrix $\boldsymbol{\Sigma}$ to scale the feature. Thus the aforementioned problems of diverging spectrum and singularity are avoided. Moreover, since there are far less parameters need to be estimated when implementing the independence rule, the error accumulation problem is much less serious when compared to the Fisher discriminant function.

Using the sample mean $\widehat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{Y_i=k}\mathbf{X}_i$, $k = 1, 2$ and sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ as estimators and letting $\widehat{\boldsymbol{D}} = \text{diag}\{\widehat{\boldsymbol{\Sigma}}\}$, we obtain the sample version of independence rule

$$\hat{\delta}_I(\mathbf{x}) = (\mathbf{x}-\widehat{\boldsymbol{\mu}})^T \widehat{\boldsymbol{D}}^{-1}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_2). \tag{3.7}$$

Fan and Fan (2008) studied the theoretical performance of $\hat{\delta}_I(\mathbf{x})$ in high dimensional setting.

Let $\mathbf{R} = \boldsymbol{D}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{D}^{-1/2}$ be the common correlation matrix and $\lambda_{\max}(\mathbf{R})$ be its largest eigenvalue, and write $\boldsymbol{\alpha} \equiv (\alpha_1, \ldots, \alpha_p)^T = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. Fan and Fan (2008) considered the parameter space

$$\Gamma_2 = \{(\boldsymbol{\alpha}, \boldsymbol{\Sigma}) : \boldsymbol{\alpha}'\boldsymbol{D}^{-1}\boldsymbol{\alpha} \geqslant C_p, \lambda_{\max}(\mathbf{R}) \leqslant b_0, \min_{1\leqslant j\leqslant p} \sigma_j^2 > 0\}, \tag{3.8}$$

where C_p is a deterministic positive sequence depending only on the dimensionality p, b_0 is a positive constant, and σ_j^2 is the j-th diagonal element of $\boldsymbol{\Sigma}$. The condition $\boldsymbol{\alpha}^T\boldsymbol{D}\boldsymbol{\alpha} \geqslant C_p$ is similar to the condition $d(\boldsymbol{\theta}) \geqslant c$ in Bickel and Levina (2004). In fact, $\boldsymbol{\alpha}'\boldsymbol{D}^{-1}\boldsymbol{\alpha}$ is the accumulated marginal signal strength of p individual features, and the condition $\boldsymbol{\alpha}'\boldsymbol{D}^{-1}\boldsymbol{\alpha} \geqslant C_p$ imposes a lower bound on it. Since there is no

restriction on the smallest eigenvalue, the condition number of $\mathbf{R}$ can diverge with sample size. The last condition $\min_{1\leqslant j\leqslant p}\sigma_j^2 > 0$ ensures that there are no deterministic features that make classification trivial and the diagonal matrix $\boldsymbol{D}$ is always invertible. It is easy to see that Γ_2 covers a large family of classification problems.

To access the impact of dimensionality, Fan and Fan (2008) studied the posterior misclassification rate and the worst case posterior misclassification rate of $\hat{\delta}_I$ over the parameter space Γ_2. Let $\mathbf{X}$ be a new observation from class 1. Define the posterior misclassification rate and the worst case posterior misclassification rate respectively as

$$W(\hat{\delta}_I, \boldsymbol{\theta}) = P(\hat{\delta}_I(\mathbf{X}) < 0 | (\mathbf{X}_i, Y_i), i = 1, \ldots, n), \tag{3.9}$$

$$W_{\Gamma_2}(\hat{\delta}_I) = \max_{\boldsymbol{\theta}\in\Gamma_2} W(\hat{\delta}_I, \boldsymbol{\theta}). \tag{3.10}$$

Fan and Fan (2008) showed that when $\log p = o(n)$, $n = o(p)$ and $nC_p \to \infty$, the following inequality holds

$$W(\hat{\delta}_I, \boldsymbol{\theta}) \leqslant \overline{\Phi}\left(\frac{\sqrt{\frac{n_1 n_2}{pn}}\boldsymbol{\alpha}'\boldsymbol{D}^{-1}\boldsymbol{\alpha}(1+o_p(1)) + \sqrt{\frac{p}{nn_1n_2}}(n_1-n_2)}{2\sqrt{\lambda_{\max}(\mathbf{R})}\big\{1+n_1n_2/(pn)\boldsymbol{\alpha}'\boldsymbol{D}^{-1}\boldsymbol{\alpha}(1+o_p(1))\big\}^{1/2}}\right). \tag{3.11}$$

This inequality gives an upper bound on the classification error. Since $\overline{\Phi}(\cdot)$ decreases with its argument, the right hand side decreases with the fraction inside $\overline{\Phi}$. The second term in the numerator of the fraction shows the influence of sample size on classification error. When there are more training data from class 1 than those from class 2, i.e., $n_1 > n_2$, the fraction tends to be larger and thus the upper bound is smaller. This is in line with our common sense, as if there are more training data from class 1, then it is less likely that we misclassify $\mathbf{X}$ to class 2.

Fan and Fan (2008) further showed that if $\sqrt{n_1n_2/(np)}C_p \to C_0$ with C_0 some positive constant, then the worst case posterior classification error

$$W_{\Gamma_2}(\hat{\delta}_I) \xrightarrow{\text{P}} \overline{\Phi}\Big(\frac{C_0}{2\sqrt{b_0}}\Big). \tag{3.12}$$

We make some remarks on the above result (3.12). First of all, the impact of dimensionality is shown as $C_p/\sqrt{p}$ in the definition of C_0. As dimensionality p increases, so does the aggregated signal C_p, but a price of the factor $\sqrt{p}$ needs to be paid for using more features. Since n_1 and n_2 are assumed to be comparable, $n_1n_2/(np) = O(n/p)$. Thus one can see that asymptotically $W_{\Gamma_2}(\hat{\delta}_I)$ increases with $\sqrt{n/p}C_p$. Note that $\sqrt{n/p}C_p$ measures the tradeoff between dimensionality p and the overall signal strength C_p. When the signal level is not strong enough to balance out the increase of dimensionality, i.e., $\sqrt{n/p}C_p \to 0$ as $n \to \infty$, then $W_{\Gamma_2}(\hat{\delta}_I) \xrightarrow{\text{P}} \frac{1}{2}$. This indicates that the independence rule $\hat{\delta}_I$ would be no better than the random guessing due to noise accumulation, and using less features can be beneficial.

The inequality (3.11) is very useful. Observe that if we only include the first m features $j = 1, \ldots, m$ in the independence rule, then (3.11) still holds with each

term replaced by its truncated version and p replaced by m. The contribution of the j feature is governed by its marginal utility α_j^2/σ_j^2. Let us assume that the importance of the features is already ranked in the descending order of $\{\alpha_j^2/\sigma_j^2\}$. Then $m^{-1}\sum_{j=1}^m \alpha_j^2/\sigma_j^2$ will most possibly first increase and then decrease as we include more and more features, and thus the right hand side of (3.11) first decreases and then increases with m. Minimizing the upper bound in (3.11) can help us to find the optimal number of features m.

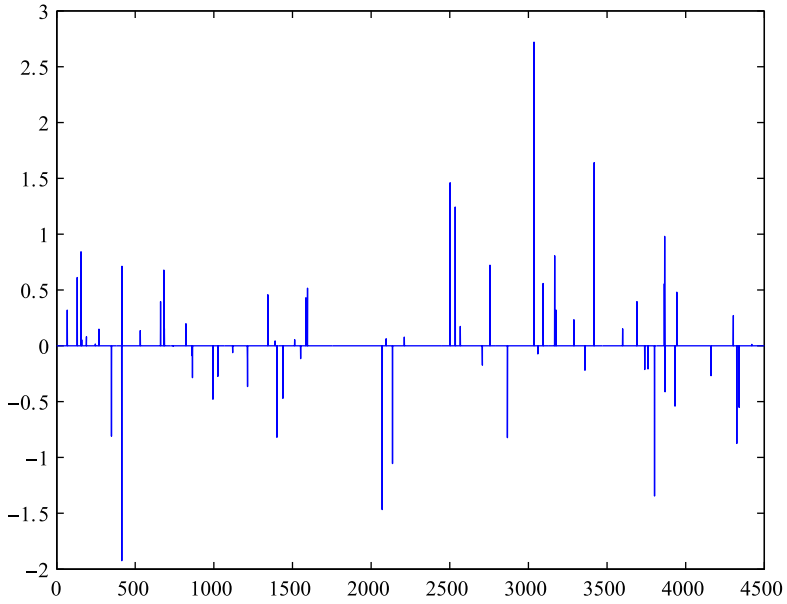

Figure 3.1 The centroid $\boldsymbol{\mu}_1$ of class 1. The heights indicate the values of non-vanishing elements.

To illustrate the impact of dimensionality, let us take $p = 4500$, $\boldsymbol{\Sigma}$ the identity matrix, and $\boldsymbol{\mu}_2 = 0$ whereas $\boldsymbol{\mu}_1$ has 98% of coordinates zero and 2% non-zero, generated from the double exponential distribution. Figure 3.1 illustrates the vector $\boldsymbol{\mu}_1$, in which the heights show the values of non-vanishing coordinates. Clearly, only about 2% of features have some discrimination power. The effective number of features that have reasonable discrimination power (excluding those with small values) is much smaller. If the best two features are used, it clearly has discrimination power, as shown in Figure 3.2(a), whereas when all 4500 features are used, they have little discrimination power (see Figure 3.2(d)) due to noise accumulation. When $m = 100$ (about 90 features are useful and 10 useless: the actual useful signals are less than 90 as many of them are weak) the signals are strong enough to overwhelm the noise accumulation, where as when $m = 500$ (at least 410 features are useless), the noise accumulation exceeds the strength of the signals so that there is no discrimination power.

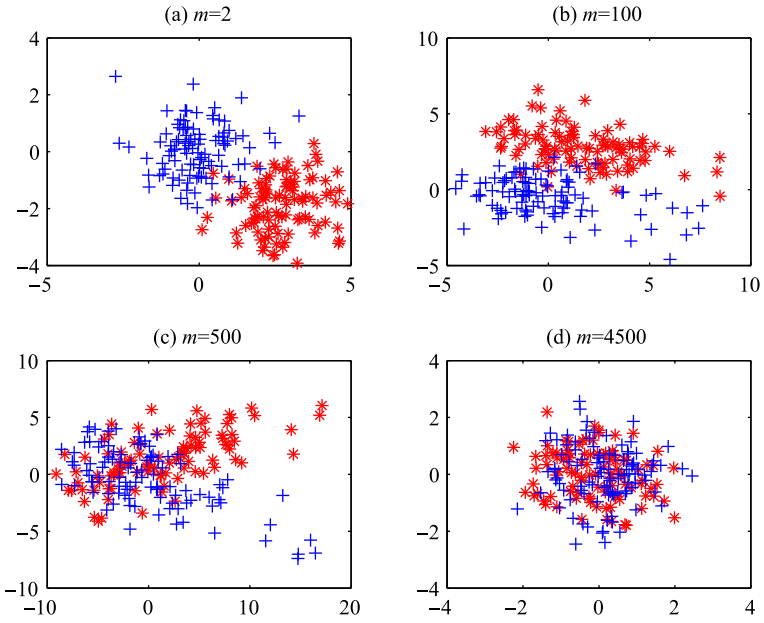

Figure 3.2 The plot of simulated data with "∗" indicated the first class and "+" the second class. The best m features are selected and the first two principal components are computed based on the sample covariance matrix. The data are then projected onto these two principal components and are shown in (a), (b) and (c). In (d), the data are projected on two randomly selected directions in the 4500-dimensional space.

3.3 Linear discriminants in high dimensions

From discussions in the previous two subsections, we see that in high dimensional setting, the performance of classifiers is very different from their performance when dimension is fixed. As we have mentioned earlier, the bad performance is largely caused by the error accumulation when estimating too many noise features with little marginal utility α_j^2/σ_j^2. Thus dimension reduction and feature selection are very important in high dimensional classification.

A popular class of dimension reduction methods is projection. See, for example, principal component analysis in Ghosh (2002), Zou et al. (2004), and Bair et al. (2006); partial least squares in Nguyen and Rocke (2002), Huang and Pan (2003), and Boulesteix (2004); and sliced inverse regression in Li (1991), Zhu et al. (2006), and Bura and Pfeiffer (2003). As pointed out by Fan and Fan (2008), these projection methods attempt to find directions that can result in small classification errors. In fact, the directions found by these methods put much more weight on features that have large classification power. In general, however, linear projection methods are likely to perform poorly unless the projection vector is sparse, namely, the effective number of selected features is small. This is due to the aforementioned noise accumulation prominently featured in high-dimensional

problems.

To formally establish the result, let $\mathbf{a}$ be a p-dimensional unit random vector coming from a uniform distribution over a $(p-1)$-dimensional sphere. Suppose that we project all observations onto the vector $\mathbf{a}$ and apply the Fisher discriminant analysis to the projected data $\mathbf{a}^T\mathbf{X}_1, \ldots, \mathbf{a}^T\mathbf{X}_n$, that is, we use the discriminant function

$$\hat{\delta}_{\mathbf{a}}(\mathbf{x}) = (\mathbf{a}^T\mathbf{x} - \mathbf{a}^T\widehat{\boldsymbol{\mu}})(\mathbf{a}^T\widehat{\boldsymbol{\mu}}_1 - \mathbf{a}^T\widehat{\boldsymbol{\mu}}_2). \tag{3.13}$$

Fan and Fan (2008) showed that under some regularity conditions, if $p^{-1}\sum_{i=1}^{p} \alpha_j^2/\sigma_j^2 \to 0$, then

$$P(\hat{\delta}_{\mathbf{a}}(\mathbf{X}) < 0 | (\mathbf{X}_i, Y_i), i = 1, \ldots, n) \xrightarrow{\text{P}} \frac{1}{2}, \tag{3.14}$$

where $\mathbf{X}$ is a new observation coming from class 1, and the probability is taken with respect to the random vector $\mathbf{a}$ and new observation $\mathbf{X}$ from class 1. The result demonstrates that almost all linear discriminants cannot perform any better than random guessing, due to the noise accumulation in the estimation of population mean vectors, unless the signals are very strong, namely the population mean vectors are very far apart. In fact, since the projection direction vector $\mathbf{a}$ is randomly chosen, it is nonsparse with probability one. When a nonsparse projection vector is used, one essentially uses all features to do classification, and thus the misclassification rate could be as high as random guessing due to the noise accumulation. This once again shows the importance of feature selection in high dimensionality classification. To illustrate the point, Figure 3.2(d) shows the projected data onto two randomly selected directions. Clearly, neither projections has discrimination power.

4 Distance-based classification rules

Many distance-based classifiers have been proposed in the literature to deal with classification problems with high dimensionality and small sample size. They intend to mitigate the "curse-of-dimensionality" in implementation. In this section, we will first discuss some specific distance-based classifiers, and then talk about the theoretical properties of general distance-based classifiers.

4.1 Naive Bayes classifier

As discussed in Section 2, the Bayes classifier predicts the class label of a new observation by comparing the posterior probabilities of the response. It follows from the Bayes theorem that

$$P(Y = k | \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} | Y = k)\pi_k}{\sum_{i=1}^{K} P(\mathbf{X} = \mathbf{x} | Y = i)\pi_i}. \tag{4.1}$$

Since $P(\mathbf{X} = \mathbf{x}|Y = i)$ and π_i, $i = 1, \ldots, K$ are unknown in practice, to implement the Bayes classifier we need to estimate them from the training data. However,

this method is impractical in high dimensional setting due to the curse of dimensionality and noise accumulation when estimating the distribution $P(\mathbf{X}|Y)$, as discussed in Section 3. The naive Bayes classifier, on the other hand, overcomes this difficulty by making a conditional independence assumption that dramatically reduces the number of parameters to be estimated when modeling $P(\mathbf{X}|Y)$. More specifically, the naive Bayes classifier uses the following calculation:

$$P(\mathbf{X} = \mathbf{x}|Y = k) = \prod_{j=1}^{p} P(X_j = x_j|Y = k), \tag{4.2}$$

where X_j and x_j are the j-th components of $\mathbf{X}$ and $\mathbf{x}$, respectively. Thus the conditional joint distribution of the p features depends only on the marginal distributions of them. So the naive Bayes rule utilizes the marginal information of features to do classification, which mitigates the "curse-of-dimensionality" in implementation. But, the dimensionality does have an impact on the performance of the classifier, as shown in the previous section. Combining (2.6), (4.1) and (4.2) we obtain that the predicted class label by naive Bayes classifier for a new observation is

$$g(\mathbf{x}) = \arg\max_{k\in\mathcal{Y}} \pi_k \prod_{j=1}^{p} P(X_j = x_j|Y = k). \tag{4.3}$$

In the case of classification between two normal distributions $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ with $\pi_1 = \pi_2 = \frac{1}{2}$, it can be derived that the naive Bayes classifier has the discriminant function

$$\delta_I(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{D}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{4.4}$$

where $\boldsymbol{D} = \text{diag}(\boldsymbol{\Sigma})$, the same as the independence rule (3.7), which assigns a new observation $\mathbf{X}$ to class 1 if $\delta_I(\mathbf{X}) \geqslant 0$; otherwise to class 2. It is easy to see that $\delta_I(\mathbf{x})$ is a distance-based classifier with distance measure chosen to be the weighted L_2-distance: $\text{dist}_I(\mathbf{x}, \boldsymbol{\mu}_i) = (\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{D}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)$.

Although in deriving the naive Bayes classifier, it is assumed that the features are conditionally independent, in practice it is widely used even when this assumption is violated. In other words, the naive Bayes classifier pretends that the features were conditionally independent with each other even if they are actually not. For this reason, the naive Bayes classifier is also called independence rule in the literature. In this chapter, we will interchangeably use the name "naive Bayes classifier" and "independence rule".

As pointed out by Bickel and Levina (2004), even when $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are assumed known, the corresponding independence rule does not lose much in terms of classification power when compared to the Bayes rule defined in (2.10). To understand this, Bickel and Levina (2004) consider the errors of Bayes rule and independence rule, which can be derived to be

$$e_1 = P(\delta_B(\mathbf{X}) \leqslant 0) = \overline{\Phi}\left(\frac{1}{2}\{\boldsymbol{\alpha}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}\}^{1/2}\right) \text{ and}$$

$$e_2 = P(\delta_I(\mathbf{X}) \leqslant 0) = \overline{\Phi}\left(\frac{1}{2}\frac{\boldsymbol{\alpha}^T\boldsymbol{D}^{-1}\boldsymbol{\alpha}}{\{\boldsymbol{\alpha}^T\boldsymbol{D}^{-1}\boldsymbol{\Sigma}\boldsymbol{D}^{-1}\boldsymbol{\alpha}\}^{1/2}}\right),$$

respectively. Since the errors e_k, $k = 1, 2$ are both decreasing functions of the arguments of $\overline{\Phi}$, the efficiency of the independence rule relative to the Bayes rule is determined by the ratio r of the arguments of $\overline{\Phi}$. Bickel and Levina (2004) showed that the ratio r can be bounded as

$$r = \frac{\overline{\Phi}^{-1}(e_2)}{\overline{\Phi}^{-1}(e_1)} = \frac{\boldsymbol{\alpha}^T \boldsymbol{D}^{-1} \boldsymbol{\alpha}}{\left\{(\boldsymbol{\alpha}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha})(\boldsymbol{\alpha}^T \boldsymbol{D}^{-1} \boldsymbol{\Sigma} \boldsymbol{D}^{-1} \boldsymbol{\alpha})\right\}^{1/2}} \geqslant \frac{2\sqrt{K_0}}{1 + K_0}. \tag{4.5}$$

where $K_0 = \max_{\Gamma_1} \frac{\lambda_{\max}(\mathbf{R})}{\lambda_{\min}(\mathbf{R})}$ with $\mathbf{R}$ the common correlation matrix defined in Section 3.2. Thus the error e_2 of the independence rule can be bounded as

$$e_1 \leqslant e_2 \leqslant \overline{\Phi}\left(\frac{2\sqrt{K_0}}{1 + K_0} \overline{\Phi}^{-1}(e_1)\right). \tag{4.6}$$

It can be seen that for moderate K_0, the performance of independence rule is comparable to that of the Fisher discriminant analysis. Note that the bounds in (4.6) represents the worst case performance. The actual performance of independence rule could be better. In fact, in practice when $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ both need to be estimated, the performance of independence rule is much better than that of the Fisher discriminant analysis.

We use the same notation as that in Section 3, that is, we use $\hat{\delta}_F$ to denote the sample version of Fisher discriminant function, and use $\hat{\delta}_I$ to denote the sample version of the independence rule. Bickel and Levina (2004) theoretically compared the asymptotical performance of $\hat{\delta}_F$ and $\hat{\delta}_I$. The asymptotic performance of Fisher discriminant analysis is given in (3.5). As for the independence rule, under some regularity conditions, Bickel and Levina (2004) showed that if $\log p/n \to 0$, then

$$\limsup_{n \to \infty} \overline{W}_{\Gamma_1}(\hat{\delta}_I) = \overline{\Phi}\Big(\frac{\sqrt{K_0}}{1 + \sqrt{K_0}} c\Big), \tag{4.7}$$

where Γ_1 is the parameter set defined in Section 3.1. Recall that (3.5) shows that the Fisher discriminant analysis asymptotically performs no better than random guessing when the dimensionality p is much larger than the sample size n. While the above result (4.7) demonstrates that for the independence rule, the worst case classification error is better than that of the random guessing, as long as the dimensionality p does not grow exponentially faster than the sample size n and $K_0 < \infty$. This shows the advantage of independence rule in high dimensional classification. Note that the impact of dimensionality can not be seen in (4.7) whereas it can be seen from (3.12). This is due to the difference of Γ_2 from Γ_1.

On the practical side, Dudoit et al. (2002) compared the performance of various classification methods, including the Fisher discriminant analysis and the independence rule, for the classification of tumors based on gene expression data. Their results show that the independence rule outperforms the Fisher discriminant analysis.

Bickel and Levina (2004) also introduced a spectrum of classification rules which interpolate between $\hat{\delta}_F$ and $\hat{\delta}_I$ under the Gaussian coloured noise model assumption. They showed that the minimax regret of their classifier has asymptotic

rate $O(n^{-\kappa}\log n)$ with κ some positive number defined in their paper. See Bickel and Levina (2004) for more details.

4.2 Centroid rule and k-nearest-neighbor rule

Hall et al. (2005) have given the geometric representation of high dimensional, low sample size data, and used it to analyze the performance of several distance-based classifiers, including the centroid rule and 1-nearest neighbor rule. In their analysis, the dimensionality $p \to \infty$ while the sample size n is fixed.

To appreciate their results, we first introduce some notations. Consider classification between two classes. Assume that within each class, observations are independent and identically distributed. Let $\mathbf{Z}_1 = (Z_{11}, Z_{12}, \ldots, Z_{1p})^T$ be an observation from class 1, and $\mathbf{Z}_2 = (Z_{21}, Z_{22}, \ldots, Z_{2p})^T$ be an observation from class 2. Assume the following results hold as $p \to \infty$

$$\frac{1}{p}\sum_{j=1}^{p}\mathrm{var}(Z_{1j}) \to \sigma^2,\ \frac{1}{p}\sum_{j=1}^{p}\mathrm{var}(Z_{2j}) \to T^2,$$
$$\frac{1}{p}\sum_{j=1}^{p}[E(Z_{1j}^2) - E(Z_{2j}^2)] \to \kappa^2, \tag{4.8}$$

where σ, T and κ are some positive constants. Let C_k be the centroid of the training data from class k, where $k = 1, 2$. Here, the centroid C_k could be the mean or median of data in class k.

The "centroid rule" or "mean difference rule" classifies a new observation to class 1 or class 2 according to its distance to their centroids. This approach is popular in genomics. To study the theoretical property of this method, Hall et al. (2005) first assumed that $\sigma^2/n_1 \geqslant T^2/n_2$. They argued that if needed, the roles for class 1 and class 2 can be interchanged to achieve this. Then under some regularity conditions, they showed that if $\kappa^2 \geqslant \sigma^2/n_1 - T^2/n_2$, then the probability that a new datum from either class 1 or class 2 is correctly classified by the centroid rule converges to 1 as $p \to \infty$; If instead $\kappa^2 < \sigma^2/n_1 - T^2/n_2$, then with probability converging to 1, a new datum from either class will be classified by the centroid rule as belonging to class 2. This property is also enjoyed by the support vector machine method which to be discussed in a later section.

The nearest-neighbor rule uses those training data closest to $\mathbf{X}$ to predict the label of $\mathbf{X}$. Specifically, the k-nearest-neighbor rule predicts the class label of $\mathbf{X}$ as

$$\delta(\mathbf{X}) = \frac{1}{k}\sum_{\mathbf{X}_i \in N_k(\mathbf{X})} Y_i, \tag{4.9}$$

where $N_k(\mathbf{X})$ is the neighborhood of $\mathbf{X}$ defined by the k closest observations in the training sample. For two-class classification problems, $\mathbf{X}$ is assigned to class 1 if $\delta(\mathbf{X}) < 1.5$. This is equivalent to the majority vote rule in the "committee" $N_k(\mathbf{X})$. For more details on the nearest-neighbor rule, see Hastie et al. (2009).

Hall et al. (2005) also considered the 1-nearest-neighbor rule. They first assumed that $\sigma^2 \geqslant T^2$. The same as before, the roles of class 1 and class 2 can be interchanged to achieve this. They showed that if $\kappa^2 > \sigma^2 - T^2$, then the probability that a new datum from either class 1 or class 2 is correctly classified by the 1-nearest-neighbor rule converges to 1 as $p \to \infty$; if instead $\kappa^2 < \sigma^2 - T^2$, then with probability converging to 1, a new datum from either class will be classified by the 1-nearest-neighbor rule as belonging to class 2.

Hall et al. (2005) further discussed the contrasts between the centroid rule and the 1-nearest-neighbor rule. For simplicity, they assumed that $n_1 = n_2$. They pointed out that asymptotically, the centroid rule misclassifies data from at least one of the classes only when $\kappa^2 < |\sigma^2 - T^2|/n_1$, whereas the 1-nearest-neighbor rule leads to misclassification for data from at least one of the classes both in the range $\kappa^2 < |\sigma^2 - T^2|/n_1$ and when $|\sigma^2 - T^2|/n_1 \leqslant \kappa^2 < |\sigma^2 - T^2|$. This quantifies the inefficiency that might be expected from basing inference only on a single nearest neighbor. For the choice of k in the nearest neighbor, see Hall, Park and Samworth (2008).

For the properties of both classifiers discussed in this subsection, it can be seen that their performances are greatly determined by the value of κ^2. However, in view of (4.8), κ^2 could be very small or even 0 in high dimensional setting due the the existence of many noise features that have very little or no classification power (i.e. those with $EZ_{1j}^2 \approx EZ_{2j}^2$). This once again shows the difficulty of classification in high dimensional setting.

4.3 Theoretical properties of distance-based classifiers

Hall, Pittelkow and Ghosh (2008) suggested an approach to accessing the theoretical performance of general distance-based classifiers. This technique is related to the concept of "detection boundary" developed by Ingster and Donoho and Jin. See, for example, Donoho and Jin (2004); Hall and Jin (2008). Ingster (2002); and Jin (2006); Hall, Pittelkow and Ghosh (2008) studied the theoretical performance of a variety distance-based classifiers constructed from high dimensional data, and obtain the classification boundaries for them. We discuss their study in this subsection.

Let $g(\cdot) = g(\cdot|(\mathbf{X}_i, Y_i), i = 1, \ldots, n)$ be a distanced-based classifier which assigns a new observation $\mathbf{X}$ to either class 1 or class 2. Hall, Pittelkow and Ghosh (2008) argued that any plausible, distance-based classifier g should enjoy the following two properties:

(a) g assigns $\mathbf{X}$ to class 1 if it is closer to each of the $\mathbf{X}_i'$s in class 1 than it is to any of the $\mathbf{X}_j'$s in class 2;
(b) If g assigns $\mathbf{X}$ to class 1 then at least one of the $\mathbf{X}_i'$s in class 1 is closer to $\mathbf{X}$ than $\mathbf{X}$ is closer to the most distant $\mathbf{X}_j'$s in class 2.

These two properties together imply that

$$\pi_{k1} \leqslant P_k(g(\mathbf{X}) = k) \leqslant \pi_{k2}, \text{ for } k = 1, 2, \tag{4.10}$$

where P_k denotes the probability measure when assuming that $\mathbf{X}$ is from class k

with $k = 1, 2$, and π_{k1} and π_{k2} are defined as

$$\pi_{k1} = P_k\Big(\max_{i\in\mathcal{G}_1} \|\mathbf{X}_i - \mathbf{X}\| \leqslant \min_{j\in\mathcal{G}_2} \|\mathbf{X}_j - \mathbf{X}\|\Big) \text{ and} \tag{4.11}$$

$$\pi_{k2} = P_k\Big(\min_{i\in\mathcal{G}_1} \|\mathbf{X}_i - \mathbf{X}\| \leqslant \max_{j\in\mathcal{G}_2} \|\mathbf{X}_j - \mathbf{X}\|\Big) \tag{4.12}$$

with $\mathcal{G}_1 = \{i : 1 \leqslant i \leqslant n, Y_i = 1\}$ and $\mathcal{G}_2 = \{i : 1 \leqslant i \leqslant n, Y_i = 2\}$. Hall, Pittelkow and Ghosh (2008) considered a family of distance-based classifiers satisfying condition (4.10).

To study the theoretical property of these distance-based classifiers, Hall, Pittelkow and Ghosh (2008) considered the following model

$$X_{ij} = \mu_{kj} + \varepsilon_{ij}, \text{ for } i \in \mathcal{G}_k, \ k = 1, 2, \tag{4.13}$$

where X_{ij} denotes the j-th component of $\mathbf{X}_i$, μ_{kj} represents the j-th component of mean vector $\boldsymbol{\mu}_k$, and ε_{ij}'s are independent and identically distributed with mean 0 and finite fourth moment. Without loss of generality, they assumed that the class 1 population mean vector $\boldsymbol{\mu}_1 = \mathbf{0}$. Under this model assumption, they showed that if some mild conditions are satisfied, $\pi_{k1} \to 0$ and $\pi_{k2} \to 0$ if and only if $p = o(\|\boldsymbol{\mu}_2\|^4)$. Then using inequality (4.10), they obtained that the probability of the classifier g correctly classifying a new observation from class 1 or class 2 converges to 1 if and only if $p = o(\|\boldsymbol{\mu}_2\|^4)$ as $p \to \infty$. This result tells us just how fast the norm of the two class mean difference vector $\boldsymbol{\mu}_2$ must grow for it to be possible to distinguish perfectly the two classes using the distance-based classifier. Note that the above result is independent of the sample size n. The result is consistent with (a specific case of) (3.12) for independent rule in which $C_p = \|\boldsymbol{\mu}_2\|^2$ in the current setting and misclassification rate goes to zero when the signal is so strong that $C_p^2/p \to \infty$ (if n is fixed) or $\|\boldsymbol{\mu}_2\|^4/p \to \infty$. The impact of dimensionality is implied by the quantity $\|\boldsymbol{\mu}_2\|^2/\sqrt{p}$.

It is well known that the thresholding methods can improve the sensitivity of distance-based classifiers. The thresholding in this setting is a feature selection method, using only features with distant away from the other. Denote by $X_{ij}^{tr} = X_{ij}I(X_{ij} > t)$ the thresholded data, $i = 1, \ldots, n$, $j = 1, \ldots, p$, with t the thresholding level. Let $\mathbf{X}_i^{tr} = (X_{ij}^{tr})$ be the thresholded vector and g^{tr} be the version of the classifier g based on thresholded data. The case where the absolute values $|X_{ij}|$ are thresholded is very similar. Hall, Pittelkow and Ghosh (2008) studied the properties of the threshold-based classifier g^{tr}. For simplicity, they assumed that $\mu_{2j} = \nu$ for q distinct indices j, and $\mu_{2j} = 0$ for the remaining $p - q$ indices, where

(a) $\nu \geqslant t$,
(b) $t = t(p) \to \infty$ as p increases,
(c) $q = q(p)$ satisfies $q \to \infty$ and $1 \leqslant q \leqslant cp$ with $0 < c < 1$ fixed, and
(d) the errors ε_{ij} has a distribution that is unbounded to the right.

With the above assumptions and some regularity conditions, they proved that the general thresholded distance-based classifier g^{tr} has a property that is analogue to the standard distance-based classifier, that is, the probability that the classifier

g^{tr} correctly classifies a new observation from class 1 or class 2 tending to 1 if and only if $p = o(T)$ as $p \to \infty$, where $T = (q\nu^2)^2/E[\varepsilon_{ij}^4 I(\varepsilon_{ij} > t)]$. Compared to the property of standard distance-based classifier, the thresholded classifier allows for higher dimensionality if $E[\varepsilon_{ij}^4 I(\varepsilon_{ij} > t)] \to 0$ as $p \to \infty$.

Hall, Pittelkow and Ghosh (2008) further compared the theoretical performance of standard distanced-based classifiers and thresholded distance-based classifiers by using the classification boundaries. To obtain the explicit form of classification boundaries, they assumed that for j-th feature, the class 1 distribution is $GN_\gamma(0,1)$ and the class 2 distribution is $GN_\gamma(\mu_{2j},1)$, respectively. Here $GN_\gamma(\mu,\sigma^2)$ denotes the Subbotin, or generalized normal distribution with probability density

$$f(x|\gamma,\mu,\sigma) = C_\gamma \sigma^{-1} \exp\left(-\frac{|x-\mu|^\gamma}{\gamma\sigma^\gamma}\right), \tag{4.14}$$

where $\gamma, \sigma > 0$ and C_γ is some normalization constant depending only on γ. It is easy to see that the standard normal distribution is just the standard Subbotin distribution with $\gamma = 2$. By assuming that $q = O(p^{1-\beta})$, $t = (\gamma r \log p)^{1/\gamma}$, and $\nu = (\gamma s \log p)^{1/\gamma}$ with $\frac{1}{2} < \beta < 1$ and $0 < r < s \leqslant 1$, they derived that the sufficient and necessary conditions for the classifiers g and g^{tr} to produce asymptotically correct classification results are

$$1 - 2\beta > 0 \text{ and} \tag{4.15}$$

$$1 - 2\beta + s > 0, \tag{4.16}$$

respectively. Thus the classification boundary of g^{tr} is lower than that of g, indicating that the distance-based classifier using truncated data are more sensible.

The classification boundaries for distance-based classifiers and for their thresholded versions are both independent of the training sample size. As pointed out by Hall, Pittelkow and Ghosh (2008), this conclusion is obtained from the fact that for fixed sample size n and for distance-based classifiers, the probability of correct classification converges to 1 if and only if the differences between distances among data have a certain extremal property, and that this property holds for one difference if and only if it holds for all of them. Hall, Pittelkow and Ghosh (2008) further compared the classification boundary of distance-based classifiers with that of the classifiers based on higher criticism. See their paper for more comparison results.

5 Feature selection by independence rule

As has been discussed in Section 3, classification methods using all features do not necessarily perform well due to the noise accumulation when estimating a large number of noise features. Thus, feature selection is very important in high dimensional classification. This has been advocated by Fan and Fan (2008) and many other researchers. In fact, the thresholding methods discussed in Hall, Pittelkow and Ghosh (2008) are also a type of feature selection.

5.1 Features annealed independence rule

Fan and Fan (2008) proposed the Features Annealed Independence Rule (FAIR) for feature selection and classification in high dimensional setting. We discuss their method in this subsection.

There is a huge literature on the feature selection in high dimensional setting. See, for example, Efron et al. (2004); Fan and Li (2001); Fan and Lv (2008, 2009); Fan et al. (2008); Lv and Fan (2009); Tibshirani (1996). Two sample t tests are frequently used to select important features in classification problems. Let $\bar{X}_{kj} = \sum_{Y_i=k} X_{ij}/n_k$ and $S^2_{kj} = \sum_{Y_i=k}(X_{kj} - \bar{X}_{kj})^2/(n_k - 1)$ be the sample mean and sample variance of j-th feature in class k, respectively, where $k = 1, 2$ and $j = 1, \ldots, p$. Then the two-sample t statistic for feature j is defined as

$$T_j = \frac{\bar{X}_{1j} - \bar{X}_{2j}}{\sqrt{S^2_{1j}/n_1 + S^2_{2j}/n_2}}, \ j = 1, \ldots, p. \tag{5.1}$$

Fan and Fan (2008) studied the feature selection property of two-sample t statistic. They considered the model (3.14) and assumed that the error ε_{ij} satisfies the Cramér's condition and that the population mean difference vector $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = (\mu_1, \ldots, \mu_p)^T$ is sparse with only the first s entries nonzero. Here, s is allowed to diverge to ∞ with the sample size n. They showed that if $\log(p - s) = o(n^\gamma)$, $\log s = o(n^{\frac{1}{2}-\beta_n})$, and $\min_{1\leqslant j\leqslant s} \frac{|\mu_j|}{\sqrt{\sigma^2_{1j}+\sigma^2_{2j}}} = o(n^{-\gamma}\beta_n)$ with some $\beta_n \to \infty$ and $\gamma \in (0, \frac{1}{3})$, then with x chosen in the order of $O(n^{\gamma/2})$, the following result holds:

$$P(\min_{j\leqslant s} |T_j| > x, \max_{j>s} |T_j| < x) \to 1. \tag{5.2}$$

This result allows the lowest signal level $\min_{1\leqslant j\leqslant s} \frac{|\mu_j|}{\sqrt{\sigma^2_{1j}+\sigma^2_{2j}}}$ to decay with sample size n. As long as the rate of decay is not too fast and the dimensionality p does not grow exponentially faster than n, the two-sample t-test can select all important features with probability tending to 1.

Although the theoretical result (5.2) shows that the t-test can successfully select features if the threshold is appropriately chosen, in practice it is usually very hard to choose a good threshold value. Moreover, even all revelent features are correctly selected by the two-sample t test, it may not necessarily be the best to use all of them, due to the possible existence of many faint features. Therefore, it is necessary to further single out the most important ones. To address this issue, Fan and Fan (2008) proposed the features annealed independence rule. Instead of constructing independence rule using all features, FAIR selects the most important ones and use them to construct independence rule. To appreciate the idea of FAIR, first note that the relative importance of features can be measured by the ranking of $\{|\alpha_j|/\sigma_j\}$. If such oracle ranking information is available, then one can construct the independence rule using m features with the largest $\{|\alpha_j|/\sigma_j\}$. The optimal

value of m is to be determined. In this case, FAIR takes the following form:

$$\delta(\mathbf{x}) = \sum_{j=1}^{p} \alpha_j (x_j - \mu_j)/\sigma_j^2 1_{\{|\alpha_j|/\sigma_j > b\}}, \tag{5.3}$$

where b is a positive constant chosen in a way such that there are m features with $|\alpha_j|/\sigma_j > b$. Thus choosing the optimal m is equivalent to selecting the optimal b. Since in practice such oracle information is unavailable, we need to learn it from the data. Observe that $|\alpha_j|/\sigma_j$ can be estimated by $|\hat{\alpha}_j|/\hat{\sigma}_j$. Thus the sample version of FAIR is

$$\hat{\delta}(\mathbf{x}) = \sum_{j=1}^{p} \hat{\alpha}_j (x_j - \hat{\mu}_j)/\hat{\sigma}_j^2 1_{\{|\hat{\alpha}_j|/\hat{\sigma}_j > b\}}. \tag{5.4}$$

In the case where the two population covariance matrices are the same, we have

$$|\hat{\alpha}_j|/\hat{\sigma}_j = \sqrt{n/(n_1 n_2)}|T_j|.$$

Thus the sample version of the discriminant function of FAIR can be rewritten as

$$\hat{\delta}_{\text{FAIR}}(\mathbf{x}) = \sum_{j=1}^{p} \hat{\alpha}_j (x_j - \hat{\mu}_j)/\hat{\sigma}_j^2 1_{\{\sqrt{n/(n_1 n_2)}|T_j| > b\}}. \tag{5.5}$$

It is clear from (5.5) that FAIR works the same way as that we first sort the features by the absolute values of their t-statistics in the descending order, and then take out the first m features to construct the classifier. The number of features m can be selected by minimizing the upper bound of the classification error given in (3.11). To understand this, note that the upper bound on the right hand side of (3.11) is a function of the number of features. If the features are sorted in the descending order of $|\alpha_j|/\sigma_j$, then this upper bound will first increase and then decrease as we include more and more features. The optimal m in the sense of minimizing the upper bound takes the form

$$m_{opt} = \arg \max_{1 \leqslant m \leqslant p} \frac{1}{\lambda_{\max}^m} \frac{[\sum_{j=1}^m \alpha_j^2/\sigma_j^2 + m(1/n_2 - 1/n_1)]^2}{nm/(n_1 n_2) + \sum_{j=1}^m \alpha_j^2/\sigma_j^2},$$

where $\lambda_{\max}^m$ is the largest eigenvalue of the correlation matrix $\mathbf{R}^m$ of the truncated observations. It can be estimated from the training data as

$$\begin{aligned} \hat{m}_{opt} &= \arg \max_{1 \leqslant m \leqslant p} \frac{1}{\hat{\lambda}_{\max}^m} \frac{[\sum_{j=1}^m \hat{\alpha}_j^2/\hat{\sigma}_j^2 + m(1/n_2 - 1/n_1)]^2}{nm/(n_1 n_2) + \sum_{j=1}^m \hat{\alpha}_j^2/\hat{\sigma}_j^2} \\ &= \arg \max_{1 \leqslant m \leqslant p} \frac{1}{\hat{\lambda}_{\max}^m} \frac{n[\sum_{j=1}^m T_j^2 + m(n_1 - n_2)/n]^2}{mn_1 n_2 + n_1 n_2 \sum_{j=1}^m T_j^2}. \end{aligned} \tag{5.6}$$

Note that the above t-statistics are the sorted ones. Fan and Fan (2008) used simulation study and real data analysis to demonstrate the performance of FAIR. See their paper for the numerical results.

5.2 Nearest shrunken centroids method

In this section, we will discuss the nearest shrunken centroids (NSC) method proposed by Tibshirani et al. (2002). This method is used to identify a subset of features that best characterize each class and do classification. Compared to the centroid rule discussed in Section 3.2, it takes into account the feature selection. Moreover, it is general and can be applied to high-dimensional multi-class classification.

Define $\bar{X}_{kj} = \sum_{i \in \mathcal{G}_k} X_{ij}/n_k$ as the j-th component of the centroid for class k, and $\bar{X}_j = \sum_{i=1}^{n} X_{ij}/n$ as the j-th component of the overall centroid. The basic idea of NSC is to shrink the class centroids to the overall centroid. Tibshirani et al. (2002) first normalized the centroids by the within class standard deviation for each feature, i.e.,

$$d_{kj} = \frac{\bar{X}_{kj} - \bar{X}_j}{m_k(S_j + s_0)}, \tag{5.7}$$

where s_0 is a positive constant, and S_j is the pooled within class standard deviation for j-th feature with

$$S_j^2 = \sum_{k=1}^{K} \sum_{i \in \mathcal{G}_k} (X_{ij} - \bar{X}_{kj})^2/(n-K)$$

and $m_k = \sqrt{1/n_k - 1/n}$ the normalization constant. As pointed out by Tibshirani et al. (2002), d_{kj} defined in (5.7) is a t-statistic for feature j comparing the k-th class to the average. The constant s_0 is included to guard against the possibility of large d_{kj} simply caused by very small value of S_j. Then (5.7) can be rewritten as

$$\bar{X}_{kj} = \bar{X}_j + m_k(S_j + s_0)d_{kj}. \tag{5.8}$$

Tibshirani et al. (2002) proposed to shrink each d_{kj} toward zero by using soft thresholding. More specifically, they define

$$d'_{kj} = \text{sgn}(d_{ik})(|d_{kj}| - \Delta)_+, \tag{5.9}$$

where $\text{sgn}(\cdot)$ is the sign function, and $t_+ = t$ if $t > 0$ and $t_+ = 0$ otherwise. This yields the new shrunken centroids

$$\bar{X}'_{kj} = \bar{X}_j + m_k(S_j + s_0)d'_{kj}. \tag{5.10}$$

As argued in their paper, since many of $\bar{X}_{kj}$ are noisy and close to the overall mean $\bar{X}_j$, using soft thresholding produces more reliable estimates of the true means. If the shrinkage level Δ is large enough, many of the d_{kj} will be shrunken to zero and the corresponding shrunken centroid $\bar{X}'_{kj}$ for feature j will be equal to the overall centroid for feature j. Thus these features do not contribute to the nearest centroid computation.

To choose the amount of shrinkage Δ, Tibshirani et al. (2002) proposed to use the cross validation method. For example, if 10-fold cross validation is used, then the training data set is randomly split into 10 approximately equal-size sub-samples. We first fit the model by using 90% of the training data, and then predict the class labels of the remaining 10% of the training data. This procedure is repeated 10 times for a fixed Δ, with each of the 10 sub-samples of the data used as the test sample to calculate the prediction error. The prediction errors on all 10 parts are then added together as the overall prediction error. The optimal Δ is then chosen to be the one that minimizes the overall prediction error.

After obtaining the shrunken centroids, Tibshirani et al. (2002) proposed to classify a new observation $\mathbf{X}$ to the class whose shrunken centroid is closest to this new observation. They define the discriminant score for class k as

$$\hat{\delta}_k(\mathbf{X}) = \sum_{j=1}^{p} \frac{(X_j - \bar{X}'_{kj})^2}{(S_j + s_0)^2} - 2\log \pi_k. \tag{5.11}$$

The first term is the standardized squared distance of $\mathbf{X}$ to the k-th shrunken centroid, and the second term is a correction based on the prior probability π_k. Then the classification rule is

$$g(\mathbf{X}) = \arg\min_k \hat{\delta}_k(\mathbf{X}). \tag{5.12}$$

It is clear that NSC is a type of distance-based classification method.

Compared to FAIR introduced in Section 5.1, NSC shares the same idea of using marginal information of features to do classification. Both methods conduct feature selection by t-statistic. But FAIR selects the number of features by using mathematical formula that is derived to minimize the upper bound of classification error, while NSC obtains the number of features by using cross validation. Practical implementation shows that FAIR is more stable in terms of the number of selected features and classification error. See Fan and Fan (2008).

6 Loss-based classification

Another popular class of classification methods is based on different (margin-based) loss functions. It includes many well known classification methods such as the support vector machine (SVM, Cristianini and Shawe-Taylor, 2000; Vapnik, 1998).

6.1 Support vector machine

As mentioned in Section 2, the zero-one loss is typically used to access the accuracy of a classification rule. Thus, based on the training data, one may ideally minimize $\sum_{i=1}^{n} I_{g(\mathbf{X}_i)\neq Y_i}$ with respect to $g(\cdot)$ over a function space to obtain an estimated classification rule $\hat{g}(\cdot)$. However the indicator function is neither convex nor smooth. The corresponding optimization is difficult, if not impossible, to

solve. Alternatively, several convex surrogate loss functions have been proposed to replace the zero-one loss.

For binary classification, we may equivalently code the categorical response Y as either -1 or $+1$. The SVM replaces the zero-one loss by the hinge loss $H(u) = [1-u]_+$, where $[u]_+ = \max\{0, u\}$ denotes the positive part of u. Note that the hinge loss is convex. Replacing the zero-one loss with the hinge loss, the SVM minimizes

$$\sum_{i=1}^{n} H(Y_i f(\mathbf{X}_i)) + \lambda J(f) \tag{6.1}$$

with respect to f, where the first term quantifies the data fitting, $J(f)$ is some roughness (complexity) penalty of f, and λ is a tuning parameter balancing the data fit measured by the hinge loss and the roughness of $f(\cdot)$ measured by $J(f)$. Denote the minimizer by $\hat{f}(\cdot)$. Then the SVM classification rule is given by $\hat{g}(\mathbf{x}) = \text{sign}(\hat{f}(\mathbf{x}))$. Note that the hinge loss is non-increasing. While minimizing the hinge loss, the SVM encourages positive $Yf(X)$ which corresponds to correct classification.

For linear SVM with $f(\mathbf{x}) = b + \mathbf{x}^T\boldsymbol{\beta}$, the standard SVM uses the 2-norm penalty $J(f) = \frac{1}{2}\sum_{j=1}^{p}\beta_j^2$. While in this exposition, we are formulating the SVM in the regularization framework. However it is worthwhile to point out that the SVM was originally introduced by V. Vapnik and his colleagues with the idea of searching for the optimal separating hyperplane. Interested readers may consult Boser, Guyon and Vapnik (1992) and Vapnik (1998) for more details. It was shown by Wahba (1998) that the SVM can be equivalently fit into the regularization framework by solving (6.1) as presented in the previous paragraph. Different from those methods that focus on conditional probabilities $P(Y|\mathbf{X} = \mathbf{x})$, the SVM targets at estimating the decision boundary $\{\mathbf{x} : P(Y = 1|\mathbf{X} = \mathbf{x}) = 1/2\}$ directly.

A general loss function $\ell(\cdot)$ is called Fisher consistent if the minimizer of $E[\ell(Yf(\mathbf{X})|\mathbf{X} = \mathbf{x})]$ has the same sign as $P(Y = +1|\mathbf{X} = \mathbf{x}) - 1/2$ (Lin, 2004). Fisher consistency is also known as classification-calibration (Bartlett, Jordan, and McAuliffe, 2006) and infinite-sample consistency (Zhang, 2004). It is a desirable property for a loss function.

Lin (2002) showed that the minimizer of $E[H(Yf(\mathbf{X})|\mathbf{X} = \mathbf{x})]$ is exactly $\text{sign}(P(Y = +1|\mathbf{X} = \mathbf{x}) - 1/2)$, the decision-theoretically optimal classification rule with the smallest risk, which is also known as the Bayes classification rule. Thus the hinge loss is Fisher consistent for binary classification.

When dealing with problems with many predictor variables, Zhu, Rosset, Hastie, and Tibshirani (2003) proposed the 1-norm SVM by using the L_1 penalty $J(f) = \sum_{j=1}^{p}|\beta_j|$ to achieve variable selection; Zhang, Ahn, Lin and Park (2006) proposed the SCAD SVM by using the SCAD penalty (Fan and Li, 2001); Liu and Wu (2007) proposed to regularize the SVM with a combination of the L_0 and L_1 penalties; and many others.

Either basis expansion or kernel mapping (Cristianini and Shawe-Taylor, 2000) may be used to accomplish nonlinear SVM. For the case of kernel learning, a bivariate kernel function $K(\cdot,\cdot)$, which maps from $\mathcal{X}\times\mathcal{X}$ to R, is employed. Then $f(\mathbf{x}) = b + \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{X}_i)$ by the theory of reproducing kernel Hilbert spaces

(RKSH), see Wahba (1990). In this case, the 2-norm of $f(\mathbf{x}) - b$ in RKHS with $K(\cdot, \cdot)$ is typically used as $J(f)$. Using the representor theorem (Kimeldorf and Wahba, 1971), $J(f)$ can be represented as $J(f) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i K(\mathbf{X}_i, \mathbf{X}_j)c_j$.

6.2 ψ-learning

The hinge loss $H(u)$ is unbounded and shoots to infinity when t goes to negative infinity. This characteristic makes the SVM tend to be sensitive to noisy training data. When there exist points far away from their own classes (namely, "outliers" in the training data), the SVM classifier tends to be strongly affected by such points due to the unboundedness of the hinge loss. In order to improve over the SVM, Shen, Tseng, Zhang, and Wong (2003) proposed to replace the convex hinge loss by a nonconvex ψ-loss function. The ψ-loss function $\psi(u)$ satisfies

$$\begin{aligned} &U \geqslant \psi(u) > 0 \text{ if } u \in [0, T]; \\ &\psi(u) = 1 - \text{sign}(u) \text{ otherwise,} \end{aligned}$$

where $0 < U \leqslant 2$ and $T > 0$ are constants. The positive values of $\psi(u)$ for $u \in [0, T]$ eliminate the scaling issue of the sign function and avoid too many points piling around the decision boundary. Their method was named the ψ-learning. They showed that the ψ-learning can achieve more accurate class prediction.

Similarly motivated, Wu and Liu (2007a) proposed to truncate the hinge loss function by defining $H_s(u) = \min(H(s), H(u))$ for $s \leqslant 0$ and worked on the more general multi-category classification. According to their Proposition 1, the truncated hinge loss is also Fisher consistent for binary classification.

6.3 AdaBoost

Boosting is another very successful algorithm for solving binary classification. The basic idea of boosting is to combine weaker learners to improve performance (Freund, 1995; Schapire, 1990). The AdaBoost algorithm, a special boosting algorithm, was first introduced by Freund and Schapire (1996). It constructs a "strong" classifier as a linear combination

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

of "simple", "weak" classifiers $h_t(\mathbf{x})$. The "weak" classifiers $h_t(\mathbf{x})$'s can be thought of as features and $H(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$ is called "strong" or final classifier. It works by sequentially reweighing the training data, applying a classification algorithm (weaker learner) to the reweighed training data, and then taking a weighted majority vote of the thus-obtained classifier sequence. This simple reweighing strategy improves performance of many weaker learners. Freund and Schapire (1996) and Breiman (1997) tried to provide a theoretic understanding based on game theory. Another attempt to investigate its behavior was made by Breiman (1998) using bias and variance tradeoff. Later Friedman, Hastie, and Tibshirani (2000) provided

a new statistical perspective, namely using additively modeling and maximum likelihood, to understand why this seemingly mysterious AdaBoost algorithm works so well. They showed that AdaBoost is equivalent to using the exponential loss $\ell(u) = e^{-u}$.

6.4 Other loss functions

There are many other loss functions in this regularization framework. Examples include the squared loss $\ell(u) = (1-u)^2$ used in the proximal SVM (Fung and Mangasarian, 2001) and the least square SVM (Suykens and Vandewalle, 1999), the logistic loss $\ell(u) = \log(1+e^{-u})$ of the logistic regression, and the modified least squared loss $\ell(u) = ([1-u]_+)^2$ proposed by Zhang and Oles (2001). In particular, the logistic loss is motivated by assuming that the probability of $Y = +1$ given $\mathbf{X} = \mathbf{x}$ is given by $e^{f(\mathbf{x})}/(1+e^{f(\mathbf{x})})$. Consequently the logistic regression is capable of estimating the conditional probability.

7 Feature selection in loss-based classification

As mentioned above, variable selection-capable penalty functions such as the L_1 and SCAD can be applied to the regularization framework to achieve variable selection when dealing with data with many predictor variables. Examples include the L_1 SVM (Zhu, Rosset, Hastie, and Tibshirani, 2003), SCAD SVM (Zhang, Ahn, Lin and Park, 2006), SCAD logistic regression (Fan and Peng, 2004). These methods work fine for the case with a fair number of predictor variables. However the remarkable recent development of computing power and other technology has allowed scientists to collect data of unprecedented size and complexity. Examples include data from microarrays, proteomics, functional MRI, SNPs and others. When dealing with such high or ultra-high dimensional data, the usefulness of these methods becomes limited.

In order to handle linear regression with ultra-high dimensional data, Fan and Lv (2008) proposed the sure independence screening (SIS) to reduce the dimensionality from ultra-high p to a fairly high d. It works by ranking predictor variables according to the absolute value of the marginal correlation between the response variable and each individual predictor variable and selecting the top ranked d predictor variables. This screening step is followed by applying a refined method such as the SCAD to these d predictor variables that have been selected. In a fairly general asymptotic framework, this simple but effective correlation learning is shown to have the sure screening property even for the case of exponentially growing dimensionality, that is, the screening retains the true important predictor variables with probability tending to one exponentially fast.

The SIS methodology may break down if a predictor variable is marginally unrelated, but jointly related with the response, or if a predictor variable is jointly uncorrelated with the response but has higher marginal correlation with the response than some important predictors. In the former case, the important feature has already been screened out at the first stage, whereas in the latter case, the unimportant feature is ranked too high by the independent screening technique.

Iterative SIS (ISIS) was proposed to overcome these difficulties by using more fully the joint covariate information while retaining computational expedience and stability as in SIS. Basically, ISIS works by iteratively applying SIS to recruit a small number of predictors, computing residuals based on the model fitted using these recruited variables, and then using the working residuals as the response variable to continue recruiting new predictors. Numerical examples in Fan and Lv (2008) have demonstrated the improvement of ISIS. The crucial step is to compute the working residuals, which is easy for the least-squares regression problem but not obvious for other problems. By sidestepping the computation of working residuals, Fan et al. (2008) has extended (I)SIS to a general pseudo-likelihood framework, which includes generalized linear models as a special case. Roughly they use the additional contribution of each predictor variable given the variables that have been recruited to rank and recruit new predictors.

In this section, we will elaborate (I)SIS in the context of binary classification using loss functions presented in the previous section. While presenting the (I)SIS methodology, we use a general loss function $\ell(\cdot)$. The *R*-code is publicly available at cran.r-project.org.

7.1 Feature ranking by marginal utilities

By assuming a linear model $f(\mathbf{x}) = b + \mathbf{x}^T\boldsymbol{\beta}$, the corresponding model fitting amounts to minimizing

$$Q(b, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i(b + \mathbf{X}_i^T\boldsymbol{\beta})) + \lambda J(f),$$

where $J(f)$ can be the 2-norm or some other penalties that are capable of variable selection. The marginal utility of the j-th feature is

$$\ell_j = \min_{b,\beta_j,\beta_{M_1}} \sum_{i=1}^{n} \ell(Y_i(b + X_{ij}^T\beta_j)).$$

For some loss functions such as the hinge loss, another term $\frac{1}{2}\beta_j^2 + \frac{1}{2}\sum_{m\in M_1}\beta_{m^2}$ may be required to avoid possible identifiability issue. In that case

$$\ell_j = \min_{b,\beta_j}\left\{\sum_{i=1}^{n} \ell(Y_i(b + X_{ij}^T\beta_j)) + \frac{1}{2}\beta_j^2\right\}. \tag{7.1}$$

The idea of SIS is to compute the vector of marginal utilities $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_p)^T$ and rank predictor variables according to their corresponding marginal utilities. The smaller the marginal utility is the more important the corresponding predictor variable is. We select d variables corresponding to the d smallest components of $\boldsymbol{\ell}$. Namely, variable j is selected if ℓ_j is one of the d smallest components of $\boldsymbol{\ell}$. A typical choice of d is $\lfloor n/\log n\rfloor$. Fan and Song (2009) provided an extensive account on the sure screening property of the independence learning and on the capacity of the model size reduction.

7.2 Penalization

With the d variables crudely selected by SIS, parameter estimation and variable selection can be further carried out simultaneously using a more refined penalization method. This step takes joint information into consideration. By reordering the variables if necessary, we may assume without loss of generality that X_1, X_2, ..., X_d are the variables that have been recruited by SIS. In the regularization framework, we use a penalty that is capable of variable selection and minimize

$$\frac{1}{n}\sum_{i=1}^{n}\ell(Y_i(b+\sum_{j=1}^{d}X_{ij}\beta_j))+\sum_{j=1}^{d}p_\lambda(|\beta_j|), \tag{7.2}$$

where $p_\lambda(\cdot)$ denotes a general penalty function and $\lambda > 0$ is a regularization parameter. For example, $p_\lambda(\cdot)$ can be chosen to be the L_1 (Tibshirani, 1996), SCAD (Fan and Li, 2001), adaptive L_1 (Zhang and Lu, 2007; Zou, 2006), or some other penalty.

7.3 Iterative feature selection

As mentioned before, the SIS methodology may break down if a predictor is marginally unrelated, but jointly related with the response, or if a predictor is jointly uncorrelated with the response but has higher marginal correlation with the response than some important predictors. To handle such difficult scenario, iterative SIS may be required. ISIS seeks to overcome these difficulties by using more fully the joint covariate information.

The first step is to apply SIS to select a set $\mathcal{A}_1$ of indices of size d, and then employ (7.2) with the L_1 or SCAD penalty to select a subset $\mathcal{M}_1$ of these indices. This is our initial estimate of the set of indices of important variables.

Next, we compute the conditional marginal utility

$$\ell_j^{(2)}=\min_{b,\beta_j}\sum_{i=1}^{n}\ell(Y_i(b+\mathbf{X}_{i,\mathcal{M}_1}^T\boldsymbol{\beta}_{\mathcal{M}_1}+X_{ij}^T\beta_j)) \tag{7.3}$$

for any $j\in\mathcal{M}_1^c=\{1,2,\ldots,p\}\backslash\mathcal{M}_1$, where $\mathbf{X}_{i,\mathcal{M}_1}$ is the sub-vector of $\mathbf{X}_i$ consisting of those elements in $\mathcal{M}_1$. If necessary, the term of $\frac{1}{2}\beta_j^2$ may be added in (7.3) to avoid identifiability issue just as the case of defining the marginal utilities in (7.1). The conditional marginal utility $\ell_j^{(2)}$ measures the additional contribution of variable X_j given that the variables in $\mathcal{M}_1$ have been included. We then rank variables in $\mathcal{M}_1^c$ according to their corresponding conditional marginal utilities and form the set $\mathcal{A}_2$ consisting of the indices corresponding to the smallest $d-|\mathcal{M}_1|$ elements.

The above prescreening step using the conditional utility is followed by solving

$$\min_{b,\ \boldsymbol{\beta}_{\mathcal{M}_1},\ \boldsymbol{\beta}_{\mathcal{A}_2}}\frac{1}{n}\sum_{i=1}^{n}\ell(Y_i(b+\mathbf{X}_{i,\mathcal{M}_1}^T\boldsymbol{\beta}_{\mathcal{M}_1}+\mathbf{X}_{i,\mathcal{A}_2}^T\boldsymbol{\beta}_{\mathcal{M}_2})+\sum_{j\in\mathcal{M}_1\cup\mathcal{A}_2}p_\lambda(|\beta_j|). \tag{7.4}$$

The penalty $p_\lambda(\cdot)$ leads to a sparse solution. The indices in $\mathcal{M}_1 \cup \mathcal{A}_2$ that have non-zero β_j yield a new estimate $\mathcal{M}_2$ of the active indices.

This process of iteratively recruiting and deleting variables may be repeated until we obtain a set of indices $\mathcal{M}_k$ which either reaches the prescribed size d or satisfies convergence criterion $\mathcal{M}_k = \mathcal{M}_{k-1}$.

7.4 Reducing false discovery rate

Sure independence screening is a simple but effective method to screen out irrelevant variables. They are usually conservative and include many unimportant variables. Next we present two possible variants of (I)SIS that have some attractive theoretical properties in terms of reducing the false discovery rate (FDR).

Denote $\mathcal{A}$ to be the set of active indices, namely the set containing those indices j for which $\beta_j \neq 0$ in the true model. Denote $\mathbf{X}_\mathcal{A} = \{X_j, j \in \mathcal{A}\}$ and $\mathbf{X}_{\mathcal{A}^c} = \{X_j, j \in \mathcal{A}^c\}$ to be the corresponding sets of active and inactive variables respectively.

Assume for simplicity that n is even. We randomly split the sample into two halves. Apply SIS separately to each half with $d = \lfloor n/\log n \rfloor$ or larger, yielding two estimates $\hat{\mathcal{A}}^{(1)}$ and $\hat{\mathcal{A}}^{(2)}$ of the set of active indices $\mathcal{A}$. Both $\hat{\mathcal{A}}^{(1)}$ and $\hat{\mathcal{A}}^{(2)}$ may have large FDRs because they are constructed by SIS, a crude screening method. Assume that both $\hat{\mathcal{A}}^{(1)}$ and $\hat{\mathcal{A}}^{(2)}$ have the sure screening property, $P(\mathcal{A} \subset \hat{\mathcal{A}}^{(j)}) \to 1$, for $j = 1$ and 2. Then

$$P(\mathcal{A} \subset \hat{\mathcal{A}}^{(1)} \cap \hat{\mathcal{A}}^{(2)}) \to 1.$$

Thus motivated, we define our first variant of SIS by estimating $\mathcal{A}$ with $\hat{\mathcal{A}} = \hat{\mathcal{A}}^{(1)} \cap \hat{\mathcal{A}}^{(2)}$.

To provide some theoretical support, we make the following assumption:
Exchangeability Condition: Let $r \in \mathbb{N}$, the set of natural numbers. The model satisfies the exchangeability condition at level r if the set of random vectors

$$\{(Y, \mathbf{X}_\mathcal{A}, X_{j_1}, \ldots, X_{j_r}) : j_1, \ldots, j_r \text{ are distinct elements of } \mathcal{A}^c\}$$

is exchangeable.

The Exchangeability Condition ensures that each inactive variable has the same chance to be recruited by SIS. Then we have the following nonasymptotic probabilistic bound.

Let $r \in \mathbb{N}$, and assume that the model satisfies the Exchangeability Condition at level r. For $\hat{\mathcal{A}} = \hat{\mathcal{A}}^{(1)} \cap \hat{\mathcal{A}}^{(2)}$ defined above, we have

$$P(|\hat{\mathcal{A}} \cup \mathcal{A}^c| \geqslant r) \leqslant \frac{\binom{d}{r}^2}{\binom{p-|\mathcal{A}|}{r}} \leqslant \frac{1}{r!}\left(\frac{d^2}{p-|\mathcal{A}|}\right)^r,$$

where there second inequality requires $d^2 \leqslant p - |\mathcal{A}|$.

When $r = 1$, the above probabilistic bound implies that, when the number of selected variables $d \leqslant n$, we have with high probability $\hat{\mathcal{A}}$ reports no 'false

positives' if the exchangeability condition is satisfied at level 1 and if p is large by comparison with n^2. It means that it is very likely that any index in the estimated active set also belongs to the active set in the true model, which, together with sure screening assumption, implies the model selection consistency. The nature of this result is somehow unusual in that it suggests that a 'blessing of dimensionality' the probability bound one false positives decreases with p. However, this is only part of the full store, because the probability of missing elements of the true active set is expected to increase with p.

The iterative version of the first variant of SIS can be defined analogously. We apply SIS to each partition separately to get two estimates of the active index set $\hat{\mathcal{A}}_1^{(1)}$ and $\hat{\mathcal{A}}_1^{(2)}$, each having d elements. After forming the intersection $\hat{\mathcal{A}}_1 = \hat{\mathcal{A}}_1^{(1)} \cap \hat{\mathcal{A}}_1^{(2)}$, we carry out penalized estimation with all data to obtain a first approximation $\hat{\mathcal{M}}_1$ to the true active index set. We then perform a second stage of the ISIS procedure to each partition separately to obtain sets of indices $\hat{\mathcal{M}}_1 \cup \hat{\mathcal{A}}_2^{(1)}$ and $\hat{\mathcal{M}}_1 \cup \hat{\mathcal{A}}_2^{(2)}$. Take their intersection and re-estimate parameters using penalized estimation to get a second approximation $\hat{\mathcal{M}}_2$ to the true active set. This process can be continued until convergence criterion is met as in the definition of ISIS.

8 Multi-category classification

Sections 6 and 7 focus on binary classifications. In this section, we will discuss how to handle classification problems with more than two classes.

When dealing with classification problems with a multi-category response, one typically label the response as $Y \in \{1, 2, \ldots, K\}$, where K is the number of classes. Define conditional probabilities $p_j(\mathbf{x}) = P(Y = j | \mathbf{X} = \mathbf{x})$ for $j = 1, 2, \ldots, K$. The corresponding Bayes rule classifies a test sample with predictor vector $\mathbf{x}$ to the class with the largest $p_j(\mathbf{x})$. Namely the Bayes rule is given by $\underset{j}{\operatorname{argmax}}\, p_j(\mathbf{x})$.

Existing methods for handling multi-category problems can be generally divided into two groups. One is to solve the multi-category classification by solving a series of binary classifications while the other considers all the classes simultaneously. Among the first group, both methods of constructing either pairwise classifiers (Krefsel, 1998; Schmidt and Gish, 1996) or one-versus-all classifiers (Hsu and Lin, 2002; Rifkin and Klautau, 2004) are popularly used. In the one-versus-all approach, one is required to train K distinct binary classifiers to separate one class from all others and each binary classifier uses all training samples. For the pairwise approach, there are $K(K-1)/2$ binary classifier to be trained with one for each pair of classes. Comparing to the one-versus-all approach, the number of classifiers is much larger for the pairwise approach but each one involves only a subsample of the training data and thus is easier to train. Next we will focus on the second group of methods.

Weston and Watkins (1999) proposed the k-class support vector machine. It

solves

$$\min \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq Y_i}(2-[f_{Y_i}(\mathbf{X}_i)-f_j(\mathbf{X}_i)])_+ + \lambda\sum_{j=1}^{K}\parallel f_j \parallel . \tag{8.1}$$

The linear classifier takes the form $f_j(\mathbf{x}) = b_j + \boldsymbol{\beta}_j^T\mathbf{x}$, whereas the penalty in (8.1) can be taken as the L_2-norm $\|f_j\| = w_j\|\boldsymbol{\beta}_j\|^2$ for some weight w_j. Let $\hat{f}_j(\mathbf{x})$ be the solution to (8.1). Then the classifier assigns a new observation $\mathbf{x}$ to class $\hat{k} = \text{argmax}_j\, \hat{f}_j(\mathbf{x})$. Zhang (2004) generalized this loss to $\sum_{k\neq Y}\phi(f_Y(\mathbf{X})-f_k(\mathbf{X}))$ and called it pairwise comparison method. Here $\phi(\cdot)$ can be any decreasing function so that a large value $f_Y(\mathbf{X}) - f_k(\mathbf{X})$ for $k \neq Y$ is favored while optimizing. In particular Weston and Watkins (1999) essentially used the hinge loss up to a scale of factor 2. By assuming the differentiability of $\phi(\cdot)$, Zhang (2004) showed that the desirable property of order preserving. See Theorem 5 of Zhang (2004). However the differentiability condition on $\phi(\cdot)$ rules out the important case of hinge loss function.

Lee, Lin, and Wahba (2004) proposed a nonparametric multi-category SVM by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq Y_i}\left(f_j(\mathbf{X}_i)+\frac{1}{k-1}\right)_+ + \lambda\sum_{j=1}^{K}\parallel f_j \parallel \tag{8.2}$$

subject to the sum-to-zero constraint in the reproducing kernel Hilbert space. Their loss function works with the sum-to-zero constraint to encourage $f_Y(\mathbf{X}) = 1$ and $f_k(\mathbf{X}) = -1/(k-1)$ for $k \neq Y$. For their loss function, they obtained Fisher consistency by proving that the minimizer of $E\sum_{j\neq Y}(f_j(\mathbf{X}) - 1/(k-1))_+$ under the sum-to-zero constraint at $\mathbf{X} = \mathbf{x}$ is given by $f_j(\mathbf{x}) = 1$ if $j = \text{argmax}_m\, p_m(\mathbf{x})$ and $-1/(k-1)$ otherwise. This formulation motivated the constrained comparison method in Zhang (2004). The constrained comparison method use the loss function $\sum_{k\neq Y}\phi(-f_k(\mathbf{X}))$. Zhang (2004) showed that this loss function in combination with the sum-to-zero constraint has the order preserving property as well (Theorem 7, Zhang 2004).

Liu and Shen (2006) proposed one formulation to extend the ψ-learning from binary to multicategory. Their loss performs multiple comparisons of class Y versus other classes in a more natural way by solving

$$\min \frac{1}{n}\sum_{i=1}^{n}\psi(\min_{j\neq Y_i}(f_{Y_i}(\mathbf{X}_i)-f_j(\mathbf{X}_i))) + \lambda\sum_{j=1}^{K}\parallel f_j \parallel \tag{8.3}$$

subject to the sum-to-zero constraint. Note that the ψ loss function is non-increasing. The minimization in (8.3) encourages $f_{Y_i}(\mathbf{X}_i)$ to be larger than $f_j(\mathbf{X}_i)$ for all $j \neq Y_i$ thus leading to correct classification. They provided some statistical learning theory for the multicategory ψ-learning methodology and obtained fast convergence rates for both linear and nonlinear learning examples.

Similarly motivated as Liu and Shen (2006), Wu and Liu (2007a) proposed the robust truncated hinge loss support vector machines. They define the truncated

hinge loss function to be $H_s(u) = \min\{H(u), H(s)\}$ for some $s \leqslant 0$. The robust truncated hinge loss support vector machine solves

$$\min \frac{1}{n} \sum_{i=1}^{n} H_s(\min_{j \neq Y_i}(f_{Y_i}(\mathbf{X}_i) - f_j(\mathbf{X}_i))) + \lambda \sum_{j=1}^{K} \parallel f_j \parallel . \tag{8.4}$$

Wu and Liu (2007a) used the idea of support vectors to show that the robust truncated hinge loss support vector machine is less sensitive to outliers than the SVM. Note that $H_s(u) = H(u) - [s-u]_+$. This decomposition makes it possible to use the difference convex algorithm (An and Tao, 1997) to solve (8.4). In this way, they showed that the robust truncated hinge loss support vector machine removes some support vectors form the SVM and consequently its corresponding support vectors are a subset of the support vectors of the SVM. Fisher consistency is also established for the robust truncated hinge loss support vector machine when $s \in [-1/(K-1), 0]$. Recall that K is the number of classes. This tells us that more truncation is needed to guarantee consistency for larger K.

The truncation idea is in fact very general. It can be applied to other loss functions such as the logistic loss in logistic regression and the exponential loss in AdaBoost. Corresponding Fisher consistency is also available. Wu and Liu (2007a) only used the hinge loss to demonstrate how the truncation works. In another work, Wu and Liu (2007b) studied the truncated hinge loss function using the formulation of Lee, Lin, and Wahba (2004).

Other formulations of multicategory classification includes those of Vapnik (1998), Bredensteiner and Bennett (1999), Crammer and Singer (2001) among many others. Due to limited space, we cannot list all of them here. Interested readers may read those papers and references therein for more formulations.

In the aforementioned different formulations of multicategory classification with linear assumption that $f_k(\mathbf{x}) = b_k + \boldsymbol{\beta}_k^T \mathbf{x}$ for $k = 1, 2, \ldots, K$, variable selection-capable penalty function can be used in place of $\parallel f_k \parallel$ to achieve variable selection. For example Wang and Shen (2007) studied the L_1 norm multiclass support vector machine by using penalty $\sum_{k=1}^{K} \sum_{j=1}^{p} |\beta_{jk}|$. Note that the L_1 norm treats all the coefficients equally. It ignores the fact that the group of $\beta_{j1}, \beta_{j2}, \ldots, \beta_{jK}$ corresponds to the same predictor variable X_j. As a result the L_1 norm SVM is not efficient in achieving variable selection. By including this group information into consideration, Zhang, Liu, Wu, and Zhu (2008) proposed the adaptive super norm penalty for multi-category SVM. They use the penalty $\sum_{j=1}^{p} w_j \max_{k=1,2,\ldots,K} |\beta_{jk}|$, where the adaptive weight w_j is based on a consistent estimate in the same way as the adaptive L_1 penalty (Zhang and Lu, 2007; Zou, 2006) does. Note that the super norm penalty encourages the entire group $\beta_{j1}, \beta_{j2}, \ldots, \beta_{jK}$ to be exactly zero for any noise variable X_j and thus achieves more efficient variable selection.

Variable selection-capable penalty works effectively when the dimensionality is fairly high. However when it comes to ultrahigh dimensionality, things may get complicated. For example, the computational complexity grows with the dimensionality. In this case, the (I)SIS method may be extended to aforementioned

multi-category classifications as they are all given in loss function based formulations. Fan et al. (2008) considered (I)SIS for the formulation by Lee, Lin, and Wahba (2004). They used a couple of microarray datasets to demonstrated its practical utilities.

References

[1] An L. T. H. and Tao P. D. (1997). Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *Journal of Global Optimization* **11**, 253-285.

[2] Bair E., Hastie T., Paul D. and Tibshirani R. (2006). Prediction by supervised principal components. *J. Amer. Statist. Assoc.* **101**, 119-137.

[3] Bartlett P., Jordan M., and McAuliffe J. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association* **101**, 138-156.

[4] Bickel P. J. and Levina E. (2004). Some theory for Fisher's linear discriminant function, "naive Bayes", and some alternatives when there are many more variables than observations. *Bernoulli* **10**, 989-1010.

[5] Boser B., Guyon I. and Vapnik V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Conference on Computational Learning Theory*, 144-152. ACM Press, Pittsburgh, PA.

[6] Boulesteix A. L. (2004). PLS dimension reduction for classification with microarray data. *Stat. Appl. Genet. Mol. Biol.* **3**, 1-33.

[7] Breiman L. (1997). Prediction games and arcing algorithms. *Technical Report* **504**, Dept. Statistics, Univ. California, Berkeley.

[8] Breiman L. (1998). Arcing classifiers (with discussion). *Ann. Statist.* **26**, 801-849.

[9] Bura E. and Pfeiffer R. M. (2003). Graphical methods for class prediction using dimension reduction techniques on DNA microarray data. *Bioinformatics* **19**, 1252-1258.

[10] Cristianini N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge.

[11] Donoho D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *Aide-Memoire of a Lecture at AMS Conference on Math Challenges of the 21st Century.*

[12] Donoho D. L. and Jin J. (2004). Feature Selection by Higher Criticism Thresholding: Optimal Phase Diagram. *Manuscript.*

[13] Dudoit S., Fridlyand J. and Speed T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Amer. Statist. Assoc.* **97**, 77-87.

[14] Efron B., Hastie T., Johnstone I. and Tibshirani R. (2004). Least angle regression (with discussion). *Ann. Statist.* **32**, 407-499.

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

[16] Fan J. and Li R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**, 1348-1360.

[17] Fan J. and Li R. (2006). Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *Proceedings of the International Congress of Mathematicians* (M. Sanz-Sole, J. Soria, J.L. Varona, J. Verdera, eds.), Vol. III, 595-622. European Mathematical Society, Zurich.

[18] Fan J. and Lv J. (2008). Sure independence screening for ultrahigh dimensional feature space (with discussion). *J. Roy. Statist. Soc. Ser. B* **70**, 849-911.

[19] Fan J. and Lv J. (2009). Properties of Non-concave Penalized Likelihood with NP-dimensionality. *Manuscript.*

[20] Fan J. and Peng H. (2004). Nonconcave penalized likelihood with diverging number of parameters. *Ann. Statist.* **32**, 928-961.

[21] Fan J., Samworth R. and Wu Y. (2008). Ultrahigh dimensional variable selection: Beyond the linear model. *Journal of Machine Learning Research.* To appear.

[22] Fan J. and Song R. (2009). Sure Independence Screening in Generalized Linear Models with NP-dimensionality. *Manuscript.*

[23] Freund Y. (1995). Boosting a weak learning algorithm by majority. *Inform. and Comput.* **121**, 256-285.

[24] Freund Y. and Schapire R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 148-156. Morgan Kaufman, San Francisco.

[25] Friedman J., Hastie T. and Tibshirani R. (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* **28**, 337-407.

[26] Fung G. and Mangasarian O. L. (2001). Proximal support vector machine classifiers. In *Proceedings KDD-2001: Knowledge Discovery and Data Mining* (Provost F. and Srikant F., eds), 77-86. Asscociation for Computing Machinery.

[27] Ghosh D. (2002). Singular value decomposition regression modeling for classification of tumors from microarray experiments. *Proceedings of the Pacific Symposium on Biocomputing.* 11462-11467.

[28] Greenshtein E. (2006). Best subset selection, persistence in high-dimensional statistical learning and optimization under l1 constraint. *Ann. Statist.* **34**, 2367-2386.

[29] Greenshtein E. and Ritov Y. (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli* **10**, 971-988.

[30] Hall P. and Jin J. (2008). Properties of higher criticism under strong dependence. *Ann. Statist.* **1**, 381-402.

[31] Hall, P., Marron, J. S. and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *J. R. Statist. Soc. B* **67**, 427-444.

[32] Hall P., Park B. and Samworth R. (2008). Choice of neighbor order in nearest-neighbor classification. *Ann. Statist.* **5**, 2135-2152.

[33] Hall P., Pittelkow Y. and Ghosh M. (2008). Theoretical measures of relative performance of classifiers for high dimensional data with small sample sizes. *J. R. Statist. Soc. B* **70**, 159-173.

[34] Hastie T., Tibshirani R. and Friedman J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*. Springer-Verlag, New York.

[35] Hsu C. and Lin C. (2002). A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Netw.* **13**, 415-425.

[36] Huang X. and Pan W. (2003). Linear regression and two-class classification with gene expression data. *Bioinformatics* **19**, 2072-2978.

[37] Ingster Yu. I. (2002). Adaptive detection of a signal of growing dimension: II. *Math. Meth. Statist.* **11**, 37-68.

[38] Jin J. (2006). Higher criticism statistic: theory and applications in non-Gaussian detection. In *Proc. PHYSTAT 2005: Statistical Problems in Particle Physics, Astrophysics and Cosmology* (L. Lyons and M. K. ünel, eds). World Scientific Publishing, Singapore.

[39] Kimeldorf G. and Wahba G. (1971). Some results on Tchebycheffian spline functions, *J. Math. Anal. Applic.* **33**, 82-95.

[40] Krefsel U. (1998). Pairwise classification and support vector machines. In *Advances in Kernel Methods - Support Vector Learning* (B. Schölkopf, C. Burges, and A. Smola, eds.) MIT Press, Cambridge, MA.

[41] Lee Y., Lin Y. and Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* **99**, 67-81.

[42] Li K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association* **414**, 316-327.

[43] Lin Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6**, 259-275

[44] Lin Y. (2004). A note on margin-based loss functions in classification. *Statistics and Probability Letters* **68**, 73-82

[45] Liu Y. and Shen X. (2006). Multicategory ψ-learning. *Journal of the American Statistical Association* **101**, 500-509.

[46] Liu Y. and Wu Y. (2007). Variable selection via a combination of the L0 and L1 penalties. *Journal of Computational and Graphical Statistics* **16** (4), 782-798.

[47] Lv J. and Fan Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *Ann. Statist.* **37**, 3498-3528.

[48] Mahalanobis P. C. (1930). On tests and measures of group divergence. *Journal of the Asiatic Society of Bengal* **26**, 541-588.

[49] Nguyen D. V. and Rocke D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18**, 39-50.

[50] Rifkin R. and Klautau A. (2004). In defence of one-versus-all classificaiton. *Journal of Machine Learning Research* **5**, 101-141.

[51] Schapire R. E. (1990). The strength of weak learnability. *Machine Learning* **5**, 197-227.

[52] Schmidt M. S. and Gish H. (1996). Speaker identification via support vector classifiers. In *Proceedings of the 21st IEEE International Conference Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)* 105-108, Atlanta, GA.

[53] Shen X., Tseng G. C., Zhang X. and Wong W. H. (2003). On ψ-Learning. *Journal of the American Statistical Association* **98**, 724-734.

[54] Suykens J. A. K. and Vandewalle J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters* **9**, 293-300.

[55] Tibshirani R. (1996). Regression shrinkage and selection via the LASSO. *J. Roy. Statist. Soc. Ser. B* **58**, 267–288.

[56] Tibshirani R., Hastie T., Narasimhan B. and Chu G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad. Sci.* **99**, 6567-6572.

[57] Vapnik V. (1998). *Statistical Learning Theory*. Wiley, New York.

[58] Wahba G. (1990). Spline models for observational data. *CBMS-NSF Regional Conference Series in Applied Mathematics* **59**. SIAM, Philadelphia.

[59] Wahba G. (1998). Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV. In *Advances in Kernel Methods: Support Vector Learning* (Schökopf, Burges, and Smola, eds.), 125-143. MIT Press, Cambridge, MA.

[60] Wang L. and Shen X. (2007). On L1-norm multi-class support vector machines: methodology and theory. *Journal of the American Statistical Association* **102**, 595-602.

[61] Weston J., and Watkins C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, 219-224.

[62] Wu Y. and Liu Y. (2007a). Robust truncated-hinge-loss support vector machines. *Journal of the American Statistical Association* **102** (479) 974-983.

[63] Wu Y. and Liu Y. (2007b). On multicategory truncated-hinge-loss support vector machines. *Contemporary Mathematics* **443**, 49-58.

[64] Zhang T. (2004), Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research* **5**, 1225-1251.

[65] Zhang H. H., Ahn J., Lin X. and Park C. (2006). Gene selection using support vector machines with nonconvex penalty. *Bioinformatics* **22**, 88-95.

[66] Zhang H. H., Liu Y., Wu. and Zhu J. (2008). Variable selection for the multicategory SVM via sup-norm regularization. *Electronic Journal of Statistics* **2**, 149-167.

[67] Zhang H. H. and Lu W. (2007). Adaptive-LASSO for Cox's proportional hazard model. *Biometrika* **94**, 691-703.

[68] Zhang T. and Oles F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval* **4**, 5-31.

[69] Zhu L., Miao B. and Peng H. (2006). On sliced inverse regression with high-dimensional covariates. *Journal of the American Statistical Association*. **101**, 630-643.

[70] Zhu J., Rosset S., Hastie T. and Tibshirani R. (2003). 1-norm support vector machines. *Neural Information Processing Systems* **16**.

[71] Zou H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418-1429.

[72] Zou H., Hastie T. and Tibshirani. R. (2004). Sparse principal component analysis. *Technical Report*.