DEPARTMENT OF STATISTICS
University of Wisconsin
1210 West Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 1044

October 29, 2001 (revised February 10, 2002)

A Note on Margin-based Loss Functions in Classification

by

Yi Lin

A Note on Margin-based Loss Functions in Classification

Yi Lin

Abstract

In many classification procedures, the classification function is obtained (or trained) by minimizing a certain empirical risk on the training sample. The classification is then based on the sign of the classification function. In recent years, there have been a host of classification methods proposed in machine learning that use different margin-based loss functions in the training. Examples include the AdaBoost procedure, the support vector machine, and many variants of them. The margin-based loss functions used in these procedures are usually motivated as upper bounds of the misclassification loss, but this can not explain the statistical properties of the classification procedures. We consider the margin-based loss functions from a statistical point of view. We first show that under general conditions, margin-based loss functions are Fisher consistent for classification. That is, the population minimizer of the loss function leads to the Bayes optimal rule of classification. In particular, almost all margin-based loss functions that have appeared in the literature are Fisher consistent. We then study margin-based loss functions in the method of sieves and the method of regularization. We show that the Fisher consistency of margin-based loss functions often leads to consistency and rate of convergence (to the Bayes optimal risk) results under general conditions. The common notion of margin-based loss functions as upper bounds of the misclassification loss is formalized and investigated. It is shown that the hinge loss is the tightest convex upper bound of the misclassification loss. Simulations are carried out to compare some commonly used margin-based loss functions.

Key Words: Consistency for classification, Bayes rule of classification, rate of convergence, method of sieves, method of regularization.

1 Introduction

We consider the binary classification problem studied extensively in statistics and machine learning. Suppose we are given a training set of observations $D_n = \{(\mathbf{x}_i, y_i), i = 1, ..., n\}$, assumed to be i.i.d. realizations of a random pair $(\mathbf{X}, Y)$. Here $\mathbf{X} \in \mathcal{X}$ is the explanatory or input vector and Y is the class label that takes values in $\{-1, 1\}$. A classification rule ϕ is a mapping from the input space $\mathcal{X}$ to $\{-1, 1\}$. The generalization error of ϕ is the expected misclassification rate $R(\phi) = P\{\phi(\mathbf{X}) \neq Y\}$. This is the evaluating criterion for the performance of classifiers. Let $p(\mathbf{x}) = P(Y = 1|\mathbf{X} = \mathbf{x})$ be the conditional probability of the positive class given $\mathbf{X} = \mathbf{x}$, and $g(\mathbf{x}) = \log[p(\mathbf{x})/(1 - p(\mathbf{x}))]$ be the log odds function. Then the decision-theoretically optimal classification rule with the smallest generalization error is $\phi^*(\mathbf{x}) = \text{sign}[p(\mathbf{x}) - 1/2] = \text{sign}[g(\mathbf{x})]$. This optimal rule is called the Bayes optimal rule. The generalization error of the Bayes optimal rule $R^* = R(\phi^*)$ is called the Bayes optimal risk. When $p(\mathbf{x}) = 1/2$, it does not matter how you classify a subject, as the misclassification rate is the same. To avoid technicality, we assume throughout this paper $p(\mathbf{x}) \neq 1/2, \quad a.s.$.

In practice we do not know the underlying probability distribution, and need to learn a good classification rule from the training sample. A common approach is to derive a classification function $f(\mathbf{x})$ by minimizing an empirical risk on the training sample. The classification rule is then taken to be sign$[f(\mathbf{x})]$. For a given training loss function $\ell(y, f(\mathbf{x}))$, the empirical risk is $1/n \sum_i \ell(y_i, f(\mathbf{x}_i))$. In the traditional logistic regression, the training loss function is the conditional negative log-likelihood $\log[1+e^{-yf(\mathbf{x})}]$. In recent years, there have been a host of classification methods proposed in the machine learning literature that use different loss functions based on the concept of the margin. Given a classification function f, the margin of a subject $(\mathbf{x}, y)$ achieved by the classification function is defined as $yf(\mathbf{x})$. It is easy to see that a subject is classified correctly by the classification rule sign$[f(\mathbf{x})]$ if and only if its margin is positive. The concept of margin is commonly used in machine learning. A large margin is often taken to indicate a confident classification, and many machine learning methods are often motivated by the notion of increasing the margin.

The misclassification loss of a classification rule sign$[f(\mathbf{x})]$ can be written as a function of the margin: $[-yf(\mathbf{x})]_*$, where the function $(\cdot)_*$ takes value 1 for positive arguments and 0 for negative arguments. Intuitively, the misclassification loss should be used as the training loss, since it is the loss function used to evaluate the performances of classifiers. However, the function $(\cdot)_*$ is not convex and not continuous, and causes problems for computation. Therefore many other margin-based loss functions are used as training loss functions in many classification procedures. Examples include the exponential loss $\exp[-yf(\mathbf{x})]$ used in AdaBoost, the hinge loss $[1-yf(\mathbf{x})]_+$ used in the support vector machine, and many others. A brief overview of these loss functions is given in section 2.

In the machine learning literature, margin-based loss functions are usually motivated as approximations to or upper bounds of the misclassification loss. However, this does not explain why such loss functions give good performance statistically, since minimizing an upper bound is quite different from minimizing the original function, especially since these upper bounds are not very tight. Many theoretical results on procedures based on margin-based loss functions have been derived in the computational learning literature. We do not attempt to give a complete list, but refer to Vapnik (1995), Schapire, Freund, Bartlett, and Lee (1998), Bartlett and Shawe-Taylor (1999), Cristianini and Shawe-Taylor (2000), and Mason, Baxter, Bartlett, and Frean (2000) for references. Such results typically bound the generalization error with quantities related to the empirical margins of the training sample points. It is not clear how the generalization error or its upper bound compare with the Bayes optimal risk R^*.

Jiang (2001) argued forcefully that when data are noisy, it is more appropriate to study the difference between the generalization error and the Bayes optimal risk, rather than the generalization error itself. In most practical situations the Bayes optimal risk R^* is greater than zero, therefore the generalization error of any classifier can not be going to zero as the sample size goes to infinity. The magnitude of the generalization error alone does not give a complete picture of the performance of the classifier. The same generalization error of say 0.2 may mean an excellent performance in situations where the classes have a serious overlap; it can also mean a very poor performance in a simple situation where the classes are almost linearly separated. Thus the Bayes optimal risk should be used as a benchmark for the performance of classifiers. A sequence of classifiers is said to be consistent if their misclassification errors converge to the Bayes optimal risk. Formally, Let ϕ_n be a sequence of classifiers based on data D_n. Let $R(\phi_n) = P\{\phi_n(\mathbf{X}, D_n) \neq Y|D_n\}$ be the generalization error of ϕ_n. Then ϕ_n is consistent for a family of distributions of $(\mathbf{X}, Y)$, if for every member of the family, we have $R(\phi_n) \longrightarrow R(\phi^*)$ in probability. See Devroye, Györfi, and Lugosi (1996). Marron (1983) and Mammen and Tsybakov (1999) studied the rate with which the generalization error of classification rules goes to the Bayes optimal risk. Marron (1983) showed the optimal rate of convergence to the Bayes optimal risk, under smoothness conditions on the class densities, is the same as that of the mean integrated squared error (going to zero) in function estimation, and the (density) plug in rule achieves the optimal rate of convergence. Mammen and Tsybakov (1999) studied the rates under smoothness conditions on the decision boundary. The optimal rates they obtained is much fast. They further showed that such rates can be achieved by decision rules based on minimization of empirical risk over the whole class of sets or over sieves. These methods, however, are hard to implement.

In this paper we consider the consistency and rate of convergence (to the Bayes optimal risk) properties of classifiers based on general margin-based loss functions. In section 3, we motivate a notion of Fisher consistency for classification, and established the Fisher consistency for a very general class of margin-based loss functions. It is also noted there an interesting connection between kernel smoothing in classification and a method of regularization with a particularly simple margin-based loss function. In section 4, we connect Fisher consistency with consistency, and use this connection to derive some theoretical results on consistency and rate of convergence (to the Bayes optimal risk) for classifiers based on margin-based loss functions.

In section 5 we look at some of the commonly used convex margin-based loss functions in more detail. In particular, the hinge loss is shown to be the tightest convex upper bound of the misclassification loss. Some simulations are carried out to compare the performance of several commonly used convex margin-based loss functions in the context of boosting and the method of regularization. A summary and discussions are given in section 6. Proofs are given in section 7.

2 Examples of margin-based loss functions

Boosting and the support vector machine are two examples of recently proposed machine learning procedures that involve margin-based training loss functions. Each has a number of variants with different margin-based loss functions.

Boosting was proposed in the Computational Learning Theory literature. See Schapire (1990), Freund (1995), and Freund and Schapire (1996). The basic idea is to combine weaker learners to improve performance. Freund and Schapire (1996) introduced the popular AdaBoost procedure. It has been noted that boosting can be seen as a gradient descent algorithm in the function space. See Breiman (1999), Mason et al (2000), Friedman, Hastie, and Tibshirani (2000), Collins, Schapire, and Singer (2000). Friedman, Hastie, and Tibshirani (2000) showed that AdaBoost can be viewed as a stage-wise additive fitting with the exponential loss function $\exp[-yf(\mathbf{x})]$. This loss function appeared in Schapire and Singer (1998), and was motivated as an upper bound on the misclassification loss. Friedman, Hastie, and Tibshirani (2000) further proposed LogitBoost procedure based on the negative log-likelihood $\log[1+e^{-yf(\mathbf{x})}]$. Friedman (2001) introduced a gradient boosting method MART for regression and classification. Mason et al (2000) proposed AnyBoost classification procedures that perform gradient descent in the function space with general loss functions of the margin. They gave a list of some of the margin-based loss functions used in existing boosting type methods. Other than the loss functions mentioned earlier, the list include $[1-yf(\mathbf{x})]^5$ used in the ARC-X4 procedure (Breiman, 1998), and $[1-yf(\mathbf{x})]^2$ used in constructive NN algorithm (Lee, Bartlett, and Williamson, 1996) [this is actually the same as the common square loss $(y-f(\mathbf{x}))^2$]. See Figure 1, top left panel, for the graphs of these two loss functions and the exponential loss function. The graphs of all the loss functions discussed in this section are given in Figure 1.

Mason, Bartlett, and Baxter (1999) introduced a notion of B-admissibility of margin-based loss functions. A family $\{C_N : N = 1, 2, ...\}$ of margin cost functions is B-admissible for $B \geq 0$ if for all N there is an interval $I \in R$ of length no more than B and a function $\Phi_N : [-1, 1] \to I$ that satisfies

$$[-\alpha]_* \leq E[\Phi_N(Z)] \leq C_N(\alpha)$$

for all $\alpha \in [-1, 1]$, where $Z = (1/N)\sum_{I=1}^{N} Z_I$ with $Z_I \in \{-1, 1\}$ i.i.d. and $P(Z_I = 1) = (1+\alpha)/2$. They motivated this notion by deriving an upper bound of the generalization error for procedures based on B-admissible margin cost functions. Based on the upper bound, they proposed a family of margin-based loss functions. These are smooth non-convex functions that follow the misclassification loss closely, and are not easy to deal with computationally. For computational considerations, Mason et al (1999) proposed using:

$$C_\theta(\alpha) = \begin{cases} (1.2-\gamma)-\gamma\alpha & : \; -1 \leq \alpha \leq 0 \\ (1.2-\gamma)-(1.2-2\gamma)\alpha/\theta & : \; 0 \leq \alpha \leq \theta \\ \gamma/(1-\theta)-\gamma\alpha/(1-\theta) & : \; \theta \leq \alpha \leq 1 \end{cases}$$

with γ fixed at 0.1. Mason et al (2000) further proposed using the so called normalized sigmoid cost function $1-\tanh[\lambda yf(\mathbf{x})]$. See Figure 1, top right panel, for the graphs of these functions.

The support vector machine was first proposed in Boser, Guyon, and Vapnik (1992). The hard margin linear support vector machine (Boser, Guyon, and Vapnik, 1992) simply finds the optimal separating hyper-plane in the simple situation of linearly separable classes. In the soft margin support vector machine (Cortes and Vapnik, 1995) nonnegative slack variables are used to deal with overlapping classes. The linear support vector machine is then extended to the nonlinear support vector machine by mapping the data into high (even infinite) dimensional feature space, and applying the linear support vector machine in the feature space. Through a reproducing kernel trick, the computation of the linear support vector machine in the high (or infinite) dimensional feature space can be carried out in the original input space, and we do not have to explicitly implement the mapping into the feature space. The support vector machine has the advantage that the solution is usually sparse, and has been used on very large datasets. Several authors, including Vapnik (1995) and Shawe-Taylor and Cristianini (1998) have derived upper bounds on the generalization error of the SVM based on the quantities related to the sample margins.

It is now well known that the nonlinear support vector machine can be seen as a special case of the method of regularization. See Wahba (1999), Evgeniou, Pontil, and Poggio (1999). For a general loss function $\ell(y, f(\mathbf{x}))$, the method of regularization solves

$$\arg\min_{f \in F} 1/n \sum_{i=1}^{n} \ell[y_i, f(\mathbf{x}_i)] + \lambda_n J(f), \tag{1}$$

where F is a reproducing kernel Hilbert space of functions, $J(\cdot)$ is a penalty (regularization) functional, often a norm or semi-norm in F. The smoothing parameter λ_n depends on the sample size n. The penalized logistic regression is one example of the method of regularization with the (logistic) loss function $\log[1 + e^{-yf(\mathbf{x})}]$. The support vector machine is another example with the corresponding training loss function being $[1 - yf(\mathbf{x})]_+$. Here $(\tau)_+ = \tau$, for $\tau > 0$; and is 0 otherwise. This loss function is also referred to as the hinge loss. Once the SVM solution f is found, the classification rule is $\text{sign}[f(\mathbf{x})]$. Some variants of the support vector machine use the loss function $[1 - yf(\mathbf{x})]_+^q$ with $q > 1$, especially with $q = 2$. See Burges (1998), Lee and Mangasarian (2001). See Figure 1, bottom left panel for a display of the logistic loss, the hinge loss, and the hinge loss with $q = 2$.

Shen, Zhang, Tseng, and Wong (2001) proposed the generalization machine, which is a method of regularization with a margin-based loss function $\psi(yf)$, where $\psi(z)$ is 2 if $z < 0$, $1 - z$ if $0 < z < 1$, and 0 if $z \geq 1$. See Figure 1, bottom right panel for a display of this loss function. They proved some interesting theoretic results for the generalization machine, including some rate of convergence (to the Bayes optimal risk) results.

3 Fisher consistency of margin-based loss functions

We show that under general conditions, margin-based loss function $\ell(y, f(\mathbf{x})) = V(yf(\mathbf{x}))$ satisfies the condition that the minimizer of $E\ell(Y, f(\mathbf{X}))$ has the same sign as $\text{sign}[2p(\mathbf{x}) - 1]$. Such a condition can be seen as the Fisher consistency for classification problems. In the traditional parameter estimation situation, Fisher consistency means that the estimation procedure in the population space will produce the target of the estimation. For example, let Z_i, $i = 1, 2, ..., n$ be a random sample from a distribution $f(z, \theta_0)$. An M-type estimate of θ_0 is the minimizer of $1/n \sum_{i=1}^{n} \ell(z_i, \theta)$, for some loss function ℓ. The estimation procedure is said to be Fisher consistent if the minimizer of $E\ell(Z, \theta)$ is θ_0. See, for example, Duan and Li (1989). Fisher consistency usually implies strong consistency under suitable regularity conditions. One obvious example is the maximum likelihood estimation. In nonparametric function estimation, Fisher consistency means that the population minimizer of the estimation criterion is the underlying true function to be estimated.

In the context of classification, in order for a classification rule $\text{sign}[f(\mathbf{x})]$ to achieve the Bayes optimal risk, the classification function f must have the same sign as $\text{sign}[p(\mathbf{x}) - 1/2]$. Therefore Fisher consistency for a classification procedure based on a loss function can be defined to be that the population minimizer of the loss function have the same sign function as $\text{sign}[p(\mathbf{x}) - 1/2]$.

Example 3.1 *The method of regularization (1). There is a large body of literature on the method of regularization in function estimation. See e.g., Silverman (1982), Wahba (1990),* *Cox and O'Sullivan (1990), Van de Geer (1990), Gu and Qiu (1993), Shen (1998) for references. Cox and O'Sullivan (1990) provided a general abstract framework for studying the asymptotic properties of such methods. In general, let the minimizer of $E\ell[Y, f(\mathbf{X})]$ be denoted by $\bar{f}$, then under the condition that $\bar{f}$ is in the reproducing kernel Hilbert space F under study, and some other regularity conditions, the solution $\hat{f}$ to (1) goes to $\bar{f}$. If $\bar{f}$ has the same sign as sign$[p(\mathbf{x}) - 1/2]$, then we can expect the method of regularization estimate approaches the Bayes optimal rule.*

Example 3.2 *The method of sieves. The method of sieves solves*

$$\arg\min_{f\in F_n} 1/n \sum_{i=1}^{n} \ell[y_i, f(\mathbf{x}_i)], \tag{2}$$

where F_n is an increasing sequence of subspaces (approximating spaces) of F. There is a large body of literature on the method of sieves in nonparametric function estimation. See e.g., Grenander (1981), Geman and Hwang (1982), Shen and Wong (1994) for references. A popular special case of the method of sieves takes the approximating spaces to be the span of polynomial spline functions and their tensor products. See Stone, Hansen, Kooperberg, and Truong (1997), Huang (1998) for the asymptotic theory of these methods. In general, the target of the estimation is the population minimizer of $\ell[y, f(\mathbf{x})]$. Therefore, for the sieve method to perform well for classification, the minimizer of $E\ell[Y, f(\mathbf{X})]$ should have the same sign as sign$[p(\mathbf{x}) - 1/2]$. Many classification and regression procedures including MARS (Friedman, 1990) are often viewed as greedy, adaptive implementations of the method of sieves, though the properties of the adaptive procedures can not be fully explained through the existing asymptotic theory of the method of sieves. In this sense boosting is related to the method of sieves with the approximating spaces being the linear spaces spanned by the weak learning functions. The method of regularization is also related to the method of sieves. The minimization problem (1) is equivalent to (2) with $F_n = \{f \in F : J(f) \leq M_n\}$ for some M_n.

Consider a function V satisfying the following assumptions:

1. $V(z) < V(-z)$, $\forall z > 0$.
2. $V'(0) \neq 0$ exists.

We have the following

Theorem 3.1 *Let V be some function satisfying assumptions 1 and 2. If $EV[Yf(\mathbf{X})]$ has a global minimizer $\bar{f}(\mathbf{x})$, then sign$[\bar{f}(\mathbf{x})] =$ sign$[p(\mathbf{x}) - 1/2]$, a.s..*

Several special cases are well known in the literature: the population minimizer of $[1-yf(\mathbf{x})]^2$ is $2p(\mathbf{x}) - 1$; the population minimizer of the logistic loss is the log odds function $g(\mathbf{x}) = \log[p(\mathbf{x})/(1 - p(\mathbf{x}))]$, which has the same sign as that of $p(\mathbf{x}) - 1/2$; Friedman, Hastie, and Tibshirani (2000) showed that the population minimizer of the exponential loss is half of the log odds function. Lin (1999) showed that the population minimizer of the hinge loss is sign$[p(\mathbf{x}) - 1/2]$.

Remark 3.1 *Notice we do not require the global minimizer to be unique. Also, the global minimizer is allowed to take on values ∞ or $-\infty$. One example is the normalized sigmoid loss $1-\tanh(\lambda y f(\mathbf{x}))$. It can be checked that the population minimizer is ∞ when $p(\mathbf{x}) > 1/2$, and is $-\infty$ when $p(\mathbf{x}) < 1/2$.*

Remark 3.2 *Assumption 2 is only used to guarantee that $\bar{f}(\mathbf{x}) \neq 0$, a.s., and can be relaxed. For example, consider the misclassification risk $E[-Yf(\mathbf{X})]_*$. It is minimized by any function that has the same sign as sign$[p(\mathbf{x}) - 1/2]$, though the function $(\cdot)_*$ is not differentiable at* 0*. Another example is the loss function used in the generalization machine (Shen et al, 2001). The loss is not differentiable at* 0*. It can be seen that the population minimizer of the loss function is any number greater than* 1 *when $p(\mathbf{x}) > 1/2$, and is any number smaller than* -1 *when $p(\mathbf{x}) < 1/2$. Thus the population minimizer has the same sign as that of sign$[p(\mathbf{x}) - 1/2]$.*

We can relax assumption 1 to

Assumption 1': there exists a positive number a such that $V(z) > V(a)$ for any $z > a$, and $V(z) > V(-a)$ for any $z < -a$, and that $V(z) < V(-z)$, $\forall z \in (0, a]$.

Theorem 3.2 *Let $\bar{f}(\mathbf{x})$ be any global minimizer of $EV[Yf(\mathbf{X})]$ for some function V satisfying assumptions 1' and 2. Then we have sign$(\bar{f})$ = sign$(p - 1/2)$, a.s..*

We can see assumption 1 can be seen as a special case of assumption 1' with $a = \infty$. It is interesting to see that loss functions that put very serious penalty on classifications that are too correct (margin larger than 1), while putting very slight penalty on misclassification are still Fisher consistent.

Here we note a connection between kernel smoothing and a method of regularization in classification. Silverman (1984) showed that in regression the smoothing spline (a method of regularization with the cubic spline reproducing kernel) corresponds approximately to smoothing by a variable bandwidth kernel. It turns out in classification the connection between kernel smoothing and the method of regularization is much more straightforward. Consider the simplest margin-based training loss function $-yf(\mathbf{x})$. The minimizer of $E[-Yf(\mathbf{X})]$ is ∞ when $p(\mathbf{x}) > 1/2$, and is $-\infty$ when $p(\mathbf{x}) < 1/2$. This loss function has not been used in any existing classification procedure in the literature. Let us consider (1) with this loss function. Let F be a reproducing kernel Hilbert space with reproducing kernel $K(\mathbf{x}, \mathbf{x}')$, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Let the penalty term $J(f)$ be the norm of f in F. By the representer theorem (Kimeldorf and Wahba, 1971), the solution to (1) is in the finite dimensional space spanned by $\{K(\mathbf{x}_i, \cdot), i = 1, ..., n\}$. Write $f(\mathbf{x}) = \sum_i c_i K(\mathbf{x}_i, \mathbf{x})$. Now let $\mathbf{y} = (y_1, ..., y_n)'$, $\mathbf{c} = (c_1, ..., c_n)'$, $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_n))'$ and $\mathbf{K}$ be the n by n matrix with ijth entry being $K(\mathbf{x}_i, \mathbf{x}_j)$. The quantity in (1) becomes

$$-\mathbf{y}'\mathbf{K}\mathbf{c}/n + \lambda_n \mathbf{c}'\mathbf{K}\mathbf{c},$$

and the minimizer is $\hat{\mathbf{c}} = \mathbf{y}/(2n\lambda_n)$. Therefore $\hat{f}(\mathbf{x}) = \sum_i y_i K(\mathbf{x}_i, \mathbf{x})/(2n\lambda_n)$, and the classification rule is $\text{sign}[\hat{f}(\mathbf{x})] = \text{sign}[\sum_i y_i K(\mathbf{x}_i, \mathbf{x})]$. This is exactly kernel smoothing in classification. For example, let the reproducing kernel be the Gaussian reproducing kernel

$K(\mathbf{x}, \mathbf{x}') = \exp[(\mathbf{x}-\mathbf{x}')^2/\sigma^2]$, then the resulting classification is the same as the classification resulted from the kernel smoothing with Gaussian kernel. Thus the kernel smoothing technique in classification corresponds to a method of regularization with a particularly simple margin-based loss function.

4 Some asymptotic results

There exist well established framework for deriving asymptotic results for function estimation problems. These frameworks are especially applicable to situations in which the loss function is convex. We demonstrate that these frameworks can be used to derive rate of convergence (to the Bayes optimal risk) results in classification, once the Fisher consistency of the training loss is established. The following lemma shows that the convergence of the generalization error to the Bayes optimal risk can be studied through studying the convergence in function estimation.

Lemma 4.1 *Let V be a function satisfying assumptions 1 [or 1'], 2, and $V''(z) > 0$, $\forall z$. If $\bar{f}$ is the global minimizer of $EV[Yf(\mathbf{X})]$, then for any function f,*

$$R[sign(f)] - R^* \leq c\int |f(\mathbf{x}) - \bar{f}(\mathbf{x})| d(\mathbf{x}) d\mathbf{x} \leq c\{\int |f(\mathbf{x}) - \bar{f}(\mathbf{x})|^2 d(\mathbf{x}) d\mathbf{x}\}^{1/2},$$

where c is a constant depending only on V, and $d(\mathbf{x})$ is the density function of $\mathbf{X}$.

Remark 4.1 *For the special case in which $\bar{f}(\mathbf{x})$ is $[p(\mathbf{x}) - 1/2]$, this lemma is well known. See Theorem 2.2 of Devroye, Györfi, and Lugosi (1996), and the references listed there. Lemma 4.1 represents a generalization that is applicable to more general target functions. Much of the proof of Lemma 4.1 is to establish the inequality*

$$|p(\mathbf{x}) - 1/2| \leq c|\bar{f}(\mathbf{x})| \tag{3}$$

for some positive c that does not depend on $\mathbf{x}$.

Remark 4.2 *The condition $V''(z) > 0$, $\forall z$, in Lemma 4.1 can be relaxed. The key of the proof is (3). For the hinge loss, we have $\bar{f}(\mathbf{x}) = sign[p(\mathbf{x}) - 1/2]$, therefore (3) is trivially satisfied, and result of Lemma 4.1 applies. Other examples include the generalization machine loss and the normalized sigmoid loss.*

Existing frameworks in function estimation problems can then be applied directly to establish asymptotic results in classification. For example, combining Lemma 4.1 and a straightforward application of the framework in Shen and Wong (1994) gives

Theorem 4.1 *Let $\ell(y, f) = V(yf)$ be the loss function in the method of sieves, with V satisfying assumptions 1 [or 1'], 2, and V'' is continuous and positive. Let $\bar{f}$ be the minimizer of $E\ell(Y, f(\mathbf{X}))$. Let F be a function space and the functions in F are bounded uniformly by a constant C in the L_∞ norm. Assume $\bar{f} \in F$. Let F_n be a sequence of approximating subspaces (sieve) of F satisfying $H(\epsilon, F_n) \leq An^{2r_0}\epsilon^{-r}$ for some constants $r_0 < \frac{1}{2}$, $A > 0$, and all small*

$\epsilon > 0$. Here $H(\epsilon, F_n)$ is the L_∞-metric entropy of the space F_n, that is, $\exp(H(\epsilon, F_n))$ is the number of ϵ-balls in the L_∞-metric needed to cover the space F_n. Then for the sieve estimate f_n, we have

$$R[sign(f_n)] - R^* = O_p(\max(n^{-\tau}, \rho(\pi_n \bar{f}, \bar{f}))).$$

Here $\rho(\pi_n \bar{f}, \bar{f}) = \inf_{f \in F_n} \int (f - \bar{f})^2 d(\mathbf{x}) d\mathbf{x}$, and

$$\tau = \begin{cases} \frac{1-2r_0}{2} - \frac{\log\log n}{2\log n}, & if \quad r = 0^+; \\ \frac{1-2r_0}{2+r}, & if \quad 0 < r < 2; \\ \frac{1-2r_0}{4} - \frac{\log\log n}{2\log n}, & if \quad r = 2; \\ \frac{1-2r_0}{2r} & if \quad r > 2, \end{cases}$$

where ϵ^{-0^+} is taken to represent $\log(1/\epsilon)$.

Remark 4.3 *The condition that the functions in F are uniformly bounded by a constant C in the L_∞ norm is introduced for convenience. This condition is reasonable when V satisfies assumption 1' with a finite $a > 0$, since in that case $\bar{f}$ is always bounded by a. In cases of exponential loss and logistic loss, additional assumptions such as that $p(\mathbf{x})$ is bounded away from 0 and 1 is required for $\bar{f}$ to be bounded.*

Remark 4.4 *Theorem 4.1 is a direct application of Theorem 1 of Shen and Wong (1994). Stronger results (with weaker conditions) can be obtained by applying Theorem 2 of Shen and Wong (1994). We do not pursue this here.*

Thus the rate of convergence to the Bayes optimal risk is controlled by the size of approximating space and the sieve approximation error $\rho(\pi_n \bar{f}, \bar{f})$. The best rate is determined by the complexity of the function space F.

Results on the method of regularization can be established through Theorem 4.1 by making use of the connection between the method of regularization and the method of sieves. However, it is usually more straightforward to combine Lemma 4.1 and the framework in Cox and O'Sullivan (1990). Suppose V is a function satisfying assumptions 1 [1'], 2, $V''(z) > 0$, $\forall z$, and V''' exists and is continuous. Then the population minimizer of the loss function $V(yf)$ is unique. Assume it is in a reproducing kernel Hilbert space F. For the classifier obtained through (1), the rate of convergence of the generalization error to the Bayes optimal risk depends on the rate of decay of the eigenvalues of the reproducing kernel corresponding to F, which can be seen as another measure of the complexity of the function space F.

Results like Theorem 4.1 concerns the plug-in method in which a smooth classification function is obtained, and then the classification is based on the sign of the classification function. The rate of convergence of such plug-in methods to the Bayes optimal risk is the same as the rate in function estimation. The results in Marron (1983) suggests that this is the optimal rate for classification problems under global smoothness conditions. Mammen and Tsybakov (1999) considered the classification problem under smoothness conditions on the classification boundary instead of the classification function, and showed that it is possible to achieve faster rate of convergence than that of function estimation. However, the methods discussed in that paper are hard to implement.

Remark 4.5 *Nonparametric methods for classification typically involve smoothing parameters. In the method of regularization (1), the smoothing parameter is λ. In the method of sieves (2), the smoothing parameter is the dimension of the approximating space (and related characteristics of the approximating space). In the following discussion let us generally denote the smoothing parameter by λ. While the training loss functions is usually different from the misclassification loss, the tuning of the smoothing parameter is typically based on the misclassification loss. A direct consequence is that the classifiers picked by the tuning may not be close to the target function of the training (i.e., the population minimizer of the training loss), but they are still consistent with respect to the misclassification loss. Consider a common scenario in which we have an independent tuning set of size m. Let the classification function obtained with the smoothing parameter λ be denoted by f_λ and the corresponding classification rule is $\phi_\lambda = sign[f_\lambda]$. We choose the classifier $\phi_{\hat{\lambda}}$ with the smallest misclassification error on the tuning set over $C_k = \{\phi_\lambda : \lambda = 2^{-k}, ..., 1, 2, ..., 2^k\}$. Devroye, Györfi, and Lugosi (1996) considered this formulation of tuning smoothing parameter and showed (in section 25.2) that if the classification rule ϕ_λ is consistent, then the tuned classifier is consistent if $n \to \infty$, $k \to \infty$, $m \to \infty$, and $\log(k) = o(m)$.*

One way to look at this is to consider the models built by the training process as a parametric family of models parametrized by λ. For margin-based loss functions, the discussions in earlier sections show that under some conditions, there exists some members of the parametric family built by the training that is close to the optimal in terms of function estimation, and thus in terms of classification by Lemma 4.1. However, these members may not be the closest to the optimal in terms of classification. This is because for good classification it is not necessary to have a good estimation of the target function. All that is needed is that the estimate has the same sign as the target function. The tuning may actually pick some other members in the parametric family that might be closer to the optimal in terms of classification.

5 Some comparison studies

As can be seen from earlier sections, many different margin-based loss functions lead to consistent classification. It is of practical interest to compare their performances in different situations. In this section we compare different loss functions with simulation. Margin-based loss functions have usually been motivated as upper bounds of the misclassification loss, and it seems there is a common notion that tighter upper bounds give better classification performance. Another goal of this section is to investigate this notion.

We concentrate on convex loss functions, as they are generally easy to work with computationally. Common examples include the square loss, the exponential loss, the logistic loss, and the hinge loss. In the context of boosting, Friedman, Hastie, and Tibshirani (2000) remarked that the square loss usually perform quite well, but are generally inferior to the monotone decreasing margin-based loss functions, as it penalizes the classifications that are too correct. The exponential loss is particularly suited to the boosting procedure computationally, and is used in the popular procedure AdaBoost. Friedman, Hastie, and Tibshirani (2000) provided manageable algorithms for LogitBoost with the logistic loss. Bühlmann and Yu (2001) studied the properties of boosting with the square loss (called L_2Boost in that paper). They found comparable performances between L_2Boost and LogitBoost. The hinge loss has the nice property that it often leads to sparse solutions as in the support vector machine. All of these convex loss function can be seen as upper bounds of the misclassification loss.

The notion of being an upper bound of the misclassification loss is not well defined, since it is possible to make any function that is bounded from below an upper bound of the misclassification loss by adding a large constant to it. However, it is possible to compare different loss functions as upper bounds of the misclassification loss if we take into account the equivalence between loss functions. Two loss functions are equivalent for a classification procedure if the procedure based on the two losses give identical classification rules. For the method of regularization and the method of sieves, it can be seen that the loss functions $\ell(y, f)$ and $a\ell(y, bf) + c$ are equivalent for any $a > 0$, $b > 0$, and $c \in R$. For example, let $J(f)$ be a semi-norm in a reproducing kernel Hilbert space F, and denote the solution to (1) by $\hat{f}$. Then the solution to (1) with ℓ replaced by $a\ell(y, bf) + c$ and λ replaced by λab^2 is $\hat{f}/b$. Therefore the two solutions have the same sign, and the resulting classification rules are identical.

Proposition 5.1 *For any convex function V that satisfies condition 1 (or 1') and 2, that is an upper bound of the misclassification loss, there exists a function W such that $W \leq V$ everywhere, and $W(yf)$ is equivalent to the hinge loss. The equal sign holds everywhere only when V is equivalent to the hinge loss.*

Proof: Let U be the tangent line of V at $(0, V(0))$. Then $U \leq V$ everywhere since V is convex. By condition 1 (or 1') and 2, it is easy to see U has a negative slope. Let $W = \max(U, 0)$. Then $W \leq V$ everywhere since $V \geq 0$ everywhere. It is easy to see that W is equivalent to the hinge loss.

Proposition 5.1 establishes the hinge loss as the tightest convex margin-based upper bound of the misclassification loss. However, whether this translates into advantages in terms of classification performance is not clear and deserves further study.

In the following we compare the classification performance of classification procedures based on different loss functions through simulation studies. We first compare boosting procedures based on different convex losses. We concentrate on MART algorithms introduced in Friedman (1999, 2001). Five different loss functions are considered: the square loss, the logistic loss, the absolute deviation loss $|1 - yf|$, the exponential loss, and the hinge loss.

MART is a boosting algorithm with regression trees as the base learner. Given a loss function $\ell(y, f)$, the procedure can be described as in the following (c.f. Friedman, 2001). Initialize with a constant function

$$\arg\min_{\rho} \sum_{i=1}^{n} \ell(y_i, \rho), \tag{4}$$

then iterate through the following steps (denote the fitted function at the m-th iteration by F_m):

1. Calculate current residuals defined as

$$\tilde{y}_i = -\left[\frac{\partial \ell(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \quad i = 1, 2, ..., n.$$

2. Fit a J-node regression tree to the current residuals to obtain the tree terminal nodes R_{jm}, $j = 1, 2, ..., J$.

3. For each terminal node, find

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{jm}} \ell(y_i, F_{m-1}(\mathbf{x}_i) + \gamma)$$

4. Update the current fit:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + r \sum_{j=1}^{J} \gamma_{jm} 1(\mathbf{x} \in R_{jm})$$

Here the constant r in the last step is a regularizing factor. The regression tree in step 2 is induced with the square loss for computational simplicity, though it is possible to fit the tree in step 2 with loss functions other than the square loss, especially $\ell(y, f)$.

The derivative in step 1 is straightforward to evaluate. For example, for the hinge loss $(1-yf)_+$, the derivative is $-y[\text{sign}(1-yf)+1]/2$. The minimization in the initialization (4) and step 3 has been worked out in the cases of the square loss, the logistic loss, the exponential loss, the absolute deviation loss, and Huber's loss function. See Friedman (2001), Hastie, Tibshirani, and Friedman (2001). In the logistic loss and the exponential loss cases, a single Newton-Raphson step is preferred as an approximation to the update in step 3. Here we give the corresponding result for the hinge loss. Consider the problem

$$\arg\min_{\gamma} \sum_{i=1}^{n} \ell(y_i, \gamma + f_i), \tag{5}$$

For the hinge loss, we have

Lemma 5.1 *The solution to (5) with $\ell(y, f) = (1 - yf)_+$ is any number in the interval $[(y-f)_{(n_+)}, (y-f)_{(n_+ +1)}]$, where n_+ is the number of 1's in $\{y_i, i = 1, 2, ..., n\}$, and $(y-f)_{(k)}$ denotes the k-th smallest number in $\{y_i - f_i, i = 1, 2, ..., n\}$.*

In our simulation we consider four conditional probability functions in the eight dimensional space:

$$p_1(\mathbf{x}) = [|\sin(2\pi x_1)|^{x_3} + x_2x_4 + x_5]/3;$$
$$p_2(\mathbf{x}) = e^g/(1+e^g) \quad \text{with} \quad g(\mathbf{x}) = 4(|sin(2\pi x_1)|^{x_3} + x_2x_4 + x_5) - 6;$$
$$p_3(\mathbf{x}) = \{\text{sign}[4(|sin(2\pi x_1)|^{x_3} + x_2x_4 + x_5) - 6] + 2\}/4;$$
$$p_4(\mathbf{x}) = \{\text{sign}[4(|sin(2\pi x_1)|^{x_3} + x_2x_4 + x_5) - 6] + 2\}/4 + 0.2.$$

These functions are simple transformations of a regression function used as an example in Friedman (1999). The transformations are used to make sure that the conditional probability function $p(\mathbf{x})$ takes value in $(0, 1)$. The first two probability functions are continuous, and the last two are discontinuous. For each of the four function, we generate 500 uniform random points in $(0, 1)^8$ for the input vector $\mathbf{X}$, and then generate the y's according to the conditional probability functions. We run MART algorithms with the square loss, the logistic loss, the exponential loss, the absolute deviation loss, and the hinge loss on the generated data. We fix the regularizing factor r at 0.1, and use 6-node regression trees. The simulations are run in R. The simulation is repeated 100 times for each of the four conditional probability functions.

In MART we need to tune the number of iterations. Since we know the true conditional probability function in the simulation, to eliminate randomness in tuning we use the true expected misclassification loss in tuning as we want to concentrate on the performance of different training loss functions. That is, we find the smoothing parameter with the smallest $E[-Yf_\lambda(X)]_*$. The evaluation of $E[-Yf_\lambda(X)]_*$ involves an integral that is sometimes hard to compute, so we actually use a discrete approximation of $E[-Yf_\lambda(X)]_*$:

$$\begin{aligned} & 1/n\sum_{i=1}^{n} E\{[-Yf_\lambda(X)]_*|X = x_i\} \\ = \quad & 1/n\sum_{i=1}^{n}[1-p(x_i)][(\text{sign}(f_\lambda(x_i))+1]/2 + p(x_i)[1-\text{sign}(f_\lambda(x_i))]/2. \end{aligned}$$

Here x_i's are the realized x values. Notice this is not the resubstitution error, and is a good approximation of the expected generalization error.

The results are summarized in Figure 2. From the boxplots we see that the square loss, logistic regression loss, and the exponential loss have very similar performances, and they perform better than the hinge loss, which in turn performs better than the absolute deviation loss. We formally conduct a Bonferonni pairwise comparison with paired t-test. The overall level of the test is 0.05. For each of the four simulations, the exponential loss, the logistic loss and the square loss perform significantly better than the hinge loss and the absolute deviation loss. In the first and third simulation, the exponential loss, the logistic loss and the square loss have comparable performances, and the hinge loss and the absolute deviation loss have comparable performances. In the second simulation, the exponential loss performs significantly better than the logistic loss and the square loss. The latter two show no significant difference. In the fourth simulation, the logistic loss performs significantly better than the exponential loss and the square loss. The latter two show no significant difference. In both the second and fourth simulations, the hinge loss performs significantly better than the absolute deviation loss.

From the above simulation we see that loss functions that are tighter upper bounds of the misclassification loss does not necessarily lead to better classification performance. One possible reason why the hinge loss and the absolute deviation loss do not perform well might be that they are not compatible with the square loss, which is used to induce the regression tree structure in MART procedures (step 2). Another possible reason might be the greedy nature of the tree building process. Breiman, Friedman, Olshen, and Stone (1984) discussed the use of several loss functions in the tree building process. The misclassification loss is found to be not a good choice. Therefore being close to the misclassification loss will not give any advantage.

Now let us turn to the method of regularization, which does not have the greedy nature of the regression tree. We conjecture that in such a situation the performance of different loss functions is related to the complexity of the corresponding target function (the population minimizer of the training loss). To illustrate this we compare the method of regularization with the square loss and with the hinge loss. These correspond to the penalized least square method and the support vector machine. We first consider two one dimensional examples. The conditional probability functions in the examples are

$$p_5(x) = \frac{\exp[g(x)]}{1+\exp[g(x)]}, \text{ where } g(x) = 2\sin(3\pi x^2) + x - 0.5.$$

$$p_6(x) = sign[g(x)]/4 + 1/2.$$

The function $g(x)$ is chosen to ensure that the conditional probability functions cross 1/2 multiple times. This is important for one dimensional examples since otherwise the optimal classification rule depends only on one cross point and is too simple. Notice we do not take p_6 to be $sign[g(x)]/2 + 1/2$. This is because in that case the probabilities are 0 or 1, the positive and negative classes are clearly separable, and the method of regularization with smoothing parameter 0 will perform perfectly.

For these two examples we take the reproducing kernel Hilbert space in (1) to be the second order Sobolev Hilbert space

$$H^2 = \{f|f,\ f' \text{ abs. cont.}, f'' \in L_2\},$$

as is usually done in the smoothing spline literature. For the penalized least square method, the common practice is not to penalize linear functions; while for the support vector machine, the common practice is to only leave the constant unpenalized. For comparison purpose, we only leave the constant unpenalized in both cases. The penalized least square method with linear functions unpenalized perform very similar to that with only constants unpenalized in our examples. The corresponding reproducing kernel is:

$$K(s,t) = k_1(s)k_1(t) + k_2(s)k_2(t) - k_4(|s-t|).$$

where $k_1(\cdot) = \cdot - 0.5$, $k_2 = (k_1^2 - 1/12)/2$, and $k_4 = (k_1^4 - k_1^2/2 + 7/240)/24$. See Wahba (1990), Gu (1993). The solution to (1) with the square loss is obtained with the standard procedure for solving the penalized least square problem. See Wahba (1990), section 1.3. The solution to the support vector machine is obtained by going to the dual setup of the problem. See Burges (1998), Wahba, Lin and Zhang (2000). The simulations are done in Matlab to make use of the quadratic programming function in Matlab.

We generate 100 uniform random numbers in $(0,1)$ for the input variable x. We then generate y's according to the conditional probability functions. We search for the tuning parameter among numbers of the form 2^j, $j \in Z$. For each conditional probability function and each training loss function we run the simulation one hundred times. For the example with p_5, a paired t-test applied to the results found no significant difference between the method of regularization with the square loss and the hinge loss. For the example with p_6, the hinge loss is significantly better.

We next look at two two-dimensional examples. The conditional probability functions are:

$$p_7(\mathbf{x}) = (x_1 + x_2)/2;$$

$$p_8(\mathbf{x}) = sign(x_1 + x_2 - 1)/4 + 0.5$$

In these examples we use the Gaussian kernel $\exp(-\frac{\|\mathbf{s}-\mathbf{t}\|^2}{\sigma^2})$. This is the reproducing kernel commonly used in the support vector machine. Again we leave the constants unpenalized. There are two parameters to be tuned: λ and σ. We search for λ among $\{2^{-14}, ..., 2^5\}$, and σ among $\{0.03125, 0.0625, ..., 1.6\}$.

We generate 100 uniform random numbers in $(0,1)^2$ for the input variable $\mathbf{x}$. We then generate y's according to the conditional probability functions. For each conditional probability function and each training loss function we run the simulation one hundred times. For the example with p_7, a paired t-test applied to the results found the square loss to be significantly better than the hinge loss. For the example with p_8, the hinge loss is significantly better. The results for examples p_5 through p_8 are given in Figure 3.

6 Summary and discussion

In the context of binary classification, the Fisher consistency of a classification procedure based on a loss function $\ell(y, f(\mathbf{x}))$ can be defined as that the minimizer of $E\ell[Yf(\mathbf{X})]$ has the same sign as $sign[p(\mathbf{x}) - 1/2]$. We showed that under very general conditions, margin-based loss functions are Fisher consistent. This gives an explanation why margin-based loss functions generally work well. The Fisher consistency of the margin-based loss functions often leads to the consistency and rate of convergence (to the Bayes optimal risk) results of the corresponding classifiers. The training loss and the tuning loss are usually different. While this does not destroy the consistency property for classifiers, it has the effect that the classification function picked by tuning may not be close to the target function of the training loss.

The hinge loss is shown to be the tightest convex upper bound of the misclassification loss. However, whether this translate into advantages in terms of classification performance depends on the classification procedure used and the complexity of the target function, as suggested by our simulations. Our simulations can only serve as the first step for comparing the loss functions in terms of classification efficiency. A lot more experience is needed before we can draw any definitive conclusion. However, our simulation results seem to indicate several interesting points. In the framework of boosting procedures, the logistic loss, exponential loss, and the square loss seem to give comparable performances, and they perform better than the hinge loss, which in turn outperforms the absolute deviation loss. The weak performance of the hinge loss might be related to the greedy nature of the tree building process. In the framework of the method of regularization, we compared the classification performance of the square loss and the hinge loss. It seems that in situations where the underlying conditional probability function $p(\mathbf{x})$ is very smooth, the square loss outperforms the hinge loss. An intuitive explanation is that in such situations the target function of the square loss is $2p(\mathbf{x}) - 1$, which is simpler than the target function of the hinge loss, which is $sign[p(\mathbf{x}) - 1/2]$. In situations such as examples 6 and 8 in our simulation, the function $p(\mathbf{x})$ is not smooth, and the complexity of $sign[p(\mathbf{x}) - 1/2]$ is similar to the complexity of $2p(\mathbf{x}) - 1$, the hinge loss outperforms the square loss.

7 Proofs

Proof of Theorem 3.1: For any fixed $\mathbf{x}$, we define

$$A(z) = p(\mathbf{x})V(z) + [1 - p(\mathbf{x})]V(-z).$$

It is easy to check that $E[V(Yf(\mathbf{X}))|\mathbf{X} = \mathbf{x}] = A[f(\mathbf{x})]$. Therefore $\bar{z} = \bar{f}(\mathbf{x})$ minimizes $A(z)$.

Since $V'(0) \neq 0$ exists, we have $A'(0) = p(\mathbf{x})V'(0) - [1 - p(\mathbf{x})]V'(0) = [2p(\mathbf{x}) - 1]V'(0) \neq 0$. Thus 0 is not a minimizer of A. Therefore $\bar{f}(\mathbf{x}) \neq 0$.

Since $\bar{f}(\mathbf{x})$ is a global minimizer of $A(z)$, we have,

$$0 \geq A[\bar{f}(\mathbf{x})] - A[-\bar{f}(\mathbf{x})] = [2p(\mathbf{x}) - 1]\{V[\bar{f}(\mathbf{x})] - V[-\bar{f}(\mathbf{x})]\}$$

Now if $p(\mathbf{x}) > 1/2$, then $V[\bar{f}(\mathbf{x})] - V[-\bar{f}(\mathbf{x})] \leq 0$. Since $\bar{f}(\mathbf{x}) \neq 0$, by assumption 1, we get $\bar{f}(\mathbf{x}) > 0$. Therefore $\text{sign}[\bar{f}(\mathbf{x})] = \text{sign}[p(\mathbf{x}) - 1/2]$.

If $p(\mathbf{x}) < 1/2$, the same line of argument as above leads to $\text{sign}[\bar{f}(\mathbf{x})] = \text{sign}[p(\mathbf{x}) - 1/2]$.

Proof of Theorem 3.2: Consider $A(z)$ as defined in the proof of Theorem 3.1. From the assumptions it is easy to check that if $z > a$, then $A(z) > A(a)$; and if $z < -a$, then $A(z) > A(-a)$. Therefore the minimizer of $A(z)$ has to be in the interval $[-a, a]$. That is, $\bar{f}(\mathbf{x})$ is in $[-a, a]$. Now apply the same argument as in the proof of Theorem 3.1.

Proof of Lemma 4.1: Lemma 4.1 is a generalization Theorem 2.2 of Devroye, Györfi, and Lugosi (1996), which states: For any function f,

$$R[sign(f)] - R^* \leq 2\int [|p(\mathbf{x}) - 1/2| 1_{\text{sign}[f(\mathbf{x})] \neq \text{sign}[p(\mathbf{x}) - 1/2]}] d(\mathbf{x}) d\mathbf{x} \leq 2\int |f(\mathbf{x}) - (p(\mathbf{x}) - 1/2)| d(\mathbf{x}) d\mathbf{x}. \tag{6}$$

Now let us prove Lemma 4.1. For any fixed $\mathbf{x}$, we have

$$E[V(Yf(\mathbf{X}))|\mathbf{X} = \mathbf{x}] = p(\mathbf{x})V[f(\mathbf{x})] + [1 - p(\mathbf{x})]V[-f(\mathbf{x})]. \tag{7}$$

This is strictly convex in $f(\mathbf{x})$, therefore there always exists a unique minimizer $\bar{f}(\mathbf{x})$, if we allow $\bar{f}(\mathbf{x})$ to be $\pm\infty$.

We now establish (3) in section 4. This is obviously true when $|\bar{f}(\mathbf{x})| = \infty$. Now we consider the case when $\bar{f}(\mathbf{x})$ is finite. In this case, since $\bar{f}(\mathbf{x})$ is the minimizer of (7), taking derivative at $\bar{f}(\mathbf{x})$, we have,

$$p(\mathbf{x})V'[\bar{f}(\mathbf{x})] - [1 - p(\mathbf{x})]V'[-\bar{f}(\mathbf{x})] = 0. \tag{8}$$

From assumptions 1 [or 1'], and that $V''(z) > 0$, $\forall z$, we know that V either has a minimizer at some $0 < b < \infty$, or it is monotone decreasing. In the former case the same argument as that in the proof of Theorem 3.2 leads to $\bar{f}(\mathbf{x}) \in [-b, b]$. Therefore in both cases we have $V'[-\bar{f}(\mathbf{x})] \leq 0$ and $V'[\bar{f}(\mathbf{x})] \leq 0$. By the strict convexity of V the two equal signs can not hold at the same time since $\bar{f}(\mathbf{x})$ is not 0 (as shown in the proof of Theorem 3.1). Therefore $V'[-\bar{f}(\mathbf{x})] + V'[\bar{f}(\mathbf{x})] \neq 0, a.s.$, and we can solve (8) to get $|p(\mathbf{x}) - 1/2| = |B[\bar{f}(\mathbf{x})]|$, where $B(\cdot)$ is defined as

$$B(z) = 1/2\frac{V'(-z) - V'(z)}{V'(-z) + V'(z)}.$$

It is easy to see that $B(z)/z \to \frac{V''(0)}{2V'(0)}$ as $z \to 0$. Therefore there exist $\delta > 0$ and $C > 0$ only depending on V such that $|B(z)| \leq C|z|$ for $z \in [-\delta, \delta]$. Therefore if $\bar{f}(\mathbf{x}) \in [-\delta, \delta]$, then $|p(\mathbf{x}) - 1/2| = |B[\bar{f}(\mathbf{x})]| \leq C|\bar{f}(\mathbf{x})|$. On the other hand, if $\bar{f}(\mathbf{x})$ is not in $[-\delta, \delta]$, then $|p(\mathbf{x}) - 1/2| \leq 1/2 \leq \frac{1}{2\delta}|\bar{f}(\mathbf{x})|$. So (3) is proved.

By Theorem 3.1, 3.2, (6), and (3), we have

$$\begin{aligned}
& R[sign(f)] - R^* \\
\leq\ & 2\int [|p(\mathbf{x}) - 1/2| 1_{\mathrm{sign}[f(\mathbf{x})] \neq \mathrm{sign}[\bar{f}(\mathbf{x})]}] d(\mathbf{x}) d\mathbf{x} \\
\leq\ & 2c \int [|\bar{f}(\mathbf{x})| 1_{\mathrm{sign}[f(\mathbf{x})] \neq \mathrm{sign}[\bar{f}(\mathbf{x})]}] d(\mathbf{x}) d\mathbf{x} \\
\leq\ & 2c \int [|\bar{f}(\mathbf{x}) - f(\mathbf{x})| 1_{\mathrm{sign}[f(\mathbf{x})] \neq \mathrm{sign}[\bar{f}(\mathbf{x})]}] d(\mathbf{x}) d\mathbf{x} \\
\leq\ & 2c \int |\bar{f}(\mathbf{x}) - f(\mathbf{x})| d(\mathbf{x}) d\mathbf{x} \\
\leq\ & 2c \left\{ \int [\bar{f}(\mathbf{x}) - f(\mathbf{x})]^2 d(\mathbf{x}) d\mathbf{x} \right\}^{1/2}.
\end{aligned}$$

Proof of Theorem 4.1: Define $\rho^2(f_1, f_2) = \int (f_1 - f_2)^2 d(\mathbf{x}) d\mathbf{x}$. By Lemma 4.1 all we need to establish is $\rho(f_n, \bar{f}) = O_p(\max(n^{-\tau}, \rho(\pi_n \bar{f}, \bar{f})))$. To do that we apply Theorem 1 in Shen and Wong (1994) with $\ell(y, f) = V(yf)$, by checking the conditions C1, C2, and C3 in that theorem.

Since $\bar{f}(\mathbf{x})$ is the minimizer of $E[V(Yf(\mathbf{X}))|\mathbf{X} = \mathbf{x}] = p(\mathbf{x})V(f(\mathbf{x})) + (1 - p(\mathbf{x}))V(-f(\mathbf{x}))$, using a Taylor expansion at $\bar{f}(\mathbf{x})$ we get

$$E[V(Yf(\mathbf{X})) - V(Y\bar{f}(\mathbf{X}))] \geq c_1 \rho^2(f, \bar{f}),$$

for all $f \in F$. Here $c_1 = \inf_{[-C,C]} V''/2 > 0$. Therefore condition C1 is satisfied with $\alpha = 1$. For any $f_1, f_2 \in F$, we have $|V(yf_1(\mathbf{x})) - V(yf_2(\mathbf{x}))| \leq c_2 |yf_1(\mathbf{x}) - yf_2(\mathbf{x})| = c_2 |f_1(\mathbf{x}) - f_2(\mathbf{x})|$, where $c_2 = \sup_{[-C,C]} V'$. Here we used the fact that $y \in \{-1, 1\}$. From this it is easy to check that condition C3 is satisfied, and condition C2 is satisfied with $\beta = 1$.

Proof of Lemma 5.1: With the hinge loss, the quantity to be minimized in (5) is a piecewise linear function of γ:

$$\sum_{i=1}^{n} [1 - y_i(\gamma + f_i)]_+. \tag{9}$$

The joint points of the piecewise linear function are $(y_i - f_i)$, $i = 1, 2, ..., n$. To the left of $(y_i - f_i)_{(1)}$, the smallest of the joint points, (9) is a linear function with derivative $-n_+$. Moving γ from left to right, every time a joint point is passed, the derivative of the piecewise linear function (9) increases by 1. This is true both for any joint points corresponding to positive examples, and for any joint points corresponding to negative examples. Therefore the derivative of the piecewise linear function (9) is 0 in the interval $[(y_i - f_i)_{n_+}, (y_i - f_i)_{n_+ + 1}]$, and (9) is minimized by any point in this interval. The minimizer is unique if and only if $(y_i - f_i)_{n_+} = (y_i - f_i)_{n_+ + 1}$.

References

[1] Bartlett, P. L., and Shawe-Taylor, J. (1999), "Generalization Performance of Support Vector Machines and Other Pattern Classifiers," in *Advances in Kernel Methods — Support Vector Learning* eds. B. Schölkopf, C. J. C. Burges, and A. J. Smola, Cambridge, MA. MIT Press.

[2] Boser, B. E., Guyon, I. M., and Vapnik, V. (1992), "A Training Algorithm for Optimal Margin Classifiers," in *Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh.

[3] Breiman, L. (1998), "Arcing Classifiers," *Annals of Statistics*, 26, 801-824.

[4] Breiman, L. (1999), "Prediction Games and Arcing Algorithms," *Neural Computation*, 11, 1493-1517.

[5] Bühlmann, P., and Yu, B. (2001), "Boosting with the L_2-Loss: Regression & Classification," Technical report, Department of Statistics, University of California, Berkeley.

[6] Burges, C. J. C. (1998), "A Tutorial on Support Vector Machines for Pattern Recognition," *Knowledge Discovery and Data Mining*, 2(2).

[7] Cortes, C., and Vapnik, C. (1995), "Support Vector Networks," *Machine Learning*, 20, 273-297.

[8] Cox, D., and O'Sullivan, F. (1990), "Asymptotic Analysis of Penalized Likelihood and Related Estimators," *Annals of Statistics*, 18, 1676-1695.

[9] Cristianini, N., and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press.

[10] Devroye, L., Györfi, L., and Lugosi, G. (1996), *A Probability Theory of Pattern Recognition*, Springer - Verlag, New York.

[11] Duan, N., and Li, K. (1989) "Regression Analysis Under Link Violation," *Annals of Statistics*, 17, 1009-1052.

[12] Evgeniou, T., Pontil, M., and Poggio, T. (1999), "A Unified Framework for Regularization Networks and Support Vector Machines," Technical Report, M.I.T. Artificial Intelligence Laboratory and Center for Biological and Computational Learning Department of Brain and Cognitive Sciences.

[13] Freund, Y. (1995), "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, 121, 256-285.

[14] Freund, Y., and Schapire, R.E. (1996), "Experiments with a New Boosting Algorithm," in *Machine Learning: Proc. Thirteenth International Conference*, Morgan Kauffman, San Francisco.

[15] Friedman, J. H. (2001), "Greedy Function Approximation: a Gradient Boosting Machine," *Annals of Statistics*, 29, 1189-1232.

[16] Friedman, J. H. (1999), "Tutorial: Getting Started with MART in Splus," Available at www-stat.stanford.edu/ jhf/#reports.

[17] Friedman, J. H., Hastie, T., and Tibshirani, R. (2000), "Additive Logistic Regression: a Statistical View of Boosting (with discussion)," *Annals of Statistics*, 28, 337-407.

[18] Gu, C., and Qiu, C. (1993), "Smoothing Spline Density Estimation: Theory," *The Annals of Statistics*, 21, 217-234.

[19] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

[20] Jiang, W. (2001), "Some Theoretical Aspects of Boosting in the Presence of Noisy Data," Technical report, Department of Statistics, Northwestern University.

[21] Kimeldorf, G., and Wahba, G. (1971), "Some Results on Tchebycheffian Spline Functions," *J. Math. Analysis Appl.*, 33, 82-95.

[22] Lee, W. S., Bartlett, P. L., and Williamson, R. C. (1996), "Efficient Agnostic Learning of Neural Networks with Bounded Fan-in," *IEEE Transactions on Information Theory*, 42, 2118-2132.

[23] Lee, Y.-J., and Mangasarian, O. L. (2001), "SSVM: A Smooth Support Vector Machine for Classification," *Computational Optimization and Applications*, 20(1).

[24] Lin, Y. (1999), "Support Vector Machines and the Bayes Rule in Classification", to appear, *Data Mining and Knowledge Discovery.*

[25] Mammen, E., and Tsybakov, A. B. (1999), "Smooth Discrimination Analysis," *The Annals of Statistics*, 27, 1808-1829.

[26] Marron, J. S. (1983), Optimal Rates of Convergence to Bayes Risk in Nonparametric Discrimination. *The Annals of Statistics*, 11, 1142-1155.

[27] Mason, L., Bartlett, P. L., and Baxter, J. (1999), "Improved Generalization through Explicit Optimization of Margins," *Machine Leaning.*

[28] Mason, L., Baxter, J., Bartlett, P., and Frean, M. R. (2000), "Boosting Algorithms as Gradient Descent in Function Space," in *Neural Information Processing Systems*, 12, 512-518. MIT Press.

[29] Schapire, R. E. (1990), "The Strength of Weak Learnability," *Machine Learning*, 5, 197-227.

[30] Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998), "Boosting the Margin: a New Explanation for the Effectiveness of Voting Methods," *Annals of Statistics*, 26, 1651-1686.

[31] Schapire, R. E., and Singer, Y. (1998), "Improved Boosting Algorithms Using Confidence-rated Predictions," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*.

[32] Shen, X., and Wong, W. H. (1994), "Convergence Rate of Sieve Estimates," *Annals of Statistics*, 22, 580-615.

[33] Shen, X., Zhang, X., Tseng, G. C., and Wong, W. H. (2001), "Generalization machine," Technical report, department of statistics, Ohio State University.

[34] Silverman, B. W. (1982), "On the Estimation of a Probability Density Function by the Maximum Penalised Likelihood Method," *Annals of Statistics*, 10, 795-810.

[35] Silverman, B. W. (1984), "Spline Smoothing: the Equivalent Variable Kernel Method," *The Annals of Statistics*, 12, 898-916.

[36] Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer, New York.

[37] Wahba, G. (1999), "Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV," in *Advances in Kernel Methods - Support Vector Learning*, eds. Schölkopf, Burges and Smola, MIT Press.

[38] Wahba, G., Lin, Y. and Zhang, H. (2000), "GACV for Support Vector Machines, or , Another Way to Look at Margin-like Quantities," *Advances in Large Margin Classifiers*, eds. A.J. Smola, P.L. Bartlett, B. Scholkopf, and D. Schurmans. MIT Press.

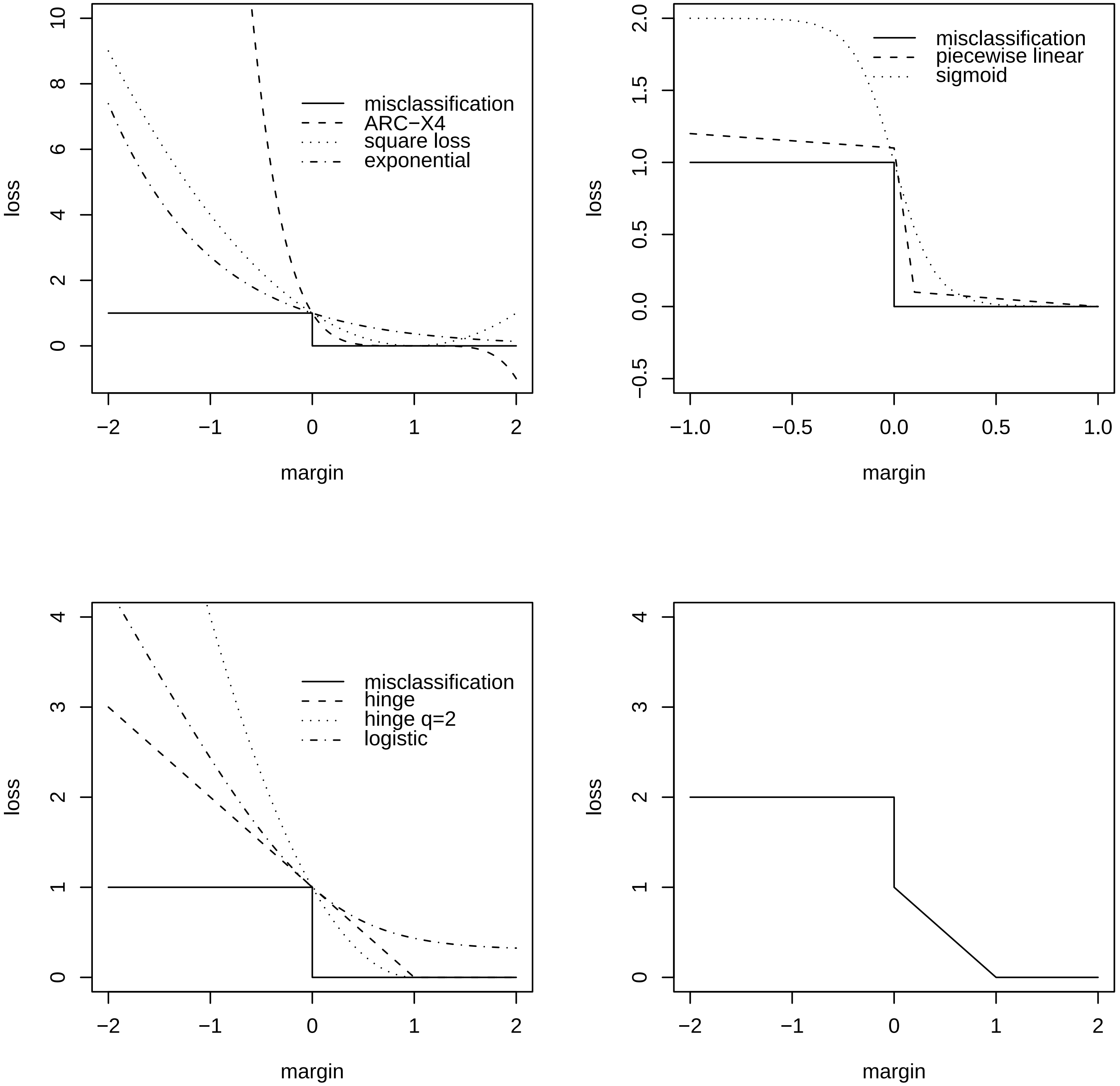

Figure 1: Examples of margin-based loss functions. Top left: The exponential loss, the square loss and the loss function in ARC-X4 [Breiman (1999)]. Top right: The normalized sigmoid loss and a piecewise linear loss. Bottom left: The hinge loss, the hinge loss with $q = 2$, and the logistic loss. Bottom right: The generalization machine loss.

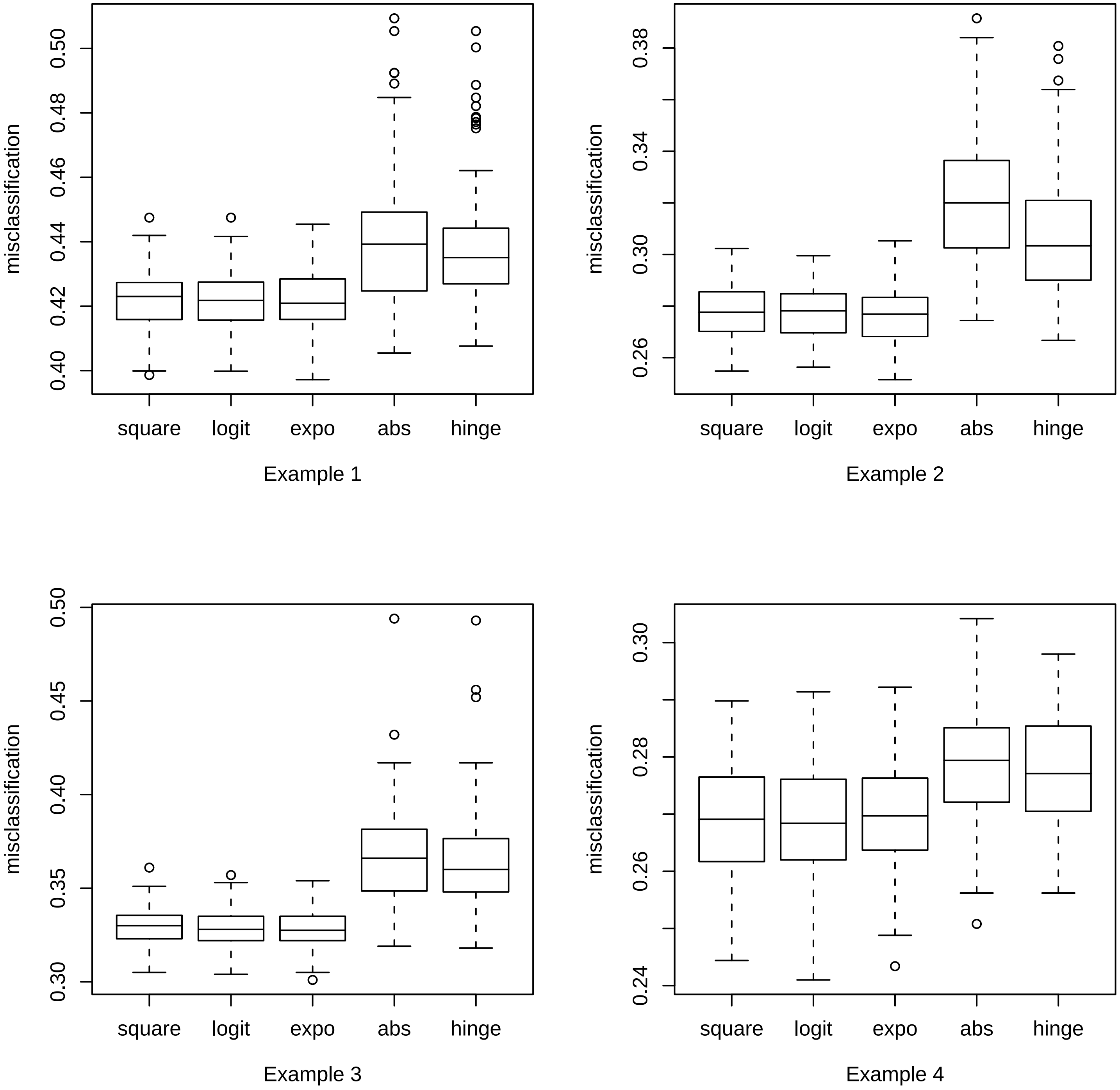

Figure 2: The generalization error of boosting procedures with the absolute deviation loss, the hinge loss, the square loss, the exponential loss and the logistic regression loss.

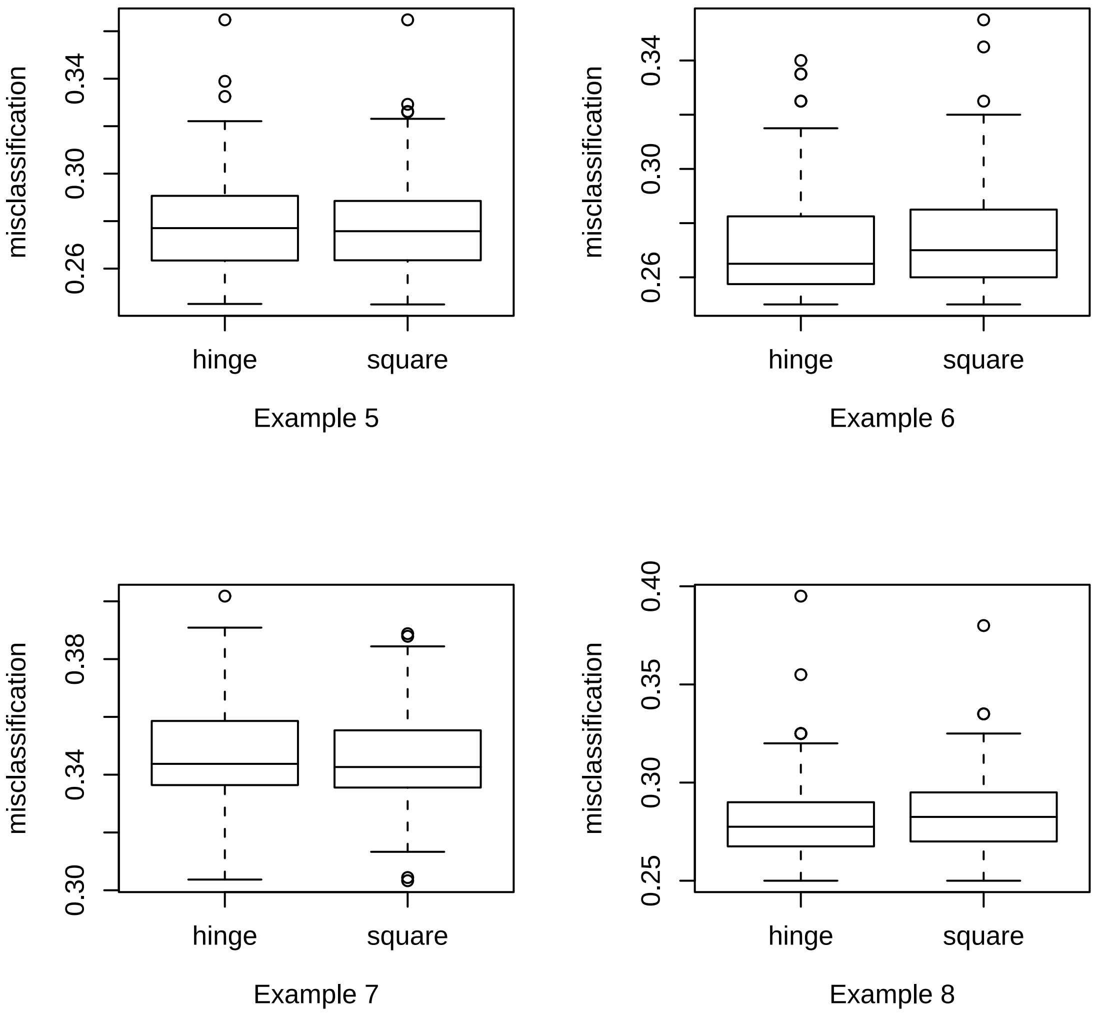

Figure 3: The generalization error of the method of regularization with the hinge loss and the square loss.