Margin Based Loss Functions in Classification and Related Topics

by

Yi Lin

1Key words: Classification, margin, the method of regularization, boosting, Fisher consistency, consistency
Margin based loss functions in classification and related topics

Yi Lin

Abstract

There has been a host of classification methods proposed in machine learning that are based on the minimization of loss functions depending only on the margin. Examples include the support vector machine and many variants of the boosting procedure. We consider the margin based loss functions from a statistical point of view. We first motivate a notion of Fisher consistency for classification, and establish the Fisher consistency for a very general class of margin based loss functions. This class contains almost all the margin based loss functions proposed in the machine learning literature as special cases. We then show that there is a close connection between Fisher consistency and consistency, and demonstrate that Fisher consistency often leads to consistency and rate of convergence (to the Bayes optimal risk) results under general conditions. The effect of choosing the smoothing parameter by the misclassification loss is also considered. Simulations are carried out to compare some important margin based loss functions that are commonly used.

1 Introduction

We consider the common binary classification problem studied extensively in statistics and machine learning. Suppose we are given a training set of observations \( D_n = \{(x_i, y_i), i = 1, ..., n\} \), assumed to be i.i.d. realizations of a random pair \((X, Y)\). Here \(X\) is the explanatory or input vector and \(Y\) is the class label that takes values in \(-1, 1\). A classification rule \(\phi\) is a mapping from the input space to \(-1, 1\). The generalization error of a classification rule \(\phi\), denoted by \(R(\phi)\) is the expected misclassification rate \(P\{\phi(X) \neq Y\}\). This is usually the evaluating criterion for the performance of classifiers. If we knew the underlying distribution, we could derive the optimal classification rule with the smallest generalization error. This optimal rule is called the Bayes optimal rule, and will be denoted as \(\phi^*\). The generalization error of the Bayes optimal rule is called the Bayes optimal risk. Denote \(p(x) = P(Y = 1|X = x)\), the Bayes optimal rule is \(\text{sign}[p(x) - 1/2]\), or equivalently, \(\text{sign}[g(x)]\), where \(g(x) = \log[p(x)/(1 - p(x))]\) is the logistic function. It does not matter how you classify a subject with an input vector \(x\) if \(p(x) = 1/2\), as the misclassification rate is always the same. To avoid technicality, we assume throughout this paper \(p(x) \neq 1/2\), \(a.s.\).

In practice we do not know the underlying probability distribution, and need to learn a good classification rule from the training sample. Traditional statistical classification methods, such as the discriminant analysis and the logistic regression, proceed by estimating \(p(x)\), or \(g(x)\). Once an estimate \(\hat{p}(x)\) or \(\hat{g}(x)\) is obtained, the classification rule is taken to be \(\text{sign}[\hat{p}(x) - 1/2]\), or \(\text{sign}[\hat{g}(x)]\). In the logistic regression, the conditional negative log-likelihood is used as the loss function in the estimation. It is well known that this loss function can be written as \(\log[1 + e^{-2y\hat{g}(x)}]\) when the values of \(y\) is 1 or \(-1\).

In recent years, there have been a host of classification methods proposed in the machine learning literature that are based on loss functions that are different from the negative log-likelihood. Almost all of these loss functions are based on the concept of the margin. For
a specific classification function $f$, if we use the classification rule $\text{sign}[f(x)]$, the margin of a subject $(x, y)$ is defined as $yf(x)$. It is easy to see that the subject is classified correctly by the classification rule $\text{sign}[f(x)]$ if and only if the 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 $\text{sign}[f(x)]$ can be written as a function of the margin: $[-yf(x)]_+$, where the function $(\cdot)_+$ takes value 1 for positive arguments and 0 for negative arguments. Intuitively, the misclassification loss should be used in the training of a classifier, since the performances of classifiers are measured according to this loss function. However, the function $(\cdot)_+$ is not convex and not continuous, and causes problems for computation. Researchers in machine learning and statistics have proposed many other margin based loss functions for the training of classifiers. Examples include the hinge loss used in the support vector machine and many other loss functions used in the variants of boosting. See Figure 1 for a graphical illustration of these loss functions. A brief overview of these loss functions is given in section 2.

In the machine learning literature, the 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 reasonable performance statistically, since minimizing an upper bound is quite different from minimizing the original function, especially since these upper bounds are not very tight.

Some theoretical results on procedures based on margin based loss functions have been derived in the computational learning literature. Examples include Vapnik (1995), Freund and Schapire (1996), Shawe-Taylor and Cristianini (1998), and Mason, Baxter, Bartlett, and Frean (2000). Such results typically give upper bounds on the expected misclassification rate of the procedures under study. The results have the advantage of being very general. However, in many situations, the upper bounds produced are very loose, and even when the sample size goes to infinity, the results only give an upper bound of the generalization error.

An alternative to deriving upper bounds of the expected misclassification rate is to compare the expected misclassification rate of the classifier with the Bayes optimal risk. Such comparison is desirable since the magnitude of the expected misclassification rate alone does not give a complete picture of the performance of the classifier. The same expected misclassification rate 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. A sequence of classifiers is said to be consistent if their misclassification errors convergence to the Bayes optimal risk. More formally, Let $\phi_n$ be a sequence of classifiers based on the data $D_n$. Let

$$R(\phi_n) = P\{\phi_n(X, D_n) \neq Y|D_n\}.$$

be the generalization error of $\phi_n$. Then $\phi_n$ is consistent for a family of distributions of $(X, Y)$, if for every member of the family, we have $R(\phi_n) \longrightarrow R(\phi^*)$ in probability. Devroye, Györfi, and Lugosi (1996) gave a comprehensive coverage of the results on the (universal) consistency of many kernel, nearest neighbor, and partitioning type procedures. 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 of the decision boundary. The optimal rate they obtained is much fast. They further showed that such rate can be achieved by decision rules based on minimization of empirical risk over the whole class of sets and 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 the classifiers based on general margin based loss functions. In section 3, we motivate a notion of Fisher consistency for classification. Fisher consistency can serve as a quick check whether a given loss function is expected to perform reasonably well. We then established the Fisher consistency for a very general class of margin based loss functions. In section 4, we formalize the connection between Fisher consistency and 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.

The performance of nonparametric classification methods depends on the tuning of the smoothing parameter(s). One common practice is to reserve an independent dataset for the tuning. While the training is usually based on an easy-to-handle margin based loss function, the tuning is based on the misclassification rate on the tuning data. Therefore different loss functions are involved in the training and the tuning of a classifier. The effect of such combination is discussed in section 5.

In section 6 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 upper bound of the misclassification loss. Some simulations are carried out to compare the performance of several commonly used convex margin-based loss functions (and their combinations). A summary is given in section 7. Proofs of some of the results are given in section 8.

2 Examples of margin based loss functions

Many recently proposed classification procedures involving loss functions can be put into two categories: the method of regularization and the boosting type of procedures. We concentrate on the loss functions for binary classification that have been used in conjunction with these two methods.

1. For a general loss function \( \ell(y, f) \), the method of regularization solves

\[
\arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \ell[y_i, f(x_i)] + \lambda J(f),
\]

where \( F \) is a reproducing kernel Hilbert space of functions, \( J(\cdot) \) is a penalty (regularization) functional, often a semi-norm in \( F \). The penalized logistic regression is one example of the method of regularization with the (logistic) loss function \( \log[1+e^{-2yf(x)}] \).

The support vector machine is another example. The support vector machine was first proposed in Boser, Guyon, and Vapnik (1992) and has received considerable interest
in the machine learning community since then. The hard margin linear support vector machine [Boser, Guyon, and Vapnik (1992)] simply finds the optimal separating hyperplane $\mathbf{x}_i \cdot \mathbf{w} + b = 0$ in the simple situation of linearly separable classes. This can be formulated as a quadratic programming problem: Minimize $\|\mathbf{w}\|^2/2$ subject to

$$\begin{align*}
\mathbf{x}_i \cdot \mathbf{w} + b &\geq +1 & \text{for } y_i = +1; \\
\mathbf{x}_i \cdot \mathbf{w} + b &\leq -1 & \text{for } y_i = -1;
\end{align*}$$

That is, maximize the distance between the two hyperplanes $\mathbf{w} \cdot \mathbf{x} + b = +1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$ under the condition that these two planes completely separate the positive and negative classes. In the more common case of overlapping classes, the above constraints can not be satisfied. In the soft margin support vector machine [Cortes and Vapnik (1995)] nonnegative slack variable are used to relax the constraints, and the problem becomes: Minimize $\|\mathbf{w}\|^2/2 + C(\sum_i \xi_i)$, subject to

$$\begin{align*}
\mathbf{x}_i \cdot \mathbf{w} + b &\geq +1 - \xi_i & \text{for } y_i = +1; \\
\mathbf{x}_i \cdot \mathbf{w} + b &\leq -1 + \xi_i & \text{for } y_i = -1; \\
\xi_i &\geq 0, \forall i.
\end{align*}$$

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 expected misclassification rate of the SVM based on the VC dimension (or the fat shattering dimension).

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 (1998), Evgeniou, Pontil, and Poggio (1999). The corresponding loss function is $(1 - yf(\mathbf{x}))_+$. 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 $\hat{f}$ is found then the classification is based upon the sign of $\hat{f}(\mathbf{x})$. 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, top 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 new 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, top 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.
2. Boosting was proposed in the Computational Learning Theory literature. Schapire (1990) developed the first simple boosting procedure in the PAC-learning framework. Schapire showed that a weak learner could always improve its performance by training two additional classifiers on filtered versions of the input data stream. Freund (1995) proposed combining many classifiers simultaneously, and improved the performance of the simple boosting algorithm of Schapire. Freund and Schapire (1996) introduced the popular Adaboost procedure. Friedman, Hastie, and Tibshirani (2000) showed that adaboost can be viewed as a stage-wise additive fitting with the exponential loss function \( \exp[-\gamma f(x)] \). This loss function appeared in Schapire and Singer (1998), and was motivated as an upper bound on the misclassification error. Friedman, Hastie, and Tibshirani (2000) further proposed a boosting procedure based on the negative log-likelihood \( \log(1 + e^{-2\gamma f(x)}) \).

It has been noted that boosting can be seen as a gradient descent algorithm in the function space. See Breiman (1998) and Friedman (1999a). Friedman (1999a) introduced a gradient boosting method MART for regression and classification. Mason, Baxter, Bartlett, and Frean (1999) proposed margin-boost 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 \([1 - \gamma f(x)]^5\) used in the ARC-X4 procedure [Breiman (1998)], and \([1 - \gamma f(x)]^2\) used in constructive NN algorithm [Lee et al. (1996)] (this is actually the same as the common \( L_2 \) loss \((y - f(x))^2\)). See Figure 1, middle left panel, for the graphs of these two loss functions and the exponential loss function.

Mason, Bartlett, and Baxter (1999) introduced a notion of \( B \)-admissibility of margin based loss functions. A family \( \{C_N : N = 1, 2, \ldots\} \) 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 \( \Phi_N : [-1, 1] \rightarrow I \) 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 using the margin based loss functions \( C_N = E[\Phi_N(Z)] \) with \( \Phi_N(\alpha) = [N^{-1/2}/2 - \alpha]_* + c \). Here \( Z \) is as in the definition of \( B \)-admissibility, and \( c \) is some constant. See Figure 2, right panel, for some members of this family. For large values of \( N \) the function \( C_N(\alpha) \) is close to \([\alpha]_* \). For computational consideration, they proposed an approximation to \( C_N(\alpha) \):

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

with \( \gamma \) fixed at 0.1. Mason, Baxter, Bartlett, and Frean (1999) further proposed using the so called normalized sigmoid cost function \( 1 - \tanh(\lambda y f(x)) \) as an approximation
of $C_N$. See Figure 1, bottom left panel, for a comparison of $C_\theta$ and the normalized sigmoid cost functions.

3 Fisher consistency for classification

In this section, we look at the classification procedures in infinite population space and introduce a notion of Fisher consistency for classification. One useful way to understand statistical procedures is to look at them in the population space. 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 $f(z, \theta_0)$. A M-type estimate is the solution to

$$\arg\min_{\theta} 1/n \sum_{i=1}^{n} \ell(z_i, \theta)$$

for some loss function $\ell$. It is clear that (2) goes to $E\ell(Z, \theta)$. The estimation procedure is Fisher consistent if the minimizer of $E\ell(Z, \theta)$ is the true parameter $\theta_0$. Fisher consistency essentially means that the procedure has no estimation bias in the limit, and usually implies strong consistency under suitable regularity conditions. One obvious example is the maximum likelihood estimation. See Duan and Li (1989) for another interesting application of Fisher consistency. This population space view has also been used to study nonparametric statistical procedures and provided much useful insight. Examples include Breiman and Friedman (1985), Hastie and Tibshirani (1990), and Breiman (2000). In nonparametric function estimation, Fisher consistency means that the population minimizer of the estimation criterion is the function to be estimated.

In the context of classification, one can define the Fisher consistency to be that the procedure in the population space achieves the Bayes optimal risk. In the binary classification problem, if the classification procedure first produce a function $f$, and then classifies according to the sign of $f$, then this notion of Fisher consistency is equivalent to that the function $f$ produced in the population space has the same sign function as $\text{sign}[p(x) - 1/2]$. For a classification procedure based on a loss function to be Fisher consistent for classification the population minimizer of the loss function should have the same sign function as $\text{sign}[p(x) - 1/2]$.

Example 3.1 The method of regularization (1). Cox and O'Sullivan (1990), extending a line of argument first introduced in Silverman (1981) in the context of log-density estimation, provided a general abstract framework for studying the asymptotic properties of such methods. In general, let the minimizer of $E\ell(Y, f(X))$ be denoted by $\hat{f}$, then under the condition that $\hat{f}$ is in the reproducing kernel Hilbert space $F$ under study, and some other regularity conditions, the solution to (1) goes to $\hat{f}$ with a certain rate. If the loss function $\ell(y, f)$ is Fisher consistent, then $\hat{f}$ have the same sign as $\text{sign}[p(x) - 1/2]$. Then we can expect the method of regularization estimate approaches the Bayes optimal rule.
Example 3.2 The sieve method. The sieve method based on a general loss function \( \ell(y, f) \) solves

\[
\arg\min_{f \in F_n} \frac{1}{n} \sum_{i=1}^{n} \ell[y_i, f(x_i)],
\]

where \( F_n \) is a finite dimensional function space (approximating space), with the dimension of \( F_n \) going up with the sample size \( n \). There is a large body of literature on the sieve method in nonparametric function estimation. In general, the target of the estimation is the population minimizer of \( \ell[y, f(x)] \). Therefore, for the sieve method to perform well for classification, the minimizer of \( E\ell[Y, f(X)] \) should have the same sign as \( \text{sign}(p(x) - 1/2) \). A popular special case of the sieve method takes the approximating space to be the span of polynomial spline functions and their tensor products. See Stone et al (1997), Huang (1998) for the asymptotic theory of these methods. The popular MARS algorithm can be seen as an adaptive version of such method. Boosting with trees is related to the sieve method with the approximating space being the linear space spanned by the tree functions. However, boosting is a stage-wise, greedy method, and its property can not be explained only by the standard theory of the sieve method.

In the following we establish the Fisher consistency for a general class of margin based loss functions. 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 \( \bar{f}(x) \) be any global minimizer of \( E\ell[Y, f(X)] \) for some function \( V \) satisfying assumptions 1 and 2. Then we have \( \text{sign}(\bar{f}) = \text{sign}(p - 1/2), \) a.s..

Notice we do not require the global minimizer to be unique. The theorem says for a function \( V \) satisfying assumptions 1 and 2, the loss function \( \ell(y f(x)) \) is Fisher consistent. Several special cases are known in the literature. It is well known that the population minimizer of the logistic loss is the logistic function \( g(x) = \log[p(x)/(1 - p(x))] \), which has the same sign as that of \( p(x) - 1/2 \). Friedman, Hastie, and Tibshirani (2000) showed that the population minimizer of the exponential loss is half of the logistic function. Lin (1999) showed that the population minimizer of the hinge loss is \( \text{sign}(p(x) - 1/2) \). (Technically, at those special points where \( p(x) \) is exactly 1 or -1, the minimizer is not unique. At such an \( x \) the minimizer can be \( c\{\text{sign}[p(x) - 1/2]\} \) for any \( c \geq 1 \).)

**Remark 3.1** The minimizer \( \bar{f} \) in the theorem is allowed to take on the values \( \infty \) or \( -\infty \). One example is the normalized sigmoid loss \( 1 - \tanh(\lambda y f(x)) \) used in Mason, Baxter, Bartlett, and Frean (2000). It can be checked that the population minimizer is \( \infty \) when \( p(x) > 1/2 \), and is \( -\infty \) when \( p(x) < 1/2 \).

**Remark 3.2** Assumption 2 is only used to guarantee that \( \bar{f}(x) \neq 0 \), a.s., therefore can be relaxed. For example, consider the misclassification risk \( E[-Y f(X)] \). It is minimized
by any function that has the same sign as $\text{sign}[p(x) - 1/2]$, 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 $p(x) > 1/2$, and is any number smaller than $-1$ when $p(x) < 1/2$. Thus the population minimizer has the same sign as that of $\text{sign}[p(x) - 1/2]$.

We can relax assumption 1 to Assumption 1' there exist 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 $\tilde{f}(x)$ be any global minimizer of $EV[Yf(X)]$ for some function $V$ satisfying assumptions 1' and 2. Then we have $\text{sign}(\tilde{f}) = \text{sign}(p - 1/2)$, a.s.

We can see assumption 1 can be seen as a special case of assumption 1' with $a = \infty$. An example of loss functions covered by Theorem 3.2 but not Theorem 3.1 is given in Figure 1, bottom right panel. It says 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 consistent.

### 4 Consistency and Fisher consistency

In this section we formalize the connection between the Fisher consistency and the consistency of classification procedures.

**Lemma 4.1** Let $\tilde{f}$ be the minimizer of $EV[Yf(X)]$ for some function $V$ satisfying assumptions 1 [or 1'], 2, and $V''(z) > 0$, $\forall z$. Then for any function $\tilde{f}$,

$$R(\tilde{f}) - R^* \leq c \int |\tilde{f}(x) - \tilde{f}(x)|d(x)dx \leq c\{E(\tilde{f} - \tilde{f})^2\}^{1/2}d(x)dx,$$

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

For the special case in which $\tilde{f}(x)$ is $[p(x) - 1/2]$, this lemma is well known. See Theorem 2.2 of Devroye, Györfi, and Lugosi (1996) (cited below for convenience), and the references listed there.

**Lemma 4.2 (Theorem 2.2 of Devroye, Györfi, and Lugosi (1996))** For any function $\tilde{f}$,

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

(3)

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 equality

$$|p(x) - 1/2| \leq C|\tilde{f}(x)|$$

for some positive $C$ that does not depend on $x$. 8
Remark 4.1 The condition $V''(z) > 0$, $\forall z$, in Lemma 4.1 can be relaxed. The key of the proof is (4). For the hinge loss, we have $\hat{f}(x) = \text{sign}[p(x) - 1/2]$, therefore (4) is trivially satisfied, and result of Lemma 4.1 applies. Other examples include the generalization machine loss and the normalized sigmoid loss.

Lemma 4.1 says that the consistency of $\hat{f}$ in estimating $f$ implies the consistency of the resulting classification rule $\text{sign}[\hat{f}(x)]$ for procedures with margin-based loss functions. The convergence of the generalization error to the Bayes optimal risk can be studied through studying the convergence in function estimation. This provides a general tool to establish consistency and rate of convergence results for classification rules. For example, considering a one dimensional problem ($d = 1$), assume $X$ is supported in $[0, 1]$ and its density is bounded away from 0 and $\infty$ in $[0, 1]$. We can have the following result.

Theorem 4.1 Suppose $V$ is a function satisfying assumptions 1 [1'], 2, $V''(z) > 0$, $\forall z$, and $V'''$ exists and is continuous. Assume that the conditional probability function $p(x)$ has absolute continuous first derivative and $\int_0^1 [p'(x)]^2 dx < \infty$. Consider the method of regularization (1) with $J(f) = \int_0^1 (f''')^2$. If $\lambda_n \sim n^{-r}$ for some $0 < r < 1$, then

$$R(\hat{f}) - R^* = O_p[\lambda + n^{-1} \lambda^{1/4}].$$

Remark 4.2 The method of regularization with $J(f) = \int_0^1 (f''')^2$ corresponds to setting the reproducing kernel Hilbert space $F$ in (1) to the second order Sobolev Hilbert space $H^2$. The $m$-th order Sobolev Hilbert space $H^m$ is defined as

$$H^m = \{ f | f^{(\nu)} \text{ abs. cont., } \nu = 0, 1, ..., m - 1; f^{(m)} \in L_2 \}.$$

The reproducing kernel of $H^m$ is given in Wahba (1990). Results similar to that in Theorem 4.1 can be obtained for the method of regularization with other reproducing kernel Hilbert spaces and reproducing kernels. For example, for multivariate problems ($d > 1$), the tensor product space $\otimes^d H^m$ is often used as the function space $F$, the corresponding reproducing kernel of such function space is simply the product of the reproducing kernel of the univariate function spaces $H^m$. For such reproducing kernels, the rate corresponding to that in Theorem 4.1 is $O_p \left[ \lambda + n^{-1} \lambda^{-1/2m} \left( \log \frac{1}{\lambda} \right)^{d-1} \right]$. Another commonly studied reproducing kernel Hilbert space is the $m$-th order Sobolev Hilbert space of $d$ dimensional functions $H^m([0, 1]^d)$ with $m > d/2$. The corresponding rate in Theorem 4.1 is $[\lambda + n^{-1} \lambda^{d/(2m)}]$. All of the above mentioned rates are based on the assumption that the population minimizer $\tilde{f}$ is in the function space $F$.

Remark 4.3 Results similar to Theorem 4.1 can be obtained for the sieve method with the approximating space being spanned by polynomial splines and their tensor products. The results in Stone et al (1997) and Huang (1998) are established for negative log-likelihood loss function, but the proofs can be applied to function estimation problems with other strictly convex loss functions. Lemma 4.1 can then be used to translate the results into classification.
Notice that the above discussion does not apply to the support vector machine. The main difficulty there is that the minimizer of $E[1 - Yf(X)]_+$ is $\text{sign}[\rho(x) - 1/2]$, which is most often discontinuous, and does not belong to any reproducing kernel Hilbert space. Thus the usual machinery for deriving asymptotic results for the method of regularization breaks down. Lin (2000) derived some rate of convergence results for the support vector machine in one dimension under general conditions.

5 Tuning set

The choice of the smoothing parameter is crucial for the success of any nonparametric method. In the asymptotic results in the nonparametric statistics literature, the smoothing parameter is usually assumed to be going to 0 at a rate that is not too slow, and not too fast. In practice, it is almost impossible to validate such conditions.

The simplest and most common approach in practice for the tuning of smoothing parameters is to use an independent tuning set. The smoothing parameters are chosen to give the smallest misclassification rate on the tuning set. Let the classification rule obtained with the smoothing parameter $\lambda$ be denoted by $f_\lambda$ and the corresponding classification rule is $\phi_\lambda = \text{sign}[f_\lambda]$. Let the tuning set be of size $m$. Let $C_k = \{\phi_\lambda : \lambda = 2^{-k}, ..., 1, 2, ..., 2^k\}$. Choose the classifier $\phi_\hat{\lambda}$ in $C_k$ with the smallest misclassification error on the tuning set. This is a formulation considered in Devroye, Györfi, and Lugosi (1996). They showed that

$$P\{R(\phi_\hat{\lambda}) - \inf_{\phi \in C_k} R(\phi) > \epsilon | D_n\} \leq (4k + 2)e^{-m\epsilon^2/2}. $$

If the classification rule $\phi_\lambda$ is consistent as $\lambda$ goes to the 0 at some polynomial rate of $n$, then $C_k$ contains a subsequence of consistent rules if $n \to \infty$, and $k \to \infty$. To ensure consistency of the tuned classifier, we need $m \to \infty$, and $\log(k) = o(m)$.

To establish results for tuning with an infinite number of possibilities, it is necessary to establish the VC dimension or entropy number of the set of possible tuning values. We will not discuss that here.

What is of interest here is that while the training of the classifier is usually based on loss functions different from the misclassification loss, the tuning is based on the misclassification loss. This has a direct consequence: the classifiers picked by the tuning may not go to the target function of the training (that is, the population minimizer of the training loss), though they are still consistent with respect to the misclassification loss.

One way to look at this is to consider the smoothing parameter $\lambda$ as the parameter for the family of parametric models built by the training process. 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. 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.
6 Some comparisons

As shown in the earlier sections, there are many consistent classification rules based on margin based loss functions. It is of practical interest to compare their performances. The most direct loss function is the misclassification loss, as it is often the final evaluating criterion. However, it is non-convex and non-continuous and is very hard to be used in training. The sigmoid like loss functions are close approximations to the misclassification loss. They are continuous (often smooth), but not convex. Working with them is possible, but still very hard. One major difficulty is the existence of multiple local minimizer.

In the following we concentrate on convex loss functions. Convex loss functions 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. The logistic loss is rooted in the likelihood theory central to the statistical methodology, and intuitively should be the first choice when estimating the conditional probability or the logistic function. However, it is not clear whether it should be the first choice for classification. As remarked in section 5, classification is different from the estimation of the target function. The hinge loss has the nice property that it often leads to sparse solutions as in the support vector machine.

Very often the margin based loss functions are motivated as upper bounds or approximations of the misclassification loss. However, the notion of being an upper bound of the misclassification loss is not well defined as it is possible to make any function that is bounded below an upper bound of the misclassification loss by adding a large constant to it. In the following we introduce a notion of equivalence between loss functions, which helps compare different loss functions as upper bounds of the misclassification rate. Two loss functions are equivalent for a classification procedure if the procedure based on the two losses give identical classifiers. For the method of regularization and the sieve method, 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 $F$, and denote the solution to (1) by $\hat{f}$. Then the solution to (1) with $\ell$ replaced by $a\ell(y, bf) + c$ and $\lambda$ replaced by $\lambda ab^2$ is $\hat{f}/b$. Therefore the two solutions have the same sign, and the resulting classifiers are identical.

**Proposition 6.1** 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 $W(y, f)$ is equivalent to the hinge loss. The equal sign hold everywhere only when $V$ is equivalent to the hinge loss.

Proof: Let $U$ be the tangent line of $V$ at $(0, V(0))$. Then $U \leq V$ 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
$W \leq V$ as $W$ is an upper bound of the misclassification loss. It is easy to see that $W$ is equivalent to the hinge loss.

Another way to look at this is that in order to compare the different equivalent classes of convex loss functions as upper bounds of the misclassification loss, we can take a representative from each class. For any convex function that satisfies condition 1 (or 1'), and 2, that is bounded from below, it is easy to see that there is a member $V$ of its equivalent class, such that $V(0) = 1$, $V'(0) = -1$, and it is an upper bound of the misclassification loss. We have

**Proposition 6.2** The hinge loss is the closest convex upper bound of the misclassification loss that satisfies $V(0) = 1$, $V'(0) = -1$.

**Proof:** Consider the tangent at $z = 0$.

The proposition establishes the hinge loss as the tightest convex upper bound of the misclassification loss. This gives the hinge loss an intuitive appeal. However, whether this translates into advantage in terms of classification performance is not clear and deserve further study.

In the following we will compare the classification performance of classification procedures based on different loss functions through a simple simulation study. We first compare the loss functions in the framework of the method of regularization, and then in the framework of the boosting procedure.

For the method of regularization we consider the penalized least square method, the penalized logistic regression method, and the support vector machine. All three methods are commonly used in practice. They are related to the square loss, the logistic loss, and the hinge loss, respectively. As we want to focus on the different loss functions, we keep other aspects such as the choice of tuning parameter(s) as simple as possible. The simulation is carried out for one dimensional problems. The simulations only serve as the first step for comparing the loss functions in terms of classification efficiency. The hope is that more comprehensive comparisons can be made later by us and others.

We take the reproducing kernel Hilbert space in (1) to be $H^2$. The corresponding reproducing kernel is the cubic spline kernel. This choice is made by convenience. There is only one tuning parameter in this case, so we can save some effort in finding the minimizing tuning parameter. Furthermore, the cubic spline kernel is commonly used in the penalized least square method, and the penalized logistic regression. There are existing statistical software that implement the penalized least square and penalized logistic regression with the cubic spline kernel. The support vector machine is usually used with the Gaussian kernel or polynomial kernel, but can be used with any other kernels. For comparison, we use the cubic spline kernel in the support vector machine as well. The cubic spline kernel is of the form

$$K(s, t) = 1 + k_1(s)k_1(t) + k_2(s)k_2(t) - k_4(|s - t|),$$

where $k_1(\cdot) = \cdot - 0.5$, $k_2 = (k_1^2 - 1/12)/2$, and $k_4 = (k_4^4 - k_4^2/2 + 7/240)/24$. See Wahba (1990), Gu (1993).

In the cases of logistic loss and the square loss, it is common practice not to penalize linear functions. Using the cubic spline kernel in (1) then results in the cubic spline procedure commonly used in the nonparametric statistics: (1) with $J(f) = \int_0^1 [f''(x)]^2dx$. The common practice of the SVM only leaves the constant unpenalized.
We consider six different conditional probability functions \( p(x) = P(Y = 1|X = x) \). The first three are smooth functions:

\[
p_1(x) = \frac{\exp(2\sin(3\pi x^2) + x - 0.5)}{1 + \exp(2\sin(3\pi x^2) + x - 0.5)}
\]

\[
p_2(x) = \frac{[\sin(3\pi x^2) + 1 + x]/3}{5 + 1}
\]

The third function is chosen to be close to \( 1/2 \), hence the classification problem is hard. The last three functions are (discontinuous) piecewise constant functions. The fourth function is \( p_4(x) = [\text{sign}(\sin(3\pi x^2))]/2 + 1]/2 + 0.2 \). The fifth function \( p_5(x) \) is \( 0.1, x \leq 0.2; 0.7, 0.2 < x \leq 0.6; 0.25, 0.6 < x \leq 0.8; 0.9, x > 0.8 \). The sixth function is similar, but with more pieces. The graphs for all six functions are shown in Figure 2.

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. For each conditional probability function and each loss function we run the simulation one hundred times.

The purpose of the simulation is to show how close the parametric family built by the different training procedure is to the truth. Since we know the true conditional probability functions, to eliminate randomness in the tuning we tune the smoothing parameter with the true expected misclassification loss. That is, we find the tuning parameter \( \lambda \) that has the smallest \( E[Y f_\lambda(X)]_* \). The evaluation of \( E[Y f_\lambda(X)]_* \) involves an integral that is sometimes hard to compute, so we actually use a discrete approximation of \( E[Y f_\lambda(X)]_* \):

\[
\frac{1}{n} \sum_{i=1}^{n} E\{[Y f_\lambda(X)]_*|X = x_i\} = \frac{1}{n} \sum_{i=1}^{n} [1 - p(x_i)][(\text{sign}(f_\lambda(x_i)) + 1)/2 + p(x_i)[1 - \text{sign}(f_\lambda(x_i)) + 1]/2.
\]

Here \( x_i \)'s are the realized \( x \) values. Notice this is not the the resubstitution error, and is a good approximation of the expected generalization error. We search for the tuning parameter among all the numbers of the form \( 2^j, j \in Z \).

The penalized least square regression and the penalized logistic regression are done in Splus, so that we can take advantage of the existing smooth.spline function. The support vector machine is done in Matlab, to take advantage of the existing quadratic programming routine quadprog.

The result is summarized in Figure 3. In all of the six examples, the support vector machine does the best. In five out of the six examples, the penalized logistic regression does better than the penalized least square regression, though the differences are very small. However, the result shown here is not conclusive: when t-test are carried out to compare the performance of the SVM and the penalized logistic regression, the difference is not significant in any of the six examples.

A second set of comparison is done for boosting procedures based on different convex losses. Boosting procedures can be viewed as an implementation of the gradient descent algorithm in the function space. Several authors have developed boosting algorithm based on
different loss functions. Friedman (1999a, b) treated regression and classification in a unified framework and introduced the software MART. Mason et al proposed anyboost algorithms for classification. The loss functions used in the later paper is typically not convex. Here we will concentrate on the MART algorithms. We compare the performance of the MART based on five different loss functions: the squared loss, the logistic loss, the absolute deviation loss, the exponential loss, and the hinge loss. All of the five losses can be seen as functions of the margin, and satisfy assumptions 1 and 2.

The MART is a boosting algorithm with regression trees as the base learner. It is an iterative algorithm. Let the fitted function at the $m$-th iteration be $F_m$. Initializing with a constant function

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

(6)

each iteration of the algorithm can be described as:

1. Calculate current residual defined as

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

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

3. For each terminal node, find

$$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$$

4. Update the current fit:

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

Here the constant $r$ in the last step is a regularizing factor. It is possible to fit the tree in step 2 with loss functions other than the squared loss, especially $L(y, f)$. MART chooses to use the squared loss to induce the tree because it is much faster to search for splits with squared loss than other losses during the tree building process.

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 and step 3 of MART algorithm has been worked out in the case of the square loss, the logistic loss, the absolute deviation loss, and Huber's loss function in Friedman (1999a). In the logistic loss case, there is no close form for the update in step 3, and a single Newton-Raphson step is used as an approximation.

Here we give the the corresponding results for the exponential loss and the hinge loss. Consider the problem

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

(7)
For the exponential loss, an explicit form of the solution to (7) can be found by setting the derivative of (7) to zero and solving for $\gamma$:

$$\gamma = \frac{\log\left(\sum_{y_i=1} e^{-f_i} \right) - \log\left(\sum_{y_i=-1} e^{f_i} \right)}{2}$$

However, this solution gives $-\infty$ or $\infty$ when $y_i$'s are all positive or all negative. Such situations are possible since the terminal nodes $R_{jm}$ can be pure. This causes trouble for computation. (The same problem would arise in the logistic loss case if we tried to use the exact solution). One possible remedy is to replace $\infty$ by a large finite number. Here we use a single step Newton-Raphson approximation in place of the exact solution. The approximate solution is

$$\gamma = \sum_{i=1}^{n} y_i e^{-y_i f_i} / \sum_{i=1}^{n} e^{-y_i f_i}.$$  

Notice this is a weighted average of the $y_i$'s.

For the hinge loss, we have

**Lemma 6.1** The solution to (7) with $L(y, f) = (1 - y f)_+$ is any number in the interval $[(y - f)_{(n+1)}, (y - f)_{(n+1)}]$, 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 each of our simulations we generate 500 uniform random points in $(0,1)^8$ for the input vector $X$. We then generate the $y$'s according to one of the following four conditional probability functions:

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

These functions are simple transformations of a regression function used as an example in the tutorial of MART [Friedman (1999b)]. The transformations are used to make sure that the conditional probability function $p(x)$ takes value in $(0,1)$. For each dataset generated we run the MART algorithm with the square loss, the logistic loss, the exponential loss, the absolute deviation loss, and the hinge loss. We fix the regularizing factor $r$ at 0.1. The simulations are run in R.

The results are summarized in Figure 4. In all four simulations, 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 conduct a Bonferroni pairwise comparison with paired t-test. The overall level of the test is 0.05. In the first and third simulation, the exponential loss, the logistic loss and the square loss have comparable performances, and they are significantly better than the performances of the hinge loss and the absolute deviation loss. 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.
7 Summary and discussion

In the context of binary classification, we define the Fisher consistency of a classification procedure based on a loss function $\ell(y, f(x))$ as that the minimizer of $E\ell(Y, f(X))$ has the same sign function as $\text{sign}[p(x) - 1/2]$. It is shown that a very general class of margin based loss functions are Fisher consistent. This gives one explanation why margin-based loss functions generally work well. The Fisher consistency often leads to the consistency 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.

It is of practical importance to compare the classification efficiency of the procedures based on different loss functions. While the logistic loss is related to the likelihood theory, hence should be very efficient for function estimation, the hinge loss is the tightest convex upper bound of the misclassification loss. It is not clear whether these translate into theoretical advantages in terms of classification performance.

We compared several commonly used convex margin-based loss functions through a simulation study. In the framework of the method of regularization, we compared the classification performance of the penalized least square regression, the penalized logistic regression, and the SVM in some simple one dimensional problems. The result is not conclusive, but seems to suggest that the support vector machine has some advantage. A lot more experience is needed before we can draw any definitive conclusion, and the conclusion may well be problem dependent. In the framework of boosting procedures, we compared the square loss, the logistic loss, the exponential loss, the hinge loss, and the absolute deviation loss. The simulations are carried out with 8-dimensional examples. The logistic loss, exponential loss, and the square loss seem to have comparable performance, and they perform better than the hinge loss and the absolute deviation loss. Part of the 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 the MART procedure. Another possible reason might be the greedy nature of the tree building process. Breiman et al (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 the hinge loss, being close to the misclassification loss, will not have any advantage. The simulations suggest that different loss functions may be suitable for different procedures.

8 Proofs

Proof of Theorem 3.1: For any fixed $x$, we define

$$A(z) = p(x)V(z) + [1 - p(x)]V(-z).$$

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

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

16
Now since \( \bar{f}(x) \) is a global minimizer of \( A(z) \), we have,
\[
0 \geq A[\bar{f}(x)] - A[-\bar{f}(x)] = [2p(x) - 1\{V[\bar{f}(x)] - V[-\bar{f}(x)]}\]
\]
Now if \( p(x) > 1/2 \), then \( V[\bar{f}(x)] - V[-\bar{f}(x)] \leq 0 \). Since \( \bar{f}(x) \neq 0 \), by assumption 1, we get \( \bar{f}(x) > 0 \). Therefore \( \text{sign}[\bar{f}(x)] = \text{sign}[p(x) - 1/2] \).

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

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, \( \bar{f}(x) \) is in \([-a, a]\). Now apply the same argument as in the proof of Theorem 3.1.

Proof of Lemma 4.1: For any fixed \( x \), we have
\[
EV[Yf(X)|X = x] = p(x)V[f(x)] + [1 - p(x)]V[-f(x)].
\] (8)

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

We now establish (4). This is obviously true when \( |\bar{f}(x)| = \infty \). Now we consider the case when \( \bar{f}(x) \) is finite. In this case, since \( \bar{f}(x) \) is the minimizer of (8), taking derivative at \( \bar{f}(x) \), we have,
\[
p(x)V'[-\bar{f}(x)] - [1 - p(x)]V'[-\bar{f}(x)] = 0.
\] (9)

From assumptions 1 [or 1'], and that \( V''(z) > 0, \forall z \), we know that \( V \) either has a minimizer at some \( 0 < a < \infty \), or it is monotone decreasing. In the former case the same argument as that in the proof of Theorem 3.2 leads to \( \bar{f}(x) \in [-a, a] \). Therefore in both cases we have \( V'[-\bar{f}(x)] \leq 0 \) and \( V'[-\bar{f}(x)] \leq 0 \). By the strict convexity of \( V \) the two equal signs can not hold at the same time since \( \bar{f}(x) \) is not 0 (as shown in the proof of Theorem 3.1). Therefore \( V'[-\bar{f}(x)] + V'[\bar{f}(x)] \neq 0 \), a.s., and we can solve (9) to get \( |p(x) - 1/2| = |B[\bar{f}(x)]| \), where \( B(\cdot) \) is defined as
\[
B(z) = \frac{1}{2} \frac{V''(-z) - V'(z)}{V'(-z) + V'(z)}.
\]

It is easy to see that \( B(z)/z \rightarrow \frac{V''(0)}{2V'(0)} \) as \( z \rightarrow 0 \). Therefore there exists \( \delta > 0 \) and \( C > 0 \) only depending on \( V \) such that \( |B(z)| \leq C|z| \) for \( z \in [-\delta, \delta] \). Therefore if \( \bar{f}(x) \in [-\delta, \delta] \), then \( |p(x) - 1/2| = |B[\bar{f}(x)]| \leq C|\bar{f}(x)| \). On the other hand, if \( \bar{f}(x) \) is not in \([-\delta, \delta] \), then \( |p(x) - 1/2| \leq 1/2 \leq \frac{1}{2\delta} |\bar{f}(x)| \). So (4) is proved.

By Theorem 3.1, 2.2, and (3), we have
\[
R(\bar{f}) - R^*
\leq 2 \int [|p(x) - 1/2| 1_{\text{sign}[\bar{f}(x)] \neq \text{sign}[\bar{f}(x)]}] d(x) dx
\leq C \int [||\bar{f}(x)| - 1_{\text{sign}[\bar{f}(x)] \neq \text{sign}[\bar{f}(x)]}] d(x) dx
\leq C \int [||\bar{f}(x) - \bar{f}(x)| - 1_{\text{sign}[\bar{f}(x)] \neq \text{sign}[\bar{f}(x)]}] d(x) dx
\leq C \int |\bar{f}(x) - \bar{f}(x)| d(x) dx
\leq C \left\{ \int [||\bar{f}(x) - \bar{f}(x)||^2 d(x)] dx \right\}^{1/2}.
\]
Outline of the proof of Theorem 4.1: Let $\bar{f}$ be the minimizer of $EV(Y f(X))$. We first show that $\bar{f} \in H^2$, the second order Sobolev Hilbert space. From the conditions we see that $p(x) \in H^2$. From (9), we get

$$V'[\bar{f}] / V'([\bar{f}] = p/(1 - p).$$

Let $U(z) = V'(z) / V'(z)$, then $U$ is a strictly increasing function. Therefore $\bar{f} = U^{-1}[p/(1 - p)]$. From this and that $V'''$ is continuous it is easy to see that $\bar{f} \in H^2$. Now we apply the theory in Cox and O'Sullivan (1990), hereafter referred to as CO, by verifying the assumptions in CO. Let $U$ be an operator defined in $H^2$ by

$$(f, Ug)_{H^2} = \int f g d(x) dx$$

for $f, g \in H^2$. The assumptions A.1 and A.2 of CO follow immediately. Let $S_f(R, \alpha)$, $0 \leq \alpha < 1$ be the ball of radius $R$ in $H^{m\alpha}$ centered at $\bar{f}$. Denote $l_n(f) = 1/n \sum_{i=1}^n V[y_i f(x_i)]$, $l(f) = E[V(Y f(X))] = \int p V[f(x)] + (1 - p) V(-f) d(x) dx$. Direct calculation gives

$$Dl(f)g = \int [p V'(f) - (1 - p) V'(-f)] g d(x);$$

$$Dl_n(f)g = 1/n \sum_{i=1}^n y_i V'[y_i f(x_i)] g(x_i);$$

$$D^2l(f)gh = \int [p V''(f) + (1 - p) V''(-f)] gh d(x);$$

$$D^2l_n(f)gh = 1/n \sum_{i=1}^n V''[y_i f(x_i)] g(x_i) h(x_i);$$

$$D^3l(f)ghq = \int [p V'''(f) - (1 - p) V'''(-f)] ghq d(x);$$

$$D^3l_n(f)ghq = 1/n \sum_{i=1}^n y_i V'''[y_i f(x_i)] g(x_i) h(x_i) q(x_i).$$

The above derivatives are well defined and continuous in $H^{m\alpha}$. It is easy to check that there are strictly positive constants $c_R$ and $C_R$ such that for any $f \in S_f(R, \alpha)$, and $g \in H^2$,

$$c_R(g, Ug)_{H^2} \leq D^2l(f)gg \leq C_R(g, Ug)_{H^2}. $$

So $l(f)$ is strictly convex for $f \in S_f(R, \alpha)$, and the minimizer $\bar{f}$ is unique. Therefore assumption A.3 and A.4 of CO holds. The verification of assumptions A.5 and A.6 is tedious, but follows the same line of computation as the computation for logistic regression in CO. Theorem 3.1 and 3.2 of CO then yields the desired result.

Proof of Lemma 6.1: With the hinge loss, (7) is a piecewise linear function of $\gamma$ with change points $(y_i - f_i)$, $i = 1, 2, ... , n$. To the left of $(y_i - f_i)$, the smallest of the changing points, (7) is a linear function with derivative $-n_+$. Moving $\rho$ from left to right, every time a change point is passed, the derivative of the piecewise linear function increases by 1. This is true both for any change points corresponding to positive examples, and for any change
points corresponding to negative examples. Therefore the derivative of the piecewise linear function (7) is 0 in the interval \([(y_i - f_i)_{n+}, (y_i - f_i)_{n+1}^]\), and (7) is minimized by any point in this interval. The minimizer is unique if and only if \((y_i - f_i)_{n+} = (y_i - f_i)_{n+1}\).

Acknowledgement: The author thanks Professor Grace Wahba for stimulating discussions.

References


Figure 1: Examples of margin based loss functions. Top left: The hinge loss, the hinge loss with \( q = 2 \), and the logistic loss. Top right: The generalization machine loss. Middle left: The exponential loss, the square loss and the loss function in ARC-X4 [Breiman (1998)]. Middle right: The family \( C_N \) of loss functions proposed in Mason, Bartlett, and Baxter (1999). Bottom left: The normalized sigmoid loss and a piecewise linear loss. Bottom right: A margin based loss function that is covered by Theorem 3.2, but not Theorem 3.1.
Figure 2: Conditional probability functions $p(x)$ used in the simulation study.
Figure 3: The expected misclassification rate of the penalized logistic regression, penalized least squares regression, and the support vector machine on the six examples in our simulation.
Figure 4: The expected misclassification rate of boosting procedures with absolute deviation loss, hinge loss, square loss, exponential loss and logistic regression loss.