Robust Truncated Hinge Loss Support Vector Machines

Yichao WU and Yufeng LIU

The support vector machine (SVM) has been widely applied for classification problems in both machine learning and statistics. Despite its popularity, however, SVM has some drawbacks in certain situations. In particular, the SVM classifier can be very sensitive to outliers in the training sample. Moreover, the number of support vectors (SVs) can be very large in many applications. To circumvent these drawbacks, we propose the robust truncated hinge loss SVM (RSVM), which uses a truncated hinge loss. The RSVM is shown to be more robust to outliers and to deliver more accurate classifiers using a smaller set of SVs than the standard SVM. Our theoretical results show that the RSVM is Fisher-consistent, even when there is no dominating class, a scenario that is particularly challenging for multicategory classification. Similar results are obtained for a class of margin-based classifiers.

KEY WORDS: Classification; d.c. algorithm; Fisher consistency; Regularization; Support vectors; Truncation.

1. INTRODUCTION

The Support vector machine (SVM) is a powerful classification tool that has enjoyed great success in many applications (Vapnik 1998; Cristianini and Shawe-Taylor 2000). It was first invented using the idea of searching for the optimal separating hyperplane with an elegant margin interpretation. The corresponding SVM classifier can be obtained by solving a quadratic programming (QP) problem, and its solution may depend only on a small subset of the training data, namely the set of support vectors (SVs).

It is now known that the SVM can be fit in the regularization framework of *loss* + *penalty* using the hinge loss (Wahba 1999). In the regularization framework, the loss function is used to keep the fidelity of the resulting model to the data. The penalty term in regularization helps avoid overfitting of the resulting model. For classification problems, one important goal is to construct classifiers with high predication accuracy, that is, good generalization ability. The most natural measure of the data fit is the classification error based on the 0–1 loss. But, optimization involving the 0–1 loss is very difficult (Shen, Tseng, Zhang, and Wong 2003). Therefore, most classification methods use convex losses as surrogates of the 0–1 loss, for example the hinge loss of the SVM, the logistic loss in the penalized logistic regression (Lin et al. 2000; Zhu and Hastie 2005), and the exponential loss function used in AdaBoost (Friedman, Hastie, and Tibshirani 2000).

Despite its wide success, however, the SVM has some drawbacks for difficult learning problems:

- The SVM classifier tends to be sensitive to noisy training data. When there exist points far away from their own classes (namely, "outliers" in the training data), the SVM classifier tends to be strongly affected by such points because of its unbounded hinge loss.
- The number of SVs can be very large for many problems, especially for difficult classification problems or problems with numerous input variables. A SVM classifier with many SVs may require longer computational time, especially for the predication phase.

In this article we propose a SVM methodology that involves truncating the unbounded hinge loss. Through this simple, yet critical modification of the loss function, we show that the resulting classifier remedies the aforementioned drawbacks of the original SVM. Specifically, the robust truncated-hinge-loss support vector machine (RSVM) is very robust to outliers in the training data. Consequently, it can deliver higher classification accuracy than the original SVM in many problems. Moreover, the RSVM retains the SV interpretation and often selects much fewer number of SVs than the SVM. Interestingly, the RSVM typically selects a subset of the SV set of the SVM. It tends to eliminate most of the outliers from the original SV set and, as a result, to deliver more robust and accurate classifiers.

Although truncation helps robustify the SVM, the associated optimization problem involves nonconvex minimization, which is more challenging than QP of the original SVM. We propose to apply the difference convex (d.c.) algorithm to solve the nonconvex problem through a sequence of convex subproblems. Our numerical experience suggests that the algorithm works effectively.

The rest of the article is organized as follows. In Section 2 we briefly review the SVM methodology and introduce the RSVM. Both binary and multicategory classification problems are considered. Some theoretical properties of the truncated margin-based losses are explored as well. In Section 3 we develop some numerical algorithms of the RSVM through the d.c. algorithm. We also give the SV interpretation of the RSVM. In Sections 4 and 5 we present numerical results on both simulated and real data to demonstrate the effectiveness of the truncated hinge loss. We provide some conclusions in Section 6, and collect proofs of the theoretical results in the Appendix.

2. METHODOLOGY

For a classification problem, we are given a training sample $\{(\mathbf{x}_i, y_i): i = 1, 2, \ldots, n\}$ that is distributed according to some unknown probability distribution function $P(\mathbf{x}, y)$. Here $\mathbf{x}_i \in \mathcal{S} \subset \Re^d$ and y_i denote the input vector and output label, where n is the sample size and d is the dimensionality of the input space. In this section we first review the method of SVM and then introduce the RSVM.

Yichao Wu is Research Associate, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544. Yufeng Liu is Assistant Professor, Department of Statistics and Operations Research, Carolina Center for Genome Sciences, University of North Carolina, Chapel Hill, NC 27599 (E-mail: *yfliu@email.unc.edu*). Liu is supported in part by grant DMS-0606577 from the National Science Foundation, the UNC Junior Faculty Development Award, and the UNC University Research Council Small Grant Program. Wu is supported in part by grant R01-GM07261 from National Institutes of Health. The authors thank the joint editors, the associate editor, and two reviewers for their constructive comments and suggestions.

Journal of the American Statistical Association
September 2007, Vol. 102, No. 479, Theory and Methods
DOI 10.1198/016214507000000617

2.1 The Support Vector Machine

For illustration, we first briefly describe the linear binary SVM. Let $y \in \{\pm 1\}$ and $f(\mathbf{x}) = \boldsymbol{w}'\mathbf{x} + b$. The standard SVM aims to find $f(\mathbf{x})$ so that $\hat{y} = \operatorname{sign}(f(\mathbf{x}))$ can be used for predication. More specifically, the SVM classifier solves the regularization problem

$$\min_f J(f) + C\sum_{i=1}^n \ell(y_i f(\mathbf{x}_i)), \tag{1}$$

with the L_2 penalty $J(f) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$, $C > 0$ a tuning parameter, and the hinge loss $\ell(u) = H_1(u) = (1-u)_+$, where $(u)_+ = u$ if $u \ge 0$ and 0 otherwise.

The optimization formulation in problem (1) is also known as the primal problem of the SVM. Using the Lagrange multipliers, (1) can be converted into an equivalent dual problem as follows:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle - \sum_{i=1}^n \alpha_i \tag{2}$$

$$\text{subject to} \quad \sum_{i=1}^n y_i\alpha_i = 0; \qquad 0 \le \alpha_i \le C, \forall i.$$

Once the solution of problem (2) is obtained, $\boldsymbol{w}$ can be calculated as $\sum_{i=1}^n y_i\alpha_i\mathbf{x}_i$, and the intercept b can be computed using the Karush–Kuhn–Tucker (KKT) complementarity conditions of the optimization theory. If nonlinear learning is needed, then the *kernel trick* can be applied by replacing the inner product $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$ by $K(\mathbf{x}_i, \mathbf{x}_j)$, where the kernel K is a positive definite function.

From problem (2), we can see that among the n training points, only points with $\alpha_i > 0$ make an impact on the SVM classifier, namely the SVs. It can be shown that these points satisfy $y_i f(\mathbf{x}_i) \le 1$. Consequently, outliers that are far from their own classes are included as SVs and influence the classifier. One important contribution of this article is a method to remove some of these outliers from the set of SVs and deliver more robust classifiers through truncating the hinge loss.

2.2 The Truncated Hinge Loss Support Vector Machine

For generality, we consider a k-class classification problem with $k \ge 2$. When $k = 2$, the methodology discussed herein reduces to the binary counterpart in Section 2.1. Let $\mathbf{f} = (f_1, f_2, \ldots, f_k)$ be the decision function vector, where each component represents one class and maps from $\mathcal{S}$ to $\Re$. To ensure uniqueness of the solution and reduce the dimension of the problem, a sum-to-0 constraint $\sum_{j=1}^k f_j = 0$ is used. For any new input vector $\mathbf{x}$, its label is estimated through a decision rule, $\hat{y} = \arg\max_{j=1,2,\ldots,k} f_j(\mathbf{x})$. Clearly, the argmax rule is equivalent to the sign function used in the binary case in Section 2.1.

Point $(\mathbf{x}, y)$ is misclassified by $\mathbf{f}$ if $y \ne \arg\max_j f_j(\mathbf{x})$, that is, if $\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y) \le 0$, where $\mathbf{g}(\mathbf{f}(\mathbf{x}), y) = \{f_y(\mathbf{x}) - f_j(\mathbf{x}), j \ne y\}$. The quantity $\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)$ is the generalized functional margin, which reduces to $yf(\mathbf{x})$ in the binary case with $y \in \{\pm 1\}$ (Liu and Shen 2006). A natural way of generalizing the binary method in Section 2.1 is to replace the term $yf(\mathbf{x})$ with $\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)$ and solve the regularization problem

$$\min_{\mathbf{f}} \sum_{j=1}^k J(f_j) + C\sum_{i=1}^n \ell\big(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)\big) \tag{3}$$

$$\text{subject to} \quad \sum_{j=1}^k f_j(\mathbf{x}) = 0.$$

For example, problem (3) becomes a multicategory SVM when we use the hinge loss H_1 for ℓ (Crammer and Singer 2001; Liu and Shen 2006). We note that our extension of the SVM from the binary case to the multicategory case is not unique; some other extensions include those of Vapnik (1998), Weston and Watkins (1999), Bredensteiner and Bennett (1999), and Lee, Lin, and Wahba (2004). Because the formulation in (3) has the margin interpretation and is closely connected with misclassification and the 0–1 loss, we use it to introduce the RSVM. In principle, the truncation operation can be applied to other multicategory SVMs as well.

Note that the hinge loss $H_1(u) = (1-u)_+$ grows linearly when u decreases with $u \le 1$. This implies that a point with large $1 - \min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)$ results in large H_1 and, consequently, greatly influences the final solution. Such points are typically far away from their own classes and tend to deteriorate the SVM performance. We propose to reduce their influence by truncating the hinge loss. In particular, we consider the truncated hinge loss function $T_s(u) = H_1(u) - H_s(u)$, where $H_s(u) = (s-u)_+$. Figure 1 displays the three functions $H_1(u)$, $H_s(u)$, and $T_s(u)$. The value of s specifies the location of truncation. We set $s \le 0$, because a truncated loss with $s > 0$ is constant for $u \in [-s, s]$ and cannot distinguish those correctly classified points with $y_i f(\mathbf{x}_i) \in (0, s]$ from those wrongly classified points with $y_i f(\mathbf{x}_i) \in [-s, 0]$. When $s = -\infty$, no truncation has been performed, and $T_s(u) = H_1(u)$. In fact, the choice of s is important and affects the performance of the RSVM.

The literature contains some previous studies on special cases of $T_s(u)$. Liu, Shen, and Doss (2005) and Liu and Shen (2006) studied the use of ψ loss, which is essentially the same as $T_0(u)$. Collobert, Sinz, Weston, and Bottou (2006) explored some advantage of $T_s(u)$ for the binary SVM. Our proposed methodology is more general and covers both binary and multicategory problems.

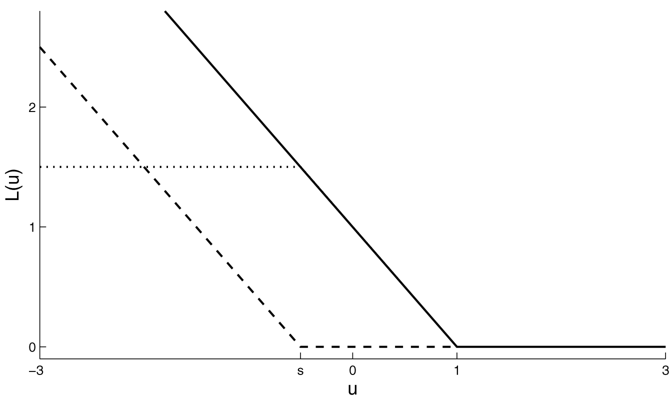

Figure 1. Plot of the functions $H_1(u)$ (——), $H_s(u)$ (- - -), and $T_s(u)$ (·····) with $T_s = H_1 - H_s$.

2.3 Theoretical Properties

In this section we study Fisher consistency of a class of truncated margin-based losses for both binary and multicategory problems. For a binary classification problem with $y \in \{\pm 1\}$ in Section 2.1, write $p(\mathbf{x}) = P(Y = +1|\mathbf{x})$. Then Fisher consistency requires that the minimizer of $E[\ell(Yf(\mathbf{X}))|\mathbf{X} = \mathbf{x}]$ has the same sign as $p(\mathbf{x}) - 1/2$ (Lin 2004). Fisher consistency, also known as classification-calibration (Bartlett, Jordan, and McAuliffe 2006), is a desirable property for a loss function. For the binary SVM, Lin (2002) showed that the minimizer of $E[H_1(Yf(\mathbf{X})|\mathbf{X} = \mathbf{x}]$ is $\operatorname{sign}(p(\mathbf{x}) - 1/2)$. As a result, the hinge loss is Fisher-consistent for binary classification. Moreover, we note that the SVM estimates only the classification boundary $\{\mathbf{x}: p(\mathbf{x}) = 1/2\}$ without estimating $p(\mathbf{x})$ itself. The following proposition establishes Fisher consistency of a class of truncated losses for the binary case.

Proposition 1. Assume that a loss function $\ell(\cdot)$ is nonincreasing and that $\ell'(0) < 0$ exists. Denote $\ell_{T_s}(\cdot) = \min(\ell(\cdot), \ell(s))$ as the corresponding truncated loss of $\ell(\cdot)$. Then $\ell_{T_s}(y \times f(\mathbf{x}))$ is Fisher-consistent for any $s \le 0$.

Proposition 1 is applicable to many commonly used losses, including the exponential loss $\ell(u) = e^{-u}$, the logistic loss $\ell(u) = \log(1 + e^{-u})$, and the hinge loss $\ell(u) = H_1(u)$. Our focus here is the truncated hinge loss T_s. Although any T_s with $s \le 0$ is Fisher-consistent, different s's in the RSVM may give different performance. A small s, close to $-\infty$, may not perform sufficient truncation to remove the effects of outliers. A large s, close to 0, may not work well either, because its penalty on wrongly classified points near the boundary may be too small to be distinguished from correctly classified points near the boundary. Our numerical experience shows that $s = 0$ used in ψ-learning is indeed suboptimal. We suggest $s = -1$ for binary problems, our numerical results show that this choice indeed works well.

For multicategory problems with $k > 2$, the issue of Fisher consistency becomes more complex. Consider $y \in \{1, \dots, k\}$ as in Section 2.2 and let $p_j(\mathbf{x}) = P(Y = j|\mathbf{x})$. In this context, Fisher consistency requires that $\arg\max_j f_j^* = \arg\max_j p_j$, where $\mathbf{f}^*(\mathbf{x}) = (f_1^*(\mathbf{x}), \dots, f_k^*(\mathbf{x}))$ denotes the minimizer of $E[\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{X}), Y))|\mathbf{X} = \mathbf{x}]$. Zhang (2004) and Tewari and Bartlett (2005) pointed out the Fisher inconsistency of $H_1(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$. Our next proposition shows that a general loss $\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ may not be always Fisher-consistent.

Proposition 2. Assume that a loss function $\ell(\cdot)$ is nonincreasing and that $\ell'(0) < 0$ exists. Then if $\mathbf{f}^*$ minimizes $E[\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{X}), Y))|\mathbf{X} = \mathbf{x}]$, it has the following properties:

a. If $\max_j p_j > 1/2$, then $\arg\max_j f_j^* = \arg\max_j p_j$.
b. If $\ell(\cdot)$ is convex and $\max_j p_j \le 1/2$, then $\mathbf{f}^* = \mathbf{0}$ is a minimizer.

Proposition 2 suggests that $\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ is Fisher-consistent when $\max_j p_j > 1/2$, that is, when there is a dominating class. Except for the Bayes decision boundary, this condition always holds for a binary problem. For a problem with $k > 2$, however, existence of a dominating class may not be guaranteed. If $\max_j p_j(\mathbf{x}) \le 1/2$ for a given $\mathbf{x}$, then $\mathbf{f}^*(\mathbf{x}) = \mathbf{0}$ can be a minimizer, and the argmax of $\mathbf{f}^*(\mathbf{x})$ cannot be uniquely determined. Interestingly, truncating $\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ can make it Fisher-consistent even in the situation of no dominating class, as shown in Theorem 1.

Theorem 1. Assume that a loss function $\ell(\cdot)$ is nonincreasing and that $\ell'(0) < 0$ exists. Let $\ell_{T_s}(\cdot) = \min(\ell(\cdot), \ell(s))$ with $s \le 0$. Then a sufficient condition for the loss $\ell_{T_s}(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ with $k > 2$ to be Fisher-consistent is that the truncation location s satisfies that $\sup_{\{u: u \ge -s \ge 0\}} (\ell(0) - \ell(u))/(\ell(s) - \ell(0)) \ge (k-1)$. This condition is also necessary if $\ell(\cdot)$ is convex.

We note that the truncation value s given in Theorem 1 depends on the class number k. For $\ell(u) = H_1(u), e^{-u}$, and $\log(1 + e^{-u})$, Fisher consistency for $\ell_{T_s}(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ can be guaranteed for $s \in [-\frac{1}{k-1}, 0], [\log(1 - \frac{1}{k}), 0]$, and $[-\log(2^{k/(k-1)} - 1), 0]$. Clearly, the larger that k is, the more truncation is needed to ensure Fisher consistency. In the binary case, Fisher consistency of ℓ_{T_s} can be established for all $s \le 0$, as shown in Proposition 2. As $k \to \infty$, the only choice of s that can guarantee Fisher consistency of $\ell_{T_s}(\min \mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ is 0 for these three losses. This is due to the fact that the difficulty of no dominating class becomes more severe as k increases. For implementation of our RSVM, we recommend choosing $s = -\frac{1}{k-1}$. Our numerical results confirm the advantage of this choice.

3. ALGORITHMS

Truncating the hinge loss produces a nonconvex loss, and as a result, the optimization problem in (3) with $\ell = T_s$ involves nonconvex minimization. Note that the truncated hinge loss function can be decomposed as the difference of two convex functions, H_1 and H_s. Using this property, we propose to apply the d.c. algorithm (An and Tao 1997; Liu et al. 2005) to solve the nonconvex optimization problem of the RSVM. The d.c. algorithm solves the nonconvex minimization problem by minimizing a sequence of convex subproblems as in Algorithm 1.

Algorithm 1. The d.c. algorithm for minimizing $Q(\Theta) = Q_{vex}(\Theta) + Q_{cav}(\Theta)$.

1. Initialize Θ_0.
2. Repeat $\Theta_{t+1} = \arg\min_\Theta (Q_{vex}(\Theta) + \langle Q'_{cav}(\Theta_t), \Theta - \Theta_t \rangle)$ until convergence of Θ_t.

Fan and Li (2001) have proposed local quadratic approximation (LQA) to handle some nonconvex penalized likelihood problems by locally approximating the nonconvex penalty function by a quadratic function iteratively. Hunter and Li (2005) showed that the LQA is a special case of the minorize-maximize or majorize-minimize (MM) algorithm and studied its convergence property. For our d.c. algorithm, because we replace H_s by its affine minorization at each iteration, the algorithm is also an instance of the MM algorithm. Note that the objective function in (3) is lower-bounded by 0. Thus, by its descent property, the d.c. algorithm converges to an ϵ-local minimizer in finite steps (An and Tao 1997; Liu et al. 2005). As shown in Sections 3.1 and 3.2, the d.c. algorithm also has a nice SV interpretation.

We derive the d.c. algorithm for linear learning in Section 3.1, and then generalize it to the case of nonlinear learning through kernel mapping in Section 3.2. We discuss implementation of the RSVM with the adaptive L_1 penalty in Section 3.3.

3.1 Linear Learning

Let $f_j(\mathbf{x}) = \boldsymbol{w}_j^T\mathbf{x} + b_j$; $\boldsymbol{w}_j \in \Re^d$, $b_j \in \Re$, and $\mathbf{b} = (b_1, b_2, \dots, b_k)^T \in \Re^k$, where $\boldsymbol{w}_j = (w_{1j}, w_{2j}, \dots, w_{dj})^T$ and $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_k)$. With $\ell = T_s$, (3) becomes

$$\min_{\boldsymbol{W},\mathbf{b}} \frac{1}{2}\sum_{j=1}^{k} \|\boldsymbol{w}_j\|_2^2 + C\sum_{i=1}^{n} T_s\big(\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i)\big) \tag{4}$$

$$\text{subject to} \quad \sum_{j=1}^{k} w_{mj} = 0; \qquad m = 1, 2, \dots, d;$$

$$\sum_{j=1}^{k} b_j = 0,$$

where the constraints are adopted to avoid any nonidentifiability issues in the solution.

Denote Θ as $(\boldsymbol{W}, \mathbf{b})$. Applying the fact that $T_s = H_1 - H_s$, the objective function in (4) can be decomposed as

$$\begin{aligned} Q^s(\Theta) &= \frac{1}{2}\sum_{j=1}^{k} \|\boldsymbol{w}_j\|_2^2 + C\sum_{i=1}^{n} H_1\big(\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i)\big) \\ &\quad - C\sum_{i=1}^{n} H_s\big(\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i)\big) \\ &= Q^s_{vex}(\Theta) + Q^s_{cav}(\Theta), \end{aligned}$$

where $Q^s_{vex}(\Theta) = \frac{1}{2}\sum_{j=1}^{k} \|\boldsymbol{w}_j\|_2^2 + C\sum_{i=1}^{n} H_1(\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i))$ and $Q^s_{cav}(\Theta) = Q^s(\Theta) - Q^s_{vex}(\Theta)$ denote the convex and concave parts.

It can be shown that the convex dual problem at the $(t+1)$th iteration, given the solution $\mathbf{f}^t$ at the tth iteration, is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{j=1}^{k} \left\| \sum_{i:y_i=j}\sum_{j'\neq y_i} (\alpha_{ij'} - \beta_{ij'})\mathbf{x}_i^T - \sum_{i:y_i\neq j} (\alpha_{ij} - \beta_{ij})\mathbf{x}_i^T \right\|_2^2 - \sum_{i=1}^{n}\sum_{j'\neq y_i} \alpha_{ij'}$$

subject to

$$\sum_{i:y_i=j}\sum_{j'\neq y_i} (\alpha_{ij'} - \beta_{ij'}) - \sum_{i:y_i\neq j} (\alpha_{ij} - \beta_{ij}) = 0, \quad j = 1, 2, \dots, k; \tag{5}$$

$$0 \le \sum_{j\neq y_i} \alpha_{ij} \le C, \qquad i = 1, 2, \dots, n; \tag{6}$$

$$\alpha_{ij} \ge 0, \qquad i = 1, 2, \dots, n;\ j \neq y_i, \tag{7}$$

where $\beta_{ij} = C$ if $f^t_{y_i}(\mathbf{x}_i) - f^t_j(\mathbf{x}_i) < s$ with $j = \arg\max(f^t_{j'}(\mathbf{x}_i) : j' \neq y_i)$ and 0 otherwise.

This dual problem is a quadratic programming (QP) problem similar to that of the standard SVM and can be solved by many optimization programs. Once its solution is obtained, the coefficients $\boldsymbol{w}_j$'s can be recovered as

$$\boldsymbol{w}_j = \sum_{i:y_i=j}\sum_{j'\neq y_i} (\alpha_{ij'} - \beta_{ij'})\mathbf{x}_i - \sum_{i:y_i\neq j} (\alpha_{ij} - \beta_{ij})\mathbf{x}_i. \tag{8}$$

More details of the derivation are provided in the Appendix.

It is interesting to note that the representation of $\boldsymbol{w}_j$'s given in (8) automatically satisfies $\sum_{j=1}^{k} w_{mj} = 0$ for each $1 \le m \le d$. Moreover, we can see that the coefficients $\boldsymbol{w}_j$ are determined by only those data points whose corresponding $\alpha_{ij} - \beta_{ij}$ is not 0 for some $1 \le j \le k$, which are the SVs of the RSVM. The set of SVs of the RSVM using the d.c. algorithm is only a subset of the set of SVs of the original SVM. The RSVM tries to remove points satisfying $f^t_{y_i}(\mathbf{x}_i) - f^t_j(\mathbf{x}_i) < s$ with $j = \operatorname{argmax}(f^t_{j'}(\mathbf{x}_i) : j' \neq y_i)$ from the original set of SVs and consequently eliminate the effects of outliers. This provides an intuitive algorithmic explanation of the robustness of the RSVM to outliers.

After the solution of $\boldsymbol{W}$ is derived, $\mathbf{b}$ can be obtained by solving either a sequence of KKT conditions as used in the standard SVM or a linear programming (LP) problem. Denote $\tilde{f}_j(\mathbf{x}_i) = \mathbf{x}_i^T\boldsymbol{w}_j$. Then $\mathbf{b}$ can be obtained through the following LP problem:

$$\min_{\boldsymbol{\eta},\mathbf{b}} C\sum_{i=1}^{n} \eta_i + \sum_{j=1}^{k}\left(\sum_{i:y_i=j}\sum_{j'\neq y_i} \beta_{ij'} - \sum_{i:y_i=j} \beta_{ij}\right) b_j$$

$$\begin{aligned} \text{subject to} \quad & \eta_i \ge 0, \qquad i = 1, 2, \dots, n \\ & \eta_i \ge 1 - \big(\tilde{f}_{y_i}(\mathbf{x}_i) + b_{y_i}\big) + \tilde{f}_j(\mathbf{x}_i) + b_j, \\ & \qquad i = 1, 2, \dots, n;\ j \neq y_i \\ & \sum_{j=1}^{k} b_j = 0. \end{aligned}$$

3.2 Nonlinear Learning

For nonlinear learning, each decision function $f_j(\mathbf{x})$ is represented by $h_j(\mathbf{x}) + b_j$ with $h_j(\mathbf{x}) \in H_K$, where H_K is a reproducing kernel Hilbert space (RKHS). Here the kernel $K(\cdot, \cdot)$ is a positive definite function mapping from $\mathcal{S} \times \mathcal{S}$ to $\Re$. Due to the representer theorem of Kimeldorf and Wahba (1971) (also see Wahba 1999), the nonlinear problem can be reduced to finding finite-dimensional coefficients v_{ij}, and $h_j(\mathbf{x})$ can be represented as $\sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i)v_{ij}$; $j = 1, 2, \dots, k$.

Denote $\mathbf{v}_j = (v_{1j}, v_{2j}, \dots, v_{nj})^T$, $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k)$, and $\mathbf{K}$ to be an $n \times n$ matrix whose (i_1, i_2) entry is $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2})$. Let $\mathbf{K}_i$ be the ith column of $\mathbf{K}$, and denote the standard basis of the n-dimensional space by $e_i = (0, 0, \dots, 1, \dots, 0)^T$, with 1 for its ith component and 0 for other components. A similar derivation as in the linear case leads to the following dual problem for nonlinear learning:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{j=1}^{k} \Bigg\langle & \sum_{i:y_i=j}\sum_{j'\neq y_i} (\alpha_{ij'} - \beta_{ij'})\mathbf{K}_i - \sum_{i:y_i\neq j} (\alpha_{ij} - \beta_{ij})\mathbf{K}_i, \\ & \sum_{i:y_i=j}\sum_{j'\neq y_i} (\alpha_{ij'} - \beta_{ij'})e_i - \sum_{i:y_i\neq j} (\alpha_{ij} - \beta_{ij})e_i \Bigg\rangle \\ & - \sum_{i=1}^{n}\sum_{j'\neq y_i} \alpha_{ij'}, \end{aligned}$$

subject to constraints (5)–(7), where the β_{ij}'s are defined similarly as in the linear case. After solving the foregoing QP problem, we can recover the coefficients $\mathbf{v}_j$'s as

$$\mathbf{v}_j = \sum_{i:y_i=j} \sum_{j' \neq y_i} (\alpha_{ij'} - \beta_{ij'}) e_i - \sum_{i:y_i \neq j} (\alpha_{ij} - \beta_{ij}) e_i.$$

The intercepts b_m's can be solved using LP as in linear learning.

3.3 Variable Selection by the L_1 Penalty

Variable selection is an important aspect in the model building process. To perform variable selection, Zhu, Rosset, Hastie, and Tibshirani (2003) investigated the SVM using the L_1 penalty. Fan and Li (2001) and Fan and Peng (2004) proposed the SCAD penalty for variable selection and studied its oracle property. Zhang, Ahn, Lin, and Park (2006) applied the SCAD penalty for the SVM. Other penalites for variable selection have been given by Yuan and Lin (2006) and Zhang (2006). Fan and Li (2006) have provided a comprehensive review of variable selection techniques and their applications.

The L_1 penalty uses the same weights for different variables in the penalty term, which may be too restrictive. Intuitively, different variables should be penalized differently according to their relative importance. A natural solution is to apply a weighted L_1 penalty. Zou (2006) proposed the adaptive L_1 penalty for variable selection and showed its oracle property for regression problems. In this section we discuss the use of the adaptive L_1 penalty in the RSVM for simultaneous classification and variable selection. In particular, we first use the L_2 penalty to derive the weights for the weighted L_1 penalty and then solve the RSVM with the new penalty. Our numerical examples indicate that the weights work well, even for the high-dimensional, low–sample size problems.

Replacing the L_2 penalty with the weighted L_1 penalty at each step of the d.c. algorithm, the objective function in (4) becomes

$$\sum_{m=1}^{d} \sum_{j=1}^{k} \delta_{mj} |w_{mj}| + C \sum_{i=1}^{n} H_1\big(\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i)\big) + \sum_{j=1}^{k} \left(\left\langle \frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta_t), \boldsymbol{w}_j \right\rangle + b_j \frac{\partial}{\partial b_j} Q^s_{cav}(\Theta_t) \right), \quad (9)$$

where δ_{mj} is the weight for coefficient w_{mj}. We suggest using $1/|w^*_{mj}|$ as the weight δ_{mj}, where w^*_{mj} is the solution of (4) using the L_2 penalty.

To solve (9), we introduce slack variable ξ_i's for the hinge loss term and obtain the following LP problem:

$$\min_{\boldsymbol{W}, \mathbf{b}, \boldsymbol{\xi}} \sum_{m=1}^{d} \sum_{j=1}^{k} \delta_{mj} |w_{mj}| + C \sum_{i=1}^{n} \xi_i + \sum_{j=1}^{k} \left\langle \frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta_t), \boldsymbol{w}_j \right\rangle + \sum_{j=1}^{k} b_j \frac{\partial}{\partial b_j} Q^s_{cav}(\Theta_t)$$

$$\begin{aligned}\text{subject to}\quad & \xi_i \geq 0, \qquad i = 1, 2, \ldots, n \\ & \xi_i \geq 1 - \big[\mathbf{x}_i^T \boldsymbol{w}_{y_i} + b_{y_i}\big] + [\mathbf{x}_i^T \boldsymbol{w}_j + b_j], \\ & \qquad i = 1, 2, \ldots, n;\ j \neq y_i; \\ & \sum_{j=1}^{k} w_{mj} = 0, m = 1, 2, \ldots, d; \\ & \sum_{j=1}^{k} b_j = 0.\end{aligned}$$

4. SIMULATIONS

In this section we investigate the performance of the proposed RSVM. Throughout our simulations, we set the sample sizes of training, tuning, and testing data as 100, 1,000, and 10,000. We generated the tuning and testing data in the same manner as the training data. Tuning sets are used to choose the regularization parameter C through a grid search, and testing errors, evaluated on independent testing data, are used to measure the accuracy of various classifiers.

4.1 Linear Learning Examples

We generated simulated datasets in the following manner. First, generate (x_1, x_2) uniformly on the unit disc $\{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$. Let ϑ denote the radian phase angle measured counterclockwise from the ray from $(0, 0)$ to $(1, 0)$ to another ray from $(0, 0)$ to (x_1, x_2). For a k-class example, the class label y is assigned to be $\lfloor \frac{k\vartheta}{2\pi} \rfloor + 1$, where $\lfloor \cdot \rfloor$ is the integer part function. Second, contaminate the data by randomly selecting *perc* ($= 10\%$ or 20%) instances and changing their label indices to one of the remaining $k - 1$ classes with equal probabilities. For the case with $p > 2$, the remaining input x_j's ($2 < j \leq p$) are generated independently from uniform$[-1, 1]$ as noise variables.

We have examined the performance of SVMs with three different loss functions (the hinge loss H_1 and the truncated hinge losses T_0 and $T_{-1/(k-1)}$) as well as three penalties (L_2, L_1, and adaptive L_1). We investigate cases with 2, 3, and 4 classes and dimensions of input variables of 2, 12, and 22. Some results of various SVMs averaging over 100 repetitions are reported in Figures 2–4 for 10% contamination. Among these three loss functions, it is very clear from these figures that truncated losses work much better than the original hinge loss using fewer SVs. This confirms our claim that truncation can help robustify the unbounded hinge loss and deliver more accurate classifiers. As to the choice of s, our suggestion is to use $-1/(k - 1)$. Our empirical results indeed show that the RSVM with $s = -1/(k - 1)$ performs better than the RSVMs with other choices of s.

Because only the first one ($k = 2$) or two ($k > 2$) input variables are relevant to classification, the remaining ones are noise. When $p > 2$, shrinkage on the coefficients can help remove some noise variables from the classifier. From Figures 2 and 3, we can conclude that methods using the L_1 and adaptive L_1 penalties give much smaller testing errors than the methods using the L_2 penalty. Between the L_1 and adaptive L_1 penalties, the adaptive procedure helps remove more noise variables and consequently works better than the original L_1-penalized methods. All methods keep the important variables in the resulting classifiers in all replications.

In terms of SVs, the average numbers of SVs corresponding to the RSVM are much smaller than those of the SVM. As discussed earlier, outliers in the training data are typically used as SVs for the standard SVM. The proposed truncated procedures tend to remove some of these points from the set of SVs and consequently have smaller sets. For a graphical visualization,

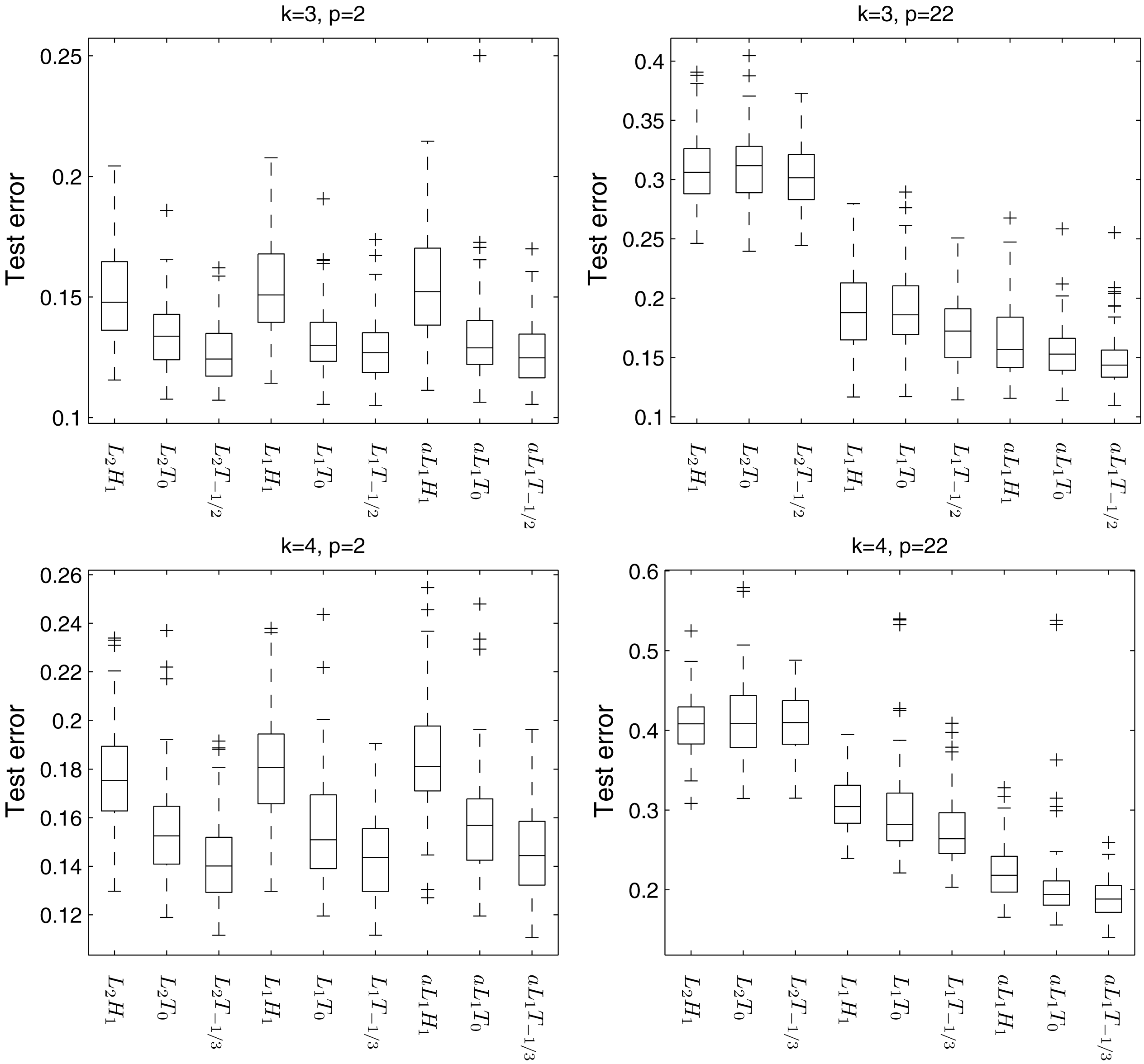

Figure 2. Boxplots of the testing errors for the linear example in Section 4.1 with $perc = 10\%$, $p = 2, 22$, and $k = 3, 4$ using nine methods (three penalties L_2, L_1, and adaptive L_1; and three losses H_1, T_0, and $T_{-1/(k-1)}$).

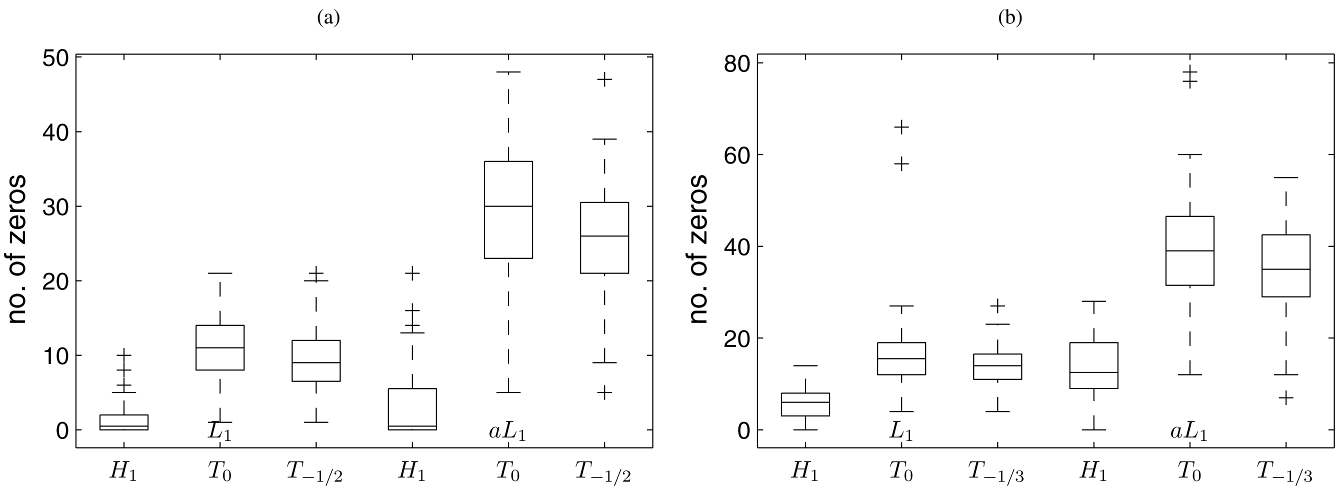

Figure 3. Boxplots of the numbers of zero coefficients for the linear example in Section 4.1, with $perc = 10\%$, $p = 22$, and $k = 3$ (a) and 4 (b) using three losses, H_1, T_0, and $T_{-1/(k-1)}$, and two penalties, L_1 and adaptive L_1.

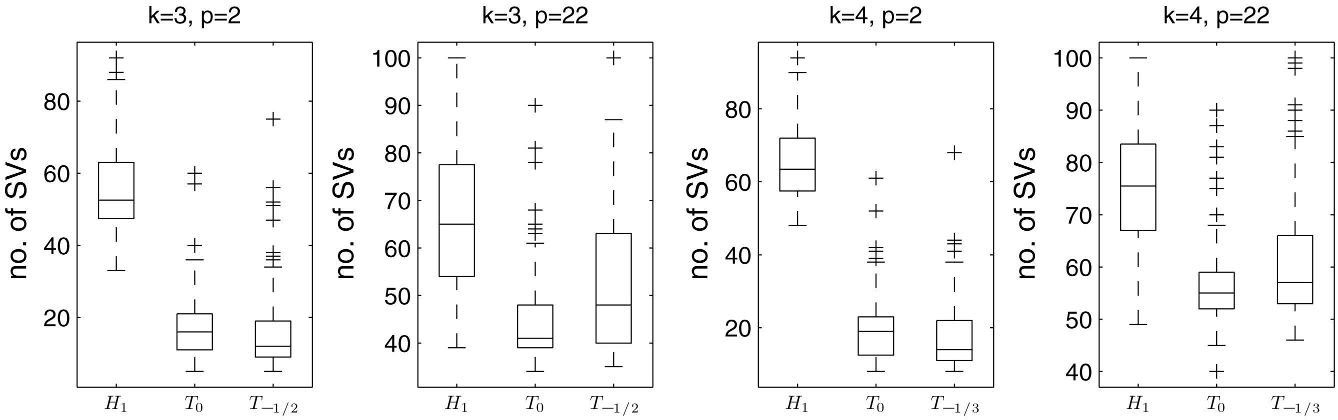

Figure 4. Boxplots of the numbers of SVs for the linear example in Section 4.1 with $perc = 10\%$, $p = 2, 22$, and $k = 3, 4$ using H_1, T_0, and $T_{-1/(k-1)}$ with the L_2 penalty.

these Figure 5 illustrates the SVs of several losses for one typical training data set with 20% contamination. Figure 5(a) shows that there are outliers in the training data generated by random contamination. Such outliers do not provide useful information for classification. The SVM, as shown in 5(b), includes all outliers in its set of SVs. The RSVM, in contrast, eliminates most outliers from the SV sets and produces more robust classifiers.

4.2 Nonlinear Learning Examples

Three-class nonlinear examples with $p = 2$ are generated in a similar way as in the linear examples in Section 4.1. First, generate (x_1, x_2) uniformly over the unit disc $\{(x_1, x_2): x_1^2 + x_2^2 \le 1\}$. Define ϑ to be the radian phase angle as in the linear case. For a three-class example, the class label y is assigned as: $y = 1$ if $\lfloor \frac{k\vartheta}{2\pi} \rfloor + 1 = 1$ or 4, $y = 2$ if $\lfloor \frac{k\vartheta}{2\pi} \rfloor + 1 = 2$ or 3, or $y = 3$ if $\lfloor \frac{k\vartheta}{2\pi} \rfloor + 1 = 5$ or 6. Next, randomly contaminate the data with $perc = 10\%$ or 20% as in the linear examples in Section 4.1.

To achieve nonlinear learning, we apply the Gaussian kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\langle \mathbf{x}_1, \mathbf{x}_2 \rangle / (2\sigma^2))$. Consequently, two parameters must be selected. The first parameter C is chosen using a grid search as in linear learning. The second parameter σ for the kernel is tuned among the first quartile, the median, and the third quartile of the between-class pairwise Euclidean distances of training inputs (Brown et al. 2000).

Results using different loss functions and different contamination percentages averaging over 100 repetitions are reported in Table 1. Similar to the linear examples, RSVMs give smaller testing errors while using fewer SVs than the standard SVM. To visualize decision boundaries and SVs of the original SVM and RSVMs, we choose one typical training sample and plot the results in Figure 6. Figure 6(a) shows the observations as well as the Bayes boundary. In the remaining three plot, boundaries using nonlinear learning with different loss functions H_1, T_0, and $T_{-0.5}$ are plotted, with their corresponding SVs displayed in the plots. From the plots, we can see that the RSVMs use many fewer SVs and at the same time yield more accurate classification boundaries than the standard SVM.

5. REAL DATA

In this section we investigate the performance of the RSVM on the real dataset Liver-disorder from UCI Machine Learning Repository. This dataset has a total of 345 observations with two classes and six input variables. (See the UCI Machine Learning Repository webpage, at *http://www.ics.uci.edu/~mlearn/MLRepository.html*, for more information on this dataset.) Before applying the methods, we standardize each input variable with mean 0 and standard deviation 1 and randomly split the dataset into training, tuning, and testing sets, each of size 115. We apply the SVM and RSVMs with $s = 0$ and -1 and with the L_2, L_1, and adaptive L_1 penalties. To further study robustness of the truncated hinge loss for both the training and tuning sets, we contaminate the class output by randomly choosing *perc* of their observations and changing the class output to the other class. We choose different contamination percentages, $perc = 0\%$, 5%, and 10%. This allows us to

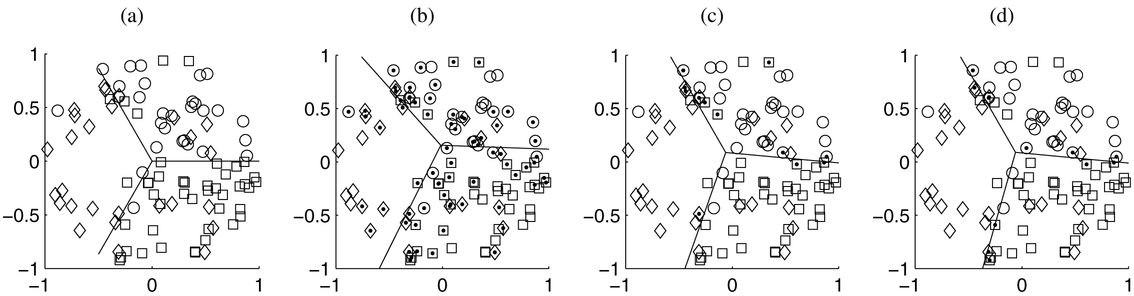

Figure 5. Plots of decision boundaries for one training set of the linear example in Section 4.1. (a) Bayes boundary; (b) H_1; (c) T_0; (d) $T_{-.5}$. The observations with black dots in the center represent SVs.

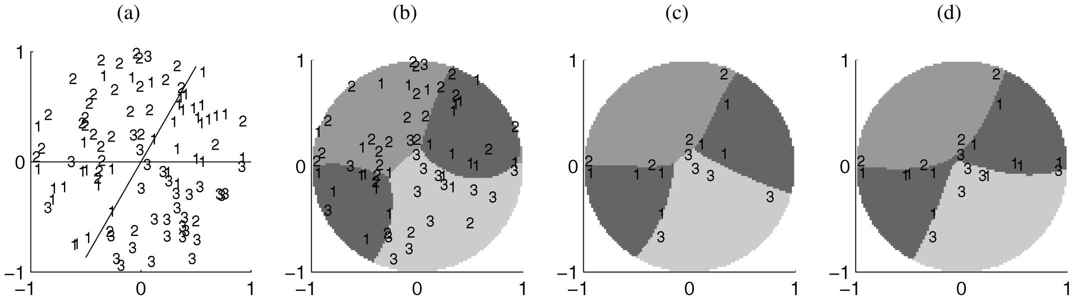

Figure 6. Plots of decision boundaries and SVs for one training set of the nonlinear example in Section 4.2 using different loss functions. (a) Bayes boundary; (b) H_1; (c) T_0; (d) $T_{-.5}$

examine the robustness of the truncated hinge loss to outliers. Results over 10 repetitions are reported in Table 2. From the table, we can conclude that the RSVM with $s=-1$ works the best among the three methods. In terms of SVs, RSVMs have much fewer SVs compared with the SVM. The RSVM with $s=0$ has more truncation and consequently fewer SVs than that of $s=-1$. Overall, contamination does not affect the testing errors of RSVMs as strongly as it does those of the SVM. This confirms our claim that the RSVM is more robust to outliers.

6. DISCUSSION

In this article we have proposed a new supervised learning method, the RSVM, which uses the truncated hinge loss and delivers more robust classifiers than the standard SVM. Our algorithm and numerical results show that the RSVM has the interpretation of SVs and tends to use a smaller, yet more stable set of SVs than that of the SVM. Our theoretical results indicate that truncation of a general class of loss functions can help make the corresponding classifiers, including RSVM, Fisher-consistent even in the absence of a dominating class for multicategory problems.

Although our focus in this article is on the SVM, the operation of truncation also can be applied to many other learning methods, as indicated by our theoretical studies. In fact, any classification methods with unbounded loss functions may suffer from the existence of extreme outliers. Truncating the original unbounded loss helps decrease the impact of outliers and consequently may deliver more robust classifiers. In future research we will explore the effect of truncation on some other loss functions, such as exponential loss and logistic loss.

APPENDIX: PROOFS AND DERIVATION OF THE DUAL PROBLEM IN SECTION 3.2

Proof of Proposition 1

Note that $E[\ell_{T_s}(Yf(\mathbf{X}))] = E[E(\ell_{T_s}(Yf(\mathbf{X}))|\mathbf{X}=\mathbf{x})]$. We can minimize $E[\ell_{T_s}(Yf(\mathbf{X}))]$ by minimizing $E(\ell_{T_s}(Yf(\mathbf{X}))|\mathbf{X}=\mathbf{x})$ for every $\mathbf{x}$. For any fixed $\mathbf{x}$, $E(\ell_{T_s}(Yf(\mathbf{X}))|\mathbf{X}=\mathbf{x})$ can be written as $p(\mathbf{x})\ell_{T_s}(f(\mathbf{x}))+(1-p(\mathbf{x}))\ell_{T_s}(-f(\mathbf{x}))$. Because ℓ_{T_s} is a nonincreasing function and $\ell'(0)<0$, the minimizer f^* must satisfy that $f^*(\mathbf{x})\geq 0$ if $p(\mathbf{x})>1/2$ and $f^*(\mathbf{x})\leq 0$ otherwise. Thus, it is sufficient to show that $f=0$ is not a minimizer. Without loss of generality, assume that $p(\mathbf{x})>1/2$. We consider two cases, $s=0$ and $s<0$. For $s=0$, $E(\ell_{T_s}(0)|\mathbf{X}=\mathbf{x}) > E(\ell_{T_s}(1)|\mathbf{X}=\mathbf{x})$, because $\ell(1)<\ell(0)$. Thus $f=0$ is not a minimizer. For $s<0$, $\frac{d}{df(\mathbf{x})}E(\ell_{T_s}(Yf(\mathbf{X}))|\mathbf{X}=\mathbf{x})\mid_{f(\mathbf{x})=0}= \frac{d}{df(\mathbf{x})}[(1-p)\ell_{T_s}(-f(\mathbf{x}))+p\ell_{T_s}(f(\mathbf{x}))]\mid_{f(\mathbf{x})=0}=(2p-1)\ell'(0)$ is less than 0, because $\ell'(0)<0$. Thus $f(\mathbf{x})=0$ is not a minimizer. We can then conclude that $f^*(\mathbf{x})$ has the same sign as $p(\mathbf{x})-1/2$.

Proof of Proposition 2

Note that $E[\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{X}),Y))]$ can be written as

$$E\Big[E\big(\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{x}),Y))|\mathbf{X}=\mathbf{x}\big)\Big] = E\left[\sum_{j=1}^{k} p_j(\mathbf{X})\ell\big(\min \mathbf{g}(\mathbf{f}(\mathbf{X}),j)\big)\right].$$

For any given $\mathbf{X}=\mathbf{x}$, assume that $j_p=\arg\max_j p_j(\mathbf{x})$ is unique and let $g_j=\min \mathbf{g}(\mathbf{f}(\mathbf{x}),j)$; $j=1,\ldots,k$. Then we can conclude that $g^*_{j_p}\geq 0$. To show this, suppose that $g^*_{j_p}<0$, which implies that $\max_j f^*_j > f^*_{j_p}$. It is easy to see that switching the largest component of $\mathbf{f}^*$ with its j_pth component will yield a smaller objective value due to the properties of ℓ. This implies that $g^*_{j_p}\geq 0$, that is, $f^*_{j_p}=\max_j f^*_j$.

To prove part a, we need to show that $g^*_{j_p}>0$. Clearly, $\mathbf{f}=\mathbf{0}$ gives the smallest objective value among solutions with $g_{j_p}=0$. Thus it is sufficient to show that $\mathbf{f}=\mathbf{0}$ is not a minimizer. Toward this end, consider a solution $\mathbf{f}_a$, whose elements are $-a$ except for the j_pth element being $(k-1)a$ for some $a\geq 0$. Then $E[E(\ell(\min \mathbf{g}(\mathbf{f}(\mathbf{x}),Y))|\mathbf{X}=\mathbf{x})]=(1-p_{j_p}(\mathbf{x}))\ell(-ka)+p_{j_p}(\mathbf{x})\ell(ka)$, and $\frac{d}{da}[(1-p_{j_p}(\mathbf{x}))\times$

Table 1. Results for the nonlinear examples in Section 4.2

	$perc=10\%$		$perc=20\%$	
Loss	Test error	Number of SVs	Test error	Number of SVs
H_1	$.1694_{(.0268)}$	$48.72_{(9.62)}$	$.2767_{(.0248)}$	$69.17_{(9.72)}$
T_0	$.1603_{(.0195)}$	$16.44_{(5.35)}$	$.2671_{(.0234)}$	$18.78_{(8.51)}$
$T_{-.5}$	$.1538_{(.0182)}$	$24.64_{(11.09)}$	$.2620_{(.0244)}$	$26.88_{(13.72)}$

Table 2. Results of the dataset Liver Disorder in Section 5

		L_2		L_1	Adaptive L_1
perc	Loss	Test error	#SV	Test error	Test error
	H_1	$.3322_{(.0395)}$	$88.10_{(10.20)}$	$.3243_{(.0369)}$	$.3200_{(.0289)}$
	T_0	$.3365_{(.0362)}$	$34.20_{(9.89)}$	$.3270_{(.0394)}$	$.3287_{(.0375)}$
0%	T_{-1}	$.3278_{(.0310)}$	$50.30_{(17.51)}$	$.3243_{(.0299)}$	$.3200_{(.0242)}$
	H_1	$.3809_{(.0809)}$	$92.20_{(8.55)}$	$.3678_{(.0754)}$	$.3574_{(.0647)}$
	T_0	$.3565_{(.0758)}$	$31.70_{(19.94)}$	$.3557_{(.0825)}$	$.3539_{(.0782)}$
5%	T_{-1}	$.3391_{(.0869)}$	$42.00_{(28.36)}$	$.3391_{(.0706)}$	$.3409_{(.0893)}$
	H_1	$.3791_{(.0754)}$	$93.90_{(8.65)}$	$.3757_{(.0759)}$	$.3835_{(.0772)}$
	T_0	$.3583_{(.0694)}$	$33.70_{(15.74)}$	$.3617_{(.0650)}$	$.3635_{(.0671)}$
10%	T_{-1}	$.3583_{(.0882)}$	$48.20_{(3.11)}$	$.3461_{(.0845)}$	$.3522_{(.0953)}$

$\ell(-ka) + p_{j_p}(\mathbf{x})\ell(ka)] \,|_{a=0} = (1 - p_{j_p}(\mathbf{x}))(-k)\ell'(0) + p_{j_p}(\mathbf{x})k\ell'(0)$ is negative when $p_{j_p} > \frac{1}{2}$.

To prove part b, we first reduce our problem to minimizing $\sum_{j=1}^k p_j \ell(g_j)$. Without loss of generality, assume that $j_p = k$ and $f_1 \le f_2 \le \cdots \le f_k$. Then $\sum_{j=1}^k p_j \ell(g_j) = \sum_{j=1}^{k-1} p_j \ell(f_j - f_k) + p_k \ell(f_k - f_{k-1}) \ge \ell(\sum_{j=1}^{k-1} p_j(f_j - f_k) + p_k(f_k - f_{k-1})) \ge \ell((1 - p_k)(f_{k-1} - f_k) + p_k(f_k - f_{k-1})) = \ell((1-2p_k)(f_{k-1} - f_k)) \ge \ell(0)$, where $\ell(0)$ is the loss of $\mathbf{f} = \mathbf{0}$. Here the first inequality is due to the convexity of ℓ, the second is because ℓ is nonincreasing, and the third is because $p_k \le 1/2$ and $f_{k-1} \le f_k$. Thus $\mathbf{f}^* = \mathbf{0}$ is a minimizer of $\sum_{j=1}^k p_j \ell(g_j)$. The desired results of the proposition then follow.

Proof of Theorem 1:

Note that $E[\ell_{T_s}(\min \mathbf{g}(\mathbf{f}(\mathbf{X}), Y))] = E[\sum_{j=1}^k \ell_{T_s}(\min \mathbf{g}(\mathbf{f}(\mathbf{X}), j) p_j(\mathbf{X}))]$. For any given $\mathbf{x}$, we need to minimize $\sum_{j=1}^k \ell_{T_s}(g_j) p_j$, where $g_j = \min \mathbf{g}(\mathbf{f}(\mathbf{x}), j)$. By definition and the fact that $\sum_{j=1}^k f_j = 0$, we can conclude that $\max_j g_j \ge 0$ and at most one of the g_j's is positive. Assume that $j_p = \arg\max_j p_j(\mathbf{x})$ is unique. Then using the nonincreasing property of ℓ_{T_s} and $\ell'(0) < 0$, the minimizer $\mathbf{f}^*$ satisfies that $g^*_{j_p} \ge 0$.

We are now left to show $g^*_{j_p} \ne 0$ or, equivalently, that $\mathbf{0}$ cannot be a minimizer. Without loss of generality, assume that $p_{j_p} > 1/k$. Then it is sufficient to show that there exists a solution with $g_{j_p} > 0$. By assumption, there exists $u_1 > 0$ such that $u_1 \ge -s$ and $(\ell(0) - \ell(u_1))/(\ell(s) - \ell(0)) \ge k - 1$. Consider a solution $\mathbf{f}^0$ with $f^0_{j_p} = u_1(k-1)/k$ and $f^0_j = -u_1/k$ for $j \ne j_p$. We want to show that $\mathbf{f}^0$ yields a smaller expected loss than $\mathbf{0}$, that is, $p_{j_p}\ell_{T_s}(u_1) + (1 - p_{j_p})\ell_{T_s}(-u_1) < \ell_{T_s}(0)$. Equivalently, $(\ell(0) - \ell(u_1))/(\ell(s) - \ell(0)) > (1 - p_{j_p})/p_{j_p}$, which holds due to the fact that $(1 - p_{j_p})/p_{j_p} < (k-1)$. This implies sufficiency of the condition.

To prove necessity of the condition, it is sufficient to show that if $(\ell(0) - \ell(u))/(\ell(s) - \ell(0)) < (k-1)$ for all u with $-u \le s \le 0$, then $\mathbf{0}$ is a minimizer of $\sum_{j=1}^k \ell_{T_s}(g_j) p_j$. Equivalently, we need to show that there exists $(p_1, \dots, p_k)$ such that $\sum_{j=1}^k \ell_{T_s}(g_j) p_j \ge \ell_{T_s}(0)$ for all $\mathbf{f}$. Without loss of generality, assume that $j_p = k$ and $f_1 \le f_2 \le \cdots \le f_k$. Then $\sum_{j=1}^k p_j \ell_{T_s}(g_j) = \sum_{j=1}^{k-1} p_j \ell_{T_s}(f_j - f_k) + p_k \ell_{T_s}(f_k - f_{k-1}) \ge (1 - p_k)\ell_{T_s}(f_{k-1} - f_k) + p_k \ell_{T_s}(f_k - f_{k-1})$, because ℓ_{T_s} is nonincreasing. Thus it is sufficient to show that $p_k \ell_{T_s}(u) + (1 - p_k)\ell_{T_s}(-u) > \ell_{T_s}(0)$ for all $u > 0$, that is, $(1 - p_k)(\ell_{T_s}(-u) - \ell(0)) > p_k(\ell(0) - \ell(u))$. Because $\ell(\min(\mathbf{g}(\mathbf{f}(\mathbf{x}), y)))$ with convex $\ell(\cdot)$ may not be Fisher-consistent for $k > 2$ (Prop. 2), we need only consider $s \ge -u$, which implies that $\ell_{T_s}(-u) = \ell(s)$. By assumption, we can set $(\ell(s) - \ell(0)) = (\ell(0) - \ell(u))/(k-1) + a$ for some $a > 0$. Denote $(\ell(0) - \ell(u)) = A$. Then we need to have $(1 - p_k)(A/(k-1) + a) > p_k A$. Let $p_k = 1/k + \epsilon$. Then it becomes $((k-1)/k - \epsilon)(A/(k-1) + a) > (1/k + \epsilon)A$ or, equivalently,

$$a\frac{k-1}{k\epsilon} > \frac{k}{k-1}A + a. \tag{A.1}$$

For any given $a > 0$ and $A > 0$, we can always find a small $\epsilon > 0$ to satisfy (A.1). The desired result then follows.

Derivation of the Dual Problem in Section 3.2

Note that $\frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta)$ and $\frac{\partial}{\partial b_j} Q^s_{cav}(\Theta)$ can be written as

$$-C\Bigg[\sum_{i:y_i=j}\big(-I_{\{\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i) < s\}}\big)\mathbf{x}_i^T + \sum_{i:y_i\ne j}\big(I_{\{j=\arg\max(f_{j'}(\mathbf{x}_i):j'\ne y_i),\, f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i) < s\}}\big)\mathbf{x}_i^T\Bigg]$$

and

$$-C\Bigg[\sum_{i:y_i=j}\big(-I_{\{\min \mathbf{g}(\mathbf{f}(\mathbf{x}_i), y_i) < s\}}\big) + \sum_{i:y_i\ne j}\big(I_{\{j=\arg\max(f_{j'}(\mathbf{x}_i):j'\ne y_i),\, f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i) < s\}}\big)\Bigg],$$

where $I_{\{A\}} = 1$ if event A is true and 0 if otherwise. Using the definition of β_{ij}, we have $\frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta) = \sum_{i:y_i=j}(\sum_{j'\ne y_i}\beta_{ij'})\mathbf{x}_i^T - \sum_{i:y_i\ne j}\beta_{ij}\mathbf{x}_i^T$ and $\frac{\partial}{\partial b_j} Q^s_{cav}(\Theta) = \sum_{i:y_i=j}(\sum_{j'\ne y_i}\beta_{ij'}) - \sum_{i:y_i\ne j}\beta_{ij}$.

Applying the first-order approximation to the concave part, the objective function at step $(t+1)$ becomes $Q^s(\Theta) = Q^s_{vex}(\Theta) + \sum_{j=1}^k \langle \frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta_t), \boldsymbol{w}_j\rangle + \sum_{j=1}^k b_j \frac{\partial}{\partial b_j} Q^s_{cav}(\Theta_t)$, where Θ_t is the current solution. Using slack variables ξ_i's for the hinge loss function, the optimization problem at step $(t+1)$ becomes

$$\min_{\boldsymbol{W},\mathbf{b},\boldsymbol{\xi}} \frac{1}{2}\sum_{j=1}^k \|\boldsymbol{w}_j\|_2^2 + C\sum_{i=1}^n \xi_i + \sum_{j=1}^k \Big\langle \frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta_t), \boldsymbol{w}_j \Big\rangle + \sum_{j=1}^k b_j \frac{\partial}{\partial b_j} Q^s_{cav}(\Theta_t)$$

$$\text{subject to}\quad \xi_i \ge 0, \qquad i = 1, 2, \dots, n,$$
$$\xi_i \ge 1 - [\mathbf{x}_i^T \boldsymbol{w}_{y_i} + b_{y_i}] + [\mathbf{x}_i^T \boldsymbol{w}_j + b_j],$$
$$i = 1, 2, \dots, n;\ j \ne y_i.$$

The corresponding Lagrangian is

$$\begin{aligned} L(\boldsymbol{W}, \mathbf{b}, \boldsymbol{\xi}) &= \frac{1}{2}\sum_{j=1}^k \|\boldsymbol{w}_j\|_2^2 + \sum_{j=1}^k \Big\langle \frac{\partial}{\partial \boldsymbol{w}_j} Q^s_{cav}(\Theta_t), \boldsymbol{w}_j \Big\rangle \\ &\quad + \sum_{j=1}^k b_j \frac{\partial}{\partial b_j} Q^s_{cav}(\Theta_t) + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n u_i \xi_i \\ &\quad - \sum_{i=1}^n \sum_{j'\ne y_i} \alpha_{ij'}\big(\mathbf{x}_i^T \boldsymbol{w}_{y_i} + b_{y_i} - \mathbf{x}_i^T \boldsymbol{w}_{j'} - b_{j'} + \xi_i - 1\big), \end{aligned} \tag{A.2}$$

subject to

$$\frac{\partial}{\partial \boldsymbol{w}_j} L = \boldsymbol{w}_j^T - \Bigg[\sum_{i:y_i=j}\sum_{j'\ne y_i}(\alpha_{ij'} - \beta_{ij'})\mathbf{x}_i^T - \sum_{i:y_i\ne j}(\alpha_{ij} - \beta_{ij})\mathbf{x}_i^T\Bigg] = 0, \tag{A.3}$$

$$\frac{\partial}{\partial b_j} L = -\Bigg[\sum_{i:y_i=j}\sum_{j'\ne y_i}(\alpha_{ij'} - \beta_{ij'}) - \sum_{i:y_i\ne j}(\alpha_{ij} - \beta_{ij})\Bigg] = 0, \tag{A.4}$$

$$\frac{\partial}{\partial \xi_i} L = C - u_i - \sum_{j\ne y_i}\alpha_{ij} = 0, \tag{A.5}$$

where the Lagrangian multipliers are $u_i \ge 0$ and $\alpha_{ij'} \ge 0$ for any $i = 1, 2, \dots, n$, $j' \ne y_i$. Substituting (A.3)–(A.5) into (A.2) yields the desired dual problem specified in Section 3.2.

[Received August 2006. Revised April 2007.]

REFERENCES

An, L. T. H., and Tao, P. D. (1997), "Solving a Class of Linearly Constrained Indefinite Quadratic Problems by D.C. Algorithms," *Journal of Global Optimization*, 11, 253–285.

Bartlett, P., Jordan, M., and McAuliffe, J. (2006), "Convexity, Classification, and Risk Bounds," *Journal of the American Statistical Association*, 101, 138–156.

Bredensteiner, E., and Bennett, K. (1999), "Multicategory Classification by Support Vector Machines," *Computational Optimizations and Applications*, 12, 53–79.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., and Haussler, D. (2000), "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines," *Proceedings of the National Academy of Science*, 97, 262–267.

Collobert, R., Sinz, F., Weston, J., and Bottou, L. (2006), "Trading Convexity for Scalability," in *Proceedings of the 23rd International Conference on Machine Learning*, Vol. 3, pp. 201–208.

Crammer, K., and Singer, Y. (2001), "On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines," *Journal of Machine Learning Research*, 2, 265–292.

Cristianini, N., and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines*, Cambridge, U.K.: Cambridge University Press.

Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360.

——— (2006), "Statistical Challenges With High Dimensionality: Feature Selection in Knowledge Discovery," in *Proceedings of the International Congress of Mathematicians*, Vol. 3, pp. 595–622.

Fan, J., and Peng, H. (2004), "Nonconcave Penalized Likelihood With a Diverging Number of Parameters," *The Annals of Statistics*, 32, 928–961.

Friedman, J., Hastie, T., and Tibshirani, R. (2000), "Additive Logistic Regression: A Statistical View of Boosting," *The Annals of Statistics*, 28, 337–407.

Hunter, D., and Li, R. (2005), "Variable Selection Using MM Algorithms," *The Annals of Statistics*, 33, 1617–1642.

Kimeldorf, G., and Wahba, G. (1971), "Some Results on Tchebycheffian Spline Functions," *Journal of Mathematical Analysis and Applications*, 33, 82–95.

Lee, Y., Lin, Y., and Wahba, G. (2004), "Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data," *Journal of the American Statistical Association*, 99, 67–81.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000), "Smoothing Spline ANOVA Models for Large Data Sets With Bernoulli Observations and the Randomized GACV," *The Annals of Statistics*, 28, 1570–1600.

Lin, Y. (2002), "Support Vector Machines and the Bayes Rule in Classification," *Data Mining and Knowledge Discovery*, 6, 259–275.

——— (2004), "A Note on Margin-Based Loss Functions in Classification," *Statistics and Probability Letters*, 68, 73–82.

Liu, Y., and Shen, X. (2006), "Multicategory ψ-Learning," *Journal of the American Statistical Association*, 101, 500–509.

Liu, Y., Shen, X., and Doss, H. (2005), "Multicategory ψ-Learning and Support Vector Machine: Computational Tools," *Journal of Computational and Graphical Statistics*, 14, 219–236.

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

Tewari, A., and Bartlett, P. (2005), "On the Consistency of Multiclass Classification Methods," in *Proceedings of the 18th Annual Conference on Learning Theory*, pp. 143–157.

Vapnik, V. (1998), *Statistical Learning Theory*, New York: Wiley.

Wahba, G. (1999), "Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV," in *Advances in Kernel Methods Support Vector Learning*, eds. B. Schoelkopf, C. Burges, and A. Smola, Cambridge, MA: MIT Press, pp. 69–88.

Weston, J., and Watkins, C. (1999), "Support Vector Machines for Multi-Class Pattern Recognition," in *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, pp. 219–224.

Yuan, M., and Lin, Y. (2006), "Model Selection and Estimation in Regression With Grouped Variables," *Journal of the Royal Statistical Society*, Ser. B, 68, 49–67.

Zhang, H. H. (2006), "Variable Selection for Support Vector Machines via Smoothing Spline ANOVA," *Statistica Sinica*, 16, 659–674.

Zhang, H. H., Ahn, J., Lin, X., and Park, C. (2006), "Gene Selection Using Support Vector Machines With Nonconvex Penalty," *Bioinformatics*, 22, 88–95.

Zhang, T. (2004), "Statistical Analysis of Some Multi-Category Large Margin Classification Methods," *Journal of Machine Learning Research*, 5, 1225–1251.

Zhu, J., and Hastie, T. (2005), "Kernel Logistic Regression and the Import Vector Machine," *Journal of Computational and Graphical Statistics*, 14, 185–205.

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003), "1-Norm Support Vector Machines," *Neural Information Processing Systems*, 16, available at *http://books.nips.cc/nips16.html*.

Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429.