On Multicategory Truncated-Hinge-Loss Support Vector Machines

Yichao Wu and Yufeng Liu

Abstract. With its elegant margin theory and accurate classification performance, the Support Vector Machine (SVM) has been widely applied in both machine learning and statistics. Despite its success and popularity, it still 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 solve these problems, [WL06] proposed a new SVM variant, the robust truncated-hinge-loss SVM (RSVM), which uses a truncated hinge loss. In this paper, we apply the operation of truncation on the multicategory hinge loss proposed by [LLW04]. We show that the proposed robust multicategory truncated-hinge-loss SVM (RMSVM) is more robust to outliers and deliver more accurate classifiers using a smaller set of SVs than the original multicategory SVM (MSVM) proposed by [LLW04].

1. Introduction

As a supervised learning technique, classification is an important tool for statistical data analysis. Among many classification methods, the Support Vector Machine (SVM) is a popular one and has enjoyed great success in many applications [Vap98, CST00]. The SVM was first invented by Vapnik and his colleagues using an elegant large margin theory. It is now known that the SVM can be fit in the regularization framework of $Loss + Penalty$ using the hinge loss [Wah99]. In the regularization framework, the loss function is used to ensure fidelity of the resulting model to the data. The penalty term in regularization helps to avoid overfitting of the resulting model. A tuning parameter is typically used to balance these two components. Besides the SVM, many other classification methods belong to the regularization framework. For example, the penalized logistic regression [LWX$^+$00, ZH05] and the AdaBoost [FHT00] use the logistic loss and the exponential loss respectively.

Despite its success, the SVM has been shown to have some drawbacks for difficult learning problems [WL06]. One drawback of the SVM classifier is that it tends to be sensitive to noisy training data. The reason is because SVM uses

1991 *Mathematics Subject Classification.* Primary 54C40, 14E20; Secondary 46E25, 20C20.

Key words and phrases. Classification, d.c. Algorithm, Fisher consistency, regularization, support vectors, truncation.

Liu is the corresponding author, and he is partially supported by Grant DMS-0606577 from the National Science Foundation and the UNC Junior Faculty Development Award.

the unbounded hinge loss and consequently the resulting classifiers may be affected by points far away from their own classes, namely "outliers" in the training data. Another drawback of the SVM is that the number of SVs can be very large for many problems, especially for difficult classification problems or problems with a large number of input variables. To overcome these problems, [WL06] suggested to truncate the hinge loss and proposed the robust truncated-hinge-loss SVM (RSVM) based on the bounded truncated hinge loss. They showed that the RSVM is more robust to outliers using a smaller set of SVs than the original SVM.

In this paper, we focus on multicategory SVM (MSVM) and apply the operation of truncation on the multicategory hinge loss proposed by [LLW04]. We show that the proposed truncated multicategory hinge loss preserves Fisher consistency. Moreover, the proposed robust multicategory truncated-hinge-loss SVM (RMSVM) is more robust to outliers in the training data than the original MSVM. Furthermore, the RMSVM retains the SV interpretation and it often selects much fewer number of SVs than the MSVM.

Although truncation helps to robustify the MSVM, the associated optimization problem becomes nonconvex minimization. We propose to apply the d.c. algorithm to solve the nonconvex problem via a sequence of convex subproblems. Our numerical experience suggests that the d.c. algorithm works effectively.

The rest of the paper is organized as follows: In Section 2, we briefly review the SVM methodology and introduce the RMSVM. In Section 3, we develop a numerical algorithm for the RMSVM via the d.c. algorithm. We also give the SV interpretation of the RMSVM. In Section 4, we present numerical examples to demonstrate effectiveness of the truncated hinge loss. We conclude the paper with Section 5.

2. Multicategory Support Vector Machine and Its Robust Variant

2.1. Multicategory Support Vector Machine. For a k-class classification problem, we are given a training sample $\{(\boldsymbol{x}_i, y_i) : i = 1, 2, \cdots, n\}$ which is distributed according to some unknown probability distribution function $P(\boldsymbol{x}, y)$, with $p_j(\boldsymbol{x}) = P(Y = j|\boldsymbol{X} = \boldsymbol{x})$. Here $\boldsymbol{x}_i \in \mathcal{S} \subset \Re^d$ and y_i; $i = 1, \ldots, n$, denote the input vectors and output labels respectively, where n is the sample size, and d is the dimensionality of the input space. We label y as $\{1, 2, \cdots, k\}$. Clearly, the Bayes rule is given by $\text{argmax}_{j=1,\ldots,k} p_j(\boldsymbol{x})$ which delivers the minimum expected misclassification rate.

Denote $\boldsymbol{f} = (f_1, f_2, \cdots, f_k)$ to be the decision function vector, where each component represents one class and maps from $\mathcal{S}$ to $\Re$. We use $\text{argmax}_{j=1,\ldots,k} f_j$ as the classifier which classifies a new input vector $\boldsymbol{x}$ into the class with the largest $f_j(\boldsymbol{x})$. Let $f_j(\boldsymbol{x}) = h_j(\boldsymbol{x}) + b_j$ with $h_j \in H_K$, where H_K is a reproducing kernel Hilbert space (RKHS) induced by the positive definite kernel $K(\cdot, \cdot)$.

To achieve multicategory classification, the standard MSVM proposed by [LLW04] solves the following optimization problem

$$\min_{\boldsymbol{f}} \frac{C}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} I(y_i \neq j)[1 + f_j(\boldsymbol{x}_i)]_+ + \frac{1}{2} \sum_{j=1}^{k} \|f_j\|^2, \tag{2.1}$$

under the sum-to-zero constraint $\sum_{j=1}^{k} f_j(\boldsymbol{x}) = 0$, where $C > 0$ is a tuning parameter. Note that nonstandard learning can be achieved by assigning different misclassification costs. For simplicity, we will focus on standard learning in this article.

The critical aspect of the MSVM formulation in (2.1) is the multicategory hinge loss $\sum_{j=1}^k I(y \neq j)[1 + f_j(\boldsymbol{x})]_+$ subject to $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$. It can be showed that the minimizer of $E[\sum_{j=1}^k I(Y \neq j)[1 + f_j(\boldsymbol{X})]_+]$ subject to the sum-to-zero constraint is $\boldsymbol{f}^*(\boldsymbol{x})$ with $f_j(\boldsymbol{x}) = k-1$ if $j = \text{argmax}_{j=1,\dots,k} P_j(\boldsymbol{x})$ and -1 otherwise [LLW04]. As a result, the MSVM formulation in (2.1) yields an extension of the binary SVM with Fisher consistency [c.f. Lin02].

Using the representer theorem [KW71, Wah99], $f_j(\boldsymbol{x})$ can represented as $b_j + \sum_{i'=1}^n K(\boldsymbol{x}, \boldsymbol{x}_{i'}) v_{i'j}$. Thus we have

$$f_j(\boldsymbol{x}_i) = b_j + \sum_{i'=1}^n K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) v_{i'j} = b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j}, \tag{2.2}$$

where $\boldsymbol{K}_i = (K(\boldsymbol{x}_i, \boldsymbol{x}_1), K(\boldsymbol{x}_i, \boldsymbol{x}_2), \cdots, K(\boldsymbol{x}_i, \boldsymbol{x}_n))^T$ and $\boldsymbol{v}_{\cdot j} = (v_{1j}, \dots, v_{nj})^T$. After plugging (2.2) into (2.1), the MSVM problem in (2.1) becomes

$$\min_{\{\boldsymbol{v}_{\cdot j}, b_j; j=1,\dots,k\}} \frac{C}{n} \sum_{i=1}^n \sum_{j \neq y_i} [b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} + 1]_+ + \sum_{j=1}^k \frac{1}{2} \boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j}, \tag{2.3}$$

$$\text{subject to } \boldsymbol{e} \sum_{j=1}^k b_j + \boldsymbol{K} \sum_{j=1}^k \boldsymbol{v}_{\cdot j} = \boldsymbol{0}. \tag{2.4}$$

where $\boldsymbol{K}$ denotes the kernel matrix with the (i, i') element being $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ and $\boldsymbol{e} = (1, 1, \cdots, 1)^T$ is a vector of length n. Problem (2.3) can be solved using quadratic programming (QP) in a similar way as the binary SVM.

2.2. Robust Truncated-Hinge-Loss MSVM. Denote $H_{-1}(u) = [1 + u]_+$. Then the multicategory hinge loss in (2.1) can expressed as $\sum_{j=1}^k I(y \neq j)[1 + f_j(\boldsymbol{x})]_+ = \sum_{j=1}^k I(y \neq j) H_{-1}(f_j(\boldsymbol{x}))$ subject to $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$. For a given training pair $(\boldsymbol{x}, y)$, this loss uses $H_{-1}(f_j(\boldsymbol{x}))$ to encourage $f_j(\boldsymbol{x})$; $j \neq y$, to be negative and thus $f_y(\boldsymbol{x})$ to be positive by the sum-to-zero constraint. Notice that $H_{-1}(u)$ increases linearly with u when $u \geq -1$. This implies that the loss will put emphasis on untypical points which are far away from their own classes, namely outliers. This is undesirable since the resulting classification boundary should not be greatly influenced by outliers.

To overcome the difficulty of outliers, we consider to decrease the impact of outliers by truncating the unbounded loss function. In particular, we consider to truncate $H_{-1}(u)$. Denote $H_s(u) = [u - s]_+$ and define a truncated loss $T_s(u) = H_{-1}(u) - H_s(u)$. Figure 1 displays the loss funcations $H_{-1}(u)$ and $T_s(u)$. As we can see from the plot, $T_s(u)$ becomes flat once $u \geq s$. Consequently, $T_s(f_j(\boldsymbol{x}))$ treats $f_j(\boldsymbol{x})$; $j \neq y$, equally once it is greater than s and therefore it may yield more robust classifiers than the original function H_{-1}. Using the truncated loss function T_s, the new proposed RMSVM solves the following optimization problem

$$\min_{\boldsymbol{f}} \frac{C}{n} \sum_{i=1}^n \sum_{j=1}^k I(y_i \neq j) T_s(f_j(\boldsymbol{x}_i)) + \frac{1}{2} \sum_{j=1}^k \|f_j\|^2, \tag{2.5}$$

under the sum-to-zero constraint $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$.

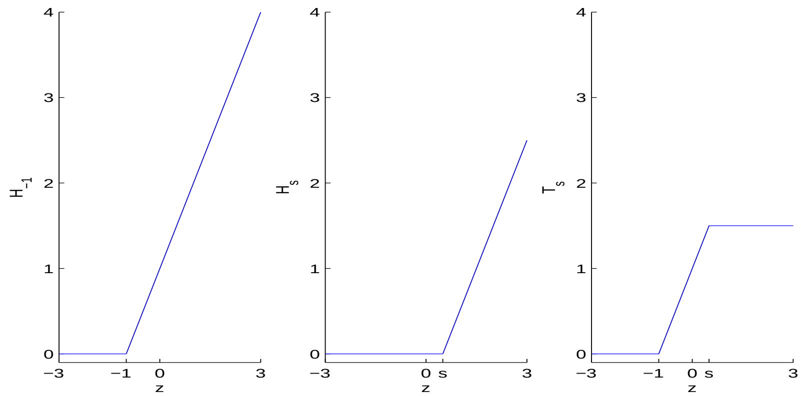

FIGURE 1. The left, middle, and right panels display functions $H_{-1}(u)$, $H_s(u)$, and $T_s(u)$ respectively.

The following theorem states Fisher consistency of the proposed truncated multicategory hinge loss:

THEOREM 2.1. *The minimizer* $\boldsymbol{f}^*$ *of* $E[\sum_{j=1}^k I(Y \neq j)T_s(f_j(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]$ *subject to* $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$ *satisfies that* $argmax_j f_j^* = argmax_j p_j$*, for any* $s \geq 0$.

PROOF. Without loss of generality, we assume $\max_j p_j > 1/k$ and $\text{argmax}_j p_j$ is unique, that is $\max_j p_j \neq$ second $\max_j p_j$. Denote $l_p = \text{argmax}_j p_j$.

Note that $E[\sum_{j=1}^k I(Y \neq j)T_s(f_j(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}] = \sum_{l=1}^k p_l(\boldsymbol{x}) \sum_{j\neq l} T_s(f_j(\boldsymbol{x}))$. Since T_s is a non-decreasing function, we can conclude that $f_{l_p}^* \geq \max_{j \neq l_p} f_j^*$. Then it is sufficient to show that $f_{l_p}^* > 0$ and $f_j^* \leq 0$ for $j \neq l_p$. We will show it in two steps: (1). $f_{l_p}^* > 0$. (2). $f_j^* \leq 0$ for $j \neq l_p$.

To show part (1), it is equivalent to showing that $f_{l_p}^* \neq 0$ since $f_{l_p}^* = \max_j f_j^*$ and $\sum_{j=1}^k f_j = 0$. Suppose $\boldsymbol{f}^* = \boldsymbol{0}$. Then $\sum_{l=1}^k p_l(\boldsymbol{x}) \sum_{j\neq l} T_s(f_j^*(\boldsymbol{x})) = k-1$. Consider another solution $\boldsymbol{f}^0$ with $f_{l_p}^0 = k-1$ and $f_j^0 = -1$ for $j \neq l_p$. Then $\sum_{l=1}^k p_l(\boldsymbol{x}) \sum_{j\neq l} T_s(f_j^0(\boldsymbol{x})) = \sum_{j\neq l_p} p_j T_s(k-1) = T_s(k-1)(1-p_{l_p}) < (k-1)$. Thus $f_{l_p}^* > 0$.

For part (2), we show that $\boldsymbol{f}^1$ with $f_{l_1}^1 > 0; l_1 \neq l_p$, cannot be the minimizer $\boldsymbol{f}^*$. To this end, consider another solution $\tilde{\boldsymbol{f}}^1$ with $\tilde{f}_{l_p}^1 = f_{l_p}^1 + f_{l_1}^1 + 1$, $\tilde{f}_{l_1}^1 = -1$, and $\tilde{f}_j^1 = f_j^1; j \neq l_p, j \neq l_1$. Then $\sum_{l=1}^k p_l(\boldsymbol{x}) \sum_{j\neq l} T_s(\tilde{f}_j^1(\boldsymbol{x})) - \sum_{l=1}^k p_l(\boldsymbol{x}) \sum_{j\neq l} T_s(f_j^1(\boldsymbol{x})) = (1-p_{l_p})[T(\tilde{f}_{l_p}) - T(f_{l_p})] - (1-p_{l_1})[T(f_{l_1}) - T(\tilde{f}_{l_1})] = A$. The desired result follows from the fact that $A < 0$ as shown in the following four cases:

- $s \geq \tilde{f}_{l_p}^1$: $A = (1-p_{l_p})(f_l^1+1) - (1-p_{l_1})(f_1^1+1) < 0$.
- $f_{l_p}^1 \leq s < \tilde{f}_{l_p}^1$: $A < (1-p_{l_p})(f_l^1+1) - (1-p_{l_1})(f_1^1+1) < 0$.
- $f_{l_1}^1 \leq s < f_{l_p}^1$: $A = -(1-p_{l_1})(f_1^1+1) < 0$.
- $0 \leq s < f_{l_1}^1$: $A = -(1-p_{l_1})(s+1) < 0$.

□

Analogous to the MSVM formulation in (2.3), the proposed RMSVM is equivalent to solving the following minimization problem:

$$\min_{\{\boldsymbol{v}_{\cdot j}, b_j\}_{j=1}^k} \frac{C}{n}\sum_{i=1}^n \sum_{j \neq y_i} \left([b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} + 1]_+ - [b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} - s]_+\right) + \frac{1}{2}\boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j} \tag{2.6}$$

$$\text{subject to } \boldsymbol{e}\sum_{j=1}^k b_j + \boldsymbol{K}\sum_{j=1}^k \boldsymbol{v}_{\cdot j} = \boldsymbol{0}, \tag{2.7}$$

where $s \geq 0$ denotes the location of the truncation.

3. D.C. Algorithm

Since the truncated loss T_s is nonconvex, the optimization problem in (2.6) involves nonconvex minimization. Notice that T_s can be decomposed as the difference of two convex functions, H_{-1} and H_s. Using this property of the new loss function, we propose to apply the the difference convex (d.c.) algorithm [AT97, LSD05] to solve the nonconvex optimization problem of the RMSVM. The d.c. algorithm solves the nonconvex minimization problem via minimizing a sequence of convex subproblems. As shown in [LSW05], the d.c. algorithm converges in finite steps and yields a local optimal solution of the original nonconvex minimization problem.

Next, we derive the d.c. algorithm for the proposed RMSVM and implement it via a sequence of QP. For simplicity of the notation, denote Θ as $\{\boldsymbol{v}_j, b_j\}_{j=1}^k$. Then we break the objective function in (2.6) into two parts:

$$\psi_{vex}(\Theta) = \frac{C}{n}\sum_{i=1}^n \sum_{j \neq y_i} [b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} + 1]_+ + \frac{1}{2}\boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j} \tag{3.1}$$

$$\psi_{cav}(\Theta) = -\frac{C}{n}\sum_{i=1}^n \sum_{j \neq y_i} [b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} - s]_+. \tag{3.2}$$

To apply the d.c. algorithm, we use a linear approximation on the concave part in the objective function. It is easy to see that

$$\frac{\partial \psi_{cav}(\Theta)}{\partial \boldsymbol{v}_{\cdot j}} = -\frac{C}{n}\sum_{i: y_i \neq j} \beta_{ij}\boldsymbol{K}_i = -\frac{C}{n}\boldsymbol{K}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j}) \tag{3.3}$$

$$\frac{\partial \psi_{cav}(\Theta)}{\partial b_j} = -\frac{C}{n}\sum_{i: y_i \neq j} \beta_{ij} = -\frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})^T \boldsymbol{e}, \tag{3.4}$$

where $\boldsymbol{L}_{\cdot j} = (L_{1j}, L_{2j}, \cdots, L_{nj})^T$, $\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j}$ denotes componentwise product, and

$$\beta_{ij} = \begin{cases} 1 & \text{if } b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} - s > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Given the solution $\boldsymbol{f}^t$ at the t-th iteration, the objective at the $(t+1)$-th iteration can be approximated by

$$\frac{C}{n}\sum_{i=1}^n \sum_{j \neq y_i} [b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} + 1]_+ + \frac{1}{2}\boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j} + \sum_{j=1}^k \left(b_j \frac{\partial \psi_{cav}(\Theta^t)}{\partial b_j} + \left\langle \frac{\partial \psi_{cav}(\Theta^t)}{\partial \boldsymbol{v}_{\cdot j}}, \boldsymbol{v}_{\cdot j}\right\rangle\right). \tag{3.5}$$

Next we will convert (3.5) into a QP problem. To this end, we define $\boldsymbol{L}$ to be a matrix with its (i,j)-th element $L_{ij} = I_{y_i \neq j}$, where I_A is the indication function which takes value 1 if A is true and 0 otherwise. Then at the $(t+1)$-th iteration, the d.c. algorithm of our RMSVM solves the following primal problem:

$$\min_{\Theta} \frac{C}{n}\sum_{j=1}^{k} \boldsymbol{L}_{\cdot j}^T \xi_{\cdot j} + \frac{1}{2}\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j} + \sum_{j=1}^{k}(b_j \frac{\partial \psi_{cav}(\Theta^t)}{\partial b_j} + \left\langle \frac{\partial \psi_{cav}(\Theta^t)}{\partial \boldsymbol{v}_{\cdot j}}, \boldsymbol{v}_{\cdot j} \right\rangle) \tag{3.6}$$

$$\text{s.t. } \xi_{ij} \geq 0 \tag{3.7}$$

$$\xi_{ij} - (b_j + \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} + 1) \geq 0 \tag{3.8}$$

$$(\sum_{j=1}^{k} b_j)\boldsymbol{e} + \boldsymbol{K}(\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}) = \boldsymbol{0}. \tag{3.9}$$

The corresponding Lagrangian function is

$$\begin{aligned}
L_D \;=\;& \frac{C}{n}\sum_{i=1}^{n}\sum_{j\neq y_i} \xi_{ij} - \sum_{i=1}^{n}\sum_{j\neq y_i} \gamma_{ij}\xi_{ij} - \sum_{i=1}^{n}\sum_{j\neq y_i} \alpha_{ij}(\xi_{ij} - b_j - \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j} - 1) \\
&+ \frac{1}{2}\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j} + \sum_{j=1}^{k}(b_j \frac{\partial \psi_{cav}(\Theta^t)}{\partial b_j} + \left\langle \frac{\partial \psi_{cav}(\Theta^t)}{\boldsymbol{v}_{\cdot j}}, \boldsymbol{v}_{\cdot j} \right\rangle) \\
&+ \boldsymbol{\delta}^T(\boldsymbol{K}(\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}) + (\sum_{j=1}^{k} b_j)\boldsymbol{e}) \\
=\;& \sum_{i=1}^{n}\sum_{j\neq y_i}(\frac{C}{n} - \gamma_{ij} - \alpha_{ij})\xi_{ij} + \sum_{j=1}^{k} b_j(\frac{\partial \psi_{cav}(\Theta^t)}{\partial b_j} + \sum_{i: y_i \neq j} \alpha_{ij} + \boldsymbol{\delta}^T \boldsymbol{e}) \\
&+ \sum_{j=1}^{k} \left\langle \sum_{i: y_i \neq j} \alpha_{ij}\boldsymbol{K}_i + \frac{\partial \psi_{cav}(\Theta^t)}{\boldsymbol{v}_{\cdot j}} + \boldsymbol{K}\boldsymbol{\delta}, \boldsymbol{v}_{\cdot j} \right\rangle + \sum_{i=1}^{n}\sum_{j: j\neq y_i} \alpha_{ij} + \frac{1}{2}\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j}, \\
=\;& \sum_{i=1}^{n}\sum_{j\neq y_i}(\frac{C}{n} - \gamma_{ij} - \alpha_{ij})\xi_{ij} + \sum_{j=1}^{k} b_j(-\frac{C}{n}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})^T \boldsymbol{e} + (\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})^T \boldsymbol{e} + \boldsymbol{\delta}^T \boldsymbol{e}) \\
&+ \sum_{j=1}^{k} \left\langle \boldsymbol{K}(\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j}) - \frac{C}{n}\boldsymbol{K}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j}) + \boldsymbol{K}\boldsymbol{\delta}, \boldsymbol{v}_{\cdot j} \right\rangle + \sum_{i=1}^{n}\sum_{j: j\neq y_i} \alpha_{ij} + \frac{1}{2}\sum_{j=1}^{k} \boldsymbol{v}_{\cdot j}^T \boldsymbol{K} \boldsymbol{v}_{\cdot j},
\end{aligned}$$

where $\alpha_{ij} \geq 0$ and $\gamma_{ij} \geq 0$, $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_n)^T$ are Lagrangian coefficients.

Solving $\frac{\partial L_D}{\partial \xi_{ij}} = 0$, $\frac{\partial L_D}{\partial b_j} = 0$, and $\frac{\partial L_D}{\partial \boldsymbol{v}_{\cdot j}} = 0$, we can get the corresponding dual QP problem. In particular, $\frac{\partial L_D}{\partial \boldsymbol{v}_{\cdot j}} = 0$ implies that $\boldsymbol{K}\boldsymbol{v}_{\cdot j} + \boldsymbol{K}(\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j}) + \boldsymbol{K}\boldsymbol{\delta} - \frac{C}{n}\boldsymbol{K}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j}) = \boldsymbol{0}$. Due to the positive definitive kernel $K(\cdot,\cdot)$, this implies that

$$\boldsymbol{v}_{\cdot j} = -(\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})). \tag{3.10}$$

Setting $\boldsymbol{\delta} = -\frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j} - \frac{C}{n}\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})$, $\bar{\boldsymbol{\alpha}} = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})$, $\bar{\boldsymbol{\beta}} = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})$, and $\boldsymbol{\delta} = -(\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}})$, we have

$$\boldsymbol{v}_{\cdot j} = -[\boldsymbol{\alpha}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j} - (\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}}) - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j}\cdot \boldsymbol{L}_{\cdot j})]. \tag{3.11}$$

After plugging (3.11) into L_D, we can rewrite the objective function of the corresponding dual problem as follows

$$
\begin{aligned}
& \sum_{j=1}^{k} \left\langle \boldsymbol{K}(\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j}) + \boldsymbol{K}\boldsymbol{\delta} - \frac{C}{n}\boldsymbol{K}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j}), -(\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})) \right\rangle \\
& + \frac{1}{2}\sum_{j=1}^{k} \left\langle -(\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})), -\boldsymbol{K}(\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})) \right\rangle \\
& + \sum_{i=1}^{n}\sum_{j \neq y_i} \alpha_{ij} \\
= \quad & -\frac{1}{2}\sum_{j=1}^{k} \left\langle (\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})), \boldsymbol{K}(\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} + \boldsymbol{\delta} - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})) \right\rangle \\
& + \sum_{i=1}^{n}\sum_{j \neq y_i} \alpha_{ij} + \text{Constant}.
\end{aligned}
$$

Thus, the dual problem of the $(t+1)$-th iteration of our DCA for RMSVM is

$$
\begin{aligned}
(3.12)\quad & \min_{\Theta} \frac{1}{2}\sum_{j=1}^{k} \Big\langle [\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} - (\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}}) - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})], \\
& \qquad \boldsymbol{K}[\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} - (\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}}) - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})] \Big\rangle - \sum_{i=1}^{n}\sum_{j \neq y_i} \alpha_{ij} \\
(3.13)\quad & \text{s.t.} \qquad 0 \leq \alpha_{ij} \leq \frac{C}{n}, i = 1, 2, \cdots, n, j \neq y_i \\
(3.14)\quad & \qquad -\frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})^T \boldsymbol{e} + (\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})^T \boldsymbol{e} - (\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}})^T \boldsymbol{e} = 0, j = 1, 2, \cdots, k.
\end{aligned}
$$

Once the solution $\boldsymbol{\alpha}$ of problem (3.12) is derived, the coefficients $\boldsymbol{v}_j$ can be recovered using the following equation:

$$
\boldsymbol{v}_{\cdot j} = -[\boldsymbol{\alpha}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j} - (\bar{\boldsymbol{\alpha}} - \frac{C}{n}\bar{\boldsymbol{\beta}}) - \frac{C}{n}(\boldsymbol{\beta}_{\cdot j} \cdot \boldsymbol{L}_{\cdot j})]. \tag{3.15}
$$

We are now left to solve b_j's using linear programming (LP) once $\boldsymbol{v}_j$'s are obtained at the $(t+1)$-th iteration. Denote $\tilde{f}_j(\boldsymbol{x}_i) = \boldsymbol{K}_i^T \boldsymbol{v}_{\cdot j}$. Then b_j's can be obtained by solving the following LP problem:

$$
\begin{aligned}
\min_{\boldsymbol{b}} \qquad & \frac{C}{n}\sum_{i=1}^{n}\sum_{j \neq y_i} [b_j + \tilde{f}_j(\boldsymbol{x}_i) + 1]_+ + \sum_{j=1}^{k} b_j \frac{\partial \psi_{cav}(\Theta^t)}{\partial b_j} \\
\text{subject to} \qquad & \sum_{j=1}^{k} b_j = 0.
\end{aligned}
$$

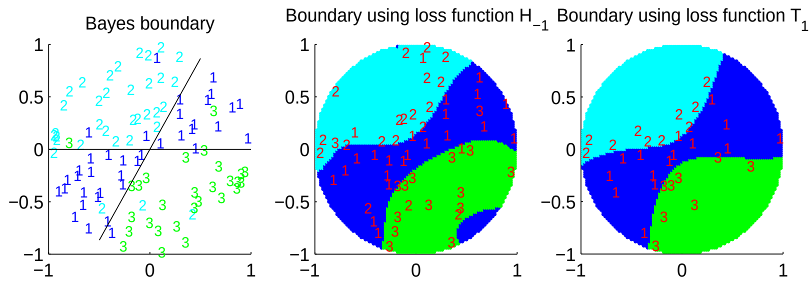

FIGURE 2. For one typical training sample in Section 4, the left panel plots all observations with the black straight lines shown as the Bayes boundary. The middle and right panels display the classification boundaries obtained by using loss functions H_{-1} and T_1, respectively, with SVs shown in red.

More explicitly,

$$\begin{aligned}
\min_{\boldsymbol{b}} \quad & \frac{C}{n}\sum_{i=1}^{n}\sum_{j\neq y_i}\xi_{ij}+\sum_{j=1}^{k}b_j\frac{\partial\psi_{cav}(\Theta^t)}{\partial b_j}\\
\text{subject to} \quad & \sum_{j=1}^{k}b_j=0\\
& \xi_{ij}\geq 0, i=1,2,\cdots,n; j\neq y_i\\
& \xi_{ij}\geq b_j+\tilde{f}_j(\boldsymbol{x}_i)+1, i=1,2,\cdots,n; j\neq y_i.
\end{aligned}$$

We stop the algorithm when the objective function value in (2.6) converges.

4. Numerical Examples

Three-class nonlinear examples with $p = 2$ are generated in the following way: First, generate (x_1, x_2) uniformly over the unit disc $\{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$. Define ϑ to be 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 3-class example, the class label y is assigned as follows: $y = 1$ 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; $y = 3$ if $\lfloor\frac{k\vartheta}{2\pi}\rfloor + 1 = 5$ or 6. Next, randomly contaminate the data by randomly selecting $perc\% = 10\%$ or 20% instances and changing their label indices to one of the remaining two classes with equal probabilities. This example was also considered in [WL06].

We apply the Gaussian kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle}{2\sigma^2})$. Here, two parameters need to be selected. The first parameter C is chosen using a grid search. The second parameter σ for the kernel is tuned among the first quartile, median, and the third quartile of the between-class pairwise Euclidean distances of training inputs ([BGL$^+$00]).

TABLE 1. Results of the nonlinear examples in Section 4

perc%	Loss	Test Error	#SV
10%	H_{-1}	0.1712 (0.0198)	62.14 (13.43)
	T_1	0.1595 (0.0203)	38.72 (16.05)
20%	H_{-1}	0.2793 (0.0270)	75.10 (10.60)
	T_1	0.2672 (0.0251)	36.96 (12.49)

We have applied MSVMs with the unbounded loss H_{-1} and the truncated loss T_1 for different contamination percentages (10% and 20%). Results based on 50 repetitions are reported in Table 1. From the table, we can see that RMSVMs give smaller testing errors while using fewer SVs than the MSVM. To visualize decision boundaries and SVs of the original MSVM and RMSVM, we choose one typical training sample and plot the results in Figure 2. The left panel shows the observations as well as the Bayes boundary. In the remaining two panels, boundaries using nonlinear learning with loss functions H_{-1} and T_1 are plotted and their corresponding SVs are labelled in red. From the plots, we can see that the RMSVM uses much fewer SVs and at the same time yields more accurate classification boundaries than the MSVM.

5. Discussion

In this paper, we propose a robust version of MSVM, namely the RMSVM. The RMSVM uses the truncated hinge loss and delivers more robust classifiers than the MSVM. Our algorithm and numerical results show that the RMSVM has the interpretation of SVs and it tends to use a smaller yet more stable set of SVs than that of the MSVM.

References

[AT97] L. T. H. An and P. D. Tao. Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. *Journal of Global Optimization*, 11:253–285, 1997.

[BGL+00] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *The Proceeding of National Academy of Sciences*, 97:262–267, 2000.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[FHT00] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28:337–407, 2000.

[KW71] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[Lin02] Y. Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275, 2002.

[LLW04] Y. Lee, Y. Lin, and G. Wahba. 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, 2004.

[LSD05] Y. Liu, X. Shen, and H. Doss. Multicategory ψ-learning and support vector machine: computational tools. *Journal of Computational and Graphical Statistics*, 14:219–236, 2005.

[LSW05] S. Liu, X. Shen, and W. Wong. Computational development of ψ-learning. *In The SIAM 2005 International Data Mining Conf.*, pages 1–12, 2005.

[LWX+00] X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *The Annals of Statistics*, 28(6):1570–1600, 2000.

[Vap98] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[Wah99] G. Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods Support Vector Learning*, pages 69–88. MIT Press, 1999.

[WL06] Y. Wu and Y. Liu. Robust truncated-hinge-loss support vector machines. *Submitted*, 2006.

[ZH05] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14:185–205, 2005.

Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544

E-mail address: yichaowu@princeton.edu

Department of Statistics and Operations Research, University of North Carolina, CB 3260, Chapel Hill, NC 27599

E-mail address: yfliu@email.unc.edu