Open access • Journal Article • DOI:10.1287/OPRE.1100.0854

Support Vector Machines with the Ramp Loss and the Hard Margin Loss — Source link

J. Paul Brooks

Institutions: Virginia Commonwealth University

Published on: 01 Mar 2011 - Operations Research (INFORMS)

Topics: Hinge loss, Huber loss, Support vector machine, Outlier and Quadratic programming

Support Vector Machines with the Ramp Loss and the Hard Margin Loss [*]

J. Paul Brooks
Department of Statistical Sciences and Operations Research
Virginia Commonwealth University

May 7, 2009

Abstract

In the interest of deriving classifiers that are robust to outlier observations, we present integer programming formulations of Vapnik's support vector machine (SVM) with the ramp loss and hard margin loss. The ramp loss allows a maximum error of 2 for each training observation, while the hard margin loss calculates error by counting the number of training observations that are misclassified outside of the margin. SVM with these loss functions is shown to be a consistent estimator when used with certain kernel functions. Based on results on simulated and real-world data, we conclude that SVM with the ramp loss is preferred to SVM with the hard margin loss. Data sets for which robust formulations of SVM perform comparatively better than the traditional formulation are characterized with theoretical and empirical justification. Solution methods are presented that reduce computation time over industry-standard integer programming solvers alone.

1 Introduction

The support vector machine (SVM) is a math programming-based binary classification method developed by Vapnik [39] and Cortes and Vapnik [12]. Math programming and classification have a long history together, dating back to the fundamental work of Mangasarian [22, 23].

The SVM formulation proposed by Vapnik and coauthors uses a continuous measure for misclassification error, resulting in a continuous convex optimization problem. Several investigators have noted that such a measure can result in an increased sensitivity to outlier observations (Figure 4(a)), and have proposed modifications that increase the robustness of SVM models.

One method for increasing the robustness of SVM is to use the *ramp loss* (Figure 1(b)), also known as the *robust hinge loss*. Training observations that fall outside the margin that are misclassified have error

[*] The author would like to gratefully acknowledge Romy Shioda for ideas arising from numerous discussions. The author would also like to acknowledge the Center for High Performance Computing at VCU for providing computational infrastructure and support.

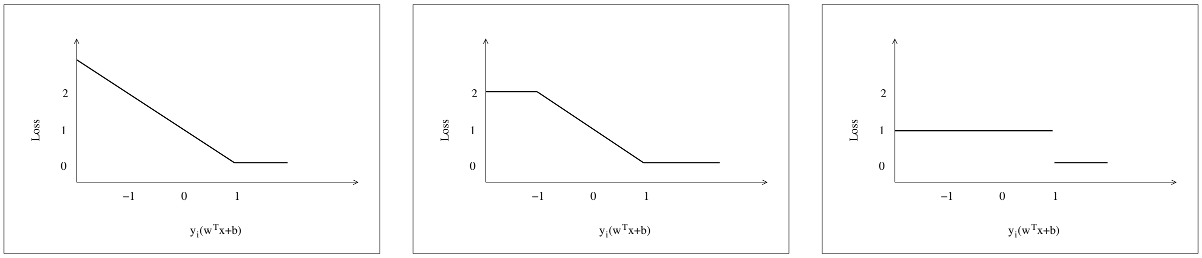
2, while observations that fall in the margin are given a continuous measure of error between 0 and 2 depending on their distance to the margin boundary. Bartlett and Mendelson [2] and Shawe-Taylor and Christianini [33] investigate some of the learning theoretic properties of the ramp loss. Shen et al. [34] and Collobert et al. [11] use optimization methods for SVM with the ramp loss that do not guarantee global optimality. Liu et al. [21] propose an outer approximation procedure for multi-category SVM with ramp loss that converges to global optima, but convergence is slow; only a single 100-observation instance is solved with the linear kernel. Xu et al. [40] solve a semidefinite programming relaxation of SVM with the ramp loss, but the procedure is computationally intensive for as few as 50 observations.

Another method for increasing the robustness of SVM is to use the *hard margin loss* (Figure 1(c)), where the number of misclassifications is minimized. Chen and Mangasarian [10] prove that minimizing misclassifications for a linear classifier is NP-Complete by reducing the OPEN HEMISPHERE [19] problem. The computational complexity of using the hard margin loss has often been used as the justification of a continuous measure of error. Orsenigo and Vercellis [27] formulate discrete SVM (DSVM) that uses the hard margin loss for SVM with a linear kernel and linearized margin term; they use heuristics for solving instances that do not guarantee global optimality. Orsenigo and Vercellis have extended their formulation and technique to soft margin DSVM (ϵ-DSVM) [29] and to fuzzy DSVM (FDSVM) [28]. Pérez-Cruz and Figueiras-Vidal [30] approximate the hard margin loss for SVM with continuous functions and use an iterative reweighted least squares method for solving instances that also does not guarantee global optimality.

Learning theory has emerged to provide a probabilistic analysis of machine learning algorithms. A method for classification is *consistent* if, in the limit as the sample size is increased, the sequence of generated classifiers converges to a *Bayes optimal rule*. A Bayes optimal rule minimizes the probability of misclassification. If convergence holds for all distributions of data, then the classification method is *universally consistent*. Due to the No Free Lunch Theorem ([13], [14, Theorem 7.2], [15, Theorem 9.1]), there cannot exist a classification method with a guaranteed rate of convergence to a Bayes optimal rule for all distributions of data; in other words, there always exists a distribution of the data for which convergence is arbitrarily slow. Steinwart [36] proves that SVM with the traditional hinge loss is universally consistent. Brooks and Lee [7] prove that an integer-programming based method for constrained discrimination, a generalization of the classification problem, is consistent.

This paper presents new integer programming formulations for SVM with the ramp loss and hard margin loss that accommodate the use of nonlinear kernel functions and the quadratic margin term. These formulations can be solved in a branch-and-bound framework, providing solutions to moderate-sized instances in reasonable time. Solution methods are presented that provide savings in computation time when incorporated with industry-standard software. The use of integer programming and branch-and-bound for deriving globally optimal solutions is not common in machine learning literature. Bennet and Demiriz [3] and Chapelle [9] use branch-and-bound algorithms to derive globally optimal solutions to a semi-supervised support vector machine (S^{3}VM). Koehler and Erenguc [20] introduce the use of integer programming to minimize misclassifications that predates the development of SVM, and their models do not incorporate the maximization of margin nor the use of kernel functions for finding nonlinear separating surfaces. Bertsimas and Shioda [4] combine ideas from SVM and classification trees in an integer programming framework that minimizes misclassifications. Gallagher et al. [16] present an integer programming model for constrained discrimination, where the number of correctly classified observations is maximized subject to limits on the number of misclassified observations.

We address the consistency of SVM with the ramp loss and hard margin loss. Relying heavily on the previous work of Steinwart [36, 37] and Bartlett and Mendelson [2], we provide proofs that SVM with the ramp loss and the hard margin loss are universally consistent procedures for estimating the Bayes



(a) (b) (c)

Figure 1: Loss functions for SVM. The loss for an observation is plotted against the "left-hand side" of primal formulations for SVM with (a) the traditional hinge loss, (b) ramp loss, and (c) hard margin loss. An observation whose left-hand side falls between -1 and 1 lies in the margin.

classification rule when used with so-called *universal kernels* [35]. We demonstrate the performance of SVM with the ramp loss and hard margin loss on simulated and real-world data for producing robust classifiers in the presence of outliers, especially when using low-rank kernels.

The remainder of the paper is structured as follows. Section 2 introduces new integer programming formulations for SVM with the hard margin loss and the ramp loss. In Section 3, we show that SVM with the ramp loss and hard margin loss is consistent. Section 4 contains solution methods for the integer programming formulations. Section 5 contains computational results on simulated and real-world data.

2 Formulations

Suppose a training set is given consisting of data points $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$, $i = 1, \ldots, n$, where y_i is the class label of the i^{th} observation. The data points are realizations of the random variables X and Y, where X has an unknown distribution and Y has an unknown conditional distribution $P(Y = h|X = x)$. A function $f : \mathbb{R}^d \to \{-1, 1\}$ is a classifier.

For a given training set, SVM balances two objectives: maximize *margin*, the distance between correctly classified sets of observations, while minimizing error. SVM can be viewed as projecting data into a higher-dimensional space and finding a separating hyperplane in the projected space that corresponds to a nonlinear separating surface in the space of the original data. As shown in [12], normalizing $\boldsymbol{w}$ and b so that $\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$ and $\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$ define the boundaries of sets of correctly classified observations, the distance between these sets is $2/||\boldsymbol{w}||$. Therefore, minimizing $\frac{1}{2}||\boldsymbol{w}||^2 = \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w}$ maximizes the margin.

2.1 Ramp Loss

Let d_i be the distance of observation $\boldsymbol{x}_i$ to the margin boundary for the class y_i. Define ξ_i as the continuous error for observation i such that

$$\xi_i = \begin{cases} d_i\|\boldsymbol{w}\| & \text{if } \boldsymbol{x}_i \text{ falls in the margin} \\ 0 & \text{otherwise} \end{cases}$$

Let z_i be a binary variable equal to 1 if observation $\boldsymbol{x}_i$ is misclassified outside of the margin and 0 otherwise. For an observation that falls in the margin, ξ_i measured in the same way that error is measured for traditional SVM. SVM with ramp loss can be formulated as

$$\begin{array}{llll} \text{[SVMIP1(ramp)]} & \min & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^n \xi_i + 2\sum_{i=1}^n z_i\right), & \\ & \text{s.t.} & y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1 - \xi_i, \text{ if } z_i = 0, & i = 1,\ldots,n, \\ & & z_i \in \{0,1\}, & i = 1,\ldots,n, \\ & & 0 \leq \xi_i \leq 2, & i = 1,\ldots,n. \end{array} \tag{1}$$

The parameter C represents the tradeoff in maximizing margin versus minimizing error. Unlike traditional SVM, the error of an observation is bounded above by 2 (Figure 1(a), (b)). This formulation can accommodate nonlinear projections of observations by replacing $\boldsymbol{x}_i$ with $\Phi(\boldsymbol{x}_i)$. The conditional constraint for observation i can be linearized by introducing a sufficiently large constant M and writing $y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1 - \xi_i - Mz_i$. The formulation is then a convex quadratic integer program, solvable by a standard branch-and-bound algorithm. By making the substitution

$$\boldsymbol{w} = \sum_{i=1}^n y_i \boldsymbol{x}_i \alpha_i, \tag{2}$$

with nonnegative α_i variables, we can obtain the following formulation for SVM with the ramp loss.

$$\begin{array}{llll} \text{[SVMIP2(ramp)]} & \min & \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n y_i y_j \boldsymbol{x}_i\cdot\boldsymbol{x}_j \alpha_i \alpha_j + C\left(\sum_{i=1}^n \xi_i + 2\sum_{i=1}^n z_i\right), & \\ & \text{s.t.} & y_i(\sum_{j=1}^n y_j \boldsymbol{x}_j\cdot\boldsymbol{x}_i \alpha_j + b) \geq 1 - \xi_i, \text{ if } z_i = 0, & i = 1,\ldots,n, \\ & & \alpha_i \geq 0, & i = 1,\ldots,n, \\ & & z_i \in \{0,1\}, & i = 1,\ldots,n, \\ & & 0 \leq \xi_i \leq 2, & i = 1,\ldots,n. \end{array} \tag{3}$$

The data occur as inner products, so that nonlinear kernel functions may be employed by replacing occurrences of $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ with $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For positive semi-definite kernels (see [33], pp. 61), the objective function for [SVMIP2(ramp)] remains convex, and the solutions are equivalent to those obtained for [SVMIP1(ramp)]. Again, the conditional constraints can be linearized by introducing a large constant M.

2.2 Hard Margin Loss

Let

$$z_i = \begin{cases} 1 & \text{if observation } i \text{ lies in the margin or is misclassified} \\ 0 & \text{o.w.} \end{cases}$$

Then an SVM formulation with the hard margin loss (Figure 1(b)) for finding a separating hyperplane in the space of the original data is

$$\begin{array}{llll} \text{[SVMIP1(hm)]} & \min & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^n z_i, & \\ & \text{s.t.} & y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1, & \text{if } z_i = 0, i = 1,\ldots,n, \\ & & z_i \in \{0,1\}, & i = 1,\ldots,n. \end{array} \tag{4}$$

The constraint for observation i can be linearized as for [SVMIP1(ramp)] [27, 8]. The formulation with linearized constraints is the same as that used by Orsenigo and Vercellis [27], except that they use a linearized version of the margin term. Making the substitution (2), the following formulation is obtained.

$$\begin{array}{llll} \text{[SVMIP2(hm)]} & \min & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \alpha_i \alpha_j + C\sum_{i=1}^{n} z_i, & \\ & \text{s.t.} & y_i(\sum_{j=1}^{n} y_j \boldsymbol{x}_j \cdot \boldsymbol{x}_i \alpha_j + b) \geq 1, \text{ if } z_i = 0, & i = 1, \ldots, n, \\ & & \alpha_i \geq 0, & i = 1, \ldots, n, \\ & & z_i \in \{0,1\}, & i = 1, \ldots, n. \end{array} \tag{5}$$

The formulation can accommodate nonlinear kernel functions in the same manner as [SVMIP2(ramp)]. The formulations [SVMIP2(hm)] and [SVMIP2(ramp)] are convex quadratic integer programs for positive-semidefinite kernel functions.

2.3 Equivalence of [SVMIP1(ramp)] and [SVMIP2(ramp)]

For a positive-semidefinite kernel function $k(\cdot,\cdot)$, there exists a function Φ such that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$ (See [33], pp. 61). In this section, we show that [SVMIP1(ramp)] and [SVMIP2(ramp)] are equivalent for positive-semidefinite kernels in the sense that an optimal solution for one formulation can be used to construct an optimal solution to the other.

In practice, the dual form of traditional SVM is solved in part because of the ability to accommodate kernel functions. The formulation [SVMIP2(ramp)] represents the ability to apply the same analysis with the ramp loss. We will now demonstrate that solutions to [SVMIP1(ramp)] can be used to construct solutions to [SVMIP2(ramp)] and vice versa.

Remark 2.1. Given a binary vector $\boldsymbol{z} \in \{0,1\}^n$, let us define the following parametric quadratic programming problem:

$$\begin{array}{llll} \text{[SVM-P}(\boldsymbol{z})] & \min & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i, & \\ & \text{s.t.} & y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i, & i : z_i = 0, \end{array} \tag{6}$$

$$\boldsymbol{\xi_i \geq 0}, \qquad \boldsymbol{i = 1, \ldots, n}. \tag{7}$$

Suppose $\boldsymbol{z} = \boldsymbol{z}^*$ is optimal for [SVMIP1(ramp)] with corresponding values $\boldsymbol{w} = \boldsymbol{w}^*$, $b = b^*$, and $\boldsymbol{\xi} = \boldsymbol{\xi}^*$. Then, $(\boldsymbol{w}', b', \boldsymbol{\xi}')$ is an optimal solution to [SVM-P($\boldsymbol{z}^*$)] if and only if $(\boldsymbol{w}', b', \boldsymbol{\xi}', \boldsymbol{z}^*)$ is an optimal solution to [SVMIP1(ramp)].

The following lemma is non-trivial because we are making a substitution for unrestricted variables in terms of a linear combination of non-negative variables. It is not immediately apparent that the optimal solution in the original problem is not excluded.

Lemma 2.1. *Given optimal solution $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ to [SVMIP1(ramp)], we can construct a feasible solution $(\boldsymbol{\alpha}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ of [SVMIP2(ramp)] with equivalent objective values (i.e., $\frac{1}{2}\|\boldsymbol{w}^*\|^2 + C(\sum_{i=1}^{n} \xi_i^* + 2\sum_{i=1}^{n} z_i^*) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \alpha_i^* \alpha_j^* + C(\sum_{i=1}^{n} \xi_i^* + 2\sum_{i=1}^{n} z_i^*)$).*

Proof. Given $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$, from Remark 2.1, $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ is an optimal solution to [SVM-P($\boldsymbol{z}^*$)]. Let $\boldsymbol{\alpha}'$ be the corresponding optimal solution for the dual of [SVM-P($\boldsymbol{z}^*$)]. Then, from the KKT conditions, we know that $\boldsymbol{w}^* = \sum_{i:z_i^*=0} y_i \boldsymbol{x}_i \alpha_i'$. Define α_i^*, $i = 1, \ldots, n$, as $\alpha_i^* = \alpha_i'$ if $z_i = 0$ and $\alpha_i^* = 0$ if $z_i = 1$. Then $(\boldsymbol{\alpha}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ is feasible for [SVMIP2(ramp)] and

$$\|\boldsymbol{w}^*\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \alpha_i^* \alpha_j^*.$$

□

Lemma 2.2. *Given optimal solution $(\boldsymbol{\alpha}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ to [SVMIP2(ramp)], we can construct a feasible solution $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ for [SVMIP1(ramp)] with equivalent objective values (i.e., $\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \alpha_i^* \alpha_j^* + C(\sum_{i=1}^{n} \xi_i^* + 2\sum_{i=1}^{n} z_i^*) = \frac{1}{2}\|\boldsymbol{w}^*\|^2 + C(\sum_{i=1}^{n} \xi_i^* + 2\sum_{i=1}^{n} z_i^*)$).*

Proof. Define $\boldsymbol{w}^*$ as

$$\boldsymbol{w}^* := \sum_{i=1}^{n} y_i \boldsymbol{x}_i \alpha_i^*.$$

Then $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{z}^*)$ is clearly a feasible solution to [SVMIP1(ramp)] with matching objective values, as this is precisely the substitution used in the creation of [SVMIP2(ramp)] from [SVMIP1(ramp)]. □

The following theorem follows immediately from Lemmas 2.1 and 2.2.

Theorem 2.1. *The optimization problems [SVMIP1(ramp)] and [SVMIP2(ramp)] are equivalent.*

This reasoning holds if we replace occurrences of $\boldsymbol{x}_i$ with $\Phi(\boldsymbol{x}_i)$, so that the result holds for all positive-semidefinite kernels. Similar reasoning shows that [SVMIP1(hm)] and [SVMIP2(hm)] are equivalent [8]. This equivalence theorem ensures that the use of [SVMIP2(ramp)] and [SVMIP2(hm)] with nonlinear kernel functions retains the same geometric interpretation as the dual for traditional SVM.

3 Consistency

We assume that X is a compact subset of $\mathbb{R}^d$ and that there exists an unknown Borel probability measure P on $X \times Y$. For a classifier $f : \mathbb{R}^d \to \{-1, 1\}$, the probability of misclassification is $\mathcal{L}(f) = P(f(X) \neq Y)$. A *Bayes classifier* f^* assigns an observation $\boldsymbol{x}$ to the group to which it is most likely to belong; i.e., $f^*(\boldsymbol{x}) = \arg \max\limits_{h \in \{-1,1\}} P(Y = h | X = \boldsymbol{x})$. It can be shown [14] that a Bayes classifier minimizes the probability of misclassification, so that $f^* = \arg \min\limits_{f} \mathcal{L}(f)$. Let $f_n(X)$ be the classifier that is selected by a method based on a sample of size n.

Definition 3.1. A classifier f is *consistent* if the probability of misclassification converges in expectation to a Bayes optimal rule as sample size is increased, or

$$\lim_{n \to \infty} E\mathcal{L}(f_n) = \mathcal{L}(f^*)$$

A classifier is *universally consistent* if it is consistent for all distributions for X and Y.

Let $C(X)$ be the space of all continuous functions $f : X \to \mathbb{R}$ on the compact metric space (X, d) with the supremum norm $||f||_\infty = \sup_{\boldsymbol{x} \in X} |f(\boldsymbol{x})|$. The following definitions are due to Steinwart [35]. A function f is *induced* by a kernel k (with projection function $\Phi : X \to H$) if there exists $\boldsymbol{w} \in \mathcal{H}$ with $f(\cdot) = \boldsymbol{w} \cdot \Phi(\cdot)$. The kernel k is *universal* if the set of all induced functions is dense in $C(X)$; i.e., for all $g \in C(X)$ and all $\epsilon > 0$, there exists a function f induced by k with $||f - g||_\infty \leq \epsilon$. Steinwart [35] showed that the Gaussian kernel, among others, is universal. We will show that [SVMIP2(ramp)] and [SVMIP2(hm)] are universally consistent for universal kernel functions.

3.1 Consistency of SVM with the Ramp Loss

Before we prove the consistency of [SVMIP2(ramp)], we need to make a few more definitions and to establish some more notation. For a training set of size n, a universal positive-semidefinite kernel k, and an objective function parameter C, we denote a classifier derived from an optimal solution to [SVMIP2(ramp)] by $f_n^{k,C}$, or by $f_n^{\Phi,C}$ where $k(\cdot,\cdot) = \Phi(\cdot) \cdot \Phi(\cdot)$. Further, let $\boldsymbol{w}_n^{\Phi,C}$ be given by the same optimal solution to [SVMIP2(ramp)] and the formula (2).

Theorem 3.1 shows that solutions to [SVMIP2(ramp)] will converge to the Bayes optimal rule as the sample size n increases.

Theorem 3.1. *Let $X \subset \mathbb{R}^d$ be compact and $k : X \times X \to \mathbb{R}$ be a universal kernel. Let $f_n^{k,C}$ be the classifier obtained by solving [SVMIP2(ramp)] for a training set with n observations. Suppose that we have a positive sequence (C_n) with $C_n/n \to 0$ and $C_n \to \infty$. Then for any $\epsilon > 0$,*

$$\lim_{n \to \infty} P(\mathcal{L}(f_n^{k,C_n}) - \mathcal{L}(f^*) > \epsilon) = 0$$

Proof. The proof is in the Appendix. □

Theorem 3.1 requires that as n is increased, the parameter C is chosen under specified conditions. The consistency of the ramp loss can also be established directly under different (and more elaborate) conditions on the choice of C using Theorem 3.5 in [37].

3.2 Consistency of SVM with the Hard Margin Loss

The proof of consistency for SVM with the hard margin loss is similar to that of ramp loss. We again assume that we have a universal kernel k with projection function Φ. Let $f_n^{k,C}$ and $f_n^{\Phi,C}$ denote optimal solutions to [SVMIP2(hm)] with kernel function k and projection function Φ, respectively. The following theorem establishes the consistency of SVM with the hard margin loss when used with universal kernels and appropriate choices for C.

Theorem 3.2. *Let $X \subset \mathbb{R}^d$ be compact and $k : X \times X \to \mathbb{R}$ be a universal kernel. Let $f_n^{k,C}$ be the classifier obtained by solving [SVMIP2(hm)] for a training set with n observations. Suppose that we have a positive sequence (C_n) with $C_n/n \to 0$ and $C_n \to \infty$. Then for any $\epsilon > 0$,*

$$\lim_{n \to \infty} P(\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^*) > \epsilon) = 0$$

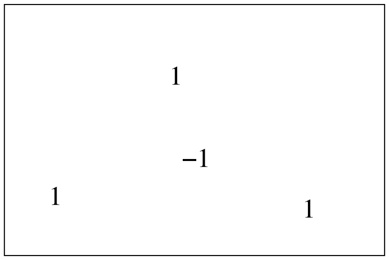


Figure 2: An observation with class label -1 falls in the convex hull of observations of class $+1$. All four observations cannot be simultaneously correctly classified by a linear hyperplane.

Proof. . The proof is in the Appendix. □

4 Solution Methods and Computation Time

To improve the computation time for solving the mixed-integer quadratic programming problems [SVMIP1(ramp)], [SVMIP2(ramp)], [SVMIP1(hm)], and [SVMIP2(hm)] in a branch and cut framework, we describe a family of facets to cut off fractional solutions for the linear kernel and introduce some heuristics to find good integer feasible solutions at nodes in the branch and cut tree. In [8], upper bounds for the constant M in the linearizations of the constraints are derived. These solution methods and computational improvements are applicable to both ramp loss and hard margin loss formulations with few adjustments; they will be presented in the form appropriate for the ramp loss.

4.1 Facets of the Convex Hull of Feasible Solutions

In this section, we discuss a class of facets for [SVMIP1(ramp)]. If in a training data set, an observation from one class lies in the convex hull of observations from the other class, then at least one of the observations must be misclassified; i.e., have error at least 1 (Figure 2).

Theorem 4.1. *[8]. Given a set of $d+1$ points $\{\boldsymbol{x}_i : y_i = 1,\ i = 1, \ldots, d+1\}$ and another point $\boldsymbol{x}_{d+2}$ with label $y_{d+2} = -1$ such that $\boldsymbol{x}_{d+2}$ falls in the convex hull of the other $d+1$ points, then*

$$\sum_{i=1}^{d+2} \xi_i + \sum_{i=1}^{d+2} z_i \geq 1$$

defines a facet for the convex hull of integer feasible solutions for [SVMIP1(ramp)].

Proof. The proof is in the Appendix. □

These *convex hull cuts* can be generated before optimization and added to a cut pool or derived by solving separation problems at nodes in the branch and bound tree. In the latter case, two separation problems can be solved, one for each class. The separation problem for the positive class has the following form

$$
\begin{array}{rrcll}
\text{[CONV-SEP]} & \mathbf{min} & \sum_{i=1}^{n} (\xi_i + z_i) h_i & & \\
\text{s.t.} & \sum_{i:y_i=1} \boldsymbol{x}_i \lambda_i & = & \sum_{i:y_i=-1} \boldsymbol{x}_i h_i, & \\
& \sum_{i:y_i=1} h_i & = & d+1, & \\
& \sum_{i:y_i=-1} h_i & = & 1, & \\
& \sum_{i:y_i=1} \lambda_i & = & 1, & \\
& \lambda_i & \leq & h_i & \forall\, i : y_i = 1, \\
& \lambda_i & \geq & 0 & \forall\, i : y_i = 1, \\
& h_i & \in & \{0,1\}, & i = 1, \ldots, n.
\end{array}
$$

Solving this mixed-integer programming problem finds an observation from the negative class that lies in the convex hull of $d+1$ points from the positive class. The h_i variables indicate whether observation $\boldsymbol{x}_i$ is one of the $d+2$ points. If the optimal objective function value is less than 1.0, then the following inequality is violated by the current fractional solution.

$$\sum_{i \in H} \xi_i + \sum_{i \in H} z_i \geq 1$$

where $H = \{i | h_i = 1\}$. Note that [CONV-SEP] may not be feasible if none of the negative class points are convex combinations of the points of the positive class. However, unless the points are linearly separable, the corresponding separation problem for the negative class would be feasible.

The convex hull cuts are implemented using ILOG CPLEX 11.1 Callable Library (http://www.ilog.com). The enhanced solver is applied to the Type A data sets described in Section 5.1 using the same computer architecture and settings, including indicator constraints. If a cut is found to be violated by 0.01, then it is added. A time limit of 2 minutes (120 CPU seconds) is imposed on the solution of each separation problem.

Adding the cuts at the root node provides good lower bounds, but the computation time per subproblem increases significantly as nodes in the branch and bound tree are explored (data not shown). No attempt at cut management is conducted, including deleting cuts that are no longer needed and controlling the number of cuts added at each node in the branch and bound tree. Should a sophisticated cut management system be employed with the convex hull cuts, we would expect savings in computational time; these savings would be in addition to the time savings observed with the solution methods in Section 4.2. In order to provide evidence that these facets are "good" in the sense that they cut off significant portions of the polytope for the linear programming relaxation, we present the lower bounds generated at the root node of the branch and bound tree that are obtained by adding violated cuts.

Results for the lower bounds at the root node provided by the convex hull cuts for instances with the linear kernel and $C = 1$ are presented in Table 1. The columns labeled *CPLEX-Generated Cuts* shows the best lower bound and the integrality gap at the root node when all CPLEX-generated cut settings are set their most aggressive level. The columns labeled *Convex Hull Cuts* shows the best lower bound and the integrality gap at the root node when the convex hull cuts are added; no CPLEX-generated cuts are added. The integrality gap is measured using the formula $(z^* - z^{LB})/z^* \times 100$, where z^* is the objective value associated with the best known integer feasible solution and z^{LB} is the lower bound at the root node.

Table 1: Best Lower Bound at Root Node for Convex Hull Cuts

		Convex Hull Cuts		
n	d	# of Cuts	Best LB	Integrality Gap (%)
60	2	36	10.3	63.5
100	2	78	18.0	62.2
200	2	197	41.1	57.8
500	2	572	107.0	55.8
60	5	13	3.0	89.1
100	5	83	11.6	78.0
200	5	234	25.8	72.1
500	5	416	56.1	75.1
60	10	0	0.0	100.0
100	10	3	1.0	97.4
200	10	7	2.0	97.9
500	10	46	9.7	96.2

For 2- and 5-dimensional data, the convex hull cuts provide lower bounds that close the integrality gap by between 11% and 44%. For 10-dimensional data, observations are less likely to fall in the convex hull of other observations, and the usefulness of the convex hull cuts degrades. Similar behavior is observed for the real-world data sets described in Section 5.2 (data not shown). When CPLEX alone is used with indicator constraints, and all cut settings at their most aggressive level, no cuts are generated, leaving an integrality gap of 100%. When CPLEX is provided linearized constraints with upper bounds for M as derived in [8], the cuts generated by CPLEX close the integrality gap to 90.3% for the $n = 60$, $d = 10$ case; for all other instances, the integrality gap is at least 94.1%.

4.2 Heuristics for Generating Integer Feasible Solutions

This section describes heuristics for generating integer feasible solutions that are implemented within a branch-and-bound framework and applicable to all four formulations. We present methods for [SVMIP2(ramp)]; minor adjustments are needed for use with the other formulations.

Before solving the root problem in the branch and bound tree, an initial solution is derived by setting $\alpha_i = 0$ for $i = 1, \ldots, n$. The variable b is set to 1 if $n_+ > n_-$ and -1 otherwise. This solution, the "zero solution", yields an objective function value of $2C \min\{n_+, n_-\}$.

When using kernel functions of high rank (for example, the Gaussian kernel function has infinite rank) and/or for well-separated data sets, a decision boundary can often be found such that no observations are misclassified outside the margin. If feasible, such a solution can be derived by fixing all z_i variables in [SVMIP2(ramp)] to zero and solving a single continuous optimization problem. The problem is equivalent to traditional SVM with the exception that the ξ_i variables are bounded above. This solution, the "zero error solution", is checked before beginning the branching procedure.

We implement another procedure for finding initial integer feasible solutions before branching. We check the use of every positive-negative pair of observations to serve as the sole support vectors such that their conditional constraints hold at equality (i.e., they define the margin boundary). For [SVMIP2(ramp)], and for observations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with $y_1 = 1$ and $y_2 = -1$, let

$$\alpha = 2/(k(\boldsymbol{x}_1, \boldsymbol{x}_1) - 2k(\boldsymbol{x}_1, \boldsymbol{x}_2) + k(\boldsymbol{x}_1, \boldsymbol{x}_2)).$$

The solution is given by

$$\alpha_i = \begin{cases} \alpha & \text{for } i = 1, 2 \\ 0 & \text{otherwise} \end{cases}$$

$$b = (1/2)\alpha(k(\boldsymbol{x}_i, \boldsymbol{x}_i) - k(\boldsymbol{x}_j, \boldsymbol{x}_j))$$

At nodes in the branch and bound tree, we employ a heuristic for deriving integer feasible solutions. Let $(\boldsymbol{\alpha}^j, \boldsymbol{\xi}^j, \boldsymbol{z}^j)$ represent the solution to the continuous subproblem at node j in the branch and bound tree. We can project the solution into the space of the ξ_i and z_i variables to derive an integer feasible solution. For any set of values for $\boldsymbol{\alpha}$, feasible values of $\boldsymbol{\xi}$ and $\boldsymbol{z}$ can be set such that the conditional constraints are satisfied.

These methods for finding integer feasible solutions are implemented using ILOG CPLEX 11.1 Callable Library. The enhanced solver is applied to the real-world data sets described in Section 5.2 using the same architecture and settings. [SVMIP1(ramp)] and [SVMIP1(hm)] are used for instances with the linear kernel; [SVMIP2(ramp)] and [SVMIP2(hm)] are used for instances with the other kernels. There are 9 data sets and 5 C values yielding 45 problem instances for each choice of kernel. For the linear kernel, the enhanced solver finds solutions at least as good as CPLEX on 40 instances, and provides time savings on 24 instances.

Figure 3 compares the computation time requirements for the enhanced solver and CPLEX. The geometric mean of the time to the best solution obtained by CPLEX is plotted for various choices of C and for the linear and polynomial kernels. As C increases, meaning that more emphasis is placed on minimizing misclassifications over maximizing margin, the computation time decreases. For small values of C, and for the linear and polynomial degree 2 kernels, the enhanced solver outperforms CPLEX on average.

As higher rank kernels are used, both solvers are able to find good solutions quickly. These results correspond with the observation in Section 5 that when higher rank kernels are used, few if any observations are misclassified so that one may solve the traditional SVM formulation. For the Gaussian kernel, both the enhanced solver and CPLEX find optimal solutions to all 45 instances in less than 3 seconds. Each training data set is capable of being separated with no observations misclassified outside of the margin. The “zero error solution” is optimal for these data sets, indicating that ramp loss SVM is equivalent to traditional SVM for the Gaussian kernel and these training sets.

5 Classification Accuracy on Simulated and Real-World Data

The classification performance of SVM with ramp loss and hard margin loss is compared to traditional SVM on simulated and real-world data sets. Results for traditional SVM are obtained by using SVM^{light} [18].

When using the linear kernel with ramp loss and hard margin loss, formulations [SVMIP1(ramp)] and [SVMIP1(hm)] are used, respectively. When using polynomial and Gaussian kernels, formulations

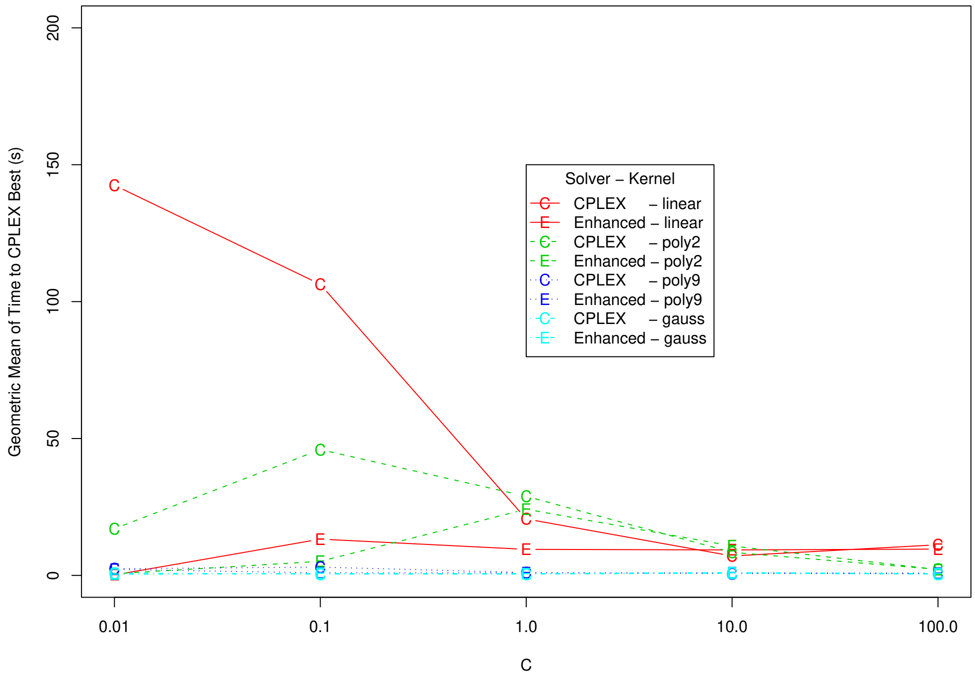


Figure 3: A comparison of computation time for instances of [SVMIP1(ramp)] and [SVMIP2(ramp)] for traditional CPLEX (CPLEX) and CPLEX with the enhancements (Enhanced) presented in the text. The linear (linear), polynomial degree 2 (poly2), polynomial degree 9 (poly9), and Gaussian/radial basis function (gauss) kernels are used. The time in CPU seconds to find a solution at least as good as the best obtained by CPLEX is plotted against values of C, the tradeoff between margin and error. For each value of C, the geometric mean across 9 real-world data sets is plotted. For the Gaussian kernel, results for $\sigma = 1.0$ are shown.

[SVMIP2(hm)] and [SVMIP2(ramp)] are used. For tests with the polynomial kernel, the form of the kernel function is $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\alpha \boldsymbol{x}_i \cdot \boldsymbol{x}_j) + \beta)^{\pi}$, $\alpha = 1$ and $\beta = 1$. The parameter π is tested with values of 2 and 9, for quadratic and ninth-degree polynomials, respectively. When using the polynomial kernel, each observation is normalized such that the magnitude of each observation vector is 1. For tests with the Gaussian kernel, the form of the kernel function is $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{\sigma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}$. Models are generated for the Gaussian kernel for values of σ at 0.1, 1, 10, 100, and 1000.

The data sets are split into training, validation, and testing data sets such that they comprise 50%, 25%, and 25% of the original data set, respectively. For real-world data sets with more than 1000 observations, a random sample of 500 observations is used for training, and the remaining observations are divided for validation and testing.

The training set is used to generate models for various values of C, the parameter that indicates the tradeoff between error and margin in each formulation. For the Gaussian kernel, models are generated for each combination of C and σ values. The impact of the choice of C for traditional SVM, ramp loss SVM, and hard margin loss SVM varies. For traditional SVM and ramp loss SVM, models are generated for $C = 0.01, 0.1, 1, 10, 100$; for hard margin loss SVM, models are generated for $C = 1, 10, 100, 1000, 10000$.

Of the models generated for a training set and loss function, the model that performs best on the validation set is used to choose the best value for C (and σ for the Gaussian kernel). This model is then applied to the testing data set, for which results are reported.

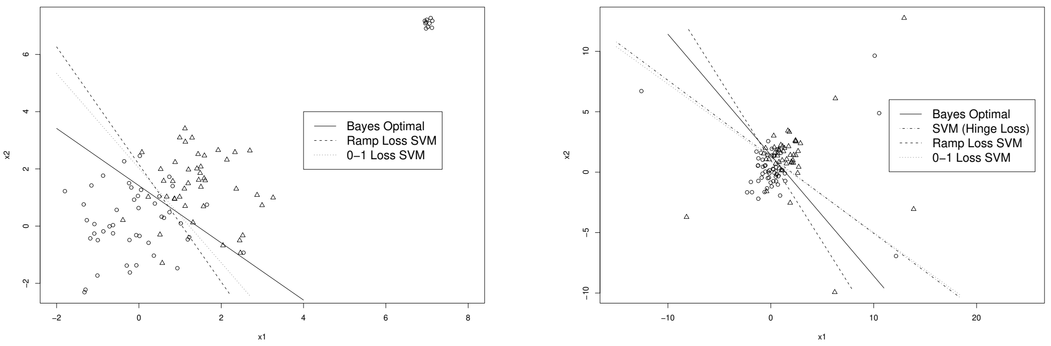
SVM^{light} instances and quadratic integer programming instances are solved on machines with 2.6 GHz Opteron processors and 4 GB RAM. All instances solved in less than 2 minutes (120 CPU seconds); the vast majority of instances were solved in a few seconds. Quadratic integer programming instances are solved using ILOG CPLEX 11.1 Callable Library (http://www.ilog.com). In all computational tests, CPLEX "indicator constraints" [17] are employed by using the function *CPXaddindconstr()* to avoid the negative effects of the M parameter required for linearization of the constraints. CPLEX implements a branching scheme for branching on disjunctions such as the indicator constraints in the proposed formulations, rather than on binary variables. For [SVMIP1(ramp)], [SVMIP1(hm)], [SVMIP2(ramp)], and [SVMIP2(hm)], CPLEX is enhanced with the heuristics for generating feasible solutions described in Section 4.2. The cuts described in Section 4.1 are not employed. If after 10 minutes (600 CPU seconds), provable optimality is not obtained, the best known solution is used.

5.1 Simulated Data

Two-group simulated data sets are sampled from Gaussian distributions, each using the identity matrix as the covariance matrix. The *mvtnorm* package in the R language and environment for statistical computing [31] is used for creating samples. The mean for group 1 is the origin, and the mean for the group 2 is $(2/\sqrt{d}, 2/\sqrt{d}, \ldots, 2/\sqrt{d})$, so that the Mahalanobis distance is 2. This configuration is equivalent to Breiman's "twonorm" benchmark model [6]. Training sample sizes n and dimensions d are given in Table 2. Non-contaminated training data are created by sampling uniformly from a pool of $2n$ observations with n observations from each group. The remaining observations are sampled uniformly to comprise the training and testing data sets. The data sets are contaminated with outliers in one of two ways. In *Type A* data sets, outlier observations are sampled for group 1, using a Gaussian distribution with covariance matrix 0.001 times the identity matrix and with a mean $(10/\sqrt{d}, 10/\sqrt{d}, \ldots, 10/\sqrt{d})$, so that the Mahalanobis distance between outliers and non-outliers is 10. In *Type B* data sets, outlier observations are sampled from both class distributions with the exception that the covariance matrix is multiplied by 100. Outliers comprise 10% of the observations in the training set, and are not present in the validation or testing data sets. Examples of the contaminated distributions are plotted in Figure 4.

The *Bayes rule* for the (non-contaminated) distributions places observations in the group for which the mean is closest because the data arises from Gaussian distributions with equal class prior probabilities [15]. For all values of d, the Bayes error is therefore $P(z > 1) \approx 15.87\%$, where $z \sim N(0, 1)$.

Misclassification rates for SVM with each of the three loss functions and four kernel functions tested and for Type A data sets are in Table 2. Using a robust loss function confers a significant advantage over the hinge loss when using the linear kernel on all 12 data sets. The type A outliers are clustered together and are able to 'pull' the separating surface for SVM with the hinge loss away from the non-contaminated data, while SVM with the robust loss functions can minimize the effect of the outliers. The advantage virtually disappears as a higher-rank kernel is used. SVM with a robust loss function outperforms SVM with the hinge loss on 9 of 12, 3 of 12, and 5 of 12 data sets with the degree-2 polynomial, degree-9 polynomial, and Gaussian kernels, respectively. When a nonlinear (and potentially discontinuous, in the



(a) (b)

Figure 4: Plots of simulated data sets contaminated with (a) Type A and (b) Type B outliers. The plots are for data sets with $n = 60$ and $d = 2$. The classifier selected by SVM, ramp loss SVM, and hard margin Loss SVM with the linear kernel and $C = 1.0$ is plotted, as well as the Bayes optimal rule. In the presence of Type A outliers, traditional SVM does not find a hyperplane but rather finds the "zero solution" as optimal, placing all observations in the "circle" group; ramp loss and hard margin Loss SVM ignore the outliers and produce classifiers that approximate the Bayes optimal rule reasonably well. For Type B outliers, the Bayes optimal rule is a combination of the robust classifiers and the traditional SVM classifier.

Gaussian case) separating surface is employed, the type B outliers can be assigned to the correct group in the training data set without affecting generalization performance. SVM with the ramp loss performs at least as well as SVM with the hard margin loss on 38 of 48 tests.

For type B outliers, using a robust loss function does not appear to confer an advantage over the hinge loss (data not shown). SVM with the robust loss functions performs at least as well as hinge-loss SVM on 32 of the 48 tests. SVM with the ramp loss outperforms SVM with the hard margin loss on 25 of 48 tests, and performs at least as well on 38 of 48 tests. This phenomenon is explained by the fact that the hard margin loss strictly penalizes observations falling in the margin - in the 'overlap' of the two groups of samples - while the ramp loss employs a continuous penalty for observations in the margin (as does the hinge loss).

5.2 Real-World Data

Nine real-world data sets from the UCI Machine Learning Repository [1] are used. The data set name, training set size, and number of attributes for each data set are given in Table 3. Observations with missing values are removed. Categorical attributes with k possible values are converted to k binary attributes, and are then treated as continuous attributes. Attributes with standard deviation 0 in the training set are removed from the training, validation, and testing data sets. Each attribute is normalized by subtracting the mean value in the training set and dividing by the standard deviation in the training set.

Results for SVM with the various loss functions and kernels on real-world data sets is in Table 4. There

Table 2: Misclassification Rates (%) for Type A Simulated Data Sets

		Linear Kernel			Deg. 2 Polynomial Kernel			Deg. 9 Polynomial Kernel			Gaussian Kernel		
			Hard			Hard			Hard			Hard	
n	d	Hinge	Margin	Ramp	Hinge	Margin	Ramp	Hinge	Margin	Ramp	Hinge	Margin	Ramp
60	2	53.3	13.3	13.3	26.7	30.0	26.7	26.7	50.0	26.7	16.7	16.7	16.7
100	2	52.0	20.0	28.0	28.0	26.0	24.0	26.0	32.0	28.0	24.0	24.0	26.0
200	2	55.0	17.0	18.0	20.0	20.0	20.0	20.0	21.0	19.0	16.0	17.0	17.0
500	2	50.0	16.0	16.4	20.4	19.6	20.0	20.0	30.0	19.6	17.2	16.0	16.0
60	5	53.3	16.7	16.7	20.0	16.7	20.0	36.7	36.7	36.7	13.3	16.7	16.7
100	5	46.0	24.0	24.0	24.0	26.0	32.0	22.0	38.0	22.0	26.0	26.0	26.0
200	5	60.0	17.0	15.0	19.0	22.0	17.0	22.0	24.0	24.0	16.0	16.0	15.0
500	5	52.8	12.0	12.8	18.0	19.6	15.2	19.6	22.4	18.8	14.8	13.6	15.2
60	10	40.0	16.7	16.7	23.3	16.7	16.7	30.0	30.0	30.0	16.7	16.7	16.7
100	10	52.0	22.0	10.0	16.0	14.0	16.0	20.0	20.0	20.0	10.0	12.0	10.0
200	10	56.0	20.0	13.0	32.0	22.0	16.0	20.0	20.0	20.0	17.0	25.0	16.0
500	10	50.8	13.6	12.0	14.8	15.2	14.0	18.8	18.8	18.8	12.8	19.2	12.4

Table 3: Real-World Training Data Sets

Label	Name in UCI Repository	n	d
adult	Adult	500	88
australian	Statlog (Australian Credit Approval)	326	46
breast[24]	Breast Cancer Wisconsin (Original)	341	9
bupa	Liver Disorders	172	6
german	Statlog (German Credit Data)	500	24
heart	Statlog (Heart)	135	19
sonar	Connectionist Bench (Sonar, Mines vs. Rocks)	104	60
wdbc[24]	Breast Cancer Wisconsin (Diagnostic)	284	30
wpbc[24]	Breast Cancer Wisconsin (Prognostic)	97	30

is no clear advantage to using one loss function over another. The ramp loss performs at least as well as the traditional SVM on 28 of 36 tests and the largest difference in misclassification rates is 4.6%. The ramp loss performs at least as well as the hard margin loss on 33 of 36 tests and outperforms the hard margin loss on 18 tests. These results give further evidence that the ramp loss is preferred to the hard margin loss. Also, the ramp loss has misclassification rates that are comparable to those of traditional SVM in the absence of outliers.

5.3 Comparisons with Other Classifiers

SVM with the ramp loss and hard margin loss is compared with other commonly-used classification methods using eleven data sets. The five real-world data sets of Section 5.2 with at least 500 observations are included as well as six simulated data sets. The simulated data sets are comprised of 1000 observations, each sampled from the distributions described in Section 5.1 for $d = 2, 5, 10$ and for type A and type B outliers. Ten percent (100) of the observations are sampled from the outlier distributions in each data set.

Each data set is partitioned into two sets, one for parameter tuning and one for testing. For each partition, 10-fold cross validation is performed. The settings with the best performance on test observations for the first partition are used for training in the second partition. Performance on the holdout data sets in the second partition is reported. Confidence intervals are constructed for the misclassification rate of each classifier.

SVM with the ramp loss and hard margin loss is compared to traditional SVM, classification trees, k-nearest neighbor, random forests, and logistic regression. The support vector machines are computed as previously described. Classification trees (CART), k-nearest neighbor, random forests, and logistic regression are trained and tested using the R language and environment for statistical computing [31] using the functions *rpart()*, *kknn()*, *randomForest()*, and *glm(family=binomial("logit"))*, respectively, which are contained in packages *rpart*[38], *kknn* [32], *randomForest* [5], and *stats* [31], respectively. SVM with the ramp loss and hard margin loss tuned for loss function (ramp loss or hard margin loss), C $(0.01, 0.1, 1, 10, 100$ for ramp loss, $1, 10, 100, 1000, 10000$ for hard margin loss), kernel (linear, degree-2 polynomial, degree-9 polynomial, Gaussian), and σ for the Gaussian kernel $(0.1, 1, 10, 100, 1000)$. Traditional SVM is tuned for the same parameter values except for the loss function. Classification trees are

Table 4: Misclassification Rates (%) for Real-World Data Sets

	Linear Kernel			Deg. 2 Polynomial Kernel			Deg. 9 Polynomial Kernel			Gaussian Kernel		
		Hard			Hard			Hard			Hard	
dataset	Hinge	Margin	Ramp	Hinge	Margin	Ramp	Hinge	Margin	Ramp	Hinge	Margin	Ramp
adult	17.5	20.3	17.7	18.1	20.3	18.3	20.9	21.9	20.8	22.6	22.7	22.7
australian	16.5	17.7	16.5	17.7	16.5	17.7	18.3	22.0	18.3	18.2	20.1	18.3
breast	2.3	2.9	2.3	2.9	4.1	3.5	5.3	6.4	5.3	4.1	4.1	4.1
bupa	36.8	34.5	32.2	32.2	37.9	29.9	36.8	37.9	33.3	27.6	33.3	31.0
german	0.0	0.0	0.0	0.0	0.0	0.0	1.6	1.6	1.6	3.6	3.6	3.6
heart	17.6	16.2	16.2	16.2	17.6	11.8	16.2	16.2	16.2	22.1	20.6	22.1
sonar	17.3	17.3	17.3	9.6	11.5	9.6	5.8	5.8	5.8	7.7	5.8	5.8
wdbc	1.4	2.1	1.4	2.1	2.8	2.1	1.4	1.4	1.4	3.5	3.5	3.5
wpbc	18.4	14.3	22.4	22.4	26.5	24.5	24.5	24.5	24.5	26.5	26.5	26.5

Table 5: Confidence Intervals for Misclassification Rate based on 10-fold Cross Validation

	Results: Average (95% CI width)					
dataset	SVM (hard margin & ramp loss)	SVM (hinge loss)	k-Nearest Neighbor	Classification Trees	Random Forest	Logistic Regression
adult	17.2(0.03)	17.0(0.03)	23.4(0.04)	20.0(0.04)	15.8(0.03)	16.8(0.03)
australian	14.1(0.04)	15.0(0.04)	14.7(0.04)	17.1(0.04)	11.8(0.04)	13.7(0.04)
breast	9.7(0.03)	3.5(0.02)	4.1(0.02)	6.2(0.03)	3.5(0.02)	5.3(0.02)
german	0.00(0.00)	0.0(0.00)	6.2(0.02)	0.0(0.00)	0.0(0.00)	0.0(0.00)
wdbc	3.9(0.02)	3.9(0.02)	5.6(0.02)	6.7 (0.03)	4.9(0.03)	7.0(0.03)
n1000d2A	19.6(0.03)	15.8(0.03)	15.8(0.03)	16.4(0.03)	17.0(0.03)	44.8(0.04)
n1000d2B	25.0(0.04)	23.0(0.04)	25.8(0.03)	25.6(0.04)	22.8(0.04)	42.4(0.04)
n1000d5A	16.6(0.03)	15.8(0.03)	17.8(0.03)	22.8(0.04)	17.8(0.03)	46.8(0.04)
n1000d5B	22.6(0.04)	21.2(0.04)	24.0(0.04)	32.4(0.04)	24.6(0.04)	29.6(0.04)
n1000d10A	24.8(0.04)	26.8(0.04)	16.8(0.04)	27.0(0.04)	17.0(0.03)	48.6(0.04)
n1000d10B	14.4(0.03)	14.4(0.03)	29.6(0.03)	34.8(0.04)	29.0(0.04)	28.8(0.04)

tuned for the split criterion (Gini or information) and k-nearest neighbor is tuned for k $(1, 3, 4, 7, 9)$ and distance function ($\mathcal{L}_1$ and $\mathcal{L}_2$). Random forests and logistic regression are used with default settings for all tests.

The 95% confidence intervals for misclassification rate are presented in Table 5. SVM with the ramp loss or hard margin loss obtains misclassification rates within 3.8% of the best classifier for all but two of the data sets, and achieves the minimum misclassification rate among the classifiers for 3 data sets. Traditional SVM achieves the minimum misclassification rate among the classifiers on 7 of 11 data sets. On the outlier-contaminated data sets, SVM with robust loss functions, traditional SVM, and k-nearest neighbor perform best. Classification trees and random forest have high misclassification rates in the presence of type B outliers, while logistic regression has high misclassification rates in the presence of both type A and type B outliers. Consistent with the results of Sections 5.1 and 5.2, SVM with the ramp loss and hard margin loss has misclassification rates that are comparable to those of traditional SVM for these data sets, and their robustness properties are not needed when a high-rank kernel is used for training.

6 Discussion

We have introduced new integer programming formulations for ramp loss and hard margin loss SVM that can accommodate nonlinear kernel functions. As traditional SVM with the hinge loss is a consistent classifier [36], we should not be too surprised that SVM with these robust loss functions is consistent as well. The formulations and solution methods for the ramp loss and hard margin loss SVM that are presented here can generate good solutions for instances that are an order of magnitude larger than previously attempted. The cuts introduced in Section 4.1 can be generalized to other math programming formulations where the number of misclassifications is minimized, and are independent of the method of regularization.

Using a branch-and-bound algorithm to solve instances of SVM with the robust loss functions is more computationally intensive than solving SVM instances with the hinge loss. In the worst case for the computational study presented here, the difference in computing time is approximately an order of magnitude. This result begs the question, “Is the extra computational time justified for the robust loss functions?” SVM with the hard margin loss can provide more robust classifiers in certain situations, but can also derive undesirable classifiers based on non-contaminated data because it strictly penalizes observations falling in the margin. SVM with the ramp loss performs no worse than SVM with the hinge loss, yet can provide more robust classifiers in the presence of outliers in certain situations.

The choice of kernel appears to be crucial as to whether SVM with the ramp loss will confer an advantage over SVM with the hinge loss. When using the linear kernel, SVM with the ramp loss is preferred to SVM with the hinge loss. As the rank of the kernel function is increased, the advantage of using a robust SVM formulation decreases. When using the most “complex” kernels, *universal kernels* [35], SVM with ramp loss provides no advantage. The reason for this can be seen in the definition of universal kernels and the property of universal kernels given in equation (8). Universal kernels project data into a space in such a way that none of the projected points are “far” from one another. Further, they are capable of learning nonlinear and discontinuous separating surfaces in the space of the original data. These properties eliminate the adverse effects of outliers, and a more robust formulation is not needed. We infer that the need for a robust formulation of SVM depends directly on the rank of the kernel function.

We conclude that when a low-rank kernel is used with SVM, it is advisable to employ the ramp loss to derive classifiers that are uninfluenced by outliers. If the number of observations is large so that computational time is a concern, we note that an ensemble classifier can be formed based on samples of the data. An open research question is to quantify the robustness of SVM as a function of kernel rank.

References

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] P. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[3] K.P. Bennett. Semi-supervised support vector machines. In *Neural Information Processing Systems*, pages 368–374, Vancouver, B.C., Canada, 1998.

[4] D. Bertsimas and R. Shioda. Classification and regression via integer optimization. *Operations Research*, 55:252–271, 2007.

[5] L. Breiman and A. Cutler. *randomForest: Breiman and Cutler’s random forests for classification and regression.* R port by A. Liaw and M. Wiener, 2009.

[6] Leo Breiman. Arcing classifiers. *Annals of Statistics*, 26:801–824, 1998.

[7] J.P. Brooks and E.K. Lee. Analysis of the consistency of a mixed integer programming-based multi-category constrained discriminant model. *to appear in Annals of Operations Research*, 2008.

[8] J.P. Brooks, R. Shioda, and A. Spencer. Discrete support vector machines. Technical Report C&O Research Report: CORR 2007-12, Department of Combinatorics and Optimization, University of Waterloo, 2007.

[9] O. Chapelle, V. Sindhwani, and S.S. Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

[10] C. Chen and O.L. Mangasarian. Hybrid misclassification minimization. *Advances in Computational Mathematics*, 5:127–136, 1996.

[11] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[12] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[13] T. Cover. Rates of convegence for nearest neighbor procedures. In *Proc. of Hawaii International Conference on System Sciences, Honolulu*, pages 413–415, 1968.

[14] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[15] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, 2001.

[16] R.J. Gallagher, E.K. Lee, and D.A. Patterson. Constrained discriminant analysis via 0/1 mixed integer programming. *Annals of Operations Research*, 74:65–88, 1997.

[17] ILOG. 2008.

[18] T. Joachims. *Advances in Kernel Methods - Support Vector Learning, B*, chapter Making large-scale SVM learning practical. B. Schölkopf and C. Burges and A. Smola (ed.). MIT-Press, 1999, http://svmlight.joachims.org.

[19] D.S. Johnson and F.P. Preparata. The densest hemisphere problem. *Theoretical Computer Science*, 6:93–107, 1978.

[20] G.J. Koehler and S.S. Erenguc. Minimizing misclassifications in linear discriminant analysis. *Decision Sciences*, 21:63–85, 1990.

[21] 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.

[22] O.L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

[23] O.L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, 14:801–807, 1968.

[24] O.L. Mangasarian and W.H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23:1–18, 1990.

[25] C. McDiarmid. *Surveys in Combinatorics 1989*, chapter On the method of bounded differences, pages 148–188. Cambridge UP, 1989.

[26] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, 1999.

[27] C. Orsenigo and C. Vercellis. Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics*, 14:221–234, 2003.

[28] C. Orsenigo and C. Vercellis. Evaluating membership functions for fuzzy discrete SVM. *Lecture Notes in Artificial Intelligence: Applications of Fuzzy Sets Theory*, 4578:187–194, 2007.

[29] C. Orsenigo and C. Vercellis. Softening the margin in discrete SVM. *Lecture Notes in Artificial Intelligence: Advances in Data Mining*, 4597:49–62, 2007.

[30] F. Pérez-Cruz and A.R. Figueiras-Vidal. Empirical risk minimization for support vector classifiers. *IEEE Transactions on Neural Networks*, 14:296–303, 2003.

[31] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[32] K. Schliep and K. Hechenbichler. *kknn: Weighted k-Nearest Neighbors Classification and Regression*, 2009.

[33] J. Shawe-Taylor and N. Christianini. *Kernel Methods for Pattern Analysis.* Cambridge UP, 2004.

[34] X. Shen, G.C. Tseng, X. Zhang, and W.H. Wong. On ψ-learning. *Journal of the American Statistical Association*, 98:724–734, 2003.

[35] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[36] I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18:768–791, 2002.

[37] I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51:128–142, 2005.

[38] T.M. Therneau and B. Atkinson. *rpart: Rrecursive Partitioning.* R port by B. Ripley, 2009.

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

[40] L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-06)*, 2006.

7 Appendix

8 Proof of Theorems 3.1 and 3.2.

Theorem 3.1 *Let $X \subset \mathbb{R}^d$ be compact and $k : X \times X \to \mathbb{R}$ be a universal kernel. Let $f_n^{k,C}$ be the classifier obtained by solving [SVMIP2(ramp)] for a training set with n observations. Suppose that we have a positive sequence (C_n) with $C_n/n \to 0$ and $C_n \to \infty$. Then for any $\epsilon > 0$,*

$$\lim_{n\to\infty} P(\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^*) > \epsilon) = 0$$

Proof. To establish the consistency of ramp loss SVM, first write the difference in population loss between $f_n^{k,C}$ and f^* as

$$\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^*) = \mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^\dagger) + \mathcal{L}(f^\dagger) - \mathcal{L}(f^*).$$

We will show that each of the differences above is bounded by $\epsilon/2$ for an appropriately-chosen $f^\dagger$.

The bound $\mathcal{L}(f^\dagger) - \mathcal{L}(f^*) < \epsilon/2$ follows directly from [36, Lemma 2]. Let

$$\begin{aligned}
B_1(P) &= \{x \in X : P(y = 1|\boldsymbol{x}) > P(y = -1|\boldsymbol{x})\},\\
B_{-1}(P) &= \{x \in X : P(y = -1|\boldsymbol{x}) > P(y = 1|\boldsymbol{x})\},\\
B_0(P) &= \{x \in X : P(y = -1|\boldsymbol{x}) = P(y = 1|\boldsymbol{x})\}.
\end{aligned}$$

Since k is universal, by [36, Lemma 2] there exists $\boldsymbol{w}^\dagger \in H$ such that $\boldsymbol{w}^\dagger \cdot \Phi(\boldsymbol{x}) \geq 1$ for all $\boldsymbol{x} \in B_1(P)$ except for a set of probability bounded by $\epsilon/4$ and $\boldsymbol{w}^\dagger \cdot \Phi(\boldsymbol{x}) \leq -1$ for all $\boldsymbol{x} \in B_{-1}(P)$ except for a set of probability bounded by $\epsilon/4$. Further, we can require that

$$\boldsymbol{w}^\dagger \cdot \Phi(\boldsymbol{x}) \in [-(1 + \epsilon/4), 1 + \epsilon/4] \tag{8}$$

for all $\boldsymbol{x}$. Setting $f^\dagger(\boldsymbol{x}) = \text{sgn}(\boldsymbol{w}^\dagger \cdot \Phi(\boldsymbol{x}))$, these conditions ensure that $\mathcal{L}(f^\dagger) - \mathcal{L}(f^*) < \epsilon/2$.

We now show that $\lim_{n\to\infty} \mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^\dagger) \leq \epsilon/2$. Let $\mathcal{R}(f)$ be the population ramp loss (with maximum value 2) for a classifier f, and let $\hat{\mathcal{R}}(f)$ be the empirical ramp loss for f. Then

$$\begin{aligned}
\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^\dagger) &\leq \mathcal{R}(f_n^{k,C}) - \mathcal{R}^\dagger + \epsilon/2 \qquad (9)\\
&\leq \hat{\mathcal{R}}(f_T^{k,C}) + \hat{C}_n(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}} - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (10)\\
&\leq \hat{\mathcal{R}}(f_T^{k,C}) + \frac{2B}{n}\sqrt{\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (11)\\
&\leq \hat{\mathcal{R}}(f_T^{k,C}) + \frac{2(2\sqrt{C})}{n}\sqrt{\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (12)\\
&\leq \hat{\mathcal{R}}(\text{sgn}(\boldsymbol{w}^\dagger \cdot \Phi)) + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{4\sqrt{C}}{n}\sqrt{\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (13)\\
&\leq \mathcal{R}(\text{sgn}(\boldsymbol{w}^\dagger \cdot \Phi)) + 2\sqrt{\frac{-\ln\gamma}{n}} + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{4\sqrt{C}}{n}\sqrt{\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (14)\\
&\leq 2\sqrt{\frac{-\ln\gamma}{n}} + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{4\sqrt{C}}{n}\sqrt{\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} + \epsilon/2 \qquad (15)
\end{aligned}$$

The right-hand side of the last line converges to $\epsilon/2$ as $C/n \to 0$ and $C \to \infty$. Inequality (9) is due to Lemma 8.1. Inequality (10) follows from [2] as stated in [33, Theorem 4.9], where $\hat{C}_n(\mathcal{F})$ is the empirical Rademacher complexity of the set of classifiers. Inequality (11) is due to [2, Theorem 21] as stated in [33, Theorem 4.12], where B^2 is an upper bound on the kernel function. Such an upper bound is guaranteed to exist because X is compact. Inequality (12) follows from the fact that $||\boldsymbol{w}|| \leq 2\sqrt{C}$ for any optimal solution of [SVMIP2(ramp)] so that $B \leq 2\sqrt{C}$. Inequality (13) is due to the fact that $f_n^{k,C}$ is optimal for [SVMIP2(ramp)], so that $1/2||\boldsymbol{w}^{k,C}||^2 + C\hat{R}(f_n^{k,C}) \leq 1/2||\boldsymbol{w}^\dagger||^2 + C\hat{R}(\boldsymbol{w}^\dagger \cdot \Phi)$. Inequality (14) follows from an application of McDiarmid's inequality [25] which implies that

$$P\left(\frac{\sum_{i=1}^n(\xi_i + 2z_i)}{n} - \mathcal{R}(\boldsymbol{w}^\dagger \cdot \Phi) \geq \gamma\right) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{2n}(2/n)^2}\right).$$

□

Lemma 8.1. *Let $\mathcal{L}(f)$ be the probability of misclassification for classifier f and let $\mathcal{R}(f)$ be the population ramp loss for classifier f. For a universal kernel k, if $f^\dagger$ is chosen as in [36, Lemma 2], then for any $\epsilon > 0$,*

$$\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^\dagger) \le \mathcal{R}(f_n^{k,C}) - \mathcal{R}(f^\dagger) + \epsilon/2.$$

Proof.

$$\begin{aligned}
\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^\dagger) &= \sum_{j\in\{\pm 1\}} (\int 1_{\{\boldsymbol{x}: jf_n^{k,C}(\boldsymbol{x})<-1\}} dx + \int 1_{\{\boldsymbol{x}: jf_n^{k,C}(\boldsymbol{x})\ge -1, \le 0\}} dx \\
&\quad - \int 1_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})<-1\}} dx - \int 1_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})\ge -1, \le 0\}} dx) \qquad (16) \\
&\le 2 \sum_{j\in\{\pm 1\}} \left(\int 1_{\{\boldsymbol{x}: jf_n^{k,C}(\boldsymbol{x})<-1\}} dx - \int 1_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})<-1\}} dx \right) \\
&\quad + \sum_{j\in\{\pm 1\}} \left(\int_{\{\boldsymbol{x}: jf_n^{k,C}(\boldsymbol{x})\ge -1, \le 0\}} (1 - jf_n^{k,C}) dx - \int_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})\ge -1, \le 0\}} (1 - jf^\dagger) dx \right) \qquad (17) \\
&= \mathcal{R}(f_n^{k,C}) - \mathcal{R}(f^\dagger) \\
&\quad - \sum_{j\in\{\pm 1\}} \int_{\{\boldsymbol{x}: jf_n^{k,C}(\boldsymbol{x})>0, \le 1\}} (1 - jf_n^{k,C}) dx + \sum_{j\in\{\pm 1\}} \int_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})>0, \le 1\}} (1 - jf^\dagger) dx \qquad (18) \\
&\le \mathcal{R}(f_n^{k,C}) - \mathcal{R}(f^\dagger) + \sum_{j\in\{\pm 1\}} \int_{\{\boldsymbol{x}: jf^\dagger(\boldsymbol{x})>0, \le 1\}} (1 - jf^\dagger) dx \qquad (19) \\
&\le \mathcal{R}(f_n^{k,C}) - \mathcal{R}(f^\dagger) + \epsilon/2 \qquad (20)
\end{aligned}$$

By [36, Lemma 2], we can select $f^\dagger$ in such a way that the last term in (18) is arbitrarily small. □

Theorem 3.2 *Let $X \subset \mathbb{R}^d$ be compact and $k : X \times X \to \mathbb{R}$ be a universal kernel. Let $f_n^{k,C}$ be the classifier obtained by solving [SVMIP2(hm)] for a training set with n observations. Suppose that we have a positive sequence (C_n) with $C_n/n \to 0$ and $C_n \to \infty$. Then for any $\epsilon > 0$,*

$$\lim_{n\to\infty} P(\mathcal{L}(f_n^{k,C}) - \mathcal{L}(f^*) > \epsilon) = 0$$

Proof. Let $\mathcal{R}(f)$ be the population ramp loss where the loss for an observation for which $yf(\boldsymbol{x}) > 0$ is 0 and the loss when $yf(\boldsymbol{x}) < -1$ is 1. Let $\hat{\mathcal{R}}(f)$ be the empirical ramp loss for f, and let $\hat{\mathcal{L}}$ be the empirical hard margin loss. Many of the steps in the proof correspond to steps in the proof of Theorem 3.1.

$$\begin{aligned}
\mathcal{L}(f_n^{k,C}) &\leq \mathcal{R}(f_n^{k,C}) && (21)\\
&\leq \hat{\mathcal{R}}(f_n^{k,C}) + \frac{2B}{n}\sqrt{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} && (22)\\
&\leq \hat{\mathcal{L}}(f_n^{k,C}) + \frac{2\sqrt{C}}{n}\sqrt{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} && (23)\\
&\leq \hat{\mathcal{L}}(\boldsymbol{w}^\dagger \cdot \Phi) + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{2\sqrt{C}}{n}\sqrt{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} && (24)\\
&\leq \mathcal{L}(\boldsymbol{w}^\dagger \cdot \Phi) + \sqrt{\frac{-\ln\gamma}{2n}} + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{2\sqrt{C}}{n}\sqrt{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} && (25)\\
&\leq \mathcal{L}(f^*) + \epsilon + \sqrt{\frac{-\ln\gamma}{2n}} + \frac{1}{2C}||\boldsymbol{w}^\dagger||^2 + \frac{2\sqrt{C}}{n}\sqrt{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} && (26)
\end{aligned}$$

The right-hand side of the last line converges to ϵ as $C/n \to 0$ and $C \to \infty$. Inequality (22) follows from [2, Theorem 21] as stated in [33, Theorems 4.9 and 4.12], where B^2 is an upper bound on the kernel function. Such an upper bound is guaranteed to exist because X is compact. Inequality (23) is due to the definitions of the losses and the upper bound for $||\boldsymbol{w}|| \leq \sqrt{C}$ for any optimal solution to [SVMIP2(hm)] so that $B \leq \sqrt{C}$. Inequality (24) is due to the fact that $f_n^{k,C}$ is optimal for [SVMIP2(ramp)], so that $1/2||\boldsymbol{w}^{k,C}||^2 + C\hat{L}(f_n^{k,C}) \leq 1/2||\boldsymbol{w}^\dagger||^2 + C\hat{L}(\boldsymbol{w}^\dagger \cdot \Phi)$. Inequality (25) follows from an application of McDiarmid's inequality [25] which implies that

$$P\left(\frac{\sum_{i=1}^{n} z_i}{n} - \mathcal{L}(\boldsymbol{w}^\dagger \cdot \Phi) \geq \gamma\right) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{n}(1/n)^2}\right).$$

Inequality (26) follows from the choice of $f^\dagger$ (and therefore $\boldsymbol{w}^\dagger \cdot \Phi$) whose existence is guaranteed by [36, Lemma 2]. □

9 Proof of Theorem 4.1.

Assumption 9.1. The observations $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ are in *general position*, meaning that no set of $d+1$ points lies in a $(d-1)$-dimensional subspace. Equivalently, every subset of $d+1$ points is *affinely independent.*

Lemma 9.1. *The convex hull of integer feasible solutions for [SVMIP1(ramp)] has dimension* $2n+d+1$.

Proof. There are $2n + d + 1$ variables in [SVMIP1(ramp)]. Let P^* be the polyhedron formed by the convex hull of integer feasible solutions to [SVMIP1(ramp)]. We will show that that no equality holds for every solution in P^* (i.e., the affine hull of the integer feasible solutions is $\mathbb{R}^{2n+d+1}$), from which we can conclude that $\dim(P^*) = 2n + d + 1$. Let ω_j be the multiplier for the w_j for each j, β be the multiplier for b, σ_i be the multiplier ξ_i for each i, and ζ_i be the multiplier for the z_i for each i. For a constant c, suppose that the following equality holds for all points in P^*.

$$\sum_{j=1}^{d} \omega_j w_j + \beta b + \sum_{i=1}^{n} \sigma_i \xi_i + \sum_{i=1}^{n} \zeta_i z_i + c = 0 \tag{27}$$

For a given training data set, consider a separating hyperplane (and assignment of resulting half-spaces to classes) that is "far" from all of the observations that classifies all observations in class +1. Such a hyperplane corresponds to a feasible solution for [SVMIP1(ramp)] with $z_i = 1$ for all i with $y_i = -1$. For an arbitrary w_j, there exists a $\delta > 0$ small enough such that adding δ to w_j results in another feasible solution with no changes to other variable values. Plugging the two solutions into (27) and taking the difference implies that $\omega_j = 0$. Because w_j is chosen arbitrarily, $\omega_j = 0$ for all j. A similar argument shows that $\beta = 0$. The equality (27) now has the form

$$\sum_{i=1}^{n} \sigma_i \xi_i + \sum_{i=1}^{n} \zeta_i z_i + c = 0 \tag{28}$$

Consider again an observation that is "far" from all observations that classifies all observations in class +1, so that $z_i = 1$ for i with $y_i = -1$. Note that for observations with $z_i = 1$, the value of ξ_i can be changed without changing the values of any z_i variables or any other ξ_i variables and remain feasible. Taking the difference of the two solutions yields $\sigma_i = 0$ for all such i. Similar reasoning yields $\sigma_i = 0$ for observations with $y_i = +1$. The equality (27) now has the form

$$\sum_{i=1}^{n} \zeta_i z_i + c = 0 \tag{29}$$

Consider an observation $\boldsymbol{x}_i$ that defines the convex hull of observations. By Assumption 9.1, there exists a hyperplane that separates $\boldsymbol{x}_i$ from the other observations. Therefore, we can find solutions to [SVMIP1(ramp)] with $z_i = 1$ and $z_i = 0$ with no other z_i variable values unchanged. Plugging these solutions into equation 29 and taking the difference yields $\zeta_i = 0$. Therefore, $\zeta_i = 0$ for any observation that defines the convex hull of observations. Discarding the observations that define the convex hull and applying the same reasoning to the observations that define the convex hull of the remaining observations yields $\zeta_i = 0$ for those observations. Continuing in the same fashion yields $\zeta_i = 0$ for all i and therefore $c = 0$. There is no equality that holds for all points in P^*, and P^* has dimension $2n + d + 1$. □

Lemma 9.2. *Given a set of $d + 1$ points $H = \{\boldsymbol{x}_i : y_i = 1,\ i = 1, \ldots, d + 1\}$ and another point $\boldsymbol{x}_{d+2}$ with label $y_{d+2} = -1$ such that $\boldsymbol{x}_{d+2}$ falls in the convex hull of the other $d + 1$ points, then*

$$F = \{(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{z}) \in P^* : \sum_{i=1}^{d+2} \xi_i + \sum_{i=1}^{d+2} z_i = 1\}$$

defines a proper face of P^*.

Proof. Consider a hyperplane "far" from all of the observations and an assignment of its half-spaces to classes that places all observations in class +1, so that $\boldsymbol{x}_{d+2}$ is the only observation in H that is misclassified with $z_{d+2} = 1$, $\xi_{d+2} = 0$, and $\xi_i = 0$ for $i = 1, \ldots, d{+}1$. There exists a corresponding solution that is feasible for [SVMIP1(ramp)], which proves that $F \neq \emptyset$. Now consider the same hyperplane with assignment of half-spaces to classes that places all observations in class -1. Then there is a corresponding solution to [SVMIP1(ramp)] that has $\sum_{i=1}^{d+2} z_i = d{+}1$. The solution does not lie on F, so $F \neq P^*$. Because $F \neq \emptyset$ and $F \neq P^*$, F is a proper face of P^*. □

Lemma 9.3. *The face F defined in Lemma 9.2 has dimension* $\dim(P^*) - 1$.

Proof. We will now show that F is a facet for P^* by showing that the inequality that defines F has dimension $\dim(P^*) - 1$, in accordance with Theorem 3.6 on page 91 of [26]. We will show that only one equality holds for all points in F. Suppose for multipliers ω_j, $j = 1, \ldots, d$; β; σ_i, $i = 1, \ldots, n$; and ζ_i, $i = 1, \ldots, n$ and a constant c that the following equality holds for all solutions in F.

$$\sum_{j=1}^{d} \omega_j w_j + \beta b + \sum_{i=1}^{n} \zeta_i z_i + c = 0 \tag{30}$$

Consider a hyperplane that is "far" from the points in the training data set that places all observations in class +1. There exists a corresponding solution with $z_{d+2} = 1$, $\xi_{d+2} = 0$, $\xi_i = 0$ for $i = 1, \ldots, d+1$, and $z_i = 0$ for $i = 1, \ldots, d+1$ in F. Choosing an arbitrary w_j and tilting the hyperplane slightly as in the proof of Lemma 9.1 produces another solution in F. Plugging the solutions into into (30) and taking the difference implies that $\omega_j = 0$. Because w_j is chosen arbitrarily, $\omega_j = 0$ for all j. A similar argument shows that $\beta = 0$.

Now consider an observation $\boldsymbol{x}_i$ that is not in the convex hull of points in H. There exists a hyperplane separating the observation from all points in H, so that there exist separate solutions placing all observations in H in class +1 with $\xi_i = 0$ and $z_i = 1$, $\xi_i > 0$ and $z_i = 1$, and $\xi_i = 0$ and $z_i = 0$, respectively. These solutions imply that $\sigma_i = \zeta_i = 0$ for all observations not in the convex hull of points in H.

By Assumption 9.1, no observation lies in the convex hull of any set of d other observations. Accordingly, observation $\boldsymbol{x}_{d+2}$ does not lie in the convex hull of any set of d other observations in H. Also, the line segment connecting $\boldsymbol{x}_1$ with $\boldsymbol{x}_{d+2}$ does not intersect the convex hull of $\{\boldsymbol{x}_i : i = 2, \ldots, d+1\}$, and therefore a hyperplane exists that separates the two sets. Assigning the half space with $\boldsymbol{x}_1$ and $\boldsymbol{x}_{d+2}$ to the class -1 can generate solutions that lie in F, as $\boldsymbol{x}_1$ is the only observation in H misclassified. Now consider an observation $\boldsymbol{x}_k$ that is not in H but is in the convex hull of $\{x_i : i = 2, \ldots, d+2\}$. There exist hyperplanes "near" $\boldsymbol{x}_k$ corresponding to solutions with $z_k = 0$ and $\xi_k = 0$, $\xi_k = 0$ and $z_k = 1$, and $\xi_k > 0$ and $z_k = 1$, respectively, while maintaining $z_1 = 1$, $z_i = 0$ for $i \in H \setminus \{1\}$, and constant values for all other z_i variables. The difference between these solutions implies that $\sigma_k = \zeta_k = 0$. Similar reasoning can be used to show that $\sigma_k = \zeta_k = 0$ for all $k \notin H$ with $\boldsymbol{x}_k$ in the convex hull of H.

We now have that (30) reduces to

$$\sum_{i \in H} \sigma_i \xi_i + \sum_{i \in H} \zeta_i z_i + c = 0. \tag{31}$$

Consider again a solution with $z_1 = 1$, $z_i = 0$ for $i \in H \setminus \{1\}$, and $\xi_i = 0$ for $i \in H$. This solution implies that $\zeta_1 = -c$. Consider also a solution with $\xi_1 = 1$, $\xi_i = 0$ for $i \in H \setminus \{1\}$, and $z_i = 0$ for $i \in H$. This solution implies that $\sigma_1 = -c$. Similar reasoning can be used to show that $\sigma_i = \zeta_i = -c$ for $i = 1, \ldots, d+1$.

Consider again a solution that places all observations in class +1 so that $z_{d+2} = 1$ and $z_i = 0$ for $i \in H \setminus \{d+2\}$ and $\xi_i = 0$ for $i \in H$. This solution implies that $\zeta_{d+2} = -c$. Consider also a solution with $\xi_{d+2} = 1$, $\xi_i = 0$ for $i \in H \setminus \{d+2\}$, and $z_i = 0$ for $i \in H$. This solution implies that $\sigma_{d+2} = -c$. Plugging the σ_i and ζ_i values into (31) produces

$$\sum_{i \in H} \xi_i + \sum_{i \in H} z_i = 1 \tag{32}$$

which is the equality that defines F. Therefore, (32) is the only equality satisfied by all points in F, and F has dimension $\dim(P^*) - 1$. □

Theorem 4.1 *[8]. Given a set of $d+1$ points $\{\boldsymbol{x}_i : y_i = 1,\ i = 1, \ldots, d+1\}$ and another point $\boldsymbol{x}_{d+2}$ with label $y_{d+2} = -1$ such that $\boldsymbol{x}_{d+2}$ falls in the convex hull of the other $d+1$ points, then*

$$\sum_{i=1}^{d+2} \xi_i + \sum_{i=1}^{d+2} z_i \geq 1$$

defines a facet for the convex hull of integer feasible solutions for [SVMIP1(ramp)].

Proof. The theorem follows directly from Lemmas 9.1-9.3. □