**Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 2**
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima,
Soo-Young Lee, and Xin Yao (Editors)

# RADIUS MARGIN BOUNDS FOR SUPPORT VECTOR MACHINES WITH THE RBF KERNEL

*Kai-Min Chung, Wei-Chun Kao, Tony Sun, Li-Lun Wang, and Chih-Jen Lin*

National Taiwan University
Department of Computer Science
Taipei 106, Taiwan

## ABSTRACT

An important approach for efficient support vector machine (SVM) model selection is to use differentiable bounds of the leave-one-out (loo) error. Past efforts focused on finding tight bounds of loo. However, their practical viability is still not very satisfactory. In [5], it has been shown that radius margin bound gives good prediction for L2-SVM. In this paper, through the analyses why this bound performs well for L2-SVM, we show that finding a bound whose minima are in a region with small loo values may be more important than its tightness. Based on this principle we propose modified radius margin bounds for L1-SVM where the original bound is only applicable to the hard-margin case. Our modification for L1-SVM achieves comparable performance to L2-SVM.

## 1. INTRODUCTION

Recently, support vector machines (SVM) [8] have been a promising tool for data classification. Its success depends on the tuning of several parameters which affect the generalization error. The error can be estimated by, for example, testing some data which are not used for training (e.g., cross validation) or by a bound from theoretical derivation. The goal of this paper is to make one of these bounds, radius margin bound, a practical tool.

Given training vectors $x_i \in R^n, i = 1, \ldots, l$, in two classes, and a vector $y \in R^l$ such that $y_i \in \{1, -1\}$, SVM solves:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w && (1.1) \\ \text{s.t.} \quad & y_i(w^T \phi(x_i) + b) \geq 1, i = 1, \ldots, l, \end{aligned}$$

where $x_i$ are mapped to a higher dimensional space by the function $\phi$. Practically, we need only $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, the kernel function. In this paper, we focus on the RBF kernel $K(x_i, x_j) = e^{-\|x_i - x_j\|^2/(2\sigma^2)}$. The parameter $\sigma$ is usually determined by an estimation of generalized error such as leave-one-out (loo) or cross validation. It was shown in [8] that the following radius margin bound holds

$$loo \leq 4R^2 \|w\|^2, \qquad (1.2)$$

where $loo$ is the number of loo errors, $w$ is the solution of (1.1), and $R$ is the radius of the smallest sphere containing all $\phi(x_i)$. It has been shown (e.g. [8]) that $R^2$ is the objective value of

$$\begin{aligned} \max_{\beta} \quad & 1 - \beta^T K \beta \\ \text{s.t.} \quad & e^T \beta = 1, 0 \leq \beta_i, i = 1, \ldots, l. && (1.3) \end{aligned}$$

Some early experiments on minimizing the right-hand side of (1.2) are in [6].

However, (1.1) is not useful in practice. It may not be feasible if $\phi(x_i)$ are not linearly separable. In addition, a highly nonlinear $\phi$ may lead to overfitting. Thus, practically we solve either

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \quad \text{or} \quad \min_{w,b,\xi} \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{l} \xi_i^2 (1.4)$$

where $\xi_i$ represents the training error and the parameter $C$ adjusts the training error and the regularization term $w^T w/2$. We refer to the two cases as L1-SVM and L2-SVM, respectively. Note that for L2-SVM we use $C/2$ instead of $C$ for easier analysis later. Then, if the RBF kernel is used, $C$ and $\sigma$ are the two tunable parameters. Usually they are solved through dual problems. For L1-SVM, its dual is

$$\begin{aligned} \max_{\alpha} \quad & e^T \alpha - \frac{1}{2} \alpha^T Q \alpha \\ \text{s.t.} \quad & y^T \alpha = 0, 0 \leq \alpha_i \leq C, i = 1, \ldots, l, && (1.5) \end{aligned}$$

where $e$ is the vector of all ones and $Q$ is an $l \times l$ matrix with $Q_{ij} = y_i y_j K(x_i, x_j)$. For primal and dual optimal solutions,

$$w = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) \text{ and } \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i = e^T \alpha - \frac{1}{2} \alpha^T Q \alpha. \qquad (1.6)$$

The dual of L2-SVM is

$$\begin{aligned} \max_{\alpha} \quad & e^T \alpha - \frac{1}{2} \alpha^T (Q + \frac{I}{C}) \alpha \\ \text{s.t.} \quad & y^T \alpha = 0, 0 \leq \alpha_i, i = 1, \ldots, l. && (1.7) \end{aligned}$$

Unfortunately, for L1-SVM, the radius margin bound cannot be used as (1.2) does not hold. However, L2-SVM can be reduced to a form of (1.1) using $\tilde{w} \equiv \begin{bmatrix} w \\ \sqrt{C}\xi \end{bmatrix}$ and the $i$th training data as $\begin{bmatrix} \phi(x_i) \\ \frac{e_i}{\sqrt{C}} \end{bmatrix}$, where $e_i$ is a zero vector of length $l$ except the $i$th component is one. Then, (1.2) can be directly used so existing work on the radius margin bound focus on L2-SVM (e.g. [3, 7]). Note that $R^2$ is now also different so we will denote the new bound as $\tilde{R}^2 \|\tilde{w}\|^2$, where $\tilde{R}^2$ is the objective value of the following problem which has the same constraints as (1.3):

$$\max_{\beta} \quad 1 + \frac{1}{C} - \beta^T (K + \frac{I}{C}) \beta. \qquad (1.8)$$

[3] is the first to use the differentiability of (1.2) and develop optimization algorithms for finding optimal $C$ and $\sigma$. This is much

$C$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| L1 test error | 10.4(0.6) | 1.9(1.0) | 9.5(3.4) | 10.2(3.2) | 10.9(3.8) |
| (1.9) | 55.9(10.0) | 25.6(10.0) | 15.5(8.1) | 15.3(8.0) | 17.0(-2.5) |
| L2 test error | 11.2(-1.4) | 2.4(0.5) | 9.5(3.3) | 9.9(2.7) | 10.5(4.6) |
| $\tilde{R}^2\|\tilde{w}\|^2$ | 11.4(-1.6) | 3.0(-0.3) | 10.0(3.1) | 10.3(2.1) | 17.0(-2.5) |

$\sigma$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| L1 test error | 10.4(5.2) | 1.9(3.9) | 9.5(0.4) | 10.2(1.4) | 10.9(8.6) |
| (1.9) | 38.4(-2.9) | 12.9(-2.9) | 19.5(-2.4) | 14.1(-2.5) | 26.1(-10.0) |
| L2 test error | 11.2(-0.0) | 2.2(2.4) | 10.1(2.1) | 10.0(-0.0) | 10.5(9.7) |
| $\tilde{R}^2\|\tilde{w}\|^2$ | 11.2(-0.9) | 3.0(0.4) | 10.2(10.0) | 10.0(-0.6) | 14.1(-1.4) |

Table 2.1: Performance of RM bounds for L1- and L2-SVM comparing to the best test accuracy (error rate in percentage and best $\ln\sigma^2$ (or $\ln C$)).

faster than a two-dimensional grid search. More implementation issues for solving large problems are discussed in [7]. For L1-SVM, as (1.2) does not hold, some modifications are necessary. In [5], following the suggestion by Chapelle, they consider

$$D^2 e^T \alpha + \sum_{i=1}^{l} \xi_i, \tag{1.9}$$

where $D = 2R$.

However, experiments in [5] showed that comparing to other methods, this bound is not good. In addition, implementation issues such as the differentiability of (1.9) have not been addressed.

Section 2 shows that different from (1.9), the RM bound for L2-SVM possesses some nice properties so that minima happen in the region where the error is small. We show that finding a bound whose minima are in a good region may be more important than its tightness. Based on the discussion for L2-SVM, in Section 3, we propose some modifications for L1-SVM which perform better than (1.9). However, these bounds, including (1.9), may not be differentiable so we propose some further modifications.

In Section 4, we show that in terms of testing accuracy as well as computational cost, the proposed modification for L1-SVM is competitive with that for L2-SVM. We also discuss several implementation issues not studied before.

Due to space limit, we leave detail of proofs and complete experimental results in [4].

## 2. RADIUS-MARGIN (RM) BOUND FOR L2-SVM

We investigate why the radius-margin (RM) bound performs well for L2-SVM. First, in Table 2.1 we list test accuracy given in [5] by comparing (1.9) for L1-SVM, and the RM bound for L2-SVM. For each problem, we fix $C$ (or $\sigma^2$) following [5], and then search for the value of $\sigma^2$ (or $C$) that minimizes the bound on ($\ln\sigma^2, \ln C$) plane. The ($C$, $\sigma^2$) is then used to train a model and predict the test data.

Some immediate observations are as follow:

1. No matter $C$ or $\sigma$ is fixed, the RM bound for L2-SVM is better.
2. When $C$ is fixed, except for problem tree, the modified RM bound for L1-SVM has minima at large $\sigma$. In other words, a good $\sigma$ should be smaller.
3. When $\sigma$ is fixed, for each problem, the modified RM bound for L1-SVM has the minimum at a smaller value than the best $C$.

Therefore, (1.9) suffers from the problem that the obtained $C$ is too small and $\sigma$ is too large. At the same time, the RM bound for L2-SVM may have inherently avoided that the minimum happens at too small $C$ or too large $\sigma$. In the following we derive two other bounds for L2-SVM and through the comparison we show the RM bound for L2-SVM really possesses such mechanisms.

Remember the RM bound for L2-SVM is derived by considering the hard margin SVM and using the following inequality (see, for example, Lemma 3 of [9]) : If in the leave-one-out procedure a support vector $x_p$ corresponding to a non-zero dual variable $\alpha_p > 0$ is recognized incorrectly, then

$$\frac{1}{2\max_{i\neq j}\|\tilde{z}_i - \tilde{z}_j\|^2} \leq \frac{\alpha_p^2}{2}\min_{\tilde{z}\in\Lambda_p^*}\|\tilde{z}_p - \tilde{z}\|^2, \tag{2.1}$$

where

$$\tilde{z}_i \equiv [\phi(x_i)^T, e_i^T/\sqrt{C}]^T \tag{2.2}$$

by changing L2-SVM to a hard-margin SVM formulation and $\Lambda_p^*$ is a subset of the convex hull by $\{\tilde{z}_1, \ldots, \tilde{z}_l\}\backslash\{\tilde{z}_p\}$. Note that we slightly modify the derivation in [9] where the left-hand side of (2.1) is $1/(2\tilde{D}^2)$. By the definition of $\Lambda_p^*$,

$$\min_{\tilde{z}\in\Lambda_p^*}\|\tilde{z}_p - \tilde{z}\|^2 \leq \max_{i\neq j}\|\tilde{z}_i - \tilde{z}_j\|^2. \tag{2.3}$$

Since $\tilde{D}$ is the diameter of the smallest sphere containing all $\tilde{z}_1, \ldots, \tilde{z}_l$,

$$\max_{i\neq j}\|\tilde{z}_i - \tilde{z}_j\|^2 \leq \tilde{D}^2 = 4\tilde{R}^2. \tag{2.4}$$

Therefore, if $x_p$ is recognized incorrectly, (2.1), (2.3), and (2.4) imply $\alpha_p^2\tilde{D}^4 \geq 1$. Then, $\alpha_p D^2 \geq 1$ so with $\sum_p \alpha_p = \|\tilde{w}\|^2$, we have

$$loo \leq \sum_{p=1}^{l}\alpha_p\tilde{D}^2 = 4\|\tilde{w}\|^2\tilde{R}^2.$$

Instead of using (2.4), from (2.2), we consider

$$\|\tilde{z}_i - \tilde{z}_j\|^2 = \frac{2}{C} + \|\phi(x_i) - \phi(x_j)\|^2.$$

With

$$\|\phi(x_i) - \phi(x_j)\|^2 \leq D^2 = 4R^2,$$

where $R^2$ is the objective value of (1.3), we obtain a different bound:

$$loo \leq (4R^2 + \frac{2}{C})\|\tilde{w}\|^2. \tag{2.5}$$

We can prove that under some conditions, (2.5) is a tighter bound:

**Theorem 1** *If $\sigma$ is fixed and $R^2 > 1/2$, then*

$$(R^2 + \frac{0.5}{C})\|\tilde{w}\|^2 \leq \tilde{R}^2\|\tilde{w}\|^2.$$

For the five problems tested earlier, when $\sigma$ is fixed as values in Table 2.1, $R^2 > 1/2$ holds.

Some immediate comparisons between the two bounds are as follows. When $\sigma$ is fixed, we can easily prove that $\lim_{C\to\infty}\tilde{R}^2 = \lim_{C\to\infty}R^2 = \lim_{C\to\infty}(R^2 + \frac{0.5}{C})$. However, for small $C$ $\tilde{R}^2 \approx \frac{1}{C}$ but $(R^2 + \frac{0.5}{C}) \approx \frac{0.5}{C}$. Therefore, when $C$ is small, $\tilde{R}^2\|\tilde{w}\|^2$ overestimates the loo error. Interestingly this becomes a good property due to the following reason. From (2.1), the RM bound seriously overestimates the loo if $\alpha_p$ is large. This happens only when $C$ is not small. Thus, large $\alpha$ happens only when $C$ is large. Therefore, the overestimation of loo at large $C$ pushes the minimum to be at a smaller value of $C$. Therefore, we can think that $\tilde{R}^2$ puts penalty at small $C$ so the minimum of the RM bound may be pushed back to the correct position. In Table 2.2, we will see

$C$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| $\bar{R}^2\|\tilde{w}\|^2$ | 11.4(-1.6) | 3.0(-0.3) | 10.0(3.1) | 10.3(2.1) | 17.0(-2.5) |
| (1.2) | 11.4(-1.6) | 3.0(-0.3) | 10.0(3.1) | 12.7(5.6) | 17.0(-2.5) |
| (2.7) | 11.4(-1.6) | 27.2(9.5) | 10.0(3.1) | 13.5(6.5) | 17.0(-2.5) |

$\sigma$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| $\bar{R}^2\|\tilde{w}\|^2$ | 11.2(-0.9) | 3.0(0.4) | 10.1(10.0) | 10.0(-0.6) | 14.1(-1.4) |
| (1.2) | 11.5(-1.5) | 3.7(-0.7) | 12.5(-1.4) | 10.5(-1.4) | 17.6(-2.3) |
| (2.7) | 11.8(-2.1) | 5.6(-1.8) | 48.0(-10.0) | 11.2(-2.2) | 26.1(-3.1) |

Table 2.2: Comparison of three RM bounds for L2-SVM (error rate and best $\ln\sigma^2$ (or $\ln C$)).

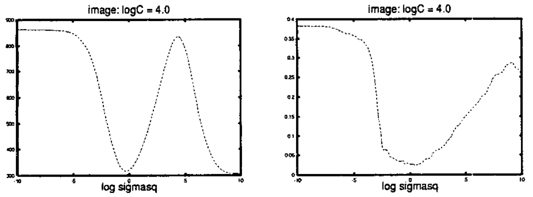

Figure 1: image dataset. The left one is the bound $(R^2 + 0.25/C)\|\tilde{w}\|^2$, and the right one is the test error. For the bound, the minimum is at $\ln\sigma^2 = 9.5$.

that when $\sigma$ is fixed, $(R^2 + \frac{0.5}{C})\|\tilde{w}\|^2$ always returns a smaller $C$ than $\bar{R}^2\|\tilde{w}\|^2$.

Next, we present another bound for L2-SVM. It motivates from the derivation of (1.9) which is a modified RM bound for L1-SVM. Remember that the original RM bound applies only to the hard-margin SVM. The derivation of (1.9) uses:

$$y_p(f^0(x_p) - f^p(x_p)) \le \alpha_p D^2, \tag{2.6}$$

where $x_p$ is any support vector, $\alpha_p$ is its corresponding dual variable, and $f^0$ and $f^p$ are the decision function trained, respectively, on the whole set and after the point $x_p$ has been removed.

When there is an loo error, $\alpha_p > 0$ so $y_p f^0(x_p) = 1 - \xi_p$. Therefore, $0 \le -y_p f^p(x_p) \le \alpha_p D^2 - 1 + \xi_p$ implies (1.1)

For L2-SVM, (2.6) still holds under the same assumption. So $D^2 e^T\alpha + \sum_{i=1}^{l}\xi_i$ can be considered as a bound of loo for L2-SVM. With $\alpha = C\xi$ and $\|\tilde{w}\|^2 = e^T\alpha$ for L2-SVM, $D^2 e^T\alpha + \sum_{i=1}^{l}\xi_i = D^2 e^T\alpha + \frac{1}{C}e^T\alpha$. Therefore, a new bound for L2 is:

$$(R^2 + \frac{0.25}{C})\|\tilde{w}\|^2. \tag{2.7}$$

In Table 2.2, it can be clearly seen that both new bounds are worse than the original RM bound, and $(R^2 + 0.25/C)\|\tilde{w}\|^2$ which is a tighter bound than (2.5) is particularly bad.

We observe that when $C$ is fixed, the accuracy of using $(R^2 + 0.5/C)\|\tilde{w}\|^2$ for waveform is not good because it obtains a too large $\sigma$. Then, for $(R^2 + 0.25/C)\|\tilde{w}\|^2$, it returns an even larger $\sigma$ so the error rate further increases. Especially for the problem image a very large $\sigma = 9.5$ is obtained.

If $\sigma \to \infty$, we can prove that $\bar{R}^2\|\tilde{w}\|^2 \approx \frac{4l_1l_2}{l}$, $(R^2 + \frac{0.5}{C})\|\tilde{w}\|^2 \approx \frac{2l_1l_2}{l}$, $(R^2 + \frac{0.25}{C})\|\tilde{w}\|^2 \approx \frac{l_1l_2}{l}$, where $l_1$ and $l_2$ are the number of data with $y_i = 1$ and $-1$. Thus, smaller values when $\sigma \to \infty$ cause the bound $(R^2 + 0.25/C)\|\tilde{w}\|^2$ to have minima at large $\sigma$. To confirm this in 1 we present the value of $(R^2 + 0.25/C)\|\tilde{w}\|^2$, using the problem image.

Clearly there are two local minima where the left one is better but is not chosen. This seems to suggest that as $\bar{R}^2\|\tilde{w}\|^2$ has larger values when $\sigma \to \infty$, it can avoid that the global minimum happens at a wrong place.

Besides, when $\sigma$ is fixed, the situation is also similar. The tighter the bound is, the worse the test accuracy is as the optimal $C$ becomes too small. We can prove that changing the bound from $\bar{R}^2$ to $R^2 + \frac{0.5}{C}$ and then $R^2 + \frac{0.25}{C}$, the optimal $C$ decreases.

Therefore, there is a benign overestimation for $\bar{R}$ when $C$ is small. In summary, finding a bound whose minima are in a good region may be more important than its tightness. In addition, a good bound should avoid that minima happen at the boundary (i.e., too small or too large $C$ and $\sigma^2$).

## 3. SOME HEURISTIC BOUNDS FOR L1-SVM

Based on the experience in the previous section, we search for better radius margin bounds for L1-SVM which can replace the $D^2 e^T\alpha + \sum_{i=1}^{l}\xi_i$ used in [5]. Our strategy is to consider that $(R^2 + \frac{0.25}{C})\|\tilde{w}\|^2$ for L2-SVM is the counterpart of $D^2 e^T\alpha + \sum_{i=1}^{l}\xi_i$ for L1-SVM as they follow from the same derivation. Then, by investigating the difference between $\bar{R}^2\|\tilde{w}\|^2$ and $(R^2 + \frac{0.25}{C})\|\tilde{w}\|^2$, we seek for the counterpart of $\bar{R}^2\|\tilde{w}\|^2$ for L1-SVM.

If we consider $\bar{R}^2 \approx R^2 + \frac{1}{C}$, $\bar{R}^2\|\tilde{w}\|^2$ is similar to $(R^2 + \frac{1}{C})\|\tilde{w}\|^2 = R^2 e^T\alpha + \sum_{i=1}^{l}\xi_i$ as $\|\tilde{w}\|^2 = e^T\alpha$ and $\alpha = C\xi$ for L2-SVM. Thus, for L1-SVM,

$$R^2 e^T\alpha + \sum_{i=1}^{l}\xi_i \tag{3.1}$$

may be a good bound. Now using (1.6), $C\sum_{i=1}^{l}\xi_i = e^T\alpha - \|w\|^2 \le e^T\alpha$, so another possibility is

$$(R^2 + \frac{1}{C})e^T\alpha = (R^2 + \frac{1}{C})(\|w\|^2 + C\sum_{i=1}^{l}\xi_i). \tag{3.2}$$

We will conduct experiments with these two new bounds later. Now we move on to another important issue: their differentiability. Then gradient-based methods can be used to find a local minimum. Unfortunately, in [4] we show that may not be differentiable.

Interestingly though $e^T\alpha$ is not a differentiable function of parameters, $e^T\alpha + C\sum\xi_i = \|w\|^2 + 2C\sum\xi_i$ is. The main reason that $\frac{1}{2}\|w\|^2 + C\sum\xi_i = e^T\alpha - \frac{1}{2}\alpha^T Q\alpha$ is differentiable is that they are primal and dual objective functions at any given parameter set. Now both primal and dual solutions are functions of $C$ and $\sigma$ in some unknown forms. So it may not be easy for these bounds to be differentiable at parameters. However, for the primal or dual objective functions, using results of perturbation analysis of optimization problems, under some conditions, they are differentiable.

To have differentiability, we propose a further modification of (3.2):

$$(R^2 + \frac{\Delta}{C})(\|w\|^2 + 2C\sum_{i=1}^{l}\xi_i), \tag{3.3}$$

where $\Delta$ is a positive constant close to one. As we discussed in Section 2, $\Delta/C$ can be thought as a penalty term for small $C$.

If we take $\Delta = 1$, the main change from (3.2) is to replace $e^T\alpha$ by a differentiable term.

Table 3.1 presents a comparison of different bounds for L1-SVM: (1.9), (3.1), (3.2), and the differentiable bound (3.3) with $\Delta = 1$ and $\Delta = 0.5$. It can be clearly seen that bounds proposed in this section is better than (1.9).

$C$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| $D^2(e^T\alpha) + \sum \xi_i$ | 55.9(10.0) | 25.6(10.0) | 15.5(8.1) | 15.3(8.0) | 17.0(-2.5) |
| $R^2(e^T\alpha) + \sum \xi_i$ | 35.2(-6.6) | 3.77(-1.2) | 9.6(3.2) | 13.5(6.8) | 17.0(-2.5) |
| $(R^2 + 1/C)e^T\alpha$ | 35.2(-6.6) | 3.8(-1.2) | 14.2(6.1) | 11.4(1.4) | 17.0(-2.5) |
| (3.3),$\Delta = 1$ | 35.2(-6.6) | 3.8(-1.2) | 10.0(3.1) | 11.4(1.4) | 17.0(-2.5) |
| (3.3),$\Delta = 0.5$ | 35.2(-6.6) | 3.8(-1.2) | 10.0(3.1) | 11.4(1.4) | 17.0(-2.5) |

$\sigma$ fixed

| Dataset | banana | image | splice | waveform | tree |
|---|---|---|---|---|---|
| $D^2(e^T\alpha) + \sum \xi_i$ | 38.4(-2.9) | 12.9(-2.9) | 19.5(-2.4) | 14.1(-2.5) | 26.1(-10.0) |
| $R^2(e^T\alpha) + \sum \xi_i$ | 22.8(-1.2) | 8.0(-1.3) | 11.4(-0.7) | 11.4(-1.1) | 14.4(-1.5) |
| $(R^2 + 1/C)e^T\alpha$ | 55.9(-10.0) | 7.2(-1.0) | 9.7(10.0) | 11.3(-0.6) | 26.1(-10.0) |
| (3.3),$\Delta = 1$ | 20.2(-0.8) | 4.9(-0.4) | 9.7(10.0) | 11.3(-0.5) | 26.1(-10.0) |
| (3.3),$\Delta = 0.5$ | 36.2(-2.3) | 9.0(-1.4) | 10.0(10.0) | 11.6(-1.2) | 26.1(-10.0) |

Table 3.1: Comparison of bounds for L1-SVM (error rate and and obtained $\ln \sigma^2$ (or $\ln C$)).

## 4. EFFICIENT IMPLEMENTATION

### 4.1. Differentiability of Bounds for L1-SVM

We have proposed to use (3.3) as the bound for L1-SVM. First, we denote it as $f(C, \sigma^2)$ and calculate its partial derivatives. For experiments in this section, we consider only $\Delta = 1$.

The differentiability of both $R^2$ and $\|w\|^2 + 2C\sum_{i=1}^{l} \xi_i$ relies on results of perturbation analysis of optimization problems. We consider Theorem 4.1 of [1] which states that for any optimization problem whose constraints are not related to parameters, if it has a unique minimizer, then the optimal value function is differentiable at parameters. Here, by the optimal value function we mean the optimal objective value as a function of parameters. However, though $1/2\|w\|^2 + C\sum_{i=1}^{l} \xi_i$ is the objective function of the primal L1-SVM, the theorem does not apply as the optimal $\xi$ may not be unique, and constraints involve with the kernel parameter $\sigma$. Therefore, we look at the dual problem as at any optimal solution(1.6) holds. For the RBF kernel, if no training data are at the same point (i.e., $x_i \neq x_j$), $Q$ is positive definite. Then, (1.5) is a strictly convex quadratic programming problem, so the optimal $\alpha$ is unique under any given parameters. Without loss of generality from now on we assume $x_i \neq x_j$. Then a remaining difficulty is that constraints of the dual form of L1-SVM are related to $C$. Therefore, we transform the dual to a problem whose constraints are independent of parameters: From (1.5), let $\alpha = C\bar{\alpha}$:

$$\begin{aligned} \min_{\bar{\alpha}} \quad & \frac{1}{2}\bar{\alpha}^T Q \bar{\alpha} - \frac{e^T\bar{\alpha}}{C} \\ \text{s.t.} \quad & y^T\bar{\alpha} = 0, 0 \leq \bar{\alpha} \leq 1, i = 1, \ldots, l. \end{aligned} \tag{4.1}$$

Then,

$$\frac{1}{2}\alpha^T Q\alpha - e^T\alpha = C^2(\frac{1}{2}\bar{\alpha}Q\bar{\alpha} - \frac{e^T\bar{\alpha}}{C}). \tag{4.2}$$

Finally, we can apply Theorem 4.1 of [1] on (4.1) so

$$\frac{\partial(\frac{1}{2}\alpha^T Q\alpha - e^T\alpha)}{\partial C} = \frac{1}{C}(\alpha^T Q\alpha - e^T\alpha).$$

Because the objective value of primal problem is the same as dual,

$$\frac{\partial(\|w\|^2 + 2C\sum_i \xi_i)}{\partial C} = \frac{2}{C}(e^T\alpha - \alpha^T Q\alpha) = 2\sum \xi_i. \tag{4.3}$$

Similarly,

$$\frac{\partial(\|w\|^2 + 2C\sum \xi_i)}{\partial(\sigma^2)} = -\sum \alpha_i\alpha_j y_i y_j \frac{\partial K(x_i, x_j)}{\partial(\sigma^2)}, \tag{4.4}$$

$$\frac{\partial R^2 + \Delta/C}{\partial C} = \frac{-\Delta}{C^2}, \frac{\partial R^2 + \Delta/C}{\partial(\sigma^2)} = -\sum \beta_i\beta_j \frac{\partial K(x_i, x_j)}{\partial(\sigma^2)}, \tag{4.5}$$

where

$$\frac{\partial K(x_i, x_j)}{\partial(\sigma^2)} = K(x_i, x_j)\frac{\|x_i - x_j\|^2}{2\sigma^4}.$$

In [4], we show that the second derivatives of bounds considered in this paper may not exist. Thus, Newton's method cannot be directly used. So quasi-Newton and nonlinear conjugate gradient methods are the remaining major candidates. Following [7], we decide to work on quasi-Newton methods.

### 4.2. Quasi-Newton Methods

Following earlier experiments, we search on the $(\ln C, \ln \sigma^2)$ space. In order to work on the $(\ln C, \ln \sigma^2)$ space, we must modify (4.3)-(4.5) by chain rules. An advantage is that the optimization problem becomes unconstrained. Otherwise, $C \geq 0$ and the property $\sigma^2 = (-\sigma)^2$ both cause problems. However, practically we still have to specify upper and lower bounds for $\ln C$ and $\ln \sigma^2$. Here, we restrict them to be in $[-10, 10]$. Once the optimization procedure has reached the boundary and still tends to move out, we stop it.

We consider the BFGS quasi-Newton method which has the following procedure: Assume $x^k$ is the current iterate and $f(x)$ is the function to be minimized:

1. Compute a search direction $p = -H_k\nabla f(x^k)$.
2. Find $x^{k+1} = x^k + \lambda p$ using a line search to ensure sufficient decrease.
3. Obtain $H_{k+1}$ by $H_{k+1} = (I - \frac{sy^T}{y^Ts})H_k(I - \frac{ys^T}{y^Ts}) + \frac{ss^T}{y^Ts}$, where $s = x^{k+1} - x^k$ and $y = \nabla f(x^{k+1}) - \nabla f(x^k)$.

Here, $H_k$ serves as the inverse of an approximate Hessian. The sufficient decrease by the line search usually means

$$f(x^k + \lambda p) \leq f(x^k) + \sigma_1\lambda\nabla f(x^k)^T p, \tag{4.6}$$

where $0 < \sigma_1 < 1$ is a positive constant. Since $p = -H_k\nabla f(x^k)$, we need $H_k$ to be positive definite to ensure that $p$ is a descent direction. A good property of the BFGS formula is that $H_{k+1}$ inherits the positive definiteness of $H_k$ as long as $y^Ts > 0$. The condition $y^Ts > 0$ is guaranteed to hold if the initial Hessian is positive definite (e.g. the identity) and the step size is determined by satisfying the second Wolfe condition:

$$\nabla f(x^k + \lambda p) \geq \sigma_2\nabla f(x^k)^T p, \tag{4.7}$$

where $0 < \sigma_1 < \sigma_2 < 1$. Note that (4.6) is usually called the first Wolfe condition.

The main disadvantage of considering (4.7) is that the line search becomes more complicated. In addition, (4.7) involves the calculation of $\nabla f(x^k + \lambda p)$ so for each trial step size, a gradient evaluation is needed. Though $\nabla f(x^k + \lambda p)$ is easily computed once $f(x^k + \lambda p)$ is computed (as pointed out in [7]), this still contributes some additional cost.

We consider an alternative approach to avoid the more complicated line search. If $y^Ts < 0$, $H_k$ is not updated. More specifically, $H_{k+1}$ is determined by

$$H_{k+1} = \begin{cases} (I - \frac{sy^T}{y^Ts})H_k(I - \frac{ys^T}{y^Ts}) + \frac{ss^T}{y^Ts} & \text{if } y^Ts > \eta, \\ H_k & \text{otherwise}, \end{cases}$$

where $\eta$ is usually a small constant. Here, we simply use $\eta = 0$. Then the second Wolfe condition is not needed.

Regarding different trials of step size to ensure the sufficient decrease condition (4.6), we can simply find the largest value in a set $\{\gamma^i | i = 0, 1, ...\}$ such that (4.6) holds ($\gamma = 1/2$ used in this paper). Also note that in early iterations, the search direction $p$ may be a long vector so, using the initial $\lambda = 1$, sometimes $x^k + p$ is far beyond the region considered. Thus, numerical instability may occur. Therefore, if $x^k + \lambda p$ is outside the $[-10, 10] \times [-10, 10]$ region, we project it back by

$$P(x_i^k + \lambda p_i) = \max(-10, \min(x_i^k + \lambda p_i, 10)).$$

We further avoid a too large step size by requiring the initial $\lambda$ to satisfy $||P(x^k + \lambda p) - x^k|| \leq 2$. This reduces the chance of going to a wrong region in the beginning.

### 4.3. Experiments

We use LIBSVM [2], which implements a decomposition method, to calculate $||w||^2$, $||\tilde{w}||^2$, $R^2$, and $\tilde{R}^2$. The computational experiments for this section were done on a Pentium III-1000 with 1024MB RAM. We keep all the default settings of LIBSVM except using a smaller stopping tolerance $10^{-6}$ (default $10^{-3}$) and increasing the cache size. It was pointed out in [7] that near the minimizer, $||\nabla f||$ is small so the error associated with finding $||w||^2, ||\tilde{w}||^2, R^2$ or $\tilde{R}^2$ may strongly affect the search direction. We have the same observation so decide to use a smaller stopping tolerance.

We compare the quasi-Newton implementation for L1- and L2-SVM. For L1-SVM, we use (3.3) with $\Delta = 1$. To demonstrate the viability for solving large problems, we include the problem ijcnn1 which has 49,990 training and 91,701 testing samples in two classes. It is from the first problem of IJCNN challenge 2001. Note that we use the winner's transformation of raw data.

Tables 4.1 presents the result using an initial point $(0, 0)$. We list the number of function and gradient evaluations, and the test accuracy. Results for L2-SVM are consistent with Table 1 of [7]. Note that the number of gradient evaluations is the same as the number of quasi-Newton iterations. Hence the average number of line searches in each iteration is very close to one. For problems image, splice, and tree, using L1-SVM, the algorithm reaches a point with $\ln C = 10$. At that point there are no bounded support vectors so we can set $\ln C$ to be the largest element of the dual variable $\alpha$ without affecting the model produced. Actually in final iterations, there are already no bounded support vectors. We have not been able to develop early stopping criteria for such a situation so decided to let the algorithm continue until it reaches the boundary. Thus, for image and waveform, L1-SVM takes more iterations than L2-SVM. On the other hand, for splice, L2-SVM also reaches $\ln C = 10$. Overall except this difference, in terms of accuracy as well as computational cost, the bound for L1-SVM is competitive with that for L2-SVM.

For the large problem ijcnn1, RM bounds for both L1- and L2-SVM reach points with error rate around 2.91% and 2.17%, respectively. Unfortunately, this is a little worse than 1.41% by $C = 16$ and $\sigma^2 = 0.125$ using cross validation. In addition, the number of support vectors is very different. There are about 17,000 support vectors comparing to 3370 when utilize cross validation. However, the total computational time is around 26,000 seconds (using the default $10^{-3}$ as the stopping tolerance of LIBSVM), much shorter than doing cross validation.

To further compare bounds for L1- and L2-SVM, in Figure 2 and 3, we present contour plots of tree, and waveform, with searching paths on them. Note that in the L1 case of Figure 3, the final solution is projected from $(8.87, 1.38)$ to $(1.25, 1.38)$ because there is already no bounded support vectors.

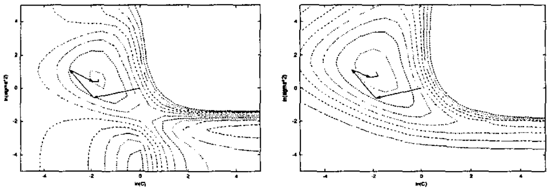

Figure 2: Contour plots of tree (left: L1 bound, right: L2 bound).

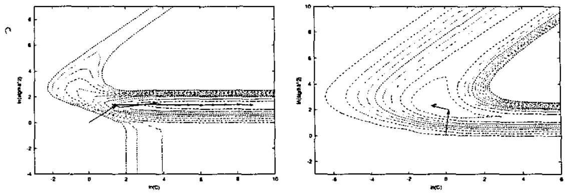

Figure 3: Contour plots of waveform (left: L1 bound, right: L2 bound). The final solution for the L1 bound is projected from $(8.87, 1.38)$ to $(1.25, 1.38)$.

## 5. REFERENCES

[1] J. F. Bonnans and A. Shapiro. Optimization problems with perturbations: a guided tour. *SIAM Review*, 40(2):228–264, 1998.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

[4] K.-M. Chung, W.-C. Kao, T. Sun, L.-L. Wang, and C.-J. Lin. Radius margin bounds for support vector machines with the rbf kernel. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2002.

[5] K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 2002. To appear.

[6] T.-T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machines. In *Proceedings of 15th Intl. Conf. Machine Learning*. Morgan Kaufman Publishers, 1998.

[7] S. S. Keerthi. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 2002. To appear.

[8] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[9] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, 2000.

| | L1 | | | L2 | | |
|---|---|---|---|---|---|---|
| | #fun | #grad | accuracy | #fun | #grad | accuracy |
| banana | 9 | 6 | 88.96 | 8 | 5 | 88.53 |
| image | 17 | 13 | 96.24 | 11 | 6 | 97.03 |
| splice | 13 | 12 | 89.84 | 21 | 19 | 89.84 |
| tree | 8 | 8 | 86.50 | 8 | 8 | 86.54 |
| waveform | 16 | 13 | 88.57 | 8 | 7 | 89.83 |
| ijcnn1 | 9 | 9 | 97.09 | 7 | 7 | 97.83 |

Table 4.1: RM bounds with $x^0 = (0, 0)$.