# THE DOUBLY REGULARIZED SUPPORT VECTOR MACHINE

Li Wang, Ji Zhu and Hui Zou

*University of Michigan and University of Minnesota*

*Abstract:* The standard $L_2$-norm support vector machine (SVM) is a widely used tool for classification problems. The $L_1$-norm SVM is a variant of the standard $L_2$-norm SVM, that constrains the $L_1$-norm of the fitted coefficients. Due to the nature of the $L_1$-norm, the $L_1$-norm SVM has the property of automatically selecting variables, not shared by the standard $L_2$-norm SVM. It has been argued that the $L_1$-norm SVM may have some advantage over the $L_2$-norm SVM, especially with high dimensional problems and when there are redundant noise variables. On the other hand, the $L_1$-norm SVM has two drawbacks: (1) when there are several highly correlated variables, the $L_1$-norm SVM tends to pick only a few of them, and remove the rest; (2) the number of selected variables is upper bounded by the size of the training data. A typical example where these occur is in gene microarray analysis. In this paper, we propose a doubly regularized support vector machine (DrSVM). The DrSVM uses the *elastic-net* penalty, a mixture of the $L_2$-norm and the $L_1$-norm penalties. By doing so, the DrSVM performs automatic variable selection in a way similar to the $L_1$-norm SVM. In addition, the DrSVM encourages highly correlated variables to be selected (or removed) together. We illustrate how the DrSVM can be particularly useful when the number of variables is much larger than the size of the training data $(p \gg n)$. We also develop efficient algorithms to compute the whole solution paths of the DrSVM.

*Key words and phrases:* Grouping effect, $p \gg n$, quadratic programming, SVM, variable selection.

## 1. Introduction

The support vector machine (SVM) is a widely used tool for classification (Vapnik (1995)). It was first motivated by the geometric consideration of maximizing the *margin* (Boser, Guyon and Vapnik (1992) and Cortes and Vapnik (1995)). If given a set of training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, where the input $\boldsymbol{x}_i \in \mathbb{R}^p$ is a vector with $p$ predictor variables, and the output $y_i \in \{1, -1\}$ denotes the class label, the SVM finds a hyperplane that separates the two classes of data points by the largest distance:

$$\max_{\beta_0, \boldsymbol{\beta}} \frac{1}{\|\boldsymbol{\beta}\|_2^2}, \quad \text{subject to } y_i(\beta_0 + \boldsymbol{x}_i^\intercal \boldsymbol{\beta}) \geq 1 - \xi_i, \xi_i \geq 0, \sum_{i=1}^n \xi_i \leq B,$$

where $\xi_i$ are slack variables that describe the overlap between the two classes, and $B$ is a tuning parameter that controls the overlap.

0.05
0.06
0.07
0.08
0.10
0.15
-1.5
-1.0
-0.5
-0.2
-0.1

Many researchers have noted the relationship between the SVM and regularized function estimation, i.e., the above optimization is equivalent to

0.1
0.2
0.3
0.4

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} [1 - y_i(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta})]_+ + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2, \tag{1}$$

0.6
0.7
0.8

where $\lambda$ is the tuning parameter, playing the same role as $B$. The classification rule for a new input $\boldsymbol{x}$ is then given by $\text{sign}(\beta_0 + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{\beta})$. Notice that (1) has the form *loss+penalty*; hence $\lambda$ controls the balance between the loss and the penalty. The function $(1 - \cdot)_+$ is called the *hinge loss*, and is plotted in Figure 1. Notice that it has a non-differentiable point at 1. The penalty is the $L_2$-norm of the coefficient vector, the same as that used in ridge regression (Hoerl and Kennard (1970)). The idea of penalizing by the sum of squares of the parameters is also used in neural networks, where it is known as *weight decay*. The ridge penalty shrinks the fitted coefficients toward zero. It is well known that this shrinkage has the effect of controlling the variance of fitted coefficients, hence possibly improving the fitted model's prediction accuracy via the bias-variance trade-off, especially when there are many highly correlated variables. An overview of the SVM as regularized function estimation can be found in Burges (1998), Evgeniou, Pontil and Poggio (1999), Wahba (1999) and Hastie, Tibshirani and Friedman (2001).

2.5

5
10
15
20
30
40
50
60
1/
$\lambda_1$
$\lambda_2$
-Norm SVM
DrSVM (fixed
)

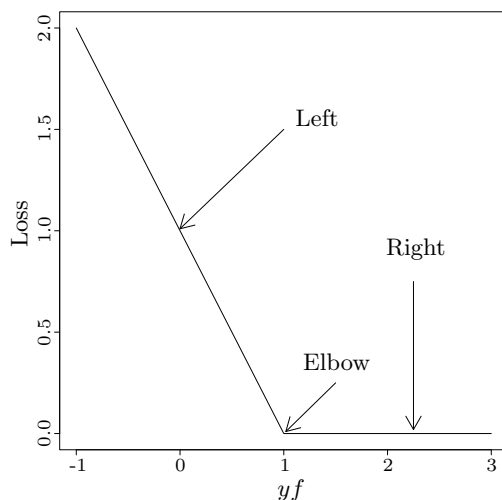$\beta$
$\beta_1$
$\beta_2$
Mixture
$L_1$
$L_2$
1.9



Figure 1. The hinge loss of the SVM. *Elbow* indicates the point $1 - yf = 0$, *Left* indicates the region to the left of the elbow, and *Right* indicates the region to the right of the elbow.

## 1.1. The $L_1$-norm support vector machine

Instead of using the $L_2$-norm penalty, several researchers have considered replacing it in (1) with the $L_1$-norm penalty, and fitting an $L_1$-norm SVM

model (Bradley and Mangasarian (1998), Song, Breneman, Bi, Sukumar, Bennett, Cramer and Tugcu (2002) and Zhu, Rosset, Hastie and Tibshirani (2004)):

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} [1 - y_i(\beta_0 + \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{\beta})]_+ + \lambda \|\boldsymbol{\beta}\|_1.$$

The $L_1$-norm penalty was first used for signal processing and regression problems by Mallat and Zhang (1993), Tibshirani (1996) and Chen, Donoho and Saunders (1998). Similar to the $L_2$-norm penalty, the $L_1$-norm penalty also shrinks the fitted coefficients toward zero, which also benefits from the reduction in the fitted coefficients' variance. Another important property of the $L_1$-norm penalty is that because of its $L_1$ nature, making $\lambda$ sufficiently large will cause some of the fitted coefficients be *exactly* zero. Thus, as $\lambda$ varies, the $L_1$-norm penalty performs a kind of continuous variable selection, while this is not the case for the $L_2$-norm penalty. It is interesting to note that the $L_2$-norm penalty corresponds to a Gaussian prior for the $\beta_j$'s, while the $L_1$-norm penalty corresponds to a double-exponential prior. The double-exponential density has heavier tails than the Gaussian density. This reflects the greater tendency of the $L_1$-norm penalty to produce some large fitted coefficients and leave others at 0, especially in high dimensional problems. See Figure 2 for contours of the $L_2$-norm penalty and the $L_1$-norm penalty.

## 1.2. The doubly regularized support vector machine

It has been argued that the $L_1$-norm penalty has advantages over the $L_2$-norm penalty under certain scenarios (Donoho, Johnstone, Kerkyachairan and Picard (1995), Friedman, Hastie, Rosset, Tibshirani and Zhu (2004) and Ng (2004)), such as when there are redundant noise variables. However, the $L_1$-norm penalty also suffers from two serious limitations (Zou and Hastie (2005)).

1. When there are several highly correlated input variables in the data set, and they are all relevant to the output variable, the $L_1$-norm penalty tends to pick only one or few of them and shrinks the rest to 0. For example, in microarray analysis, expression levels for genes that share one biological pathway are usually highly correlated, and these genes all contribute to the biological process, but the $L_1$-norm penalty usually selects only one gene from the group and does not care which one is selected. The ideal method should be able to eliminate trivial genes, and automatically include the whole group of relevant genes.

2. In the $p > n$ case, as shown in Rosset, Zhu and Hastie (2004), the $L_1$-norm penalty can keep at most $n$ input variables. Again, we use microarray as an example: the sample size $n$ is usually on the order of 10 or 100, while the

LI WANG, JI ZHU AND HUI ZOU

dimension of the input $p$ is typically on the order of $1,000$ or even $10,000$. Using the $L_1$-norm penalty can, at most, identify $n$ non-zero fitted coefficients, but it is unlikely that only 10 genes are involved in a complicated biological process.

Zou and Hastie (2005) proposed the *elastic-net* penalty to fix these two limitations. The elastic-net penalty is a mixture of the $L_1$-norm penalty and the $L_2$-norm penalty, combining good features of the two. Similar to the $L_1$-norm penalty, the elastic-net penalty simultaneously performs automatic variable selection and continuous shrinkage; the new advantages are that groups of correlated variables now can be selected together, and the number of selected variables is no longer limited by $n$.

In this paper, we apply the elastic-net penalty to the support vector machine. Specifically, we consider the following doubly regularized support vector machine, which we call the DrSVM:

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} [1 - y_i(\beta_0 + \boldsymbol{x}_i^{\top}\boldsymbol{\beta})]_+ + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1, \tag{2}$$

where both $\lambda_1$ and $\lambda_2$ are tuning parameters. The role of the $L_1$-norm penalty is to allow variable selection, and the role of the $L_2$-norm penalty is to help groups of correlated variables get selected together. We show that for classification problems, the $L_2$-norm penalty tends to make highly correlated input variables have similar fitted coefficients, which is the *grouping effect*. We also see that the number of selected input variables is not limited by $n$ anymore. Figure 2 compares contours of the $L_2$-norm, the $L_1$-norm, and the elastic-net penalty.
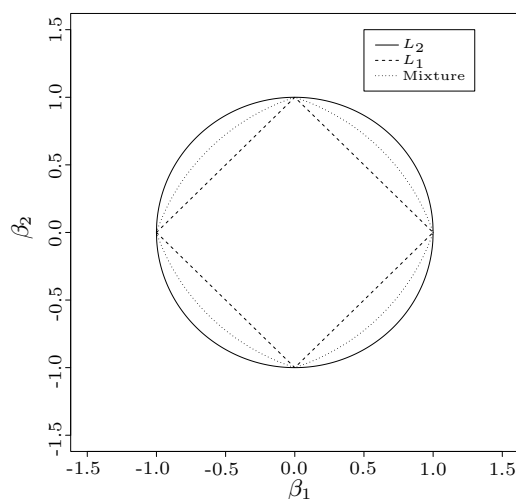


Figure 2. 2-dimensional contour plots. The $L_2$-norm $\|\boldsymbol{\beta}\|_2^2 = 1$, the $L_1$-norm $\|\boldsymbol{\beta}\|_1 = 1$, and the elastic-net $0.5\|\boldsymbol{\beta}\|_2^2 + 0.5\|\boldsymbol{\beta}\|_1 = 1$.

To get a good classification rule that performs well on future data, it is important to select appropriate tuning parameters $\lambda_1$ and $\lambda_2$. In practice, people can pre-specify a finite grid of values for $\lambda_1$ and $\lambda_2$ that covers a wide range, then use either a separate validation data set or cross-validation to do a grid search to find values for the $(\lambda_1, \lambda_2)$ pair that give the best performance across the given grid. In this paper, we illustrate that the solution path for a fixed value of $\lambda_2$, denoted as $\boldsymbol{\beta}_{\lambda_2}(\lambda_1)$, is *piece-wise linear* as a function of $\lambda_1$ (in the $\mathbb{R}^p$ space); and for a fixed value of $\lambda_1$, the solution path, denoted as $\boldsymbol{\beta}_{\lambda_1}(\lambda_2)$, is piece-wise linear as a function of $1/\lambda_2$. We further propose efficient algorithms to compute the exact solution paths. This helps us understand how the solution changes with $\lambda_1$ and $\lambda_2$, and facilitates the adaptive selection of the tuning parameters.

Before delving into the technical details, we illustrate the concept of grouping effect and piece-wise linearity of the solution paths $\boldsymbol{\beta}_{\lambda_2}(\lambda_1)$ and $\boldsymbol{\beta}_{\lambda_1}(\lambda_2)$ with a simple example. We generate 30 training data in each of two classes. Each input $\boldsymbol{x}_i$ is a $p = 30$ dimensional vector. For the "+" class, $\boldsymbol{x}_i$ has a normal distribution with mean and covariance matrix

$$\boldsymbol{\mu}_+ = (\underbrace{1, \ldots, 1}_{5}, \underbrace{0, \ldots, 0}_{25})^{\mathsf{T}},$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^*_{5\times5} & \mathbf{0}_{5\times25} \\ \mathbf{0}_{25\times5} & \boldsymbol{I}_{25\times25} \end{pmatrix},$$

where the diagonal elements of $\boldsymbol{\Sigma}^*$ are 1 and the off-diagonal elements are all equal to $\rho = 0.8$. The "$-$" class has a similar distribution, except that

$$\boldsymbol{\mu}_- = (\underbrace{-1, \ldots, -1}_{5}, \underbrace{0, \ldots, 0}_{25})^{\mathsf{T}}.$$

So $x_1, \ldots, x_5$ are highly correlated, the Bayes optimal classification boundary is given by $x_1 + \cdots + x_5 = 0$ and the Bayes error is 0.138. Figure 3 compares the result from the standard $L_2$-norm SVM, the $L_1$-norm SVM, and the DrSVM. The solid paths are for $x_1, \ldots, x_5$, which are the relevant variables; the dashed paths are for $x_6, \ldots, x_{30}$, which are the irrelevant variables. As we can see, the $L_2$-norm SVM kept all variables in the fitted model, the $L_1$-norm SVM did variable selection, but failed to identify the group of correlated variables; the DrSVM successfully selected all five relevant variables, and shrunk their coefficients close to each other.

In Section 2, we show the grouping effect of the DrSVM. In Section 3, we describe algorithms that compute the whole solution paths of the DrSVM. In Section 4, we present numerical results on both simulation and real-world data. We conclude the paper with a discussion section.
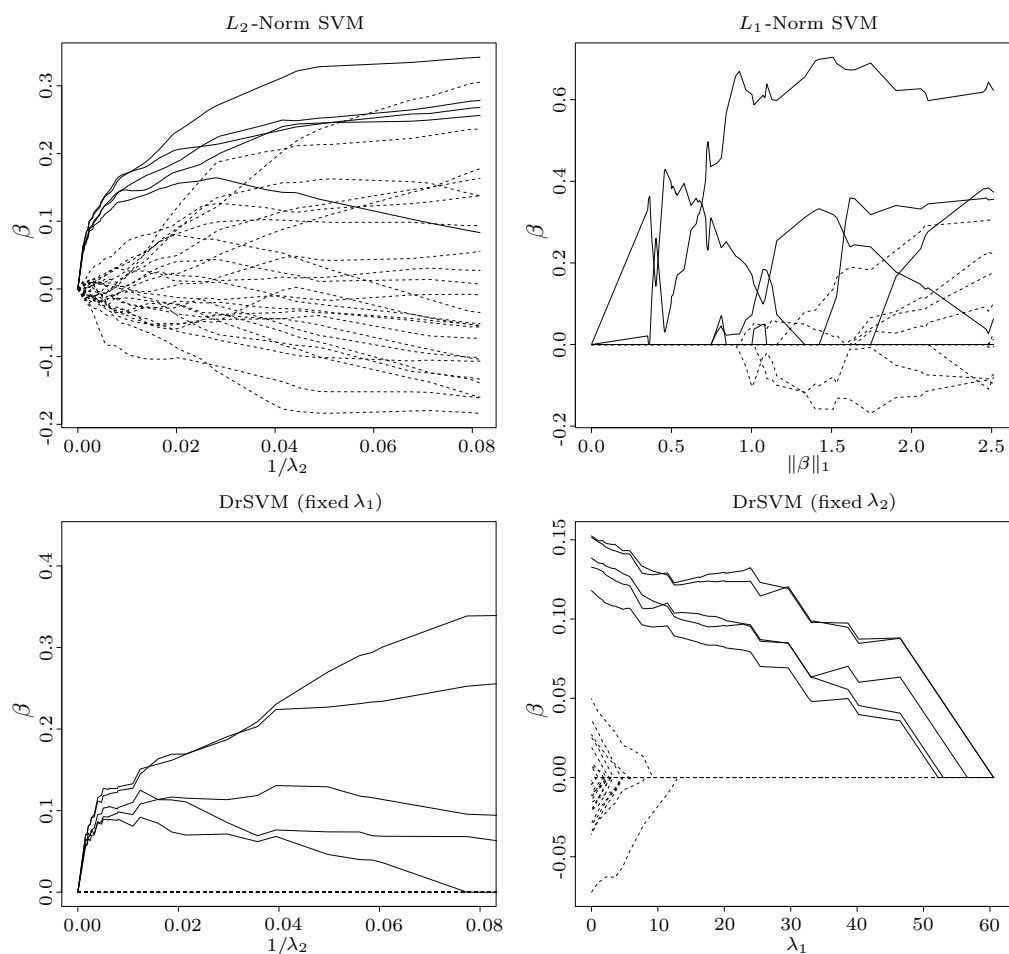
Figure 3. Comparison of different SVMs on a simple simulation data. The solid curves correspond to relevant variables, and the dashed curves correspond to irrelevant variables. The relevant variables are highly correlated. The upper left panel is for the $L_2$-norm SVM, the upper right panel is for the $L_1$-norm SVM, the bottom panels are for the DrSVM. The bottom left panel fixes $\lambda_1 = 15$, and changes $\lambda_2$; the bottom right panel fixes $\lambda_2 = 160$, and changes $\lambda_1$. We can see the DrSVM identifies all (correlated) relevant variables, and shrinks their coefficients close to each other.

## 2. Grouping Effect of the DrSVM

In this section, we illustrate how the DrSVM has the grouping effect for correlated variables. The result holds not only for the hinge loss function of the SVM, but also for general Lipschitz continuous loss functions.

Consider the following more general optimization problem:

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} \phi(y_i, f(\boldsymbol{x}_i)) + \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1, \tag{3}$$

where $f(\boldsymbol{x}) = \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}$, $\phi(y, f) = \phi(yf)$ is a function of the *margin*. We further assume that $\phi(t)$ is Lipschitz continuous, i.e., $|\phi(t_1) - \phi(t_2)| \leq M|t_1 - t_2|$ for some positive finite $M$. It is simple to verify that this condition holds for many commonly used loss functions for classification, for example, the hinge loss function (SVM) and the binomial deviance (logistic regression). Then we have the following theorem.

**Theorem 1.** *Denote the solution to (3) as $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}$. If the loss function $\phi$ is Lipschitz continuous then, for any pair $(j, l)$, we have*

$$\left| \hat{\beta}_j - \hat{\beta}_l \right| \leq \frac{M}{\lambda_2} \|\boldsymbol{x}_j - \boldsymbol{x}_l\|_1 = \frac{M}{\lambda_2} \sum_{i=1}^{n} |x_{ij} - x_{il}|. \tag{4}$$

*Furthermore, if the input variable $\boldsymbol{x}_j, \boldsymbol{x}_l$ are centered and normalized, then*

$$\left| \hat{\beta}_j - \hat{\beta}_l \right| \leq \frac{\sqrt{n}M}{\lambda_2} \sqrt{2(1 - \rho)}, \tag{5}$$

*where $\rho = cor(\boldsymbol{x}_j, \boldsymbol{x}_l)$ is the sample correlation between $\boldsymbol{x}_j$ and $\boldsymbol{x}_l$.*

**Proof.** Consider another set of coefficients

$$\hat{\beta}_0^* = \hat{\beta}_0, \quad \hat{\beta}_{j'}^* = \begin{cases} \frac{1}{2}(\hat{\beta}_j + \hat{\beta}_l), & \text{if } j' = j \text{ or } j' = l, \\ \hat{\beta}_{j'}, & \text{otherwise.} \end{cases}$$

By the definition of $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}$, we have

$$\sum_{i=1}^{n} \phi(y_i, \hat{\beta}_0^* + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^*) + \frac{\lambda_2}{2} \|\hat{\boldsymbol{\beta}}^*\|_2^2 + \lambda_1 \|\hat{\boldsymbol{\beta}}^*\|_1$$

$$- \sum_{i=1}^{n} \phi(y_i, \hat{\beta}_0 + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}) - \frac{\lambda_2}{2} \|\hat{\boldsymbol{\beta}}\|_2^2 - \lambda_1 \|\hat{\boldsymbol{\beta}}\|_1 \geq 0, \tag{6}$$

where

$$\sum_{i=1}^{n} \left[ \phi(y_i, \hat{\beta}_0^* + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^*) - \phi(y_i, \hat{\beta}_0 + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}) \right]$$

$$\leq \sum_{i=1}^{n} \left| \phi(y_i, \hat{\beta}_0^* + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^*) - \phi(y_i, \hat{\beta}_0 + \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}) \right|$$

$$\leq \sum_{i=1}^{n} M \left| y_i(\hat{\beta}_0^* + \boldsymbol{x}_i^\intercal \hat{\boldsymbol{\beta}}^*) - y_i(\hat{\beta}_0 + \boldsymbol{x}_i^\intercal \hat{\boldsymbol{\beta}}) \right| \tag{7}$$

$$= \sum_{i=1}^{n} M \left| \boldsymbol{x}_i^\intercal (\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}}) \right|$$

$$= \sum_{i=1}^{n} M \left| \frac{1}{2}(x_{ij} - x_{il})(\hat{\beta}_j - \hat{\beta}_l) \right|$$

$$= \frac{M}{2} \left| \hat{\beta}_j - \hat{\beta}_l \right| \sum_{i=1}^{n} |x_{ij} - x_{il}|$$

$$= \frac{M}{2} \left| \hat{\beta}_j - \hat{\beta}_l \right| \cdot \|\boldsymbol{x}_j - \boldsymbol{x}_l\|_1. \tag{8}$$

We also have

$$\|\hat{\boldsymbol{\beta}}^*\|_1 - \|\hat{\boldsymbol{\beta}}\|_1 = |\hat{\beta}_j^*| + |\hat{\beta}_l^*| - |\hat{\beta}_j| - |\hat{\beta}_l| = |\hat{\beta}_j + \hat{\beta}_l| - |\hat{\beta}_j| - |\hat{\beta}_l| \leq 0, \tag{9}$$

$$\|\hat{\boldsymbol{\beta}}^*\|_2^2 - \|\hat{\boldsymbol{\beta}}\|_2^2 = |\hat{\beta}_j^*|^2 + |\hat{\beta}_l^*|^2 - |\hat{\beta}_j|^2 - |\hat{\beta}_l|^2 = -\frac{1}{2}|\hat{\beta}_j - \hat{\beta}_l|^2. \tag{10}$$

Now combining (8), (9) and (10), (6) implies that

$$\frac{M}{2} \left| \hat{\beta}_j - \hat{\beta}_l \right| \cdot \|\boldsymbol{x}_j - \boldsymbol{x}_l\|_1 - \frac{\lambda_2}{2}|\hat{\beta}_j - \hat{\beta}_l|^2 \geq 0. \tag{11}$$

Hence, (4) is obtained.

For (5), we simply use the inequality

$$\|\boldsymbol{x}_j - \boldsymbol{x}_l\|_1 \leq \sqrt{n}\sqrt{\|\boldsymbol{x}_j - \boldsymbol{x}_l\|_2^2} = \sqrt{n}\sqrt{2(1 - \rho)}. \tag{12}$$

We used Lipschitz continuity in (7), where it was applied to loss functions for classification, i.e., functions of the margin. For the hinge loss, it is easy to see that the Lipschitz constant $M = 1$, hence Theorem 1 holds for the DrSVM. It is also worth noting that the theorem holds for all $\lambda_1 \geq 0$, so the grouping effect is from the $L_2$-norm penalty.

## 3. The DrSVM Algorithms

In this section, we propose efficient algorithms that can solve the whole solution path $\boldsymbol{\beta}_{\lambda_2}(\lambda_1)$ (when $\lambda_2$ is fixed) and $\boldsymbol{\beta}_{\lambda_1}(\lambda_2)$ (when $\lambda_1$ is fixed). Our algorithms hinge on the following two results.

**Theorem 2.** *When $\lambda_2$ is fixed, the solution $\boldsymbol{\beta}_{\lambda_2}(\lambda_1)$ is a piecewise linear function of $\lambda_1$.*

**Theorem 3.** *When $\lambda_1$ is fixed, the solution $\boldsymbol{\beta}_{\lambda_1}(\lambda_2)$ is a piecewise linear function of $1/\lambda_2$.*

Figure 4 illustrates the point. Any segment between two adjacent vertical lines is linear. When $\lambda_2$ is fixed, the basic idea of our algorithm is to start with $\lambda_1$ equal to $\infty$, find the direction of the linear path, move the solution in that direction until it hits a *joint* (the asterisk points in Figure 4, then adjust the direction of the path, and move on. The algorithm when $\lambda_1$ is fixed operates in a similar manner (the right panel of Figure 4).
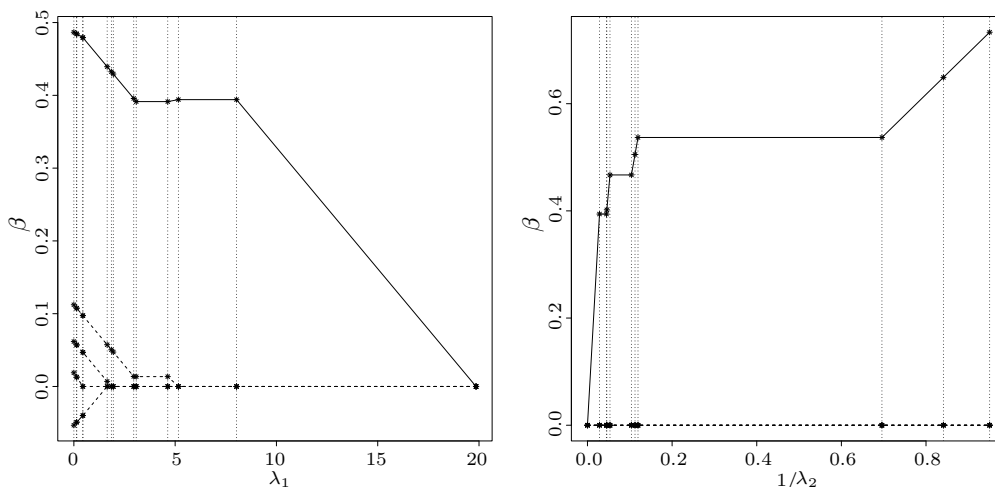


Figure 4. The simulation setup is the same as in Section 1, except the size of the training data is $n = 8 + 8$, the number of input variables is $p = 5$, and only the first variable $x_1$ is relevant to the optimal classification boundary. The solid line corresponds to $\hat{\beta}_1$, the dashed lines correspond to $\hat{\beta}_2, \ldots, \hat{\beta}_5$. The left panel is for $\hat{\boldsymbol{\beta}}_{\lambda_2}(\lambda_1)$ (with $\lambda_2 = 30$), and the right panel is for $\hat{\boldsymbol{\beta}}_{\lambda_1}(\lambda_2)$ (with $\lambda_1 = 6$).

### 3.1. Proof of Theorem 2

The optimization problem (2) for the DrSVM is equivalent to the quadratic programming problem:

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} \epsilon_i + \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2 \qquad (13)$$

$$\text{subject to } 1 - y_i f_i \leq \epsilon_i, \qquad (14)$$

$$\epsilon_i \geq 0, \quad i = 1, \ldots, n, \qquad (15)$$

$$\|\boldsymbol{\beta}\|_1 = |\beta_1| + \cdots + |\beta_p| \leq s, \qquad (16)$$

where $f_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$. Notice the hinge loss is replaced by a linear constraint, the $L_1$-norm penalty is replaced by an $L_1$-norm constraint, and the tuning

parameter $\lambda_1$ is replaced by $s$. The optimization problem (2) and the quadratic programming problem are equivalent in the sense that, for any value of $\lambda_1$, there exists a value of $s$ such that the solution to (2) and the solution to the quadratic programming problem are identical. To solve the quadratic programming problem, we write:

$$\sum_{i=1}^{n} \epsilon_i + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 + \sum_{i=1}^{n} \alpha_i(1 - y_i f_i - \epsilon_i) - \sum_{i=1}^{n} \gamma_i \epsilon_i + \eta(\sum_{j=1}^{p}|\beta_j| - s),$$

where $\alpha_i \geq 0$, $\gamma_i \geq 0$ and $\eta \geq 0$ are Lagrange multipliers. Taking derivatives with respect to $\beta_0, \boldsymbol{\beta}$ and $\epsilon_i$, we have

- $\sum_{i=1}^{n} \alpha_i y_i = 0,$
- $\lambda_2 \beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} + \eta \text{sign}(\beta_j) = 0$ for $j \in \mathcal{V}$,
- $1 - \alpha_i - \gamma_i = 0, \quad i = 1, \dots, n,$

where $\mathcal{V} = \{j : \beta_j \neq 0\}$. Notice the value of $\beta_j$ is fully determined by the values of $\alpha_i$ and $\eta$. We also have the Karush-Kuhn-Tucker (KKT) conditions from quadratic programming:

- $\alpha_i(1 - y_i f_i - \epsilon_i) = 0, \quad i = 1, \dots, n,$
- $\gamma_i \epsilon_i = 0, \quad i = 1, \dots, n,$
- $\eta(\sum_{i=1}^{p}|\beta_j| - s) = 0.$

We use $\mathcal{L}$ (Left) to denote the set of data points for which $1 - y_i f_i > 0$, $\mathcal{R}$ (Right) for $1 - y_i f_i < 0$, and $\mathcal{E}$ (Elbow) for $1 - y_i f_i = 0$ (See Figure 1). Inspecting the KKT conditions, we find

- $i \in \mathcal{L} \implies \gamma_i = 0, \alpha_i = 1;$
- $i \in \mathcal{R} \implies \gamma_i = 1, \alpha_i = 0;$
- $i \in \mathcal{E} \implies 0 \leq \gamma_i, \alpha_i \leq 1$ and $\gamma_i + \alpha_i = 1.$

So, for data points in $\mathcal{L}$ and $\mathcal{R}$, their $\alpha_i$ are determined. To solve for $\beta_j$, we also need $\alpha_i$ values for data points in $\mathcal{E}$, and especially how these values change (between 0 and 1) when $s$ increases.

When $s$ is small enough, the constraint (16) is active, i.e., $\|\boldsymbol{\beta}\|_1 = s$. When $s$ increases to a certain value, say $s^*$, this constraint will become inactive, and the solution will not change beyond the value of $s^*$. This corresponds to $\lambda_1 = 0$ in (2). Suppose for a value $s < s^*$, the solution is $(\beta_0, \boldsymbol{\beta})$, hence $\mathcal{V}, \mathcal{L}, \mathcal{R}$ and $\mathcal{E}$

are also known. Then $(\beta_0, \boldsymbol{\beta})$ have to satisfy the following equations:

$$\lambda_2 \beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} + \eta \operatorname{sign}(\beta_j) = 0, \quad j \in \mathcal{V}, \tag{17}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \tag{18}$$

$$y_i \Big( \beta_0 + \sum_{j \in \mathcal{V}} \beta_j x_{ij} \Big) = 1, \quad i \in \mathcal{E}, \tag{19}$$

$$\|\boldsymbol{\beta}\|_1 = \sum_{j \in \mathcal{V}} \operatorname{sign}(\beta_j)\beta_j = s. \tag{20}$$

This linear system consists of $|\mathcal{E}| + |\mathcal{V}| + 2$ equations and $|\mathcal{E}| + |\mathcal{V}| + 2$ unknowns: $\alpha_i$'s, $\beta_j$'s, $\beta_0$ and $\eta$. They can be further reduced to $|\mathcal{E}| + 2$ equations in $|\mathcal{E}| + 2$ unknowns by plugging (17) into (19) and (20). If the system is nonsingular, the solution is unique. In the case of singularity, the optimal solution is not unique, but the optimal region can still be determined.

When $s$ increases by a small enough amount, by continuity, the sets $\mathcal{V}, \mathcal{L}, \mathcal{R}$ and $\mathcal{E}$ will not change, and the structure of the above linear system will not change. Taking right derivatives with respect to $s$, we have

$$\lambda_2 \frac{\Delta \beta_j}{\Delta s} - \sum_{i \in \mathcal{E}} \frac{\Delta \alpha_i}{\Delta s} y_i x_{ij} + \operatorname{sign}(\beta_j) \frac{\Delta \eta}{\Delta s} = 0, \quad j \in \mathcal{V}, \tag{21}$$

$$\sum_{i \in \mathcal{E}} \frac{\Delta \alpha_i}{\Delta s} y_i = 0, \tag{22}$$

$$\frac{\Delta \beta_0}{\Delta s} + \sum_{j \in \mathcal{V}} \frac{\Delta \beta_j}{\Delta s} x_{ij} = 0, \quad i \in \mathcal{E}, \tag{23}$$

$$\sum_{j \in \mathcal{V}} \operatorname{sign}(\beta_j) \frac{\Delta \beta_j}{\Delta s} = 1, \tag{24}$$

which does not depend on the value of $s$. This implies that the solution, $\alpha_i$'s, $\beta_j$'s, $\beta_0$ and $\eta$, will change linearly in $s$. When the increase in $s$ is big enough, one of the $\mathcal{V}, \mathcal{L}, \mathcal{R}$ and $\mathcal{E}$ sets will change and the structure of the linear system will change, corresponding to a different linear piece on the solution path. Hence, the solution path is piecewise linear in $s$. Notice that $\eta$ is equivalent to $\lambda_1$; therefore $\beta_0, \boldsymbol{\beta}$ and $\alpha_i$ are also piecewise linear in $\lambda_1$, and Theorem 2 holds.

To identify changes in the structure of the linear system (or the asterisk points in Figure 4), we define four types of *events*, corresponding to the changes in $\mathcal{V}, \mathcal{L}, \mathcal{R}$ and $\mathcal{E}$.

1. A data point leaves $\mathcal{E}$ to $\mathcal{L}$ or $\mathcal{R}$. This happens when an $\alpha_i$ changes from within the region $(0, 1)$ to the boundary 1 or 0.
2. A data point reaches $\mathcal{E}$ from $\mathcal{L}$ or $\mathcal{R}$. This happens when a residual $(1 - y_i f_i)$ reaches 0.
3. An active variable in $\mathcal{V}$ becomes inactive. This happens when a non-zero coefficient $\beta_j \neq 0$ becomes 0.
4. An inactive variable joins the active variable set $\mathcal{V}$. To identify this event, we define the *generalized correlation* for variable $j$ as

$$c_j = \lambda_2 \beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij}. \tag{25}$$

From (17), we can see that all active variables in $\mathcal{V}$ have the same absolute generalized correlation value, which is $\eta$. Therefore, an inactive variable will join the active variable set when its absolute generalized correlation reaches $\eta$.

In the next two sections, we describe the algorithm that computes the whole solution path $\beta_{\lambda_2}(\lambda_1)$ in detail. The basic idea is to start at $s = 0$ (or equivalently $\lambda_1 = \infty$), find the right derivatives of $\beta_0$ and $\beta_j$ with respect to $s$, increase $s$ until an event happens, then adjust the linear system $(21)-(24)$, and find the new right derivatives. The algorithm stops when no further events happen.

### 3.2. Initial solution

In this section, we compute the initial right derivatives of $\beta_0$ and $\beta_j$. Let $n_+$ be the number of training data points in the "+" class, and $n_-$ be the number of training data points in the "−" class. We distinguish between two cases: the balanced case $(n_+ = n_-)$ and the unbalanced case $(n_+ \neq n_-)$.

**The Balanced Case**

When $s = 0$, $\boldsymbol{\beta} = 0$, and the objective function (13) becomes

$$\min_{\beta_0} \sum_{i=1}^{n} (1 - y_i \beta_0)_+.$$

Since $n_+ = n_-$, there is no unique solution for $\beta_0$, and any value of $\beta_0 \in [-1, 1]$ will give the same minimum.

Although $\beta_0$ is not unique, all the $\alpha_i$ are equal to 1. Using (25), the generalized correlation of variable $x_j$ is $-\sum_{i=1}^{n} y_i x_{ij}$. When $s$ increases by an infinitesimal amount, some variable(s) will join the active variable set $\mathcal{V}$, and $\mathcal{V}$ can be identified as

$$\mathcal{V} = \left\{ j : \left| \sum_{i=1}^{n} y_i x_{ij} \right| = \max_j \left| \sum_{i=1}^{n} y_i x_{ij} \right| \right\}.$$

The signs of the corresponding coefficients are given by $\mathrm{sign}(\sum_{i=1}^{n} y_i x_{ij})$. Then, using (21) and (24), one can solve for the right derivatives of $\beta_j, j \in \mathcal{V}$ and $\eta$ with respect to $s$.

When $s$ increases, by continuity and the balance between $n_+$ and $n_-$, all $\alpha_i$ will stay at 1 before an event happens. Therefore $1 - (\beta_0 + \sum_{j \in \mathcal{V}} \beta_j x_{ij}) \geq 0$, $i \in I_+$, and $1 + (\beta_0 + \sum_{j \in \mathcal{V}} \beta_j x_{ij}) \geq 0$, $i \in I_-$, where $I_+$ and $I_-$ contain indices of the "+" class points and the "−" class points, respectively. The above inequalities imply that the solution for $\beta_0$ is not unique, and $\beta_0$ can be any value in the interval

$$\left[ \max_{i \in I_-}(-1 - \sum_{j \in \mathcal{V}} \beta_j x_{ij}), \quad \min_{i \in I_+}(1 - \sum_{j \in \mathcal{V}} \beta_j x_{ij}) \right].$$

When $s$ increases, $\beta_j$ changes, and the length of this interval will shrink toward zero, which corresponds to two data points (from different classes) hitting the elbow simultaneously.

**The Unbalanced Case**

Without loss of generality, we assume $n_+ > n_-$. When $s = 0$, the solution is $\beta_0 = 1$ and $\boldsymbol{\beta} = 0$, which implies all the $I_-$ points are in $\mathcal{L}$ and all the $I_+$ points are in $\mathcal{E}$. When $s$ increases by an infinitesimal amount, some variable(s) will join the active variable set $\mathcal{V}$. By continuity, all the $I_-$ points will still stay in $\mathcal{L}$, but the $I_+$ points will split: some will join $\mathcal{L}$, some will join $\mathcal{R}$, and the rest will stay at $\mathcal{E}$. From (21)−(24), we can see in order to determine the right derivatives of $\beta_0$, $\beta_j$, $\alpha_i$ and $\eta$ with respect to $s$, it is crucial to identify the active variable set $\mathcal{V}$ and the elbow set $\mathcal{E}$. For the initial solution, it turns out that $\mathcal{V}$ and $\mathcal{E}$ can be identified via the following linear programming problem:

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i \in I_+} \epsilon_i + \sum_{i \in I_-} (1 - y_i f_i)$$
$$\text{subject to } 1 - y_i f_i \leq \epsilon_i, \quad i \in I_+,$$
$$\epsilon_i \geq 0, \quad i \in I_+,$$
$$\|\boldsymbol{\beta}\|_1 = |\beta_1| + \cdots + |\beta_p| \leq s.$$

Notice the loss for "−" class points has been changed from $(1 - yf)_+$ to $(1 - yf)$. This allows us to use *any* value of $s$, and always get the same $\mathcal{V}$ and $\mathcal{E}$ via the linear programming. Once the $\mathcal{V}$ and the $\mathcal{E}$ are identified, the initial right derivatives of $\beta_0$, $\beta_j$, $\alpha_i$ and $\eta$ with respect to $s$ can be solved from (21)−(24).

### 3.3. Main program

After the initial right derivatives of $\beta_0$, $\beta_j$, $\alpha_i$ and $\eta$ are identified, the main algorithm proceeds as follows.

1. Compute
   - the derivative of the residual for every non-elbow point

$$\frac{\Delta r_i}{\Delta s} = -y_i\Big(\frac{\Delta\beta_0}{\Delta s} + \sum_{j\in\mathcal{V}} x_{ij}\frac{\Delta\beta_j}{\Delta s}\Big), \quad i\notin\mathcal{E},$$

   where $r_i$ is the current residual $(1 - y_i f_i)$ for point $i$.
   - the derivative of the generalized correlation for every inactive variable

$$\frac{\Delta c_j}{\Delta s} = -\sum_{i\in\mathcal{E}} \frac{\Delta\alpha_i}{\Delta s} y_i x_{ij}, \quad j\notin\mathcal{V},$$

   where $c_j$ is the current generalized correlation value for variable $x_j$, given by (25).
2. Compute how much increase of $s$ is needed to get to each type of event:
   - an elbow point leaves the elbow, $\delta_s^1 = \min_{i\in\mathcal{E}} \max((0 - \alpha_i)/(\Delta\alpha_i/\Delta s),$ $(1 - \alpha_i)/(\Delta\alpha_i/\Delta s))$;
   - a non-elbow point hits the elbow, $\delta_s^2 = \min_{i\in\mathcal{E}_+^c} ((0 - r_i)/(\Delta r_i/\Delta s))$, where $\mathcal{E}_+^c = \{i : (0 - r_i)/(\Delta r_i/\Delta s) > 0, i\notin\mathcal{E}\}$;
   - an active variable becomes inactive, $\delta_s^3 = \min_{j\in\mathcal{V}_+} ((0 - \beta_j)/(\Delta\beta_j/\Delta s))$, where $\mathcal{V}_+ = \{j : (0 - \beta_j)/(\Delta\beta_j/\Delta s) > 0, j\in\mathcal{V}\}$;
   - an inactive variable joins the active set, $\delta_s^4 = \min_{j\notin\mathcal{V}} \max((-\eta - c_j)/(\Delta c_j/\Delta s + \Delta\eta/\Delta s), (\eta - c_j)/(\Delta c_j/\Delta s - \Delta\eta/\Delta s))$;
   - the generalized correlation of active variables reduces to zero, $\delta_s^5 = (0-\eta)/(\Delta\eta/\Delta s)$.

   The first four items correspond to the four types of events introduced in Section 3.1; the last item corresponds to one termination criterion of the algorithm.
3. Find which event happens first, $\delta_s = \min(\delta_s^1, \delta_s^2, \delta_s^3, \delta_s^4, \delta_s^5)$, and update

$$\alpha_i \longleftarrow \alpha_i + \delta_s\frac{\Delta\alpha_i}{\Delta s}, \quad i\in\mathcal{E},$$

$$\beta_0 \longleftarrow \beta_0 + \delta_s\frac{\Delta\beta_0}{\Delta s},$$

$$\beta_j \longleftarrow \beta_j + \delta_s\frac{\Delta\beta_j}{\Delta s}, \quad j\in\mathcal{V},$$

$$\eta \longleftarrow \eta + \delta_s\frac{\Delta\eta}{\Delta s}.$$

4. Update $\mathcal{L}, \mathcal{R}, \mathcal{E}$ and $\mathcal{V}$.
5. If any one of the following termination criterion is satisfied, stop the algorithm.
   - The generalized correlation reduces to zero.

- Two classes have been perfectly separated.
- A pre-specified maximum iteration number is reached.

Otherwise, use (21)−(24) to compute the new derivatives, and go back to step 1.

### 3.4. Solution path for fixed $\lambda_1$

We have described an algorithm for solving the solution path of the DrSVM when $\lambda_2$ is fixed. In this section, we briefly describe a similar algorithm that solves the solution path of the DrSVM when $\lambda_1$ is fixed. We first prove Theorem 3.

**Proof of Theorem 3.** When $\lambda_1$ is fixed and $\lambda_2$ changes, the solution has to satisfy (17)−(19) in Section 3.1, which are derived from the Lagrange and KKT conditions. If $D = 1/\lambda_2$ and $\alpha_i^* = D\alpha_i$, (17)−(19) become

$$\beta_j - \sum_{i=1}^n \alpha_i^* y_i x_{ij} = -\lambda_1 D \text{sign}(\beta_j), \quad j \in \mathcal{V},$$

$$\sum_{i=1}^n \alpha_i^* y_i = 0,$$

$$y_i \Big( \beta_0 + \sum_{j \in \mathcal{V}} x_{ij} \beta_j \Big) = 1, \quad i \in \mathcal{E}.$$

This system consists of $|\mathcal{E}| + |\mathcal{V}| + 1$ equations in $|\mathcal{E}| + |\mathcal{V}| + 1$ unknowns: $\beta_0$, $\beta_j$ $(j \in \mathcal{V})$, $\alpha_i^*$ $(i \in \mathcal{E})$. Therefore, using the same argument as in Section 3.1, one can show the solution $(\beta_0, \boldsymbol{\beta})$ is piecewise linear in $D$ (or $1/\lambda_2$).

Similarly, one can show that $\alpha_i$ are also piecewise linear, but piecewise linear in $\lambda_2$, rather than $1/\lambda_2$. These facts help us design an efficient algorithm to compute the solution path of the DrSVM when $\lambda_1$ is fixed. The idea is similar to the algorithm in Section 3.3, with minor modifications in computing how much increase of $D$ (or decrease of $\lambda_2$) is needed to get to the next event:

- an elbow point leaves the elbow, $\delta_{\lambda_2}^1 = \max_{i \in \mathcal{E}} \min((0 - \alpha_i)/(\Delta\alpha_i/\Delta\lambda_2), (1 - \alpha_i)/(\Delta\alpha_i/\Delta\lambda_2))$;
- a non-elbow point hits the elbow, $\delta_D^2 = \min_{i \in \mathcal{E}_+^c} ((0 - r_i)/(\Delta r_i/\Delta D))$, where $\mathcal{E}_+^c = \{i : (0 - r_i)/(\Delta r_i/\Delta s) > 0, i \notin \mathcal{E}\}$;
- an active variable becomes inactive, $\delta_D^3 = \min_{j \in \mathcal{V}_+} ((0 - \beta_j)/(\Delta\beta_j/\Delta D))$, where $\mathcal{V}_+ = \{j : (0 - \beta_j)/(\Delta\beta_j/\Delta s) > 0, j \in \mathcal{V}\}$;
- an inactive variable joins the active set, $\delta_{\lambda_2}^4 = \max_{j \notin \mathcal{V}} \min((-\lambda_1 - c_j)/(\Delta c_j/\Delta\lambda_2), (\lambda_1 - c_j)/(\Delta c_j/\Delta\lambda_2))$;
- the tuning parameter $\lambda_2$ reduces to zero, $\delta_{\lambda_2}^5 = -\lambda_2$.

Again, the first four items correspond to the four types of events in Section 3.1; the last item corresponds to one termination criterion of the algorithm. Which event happens first can then be determined by

$$\delta_{\lambda_2} = \max\left(\frac{-\lambda_2^2\delta_D^1}{1 + \lambda_2\delta_D^1}, \delta_{\lambda_2}^2, \frac{-\lambda_2^2\delta_D^3}{1 + \lambda_2\delta_D^3}, \delta_{\lambda_2}^4, \delta_{\lambda_2}^5\right).$$

The rest of the algorithm is the same as in Section 3.3.

### 3.5. Computational complexity

The major computational cost is associated with solving the linear system $(21)-(24)$ at each step, which involves $|\mathcal{E}| + 2$ equations and unknowns (after plugging (21) in (23) and (24)). Solving such a system involves $O(|\mathcal{E}|^3)$ computations. However, for any two consecutive steps, the linear systems usually differ by only one row or one column (corresponding to one of the four types of events); therefore, the computational cost can be reduced to $O(|\mathcal{E}|^2)$ via inverse updating/downdating. The computation of $\Delta\beta_j/\Delta s$ in (21) requires $O(|\mathcal{E}| \cdot |\mathcal{V}|)$ computations after getting $\Delta\alpha_i/\Delta s$. Notice, due to the nature of $(21)-(24)$, $|\mathcal{E}|$ is always less than or equal to $\min(n, p)$ and, since $|\mathcal{V}| \leq p$, the computational cost at each step can be estimated (bounded) as $O(\min^2(n, p) + p\min(n, p))$.

It is difficult to predict the number of steps on the solution path for arbitrary data. Our experience so far suggests that the total number of steps is $O(\min(n, p))$. This can be heuristically understood in the following way: if $n < p$, the training data are perfectly separable by a linear model, then it takes $O(n)$ steps for every data point to pass through the elbow to achieve the zero loss; if $n > p$, then it takes $O(p)$ steps to include every variable in the fitted model. Overall, this suggests the total computational cost is $O(p\min^2(n, p)+\min^3(n, p))$.

## 4. Numerical Results

In this section, we use both simulation data and real-world data to illustrate the DrSVM. In particular, we want to show that with high dimensional data, the DrSVM is able to remove irrelevant variables, and identify relevant (sometimes correlated) variables.

### 4.1. Simulation

We first consider the scenario where all input variables are independent. The "+" class has a normal distribution with mean $\boldsymbol{\mu}_+ = (\underbrace{0.5, \ldots, 0.5}_{5}, \underbrace{0, \ldots, 0}_{p-5})^\top$, $\boldsymbol{\Sigma} = \boldsymbol{I}_{p \times p}$. The "−" class has a similar distribution except that $\boldsymbol{\mu}_- = (\underbrace{-0.5, \ldots, -0.5}_{5}, \underbrace{0, \ldots, 0}_{p-5})^\top$. So the Bayes optimal classification rule only depends on $x_1, \ldots, x_5$, and the Bayes error is 0.132, independent of the dimension $p$.

We consider both the $n > p$ and $n \ll p$. In the $n > p$ case, we generate $100 = 50 + 50$ training data, each input $\boldsymbol{x}_i$ is a $p = 10$-dimensional vector; in the $n \ll p$ case, we generate $50 = 25 + 25$ training data, each input $\boldsymbol{x}_i$ is a $p = 300$-dimensional vector. We compare the $L_1$-norm SVM, the $L_2$-norm SVM, and the DrSVM. For fairness, we use $20,000$ validation data to select the tuning parameters for each method, then apply the selected models to a separate $20,000$ testing data set. Each experiment is repeated 30 times. The means of the prediction errors and the corresponding standard errors (in parentheses) are summarized in Table 1. As we can see, the prediction errors of the $L_1$-norm SVM and the DrSVM are similar: both are close to the optimal Bayes error when $n > p$, and degrade a little bit when $n \ll p$. This is not the case for the $L_2$-norm SVM: in the $n > p$ case, the prediction error is only slightly worse than that of the $L_1$-norm SVM and the DrSVM, but it degrades dramatically in the $n \ll p$ case. This is due to the fact that the $L_2$-norm SVM uses all input variables, and its prediction accuracy is polluted by the noise variables.

Table 1. Comparison of the prediction performance when all input variables are independent; $p_0$ is the number of relevant variables.

|          | $n$ | $p$ | $p_0$ | Test Error |
|----------|-----|-----|-------|------------|
| $L_2$ SVM |     |     |       | 0.145 (0.007) |
| $L_1$ SVM | 100 | 10  | 5     | 0.142 (0.008) |
| DrSVM    |     |     |       | 0.139 (0.005) |
| $L_2$ SVM |     |     |       | 0.323 (0.018) |
| $L_1$ SVM | 50  | 300 | 5     | 0.199 (0.031) |
| DrSVM    |     |     |       | 0.178 (0.021) |

Besides the prediction error, we also compare the selected variables of the $L_1$-norm SVM and the DrSVM (The $L_2$-norm SVM keeps all input variables). In particular, we consider $q_{\text{signal}}$ = number of selected relevant variables, and $q_{\text{noise}}$ = number of selected noise variables. The results are in Table 2. Again, we see that the $L_1$-norm SVM and the DrSVM perform similarly; both are able to identify the relevant variables (the $L_1$-norm SVM missed one on average) and remove most of the irrelevant variables.

Now we consider the scenario when the relevant variables are correlated. Similar as the independent scenario, the "+" class has a normal distribution, with mean and covariance $\boldsymbol{\mu}_+ = (\underbrace{1, \ldots, 1}_{5}, \underbrace{0, \ldots, 0}_{p-5})^{\mathsf{T}}$,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^*_{5 \times 5} & \boldsymbol{0}_{5 \times (p-5)} \\ \boldsymbol{0}_{(p-5) \times 5} & \boldsymbol{I}_{(p-5) \times (p-5)} \end{pmatrix},$$

where the diagonal elements of $\mathbf{\Sigma}^*$ are 1 and the off-diagonal elements are all equal to $\rho = 0.8$. The "$-$" class has a similar distribution except that $\boldsymbol{\mu}_- = (\underbrace{-1,\ldots,-1}_{5},\underbrace{0,\ldots,0}_{p-5})^{\mathsf{T}}$. So the Bayes optimal classification rule depends on $x_1,\ldots,x_5$, which are highly correlated. The Bayes error is 0.138, independent of the dimension $p$.

Table 2. Comparison of variable selection when all input variables are independent; $p_0$ is the number of relevant variables; $q_{\text{signal}}$ is the number of selected relevant variables, and $q_{\text{noise}}$ is the number of selected noise variables.

|          | $n$  | $p$  | $p_0$ | $q_{\text{signal}}$ | $q_{\text{noise}}$ |
|----------|------|------|-------|---------------------|--------------------|
| $L_1$ SVM | 100  | 10   | 5     | 5.00 (0.00)         | 2.43 (1.52)        |
| DrSVM    |      |      |       | 5.00 (0.00)         | 1.80 (1.30)        |
| $L_1$ SVM | 50   | 300  | 5     | 3.87 (0.82)         | 4.33 (4.86)        |
| DrSVM    |      |      |       | 4.53 (0.57)         | 6.37 (4.35)        |

Again, we consider both the $n > p$ and $n \ll p$. In the $n > p$ case, $n = 50+50$ and $p = 10$. In the $n \ll p$ case, $n = 25 + 25$ and $p = 300$. Each experiment is repeated 30 times. The result for the prediction errors are shown in Table 3. Now when changing from the $n > p$ case to the $n \ll p$ case, the performance of the $L_1$-norm SVM, as well as the $L_2$-norm SVM, degrades, but the DrSVM performs about the same. Table 4 compares the variables selected by the $L_1$-norm SVM and the DrSVM, which sheds some light on what happened. Both the $L_1$-norm SVM and the DrSVM are able to identify relevant variables. However, when the relevant variables are highly correlated, the $L_1$-norm SVM tends to keep only a small subset of the relevant variables, and overlook the others, while the DrSVM tends to identify all of them, due to the grouping effect. Both methods seem to work well in removing irrelevant variables.

Table 3. Comparison of the prediction performance when the relevant variables are highly correlated; $p_0$ is the number of relevant variables.

|          | $n$  | $p$  | $p_0$ | Test Error     |
|----------|------|------|-------|----------------|
| $L_2$ SVM |      |      |       | 0.142 (0.003)  |
| $L_1$ SVM | 100  | 10   | 5     | 0.144 (0.003)  |
| DrSVM    |      |      |       | 0.140 (0.001)  |
| $L_2$ SVM |      |      |       | 0.186 (0.012)  |
| $L_1$ SVM | 50   | 300  | 5     | 0.151 (0.007)  |
| DrSVM    |      |      |       | 0.139 (0.004)  |

Table 4. Comparison of variable selection when the relevant variables are highly correlated; $p_0$ is the number of relevant variables; $q_{\text{signal}}$ is the number of selected relevant variables, and $q_{\text{noise}}$ is the number of selected noise variables.

|          | $n$ | $p$ | $p_0$ | $q_{\text{signal}}$ | $q_{\text{noise}}$ |
|----------|-----|-----|-------|---------------------|--------------------|
| $L_1$ SVM | 100 | 10  | 5     | 3.73 (0.69)         | 0.30 (0.53)        |
| DrSVM    |     |     |       | 5.00 (0.00)         | 0.10 (0.31)        |
| $L_1$ SVM | 50  | 300 | 5     | 2.17 (0.83)         | 0.30 (0.60)        |
| DrSVM    |     |     |       | 4.90 (0.40)         | 0.97 (2.03)        |

In the last, we consider a scenario where the relevant variables have different contributions to the classification, and the pairwise correlations are not all equal. The basic setup is similar to the above two scenarios, except that $\boldsymbol{\mu}_+ = (\underbrace{1,\ldots,1}_{5},\underbrace{0,\ldots,0}_{p-5})^\top$, $\boldsymbol{\mu}_- = (\underbrace{-1,\ldots,-1}_{5},\underbrace{0,\ldots,0}_{p-5})^\top$, and

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} 1 & 0.8 & 0.8^2 & 0.8^3 & 0.8^4 \\ 0.8 & 1 & 0.8 & 0.8^2 & 0.8^3 \\ 0.8^2 & 0.8 & 1 & 0.8 & 0.8^2 \\ 0.8^3 & 0.8^2 & 0.8 & 1 & 0.8 \\ 0.8^4 & 0.8^3 & 0.8^2 & 0.8 & 1 \end{pmatrix}.$$

The Bayes optimal classification boundary is given by $1.11x_1 + 0.22x_2 + 0.22x_3 + 0.22x_4 + 1.11x_5 = 0$, and the Bayes error is 0.115. Notice that the true coefficients $\beta_2, \beta_3$ and $\beta_4$ are small compared with $\beta_1$ and $\beta_5$. To test our algorithm for the unbalanced case, we let $n = 60 + 40$ when $p = 10$, and $n = 30 + 20$ when $p = 300$. Each experiment is repeated 30 times. The results are summarized in Tables 5 and 6. As we can see, the DrSVM still dominates the $L_1$-norm SVM in terms of identifying relevant variables.

Table 5. Comparison of the prediction performance when the relevant variables have different class means and the pairwise correlations are not all equal; $p_0$ is the number of relevant variables.

|          | $n$ | $p$ | $p_0$ | Test Error    |
|----------|-----|-----|-------|---------------|
| $L_2$ SVM |     |     |       | 0.128 (0.008) |
| $L_1$ SVM | 100 | 10  | 5     | 0.117 (0.004) |
| DrSVM    |     |     |       | 0.115 (0.003) |
| $L_2$ SVM |     |     |       | 0.212 (0.022) |
| $L_1$ SVM | 50  | 300 | 5     | 0.125 (0.010) |
| DrSVM    |     |     |       | 0.120 (0.006) |

Table 6. Comparison of variable selection when the relevant variables have different class means and the pairwise correlations are not all equal; $p_0$ is the number of relevant variables; $q_{signal}$ is the number of selected relevant variables, and $q_{noise}$ is the number of selected noise variables.

|            | $n$ | $p$ | $p_0$ | $q_{signal}$ | $q_{noise}$ |
|------------|-----|-----|-------|--------------|-------------|
| $L_1$ SVM  | 100 | 10  | 5     | 3.70 (0.84)  | 1.48 (0.67) |
| DrSVM      |     |     |       | 4.53 (0.57)  | 0.53 (1.04) |
| $L_1$ SVM  | 50  | 300 | 5     | 3.03 (0.72)  | 1.23 (1.87) |
| DrSVM      |     |     |       | 4.23 (0.94)  | 2.93 (4.72) |

## 4.2. Microarray analysis

In this section, we apply the DrSVM to classification of gene microarrays. Classification of patient samples is an important aspect of cancer diagnosis and treatment. The $L_2$-norm SVM has been successfully applied to microarray cancer diagnosis problems (Guyon, Weston, Barnhill and Vapnik (2002) and Mukherjee, Tamayo, Slonim, Verri, Golub, Mesirov and Poggio (1999)). However, one weakness of the $L_2$-norm SVM is that it only predicts a cancer class label but does not automatically select relevant genes for the classification. Often a primary goal in microarray cancer diagnosis is to identify the genes responsible for the classification, rather than class prediction. The $L_1$-norm SVM has an inherent gene (variable) selection property due to the $L_1$-norm penalty, but the maximum number of genes that the $L_1$-norm SVM can select is upper bounded by $n$, which is typically much smaller than $p$ in microarray problems. Another drawback of the $L_1$-norm SVM, as seen in the simulation study, is that it usually fails to identify groups of genes that share the same biological pathway, which have correlated expression levels. The DrSVM overcomes these difficulties, and achieves the goals of classification of patients and (group) selection of genes simultaneously.

We use a leukemia dataset (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing and Caligiuri (2000)) to illustrate the point. This dataset consists of 38 training data and 34 test data for two types of acute leukemia, *acute myeloid leukemia* (AML) and *acute lymphoblastic leukemia* (ALL). Each datum is a vector of $p = 2,308$ genes. The tuning parameters are chosen according to 10-fold cross-validation, then the final model is fitted on all the training data and evaluated on the test data. The results are summarized in Table 7. As we can see, the DrSVM seems to have the best prediction performance. However, notice this is a very small (and "easy") dataset, so the difference may not be significant. It is also worth noting that the 22 genes selected by the $L_1$-norm SVM is a subset of the 78 genes selected by the DrSVM. Figure 5 shows the heatmap of the selected 78 genes. We have ordered the genes by hierarchical clustering, and similarly for all $38 + 34$ samples (based on the

Table 7. Results on the Leukemia dataset

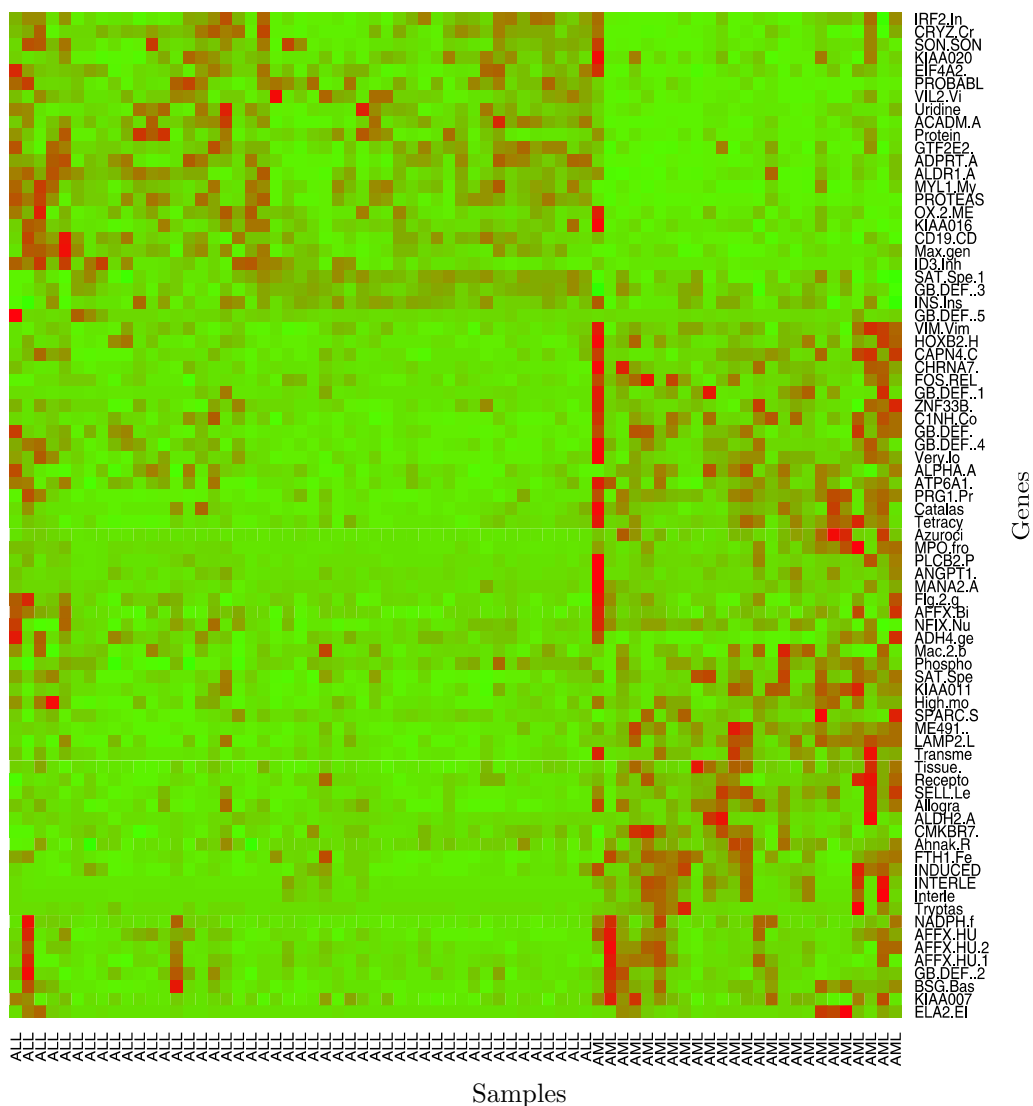|  | CV Error | Test Error | # of Genes |
|---|---|---|---|
| Golub | 3/38 | 4/34 | 50 |
| $L_2$-norm SVM | 0/38 | 1/34 | 2,308 |
| $L_1$-norm SVM | 3/38 | 1/34 | 22 |
| DrSVM | 0/38 | 0/34 | 78 |



Figure 5. Heatmap of the selected 78 genes. The genes are ordered by hierarchical clustering, and similarly for all $38 + 34$ samples.

selected genes). Clear separation of the two classes is evident. Roughly speaking, the top set of genes overexpress for ALL and underexpress for AML; for the bottom set of genes, this is reversed.

## 4.3. Handwritten digit recognition

In this section, we consider a classical handwritten digit recognition example. Recognition of generic object categories is one of the major challenges in computer vision. Most current methods can roughly be divided into brightness-based and feature-based. Brightness-based methods use the raw pixels as the input vector (usually after certain normalization so that all images have approximately the same size and orientation). Feature-based methods first extract information from the image, such as shape, texture and color, then use this information as the input vector.

We work with feature-based methods for they are closer to the biological vision system than brightness-based methods. In particular, we concentrate on shape features, for in biological vision, shape cue is arguably the strongest among all types of cues (shape, texture, color, etc.) for visual object recognition (Palmer (1999)).

There are two distinct challenges here.

- The size of the training dataset is small (a person does not need to see many images to generalize the notion of the new object to a novel image). On the other hand, due to the richness of information from a single image, shape features alone could be on the order of $1,000$ real numbers, such as edge locations and directions, corner locations, arrangement of features, etc. So the feature set is rich. This falls into the $p > n$ framework.
- Besides predicting the correct object category for a given image, another challenge is to identify the relevant features that contribute most to the classification. For example, when looking at a slate of handwritten digits (Figure 6), despite the variation in writing style, one can still see the distinctive parts of the shape which best separate one class from the other.

The proposed DrSVM method naturally applies here. We compare the $L_2$-norm SVM, the $L_1$-norm SVM and the DrSVM on a subset of the MINIST database (LeCun, Jackel, Bottou, Cortes, Denker, Drucker, Guyon, Muller, Sackinger, Simard and Vapnik (1995)). The standard $L_2$-norm SVM has been applied to this database before and shown good performance (LeCun et al. (1995)), on par with human performance. However, once again, a weakness of the $L_2$-norm SVM is that it only predicts an object class but does not automatically

-1
0
1
2
3
5
10
15
20
30
40
50
60
1/
$\lambda_1$
$\lambda_2$
-Norm SVM
DrSVM (fixed )
Loss
$yf$
Right
Elbow
Left
$\beta$
$\beta_1$
$\beta_2$
Mixture
$L_1$
$L_2$
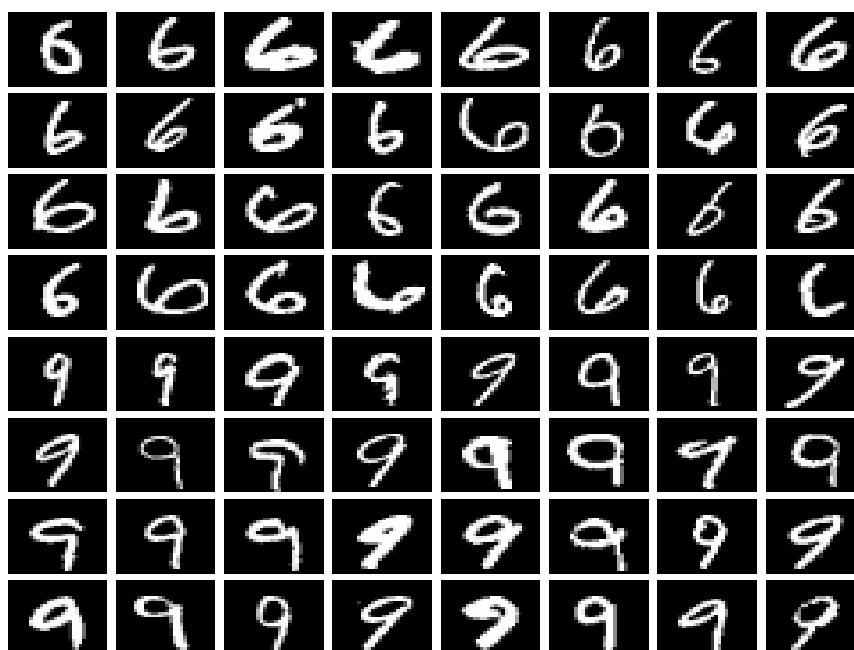1.9
$L_1$
$L_2$
$\|\beta\|_1$

Figure 6. Some examples of handwritten digits 6 and 9.

select relevant features for the classification. The $L_1$-norm SVM has the inherent feature selection property, but tends to overlook correlated features. The DrSVM overcomes these two difficulties. We use digit 6 versus digit 9 as an example. We randomly sampled $250 + 250$ training data and $750 + 750$ test data from the MINIST database. Each digit (sample) consists of 804 shape features based on the so called *shape context distance* (Belongie, Malik and Puzicha (2002) and Zhang and Malik (2003)). Tuning parameters were selected using 10-fold cross-validation. Numerical results are summarized in Table 8. The DrSVM performed a little better than the standard $L_2$-norm SVM in terms of misclassification error, and it automatically selected 128 shape features, while the standard $L_2$-norm SVM used all 804 features. See Figure 7.

Table 8. Results for 6 versus 9 from the MINIST database. Training size is $250+250$, test size is $750+750$, and each digit consists of 804 shape features.

|          | CV Error | Test Error | # of Features |
|----------|----------|------------|---------------|
| $L_2$ SVM | 6/500   | 22/1500    | 804           |
| $L_1$ SVM | 6/500   | 17/1500    | 49            |
| DrSVM    | 1/500    | 15/1500    | 128           |

$1/$
$\lambda_1$
$\lambda_2$
-Norm SVM
DrSVM (fixed
)
Loss
$yf$
Right
Elbow
Left
$\beta$
$\beta_1$
$\beta_2$
Mixture
$L_1$
$L_2$
1.9
$L_1$
$L_2$
$\|\beta\|_1$

Selected Features

$1/$
$\lambda_1$
$\lambda_2$
-Norm SVM
DrSVM (fixed
)
Loss
$yf$
Right
Elbow
Left
$\beta$
$\beta_1$
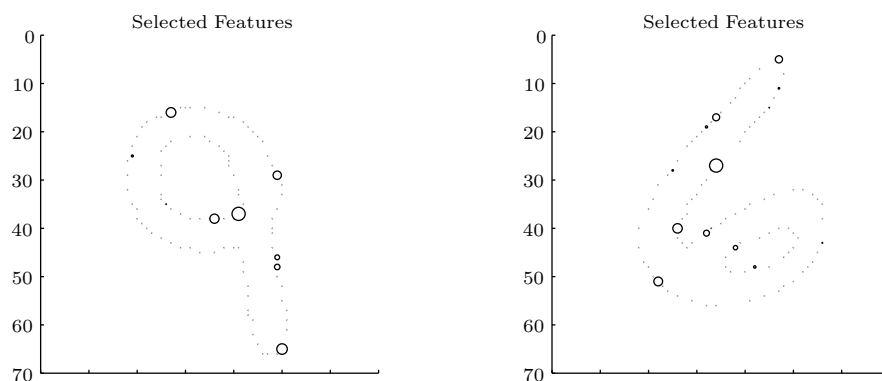$\beta_2$
Mixture
$L_1$
$L_2$
1.9
$L_1$
$L_2$
$\|\beta\|_1$



Figure 7. The center of each circle indicates the position of the selected feature, and the size of each circle is proportional to the corresponding feature's *importance*.

## 5. Discussion

We have applied the elastic-net penalty to the hinge loss, and proposed the DrSVM method for classification problems. This method is especially useful with high dimensional data, with respect to effectively removing irrelevant variables and identifying relevant variables. Unlike previously developed support vector machines, e.g. the $L_1$-norm SVM, the DrSVM is able to select groups of variables that are correlated, and the number of selected variables is no longer bounded by the size of the training data, thus being able to deal with the $p \gg n$ problem. We also proposed efficient algorithms that can compute the whole solution paths of the DrSVM, which facilitates selection of the tuning parameters.

There are several interesting directions in which the DrSVM can be extended:

- How to efficiently utilize the solution paths? Computing one solution path is very efficient, but cross-validation can be computationally expensive. The ideal situation is to have a model selection criterion that can be calculated along the solution path; then once the solution path is computed, the best tuning parameter can be identified. The GACV (Wahba, Lin and Zhang (2000)) is a computable proxy for the generalized Kullback-Liebler distance, and seems to be a good model selection criterion for the support vector machine. Since the DrSVM has two tuning parameters, $\lambda_1$ and $\lambda_2$, combining the GACV criterion and a two-step procedure seems to be promising: In the first step, one can fix a (relatively large) value for $\lambda_2$, compute the solution path $\beta_{\lambda_2}(\lambda_1)$, and use the GACV criterion to select a value for $\lambda_1$; this step corresponds to setting an appropriate threshold to remove noise variables. In the second step, one fixes $\lambda_1$ at the value selected in step one, computes the

solution path $\hat{\beta}_{\lambda_1}(\lambda_2)$, and again uses the GACV criterion to identify a value for $\lambda_2$; this corresponds to getting the correct grouping effect for correlated variables. The advantage of this strategy is that it avoids a two-dimensional grid search. Our preliminary results are encouraging, and we will explore this further.

- The algorithm proposed in Section 3 is efficient. However, when both $n$ and $p$ are large, the initial solution (in the balanced case) may require substantial computational efforts. This is due the fact that the hinge loss function is not differentiable at the point $yf = 1$, and a linear programming is called upon to solve the initial solution. So the question is how one can modify the DrSVM to improve the computational efficiency? We can consider placing the hinge loss with the *Huberized hinge loss* (Rosset and Zhu (2004)). The Huberized hinge loss is

$$\phi(yf) = \begin{cases} \frac{1-\delta}{2} + (\delta - yf), & \text{if } yf \leq \delta, \\ \frac{(1-yf)^2}{2(1-\delta)}, & \text{if } \delta < yf \leq 1, \\ 0, & \text{otherwise,} \end{cases}$$

where $\delta < 1$. Figure 8 compares the Huberized hinge loss and the hinge loss. The Huberized hinge loss is differentiable everywhere, and it is straightforward to get the initial solution. The Huberized hinge loss has a similar shape as the hinge loss; therefore, one might expect the prediction performance of the Huberized hinge loss to be similar to that of the hinge loss.
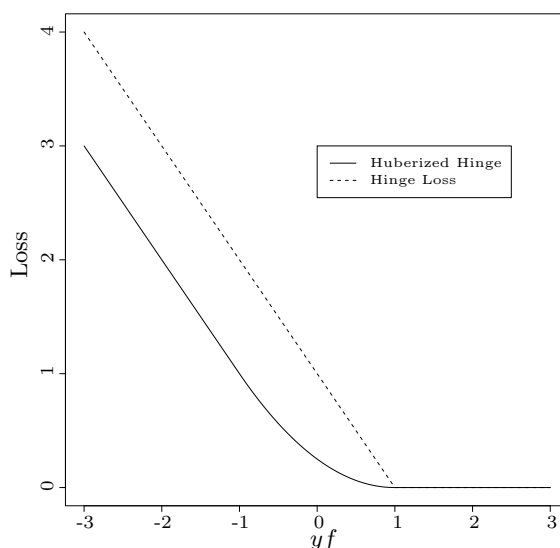


Figure 8. The hinge loss and the Huberized hinge loss (with $\delta = -1$).

## Acknowledgement

## References

Belongie, S., Malik, J. and Puzicha J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 509-522.

Boser, B., Guyon, I. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh.

Bradley, P. and Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines. In *International Conference on Machine Learning*. Morgan Kaufmann.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2**, 121-167.

Chen, S., Donoho, D. and Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM, J. Sci. Comput.* **20**, 33-61.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning* **20**, 273-297.

Donoho, D., Johnstone, I., Kerkyachairan, G. and Picard, D. (1995). Wavelet shrinkage: asymptopia? (with discussion). *J. Roy. Statist. Soc. Ser. B* **57**, 201-337.

Evgeniou, T., Pontil, M. and Poggio, T. (1999). Regularization networks and support vector machines. In *Advances in Large Margin Classifiers*. MIT Press.

Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004). Discussion of "Consistency in boosting" by W. Jiang, G. Lugosi, N. Vayatis and T. Zhang. *Ann. Statist.* **32**.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J. and Caligiuri, M. (2000). Molecular classification of cancer: class discovery and class predicion by gene expression monitoring. *Science* **286**, 531-536.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning* **46**, 389-422.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer-Verlag, New York.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**, 55-67.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P. and Vapnik, V. (1995). Learning algorithms for classification: a comparison on handwritten digit recognition. In *Neural Networks: The Statistical Mechanics Perspective*.

Mallat, S. and Zhang, Z. (1993). Matching pursuit in a time-frequency dictionary. *IEEE Trans. Signal Process.* **41**, 3397-3415.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J. and Poggio, T. (1999). Support vector machine classification of microarray data. Technical Report, AI Memo 1677, MIT.

Ng, A. (2004). Feature selection, $L_1$ vs. $L_2$ regularization, and rotational invariance. In *International Conference on Machine Learning*. Morgan Kaufmann.

Palmer, S. (1999). *Vision Science – Photons to Phenomenology*. MIT Press.

Rosset, S. and Zhu, J. (2004). Piecewise linear regularized solution paths. Technical Report, Department of Statistics, University of Michigan.

Rosset, S., Zhu, J. and Hastie, T. (2004). Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.* **5**, 941-973.

Song, M., Breneman, C., Bi, J., Sukumar, N., Bennett, K., Cramer, S. and Tugcu, N. (2002). Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *J. Chemical Information and Computer Sciences*.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58**, 267-288.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert space and the randomized GACV. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Wahba, G., Lin, Y. and Zhang, H. (2000). Generalized approximate cross validation for support vector machines. In *Advances in Large Margin Classifiers*. MIT Press.

Zhang, H. and Malik, J. (2003). Learning a discriminative classifier using shape context distances. In *IEEE Computer Vision and Pattern Recognition* **1**, 242-247.

Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2004). 1-norm SVMs. *Neural Inf. Process. Systems* **16**.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. Roy. Statist. Soc. Ser. B* **67**, 301-320.

Department of Statistics, University of Michigan, Ann Arbor, MI 48109, U.S.A.

E-mail: wang@umich.edu

Department of Statistics, University of Michigan, Ann Arbor, MI 48109, U.S.A.

E-mail: jizhu@umich.edu

School of Statistics, University of Minnesota, Minneapolis, MN 55455, U.S.A.

E-mail: hzou@stat.umn.edu