Chapter 2

VARIABLE SELECTION FOR THE LINEAR SUPPORT VECTOR MACHINE

Ji Zhu
Department of Statistics
University of Michigan
jizhu@umich.edu

Hui Zou
School of Statistics
University of Minnesota
hzou@stat.umn.edu

Abstract The standard L_2-norm support vector machine (SVM) is a widely used tool for the classification problem. 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. In this chapter, 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 also develop efficient algorithms to compute the whole solution paths of the DrSVM.

Keywords: SVM; Variable Selection; Quadratic Programming

J. Zhu and H. Zou: *Variable Selection for the Linear Support Vector Machine*, Studies in Computational Intelligence (SCI) **35**, 35–59 (2007)
www.springerlink.com

1. Introduction

In a standard two-class classification problem, we are given a set of training data (x_1, y_1), (x_2, y_2), $\ldots (x_n, y_n)$, where the input (predictor variable) $x_i \in \mathbb{R}^p$ is a p-dimensional vector and the output (response variable) $y_i \in \{1, -1\}$ is a binary categorical variable. The aim is to find a classification rule from the training data, so that when given a new input x, we can assign a class label, either 1 or -1, to it.

The support vector machine (SVM) has been a popular tool for the two-class classification problem in the machine learning field. Recently, it has also gained increasing attention from the statistics community. Below we briefly review the SVM from these two perspectives. We refer the readers to [2], [6], [11] and [21] for details.

Let us first consider the case when the training data can be perfectly separated by a hyperplane in $\mathbb{R}^p$. Define the hyperplane by

$$\{x : f(x) = \beta_0 + x^T\beta = 0\},$$

where β is a unit vector: $\|\beta\|_2 = 1$, then $f(x)$ gives the signed distance from a point x to the hyperplane. Since the training data are linearly separable, we are able to find a hyperplane such that

$$y_i f(x_i) > 0, \quad \forall i. \tag{2.1}$$

Indeed, there are infinitely many such hyperplanes. Among the hyperplanes satisfying (2.1), the SVM looks for the one that maximizes the *margin*, where the margin is defined as the smallest distance from the training data to the hyperplane. Hence we can write the SVM problem as:

$$\begin{aligned} \max_{\beta,\beta_0,\|\beta\|_2=1} \quad & C, \\ \text{subject to} \quad & y_i(\beta_0 + x_i^T\beta) \geq C, \quad i = 1, \ldots, n, \end{aligned}$$

where C is the margin.

When the training data are not linearly separable, we allow some training data to be on the wrong side of the edges of the margin and introduce slack variables $\xi_i, \xi_i \geq 0$. The SVM problem then becomes

$$\max_{\beta,\beta_0,\|\beta\|_2=1} \quad C, \tag{2.2}$$

$$\text{subject to} \quad y_i(\beta_0 + x_i^T\beta) \geq C(1 - \xi_i), \quad i = 1, \ldots, n, \tag{2.3}$$

$$\xi_i \geq 0, \sum \xi_i \leq B, \tag{2.4}$$

where B is a pre-specified positive number, which can be regarded as a *tuning parameter*. Figure (2.1) illustrates both the linearly separable

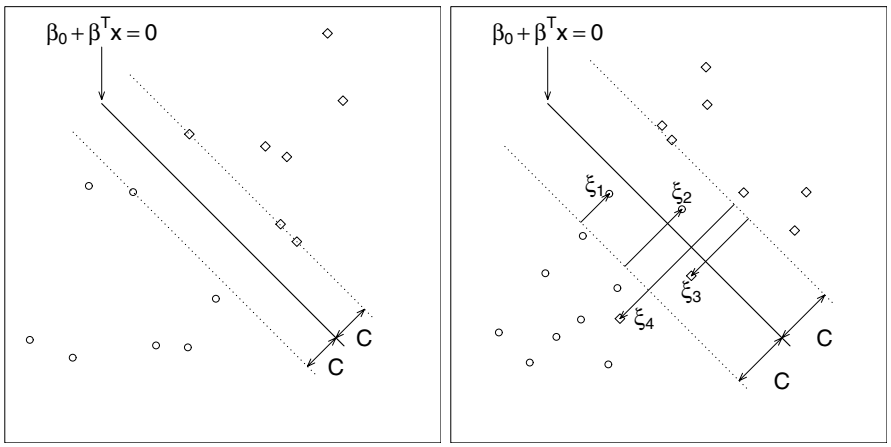

Figure 2.1. Linear support vector machine classifiers.

and non-separable cases. This presents the geometric view of the linear SVM, i.e., a hyperplane that maximizes the margin of the training data.

It turns out that the SVM is also equivalent to a regularized function fitting problem. With $f(x) = \beta_0 + x^T\beta$, consider the optimization problem:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|\beta\|_2^2, \tag{2.5}$$

where the subscript "+" indicates the positive part and λ is a tuning parameter. One can show that the solutions to (2.5) have one-to-one correspondence to those of the SVM (2.2) – (2.4).

Notice that (2.5) has the form *loss* + *penalty*, which is a familiar paradigm to statisticians in function estimation. The loss function $(1 - yf)_+$ is called the *hinge* loss (Figure 2.2). [13] shows that:

$$\arg\min_f E_Y \left[(1 - Yf(x))_+\right] = \text{sign}\left(p_1(x) - \frac{1}{2}\right),$$

where $p_1(x) = P(Y = 1|X = x)$ is the conditional probability of a point being in class 1 given $X = x$. Hence the SVM can be interpreted as trying to implement the optimal Bayes classification rule without estimating the actual conditional probability $p_1(x)$. The penalty is the L_2-norm of the coefficient vector, the same as that used in the ridge regression [12]. 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 towards zero. It is well known that this shrinkage has the effect of controlling the variance of

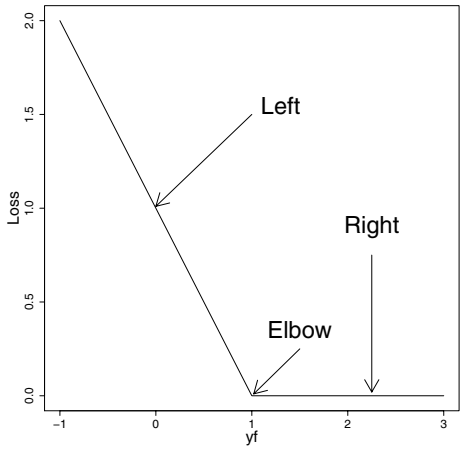

Figure 2.2. The hinge loss of the SVM. *Elbow* indicates the point $1 - yf = 0$, *Left* indicates the region to the left of the elbow, and *Right* indicates the region to the right of the elbow.

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.

The L_2-norm penalty shrinks the fitted coefficients *towards* zero, but never *exactly* equal to zero. All predictor variables are kept in the fitted model, thus there is no variable selection. Instead of using the L_2-norm penalty, in this chapter, we will consider using other forms of the penalty for the linear SVM. Our goal is to remove trivial variables that are not helpful in classification. The rest of the chapter are organized as follows: in Section 2, we introduce the L_1-norm SVM and the doubly regularized SVM; in Section 3, we describe efficient algorithms that compute entire solution paths of the doubly regularized SVM; numerical results are presented in Section 4, and we conclude the chapter with a discussion section.

2. Variable Selection for the Linear SVM

The L_1-norm SVM

We first consider an L_1-norm SVM model ([1], [19], [23]):

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+ + \lambda\|\beta\|_1, \tag{2.6}$$

where we use the L_1-norm of the coefficient vector to replace the L_2-norm. A canonical example that uses the L_1-norm penalty is the Lasso

[20] for the regression problem, where the response y is continuous rather than categorical:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T\beta)^2 + \lambda\|\beta\|_1.$$

[14] and [3] also apply the idea to signal processing, where the bases functions are orthogonal to each other.

Similar to the L_2-norm penalty, the L_1-norm penalty also shrinks the fitted coefficients toward zero, hence (2.6) also benefits from the reduction in the fitted coefficients' variance. Another important property of the L_1-norm penalty is that because of the L_1 nature of the penalty, with sufficiently large λ, some of the fitted coefficients will be *exactly* zero, i.e., *sparse* solution. Therefore, the L_1-norm penalty has an inherent variable selection property, while this is not the case for the L_2-norm penalty. Furthermore, as λ varies, different fitted coefficients will be set to zero, hence the L_1-norm penalty also performs a kind of continuous variable selection.

We illustrate the concept of sparsity of β with a simple example. We generate 30 training data in each of two classes. Each input x_i is a $p = 30$ dimensional vector. For the "+" class, x_i has a normal distribution with mean and covariance matrix

$$\begin{aligned} \mu_+ &= (\underbrace{1,\ldots,1}_{5},\underbrace{0,\ldots,0}_{25})^T, \\ \Sigma &= \begin{pmatrix} \Sigma^*_{5\times 5} & 0_{5\times 25} \\ 0_{25\times 5} & I_{25\times 25} \end{pmatrix}, \end{aligned}$$

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

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

So the Bayes optimal classification boundary is given by

$$x_1 + \cdots + x_5 = 0,$$

and it only depends on the first five inputs $x_1,\ldots,x_5$. We compare the fitted coefficient paths for the L_1-norm SVM and the standard L_2-norm SVM as λ varies. In the upper panels of Figure 2.3, the five solid paths are for $x_1,\ldots,x_5$ (or $\beta_1,\ldots,\beta_5$), which are the relevant variables; the dashed lines are for $x_6,\ldots,x_{30}$, which are the irrelevant noise variables.

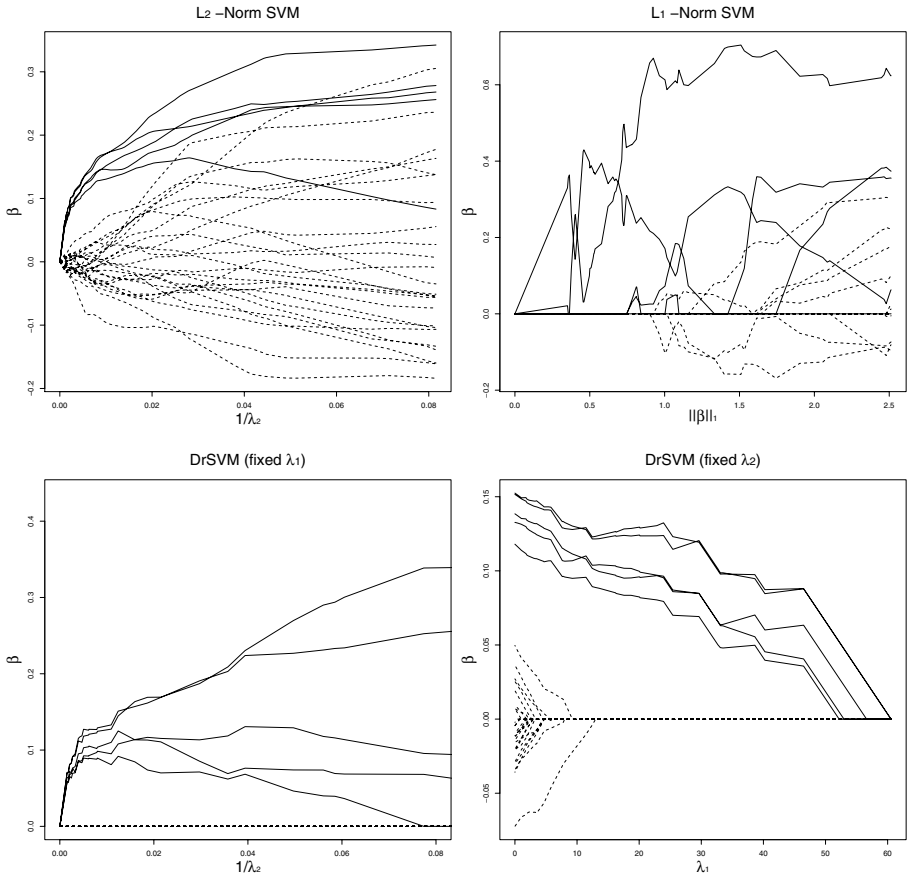

Figure 2.3. Comparison of different SVMs on a simple simulation data. The solid curves correspond to relevant variables, and the dashed curves correspond to irrelevant variables. The relevant variables are highly correlated. The upper left panel is for the L_2-norm SVM, the upper right panel is for the L_1-norm SVM, the bottom panels are for the DrSVM. The bottom left panel fixes $\lambda_1 = 15$, and changes λ_2; the bottom right panel fixed $\lambda_2 = 160$, and changes λ_1. We can see the DrSVM identified all (correlated) relevant variables, and shrunk their coefficients close to each other.

As we can see in the upper right panel, when $\|\beta\|_1 \leq 0.8$, only the relevant variables have non-zero fitted coefficients, while the noise variables have zero coefficients. Thus when the regularization parameter varies, the L_1-norm penalty does a kind of continuous variable selection. This is not the case for the standard L_2-norm penalty (upper left panel): none of the β_j is equal to zero.

It is interesting to note that the L_2-norm penalty corresponds to a Gaussian prior for the β_j's, while the L_1-norm penalty corresponds to

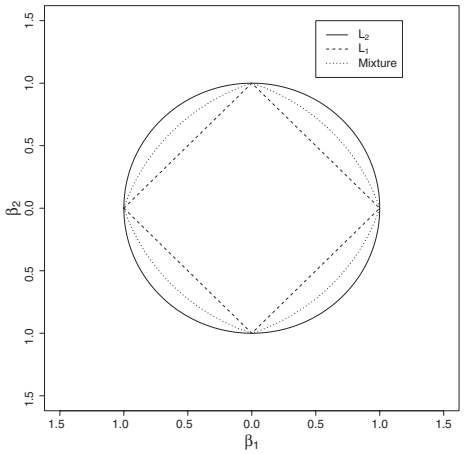

Figure 2.4. Two dimensional contour plots of the penalty function. The L_2-norm $\|\beta\|_2^2 = 1$, the L_1-norm $\|\beta\|_1 = 1$, and the elastic-net $0.5\|\beta\|_2^2 + 0.5\|\beta\|_1 = 1$.

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. Another way to look at these two penalties is that the equal penalty contours for the double exponential density in the p-dimensional Euclidean space spanned by the coefficients are hyper-diamonds, as illustrated in Figure 2.4, compared to hyper-spheres for the Gaussian density. Observing that a hyper-diamond has the vast majority of its volume in the corners gives us an intuitive sense of why we may expect the L_1-norm penalty to give sparse models.

The Doubly Regularized SVM

It has been argued that the L_1-norm penalty has advantages over the L_2-norm penalty under certain scenarios ([4], [7], [16]) such as when there are redundant noise variables. However, the L_1-norm penalty also suffers from two serious limitations [24]:

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 [18], 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 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.

[24] 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.

We apply the elastic-net penalty to the SVM. Specifically, we consider the following doubly regularized SVM, which we call the DrSVM:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+ + \frac{\lambda_2}{2}\|\beta\|_2^2 + \lambda_1\|\beta\|_1, \tag{2.7}$$

where both λ_1 and λ_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. Figure 2.4 compares contours of the L_2-norm, the L_1-norm, and the elastic-net penalty.

Grouping Effect

In this subsection, we show that the DrSVM tends to make highly correlated input variables have similar fitted coefficients, which we refer to as the *grouping effect*. 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:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \phi(y_i, f(x_i)) + \frac{\lambda_2}{2}\|\beta\|_2^2 + \lambda_1\|\beta\|_1, \tag{2.8}$$

where $f(x) = \beta_0 + x_i^T\beta$, $\phi(y, f) = \phi(yf)$ is a function of the *margin*. We further assume that $\phi(t)$ is Lipschitz continuous, i.e.

$$|\phi(t_1) - \phi(t_2)| \le M|t_1 - t_2| \quad \text{for some positive finite } M.$$

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 2.1 *Denote the solution to (2.8) as β. Assume the loss function ϕ is Lipschitz continuous, then for any pair (j, ℓ), we have*

$$|\beta_j - \beta_\ell| \leq \frac{M}{\lambda_2}\|x_j - x_\ell\|_1 = \frac{M}{\lambda_2}\sum_{i=1}^{n}|x_{ij} - x_{i\ell}|. \tag{2.9}$$

Furthermore, if the input variable x_j, x_ℓ are centered and normalized, then

$$|\beta_j - \beta_\ell| \leq \frac{\sqrt{n}M}{\lambda_2}\sqrt{2(1-\rho)}, \tag{2.10}$$

where $\rho = cor(x_j, x_\ell)$ is the sample correlation between x_j and x_ℓ.

For lack of space, we omit the proof of the theorem, and illustrate the grouping effect with a simple example. This is the same example used earlier to compare the L_1-norm SVM and the L_2-norm SVM. The results of the DrSVM are shown in the lower panels of Figure 2.3. Notice that the relevant variables $x_1, \ldots, x_5$ are highly correlated. As we can see, although the L_1-norm SVM did variable selection and were able to remove the noise variables, but it failed to identify the group of correlated variables, while the DrSVM successfully selected all five relevant variables, and shrunk their coefficients close to each other.

3. Piecewise Linear Solution Paths

Since the L_2-norm SVM and the L_1-norm SVM are special cases of the DrSVM, we only focus on the DrSVM in this section. To get a good classification rule that performs well on future data, it is important to select appropriate tuning parameters λ_1 and λ_2. In practice, people can pre-specify a finite grid of values for λ_1 and λ_2 that covers a wide range, then use either a separate validation dataset or cross-validation to do a grid search, and find values for the (λ_1, λ_2) pair that give the best performance among the given grid. In this section, we show that the solution path for a fixed value of λ_2, denoted as $\beta_{\lambda_2}(\lambda_1)$, is *piece-wise linear* as a function of λ_1 (in the $\mathbb{R}^p$ space); and for a fixed value of λ_1, the solution path, denoted as $\beta_{\lambda_1}(\lambda_2)$, is piece-wise linear as a function of $1/\lambda_2$. A canonical example for piecewise linear solution path is the Lasso [5]. The piecewise linearity property allows us to design efficient algorithms to compute the exact whole solution paths; furthermore, it helps us understand how the solution changes with the tuning parameter

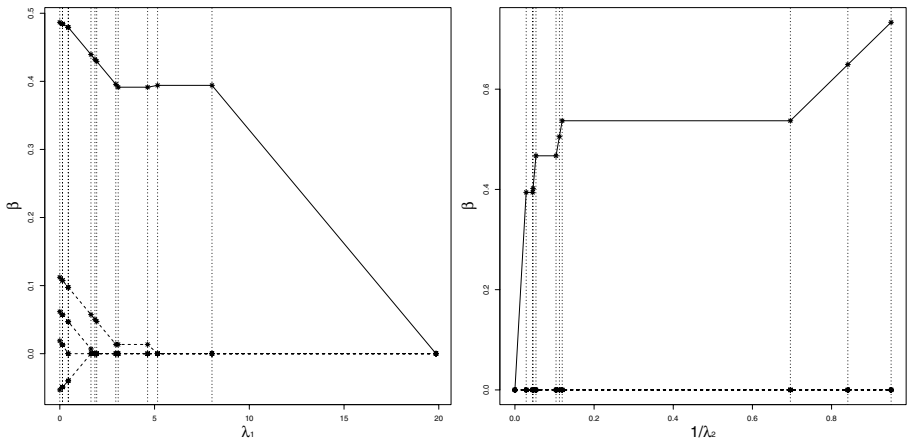

Figure 2.5. The simulation setup is the same as in Section 2, except the size of the training data is $n = 8 + 8$, the number of input variables is $p = 5$, and only the first variable x_1 is relevant to the optimal classification boundary. The solid line corresponds to β_1, the dashed lines correspond to $\beta_2, \ldots, \beta_5$. The left panel is for $\beta_{\lambda_2}(\lambda_1)$ (with $\lambda_2 = 30$), and the right panel is for $\beta_{\lambda_1}(\lambda_2)$ (with $\lambda_1 = 6$).

and facilitates the adaptive selection of the tuning parameter. Figure 2.5 illustrates the piecewise linearity property; any segment between two adjacent vertical lines is linear.

THEOREM 2.2 *When λ_2 is fixed, the solution $\beta_{\lambda_2}(\lambda_1)$ for (2.7) is a piecewise linear function of λ_1.*

THEOREM 2.3 *When λ_1 is fixed, the solution $\beta_{\lambda_1}(\lambda_2)$ for (2.7) is a piecewise linear function of $1/\lambda_2$.*

Proof of Theorem 2.2

The optimization problem (2.7) for the DrSVM is equivalent to a quadratic programming problem:

$$\min_{\beta_0,\beta} \quad \sum_{i=1}^{n} \epsilon_i + \frac{\lambda_2}{2}\|\beta\|_2^2, \tag{2.11}$$

$$\text{subject to} \quad 1 - y_i f_i \le \epsilon_i, \tag{2.12}$$

$$\epsilon_i \ge 0, \quad i = 1, \ldots, n, \tag{2.13}$$

$$\|\beta\|_1 = |\beta_1| + \cdots + |\beta_p| \le s, \tag{2.14}$$

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 λ_1 is replaced by s. The optimization problem

(2.7) and this quadratic programming problem are equivalent in the sense that for any value of λ_1, there exists a value of s, such that the solution to (2.7) and the solution to the quadratic programming problem are identical. To solve for the quadratic programming problem, we write the Lagrange:

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

where $\alpha_i \geq 0$, $\gamma_i \geq 0$ and $\eta \geq 0$ are Lagrange multipliers. Taking derivative of the Lagrange with respect to β_0, β and ϵ_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 $j \in \mathcal{V}$
- $1 - \alpha_i - \gamma_i = 0, \quad i = 1, \ldots, n$

where $\mathcal{V}$ contains the indices of non-zero coefficients, i.e. $\mathcal{V} = \{j : \beta_j \neq 0\}$. Notice the value of β_j is fully determined by the values of α_i and η. We also have the Karush-Kuhn-Tucker (KKT) conditions from the quadratic programming:

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

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 2.2). Inspecting the KKT conditions, we find

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

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

When s is small enough, the constraint (2.14) is active, i.e. $\|\beta\|_1 = s$. 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 (2.7). Suppose for a value $s < s^*$, the solution is

(β_0, β), hence $\mathcal{V}, \mathcal{L}, \mathcal{R}$ and $\mathcal{E}$ are also known. Then (β_0, β) have to satisfy the following equations derived from the Lagrange and KKT conditions:

$$\lambda_2 \beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} + \eta \text{sign}(\beta_j) = 0, \quad j \in \mathcal{V}, \tag{2.15}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \tag{2.16}$$

$$y_i(\beta_0 + \sum_{j \in \mathcal{V}} \beta_j x_{ij}) = 1, \quad i \in \mathcal{E}, \tag{2.17}$$

$$\|\beta\|_1 = \sum_{j \in \mathcal{V}} \text{sign}(\beta_j)\beta_j = s. \tag{2.18}$$

This linear system consists of $|\mathcal{E}| + |\mathcal{V}| + 2$ equations and $|\mathcal{E}| + |\mathcal{V}| + 2$ unknowns: α_i's, β_j's, β_0 and η. They can be further reduced to $|\mathcal{E}| + 2$ equations and $|\mathcal{E}| + 2$ unknowns by plugging (2.15) into (2.17) and (2.18). 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, such that the structure of the above linear system will not change. Taking right derivatives with respect to s, we have

$$\lambda_2 \frac{\Delta \beta_j}{\Delta s} - \sum_{i \in \mathcal{E}} \frac{\Delta \alpha_i}{\Delta s} y_i x_{ij} + \text{sign}(\beta_j) \frac{\Delta \eta}{\Delta s} = 0, \quad j \in \mathcal{V}, \tag{2.19}$$

$$\sum_{i \in \mathcal{E}} \frac{\Delta \alpha_i}{\Delta s} y_i = 0, \tag{2.20}$$

$$\frac{\Delta \beta_0}{\Delta s} + \sum_{j \in \mathcal{V}} \frac{\Delta \beta_j}{\Delta s} x_{ij} = 0, \quad i \in \mathcal{E}, \tag{2.21}$$

$$\sum_{j \in \mathcal{V}} \text{sign}(\beta_j) \frac{\Delta \beta_j}{\Delta s} = 1, \tag{2.22}$$

which does not depend on the value of s. This implies that the solution, α_i's, β_j's, β_0 and η, 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, so the structure of the linear system will change, which corresponds to a different linear piece on the solution path. Hence, the solution path is piecewise linear in s. Notice that η is equivalent to λ_1; therefore β_0, β and α_i are also piecewise linear in λ_1, and Theorem 2.2 holds. □

To identify changes in the structure of the linear system (or the asterisk points in Figure 2.5), 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 α_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:

$$c_j = \lambda_2 \beta_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij}. \tag{2.23}$$

From (2.15), we can see that all active variables in $\mathcal{V}$ have the same absolute generalized correlation value, which is η. Therefore, an inactive variable will join the active variable set when its absolute generalized correlation reaches η.

Proof of Theorem 2.3

When λ_1 is fixed and λ_2 changes, the solution has to satisfy (2.15) – (2.17), which are derived from the Lagrange and KKT conditions. Let $D = 1/\lambda_2$ and $\alpha_i^* = D\alpha_i$, (2.15) – (2.17) become:

$$\begin{aligned} \beta_j - \sum_{i=1}^{n} \alpha_i^* y_i x_{ij} &= -\lambda_1 D \text{sign}(\beta_j), \quad j \in \mathcal{V}, \\ \sum_{i=1}^{n} \alpha_i^* y_i &= 0, \\ y_i \left(\beta_0 + \sum_{j \in \mathcal{V}} x_{ij} \beta_j \right) &= 1, \quad i \in \mathcal{E}. \end{aligned}$$

This system consists of $|\mathcal{E}|+|\mathcal{V}|+1$ equations and $|\mathcal{E}|+|\mathcal{V}|+1$ unknowns: β_0, β_j $(j \in \mathcal{V})$, α_i^* $(i \in \mathcal{E})$. Therefore, using the same argument as in the proof of Theorem 2.2, one can show the solution (β_0, β) is piecewise linear in D (or $1/\lambda_2$). □

For lack of space, we omit the details of the algorithms that compute the whole solution paths $\beta_{\lambda_2}(\lambda_1)$ (when λ_2 is fixed) and $\beta_{\lambda_1}(\lambda_2)$ (when λ_1 is fixed) of the DrSVM. We refer the readers to [22]. When λ_2 is fixed, the basic idea of our algorithm is to start with $s = 0$ (or equivalently $\lambda_1 = \infty$), find the right derivatives of β_0 and β_j with respect to s, increase s and move the solution along the right derivative direction until an event happens (the asterisk points in Figure 2.5), then adjust the linear system (2.19) – (2.22), and find out the new right derivatives. The algorithm stops when no further event will happen. The algorithm when λ_1 is fixed functions in a similar manner (the right panel of Figure 2.5).

Computational Complexity

The major computational cost is associated with solving the linear system (2.19) – (2.22) at each step, which involves $|\mathcal{E}| + 2$ equations and unknowns (after plugging (2.19) in (2.21) and (2.22)). Solving such a system involves $O(|\mathcal{E}|^3)$ computations. However, for any two consecutive steps, the two 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 the inverse updating/downdating. The computation of $\Delta\beta_j/\Delta s$ in (2.19) requires $O(|\mathcal{E}|\cdot|\mathcal{V}|)$ computations after getting $\Delta\alpha_i/\Delta s$. Notice due to the nature of (2.19) – (2.22), $|\mathcal{E}|$ 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 any 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 into 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 L_1-norm SVM and 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.

Simulation

We first consider the scenario when all input variables are independent. The "+" class has a normal distribution with mean and covariance

$$\begin{aligned} \mu_+ &= (\underbrace{0.5, \ldots, 0.5}_{5}, \underbrace{0, \ldots, 0}_{p-5})^T, \\ \Sigma &= I_{p\times p}. \end{aligned}$$

The "−" class has a similar distribution except that

$$\mu_- = (\underbrace{-0.5, \ldots, -0.5}_{5}, \underbrace{0, \ldots, 0}_{p-5})^T.$$

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 $n > p$ case and the $n \ll p$ case. In the $n > p$ case, we generate $100 = 50 + 50$ training data, each input x_i is a $p = 10$ dimensional vector; in the $n \ll p$ case, we generate $50 = 25 + 25$ training data, each input x_i is a $p = 300$ dimensional vector. We compare the L_2-norm SVM, the L_1-norm SVM, and the DrSVM. We use 200 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 for 30 times. The means of the prediction errors and the corresponding standard errors (in parentheses) are summarized in Table 2.1. 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.

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_{signal} = number of selected relevant variables
- q_{noise} = number of selected noise variables

The results are in Table 2.2. 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 1 on average) and remove most of the irrelevant variables.

Table 2.1. 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	100	10	5	0.145 (0.007)
L_1 SVM				0.142 (0.008)
DrSVM				0.139 (0.005)
L_2 SVM	50	300	5	0.323 (0.018)
L_1 SVM				0.199 (0.031)
DrSVM				0.178 (0.021)

Table 2.2. Comparison of variable selection when all input variables are independent. p_0 is the number of relevant variables. q_{signal} is the number of selected relevant variables. q_{noise} is the number of selected noise variables.

	n	p	p_0	q_{signal}	q_{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)

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

$$\begin{aligned} \mu_+ &= (\underbrace{1,\ldots,1}_{5},\underbrace{0,\ldots,0}_{p-5})^T, \\ \Sigma &= \begin{pmatrix} \Sigma^*_{5\times 5} & 0_{5\times(p-5)} \\ 0_{(p-5)\times 5} & I_{(p-5)\times(p-5)} \end{pmatrix}, \end{aligned}$$

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

$$\mu_- = (\underbrace{-1,\ldots,-1}_{5},\underbrace{0,\ldots,0}_{p-5})^T.$$

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.

Again, we consider both the $n > p$ case and the $n \ll p$ case. 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 for 30 times. The result for the prediction errors are shown in Table 2.3. 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 2.4 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 2.3. 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	100	10	5	0.142 (0.003)
L_1 SVM				0.144 (0.003)
DrSVM				0.140 (0.001)
L_2 SVM	50	300	5	0.186 (0.012)
L_1 SVM				0.151 (0.007)
DrSVM				0.139 (0.004)

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,

Table 2.4. Comparison of variable selection when the relevant variables are highly correlated. p_0 is the number of relevant variables. q_{signal} is the number of selected relevant variables. q_{noise} is the number of selected noise variables.

	n	p	p_0	q_{signal}	q_{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)

except that

$$\begin{aligned}
\mu_+ &= (\underbrace{1,\ldots,1}_{5},\underbrace{0,\ldots,0}_{p-5})^T,\\
\mu_- &= (\underbrace{-1,\ldots,-1}_{5},\underbrace{0,\ldots,0}_{p-5})^T,\\
\Sigma^* &= \begin{pmatrix} 1 & 0.8 & 0.8^2 & 0.8^3 & 0.8^4 \\ 0.8 & 1 & 0.8 & 0.8^2 & 0.8^3 \\ 0.8^2 & 0.8 & 1 & 0.8 & 0.8^2 \\ 0.8^3 & 0.8^2 & 0.8 & 1 & 0.8 \\ 0.8^4 & 0.8^3 & 0.8^2 & 0.8 & 1 \end{pmatrix}.
\end{aligned}$$

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 β_2, β_3 and β_4 are small compared with β_1 and β_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 for 30 times. The results are summarized in Table 2.5 and 2.6. As we can see, the DrSVM still dominates the L_1-norm SVM in terms of identifying relevant variables.

Microarray Analysis

In this section, we apply the L_1-norm SVM and 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 ([9], [15]). However, one weakness of the L_2-norm SVM is that it only

Table 2.5. 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	100	10	5	0.128 (0.008)
L_1 SVM				0.117 (0.004)
DrSVM				0.115 (0.003)
L_2 SVM	50	300	5	0.212 (0.022)
L_1 SVM				0.125 (0.010)
DrSVM				0.120 (0.006)

Table 2.6. 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. 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)

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 group of genes that share the same biological pathway, which have correlated expression levels. The DrSVM naturally overcomes these difficulties, and achieves the goals of classification of patients and (group) selection of genes simultaneously.

We use a leukemia dataset [8] to illustrate the point. This dataset consists of 38 training data and 34 test data of 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 2.7. 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 2.6 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 selected genes). Clear separation of the two classes is evident. Roughly speaking, the top set of genes over-express for ALL and under-express for AML; vice versa for the bottom set of genes.

Table 2.7. Results on the Leukemia Dataset

	CV Error	Test Error	# of Genes
Golub et al.	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

5. Conclusion

We have applied the L_1-norm penalty and the elastic-net penalty to the hinge loss, and proposed the L_1-norm SVM and the DrSVM methods for classification problems. These methods are especially useful with high dimensional data, with respect to effectively removing irrelevant variables and identifying relevant variables. Compared with 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 facilitate selection of the tuning parameters.

There are other interesting directions in which the SVM can be extended:

- **Huberized SVMs** The algorithm proposed in Section 3 is efficient. However, when both n and p are large, the initial derivative

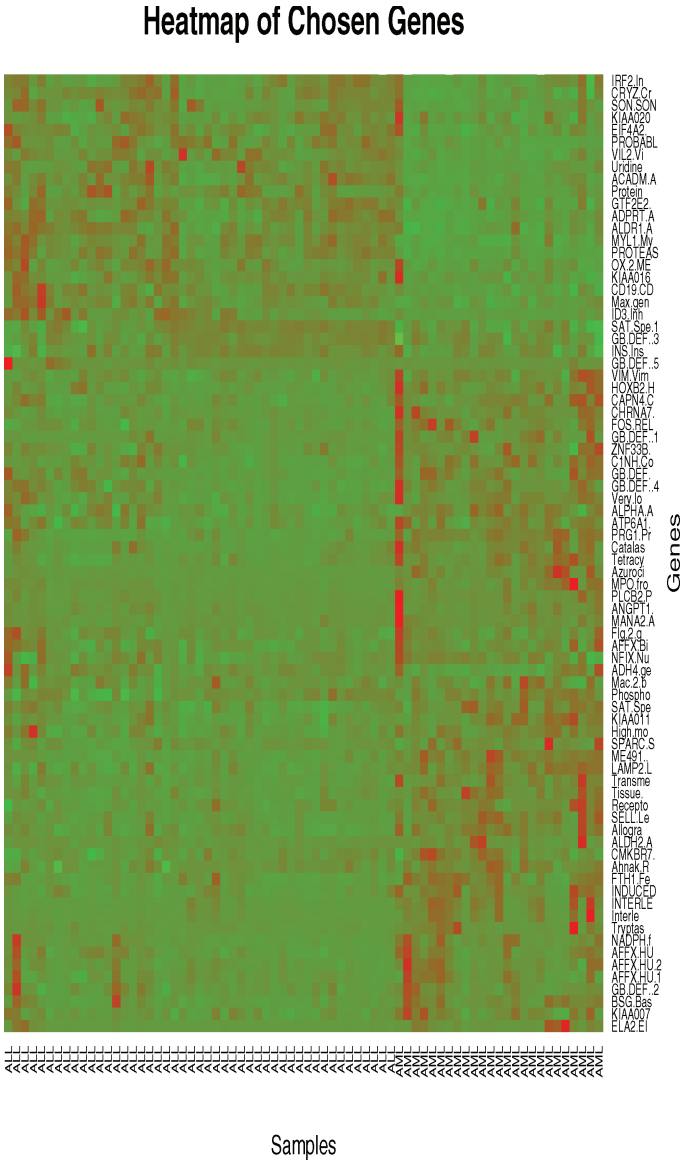

Figure 2.6. Heatmap of the selected 78 genes. We have ordered the genes by hierarchical clustering, and similarly for all $38 + 34$ samples.

of the path may require substantial computational efforts. This is due the fact that the hinge loss function is not differentiable at the point $yf = 1$. So the question is how one can modify the hinge loss to improve the computational efficiency? We consider to replace the hinge loss with the *Huberized hinge loss* [17]. The Huberized hinge loss is defined as

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

where $\delta < 1$. Figure 2.7 compares the Huberized hinge loss and the hinge loss. As we can see, the Huberized hinge loss is differentiable everywhere. The Huberized hinge loss also has a similar shape as the hinge loss; therefore, one can expect the prediction performance of the Huberized hinge loss would be similar to the hinge loss.

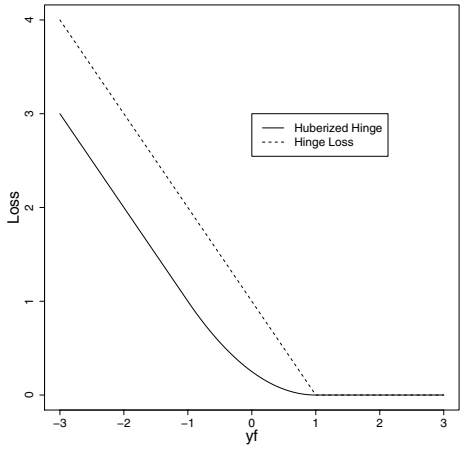

Figure 2.7. The hinge loss and the Huberized hinge loss (with $\delta = -1$). The Huberized hinge loss is differentiable everywhere, and has a similar shape as the hinge loss.

- **Factor selection in the SVM** In some problems, the input features are generated by factors, and the model is best interpreted in terms of significant factors. For example, a categorical factor is often represented by a set of dummy variables. Another familiar example is the use of a set of basis functions of a continuous variable in function estimation, e.g., univariate splines in generalized additive models [10]. As one can see, variable selection results can be directly translated to factor selection. On the other hand,

with the presence of the factor-feature hierarchy, a factor is considered as irrelevant unless its child features are all excluded from the fitted model, which we call *simultaneous elimination*.

To enforce the simultaneous elimination, [25] propose the F_∞-norm SVM which penalizes the empirical hinge loss by the sum of the L_∞ norm of factors. Here is how it works. Suppose that the p input variables can be further segmented into G groups without overlap, and the variables in the gth group are generated by factor F_g. Let $S_g = \{j : x_j \in \text{group g}\}$. Then $\{1, \ldots, p\} = \cup_{g=1}^{G} S_g$ and $S_g \cap S_{g'} = \emptyset$, $\forall g \neq g'$. We denote $x_{(g)} = (\cdots x_j \cdots)_{j \in S_g}^T$ and $\beta_{(g)} = (\cdots \beta_j \cdots)_{j \in S_g}^T$, where β is the coefficient vector in the classifier $(\beta_0 + x^T\beta)$ for separating class "1" and class "-1". For convenience, we write

$$\beta_0 + x^T\beta = \beta_0 + \sum_{g=1}^{G} x_{(g)}^T \beta_{(g)},$$

and we define the infinity norm of F_g as follows

$$\|F_g\|_\infty = \|\beta_{(g)}\|_\infty = \max_{j \in S_g} \{|\beta_j|\} \ .$$

Now given the training samples $(x_1, y_1), \ldots, (x_n, y_n)$, we can write the F_∞-norm SVM as the following

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} \left[1 - y_i \left(\beta_0 + \sum_{g=1}^{G} x_{i,(g)}^T \beta_{(g)} \right) \right]_+ + \lambda \sum_{g=1}^{G} \|\beta_{(g)}\|_\infty \ .$$

Notice that if $\|\beta_{(g)}\|_\infty$ is equal to zero, the whole factor F_g is removed from the fitted model.

Acknowledgments

The authors wish to thank Trevor Hastie, Saharon Rosset, Rob Tibshirani and Li Wang for their help. Zhu is partially supported by grant DMS-0505432 from the National Science Foundation.

References

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

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

[3] Chen, S., Donoho, D. & Saunders, M. (1998) Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing* **20**, 33–61.

[4] Donoho, D., Johnstone, I., Kerkyachairan, G. & Picard, D. (1995) Wavelet shrinkage: asymptopia? (with discussion). *Journal of the Royal Statistical Society: Series B* **57**, 201–337.

[5] Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004) Least angle regression (with discussion). *Annals of Statistics* **32**, 407–499.

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

[7] Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004) Discussion of "Consistency in boosting" by W. Jiang, G. Lugosi, N. Vayatis and T. Zhang. *Annals of Statistics* **32**, 102–107.

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

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

[10] Hastie, T. & Tibshirani, R. (1990) *Generalized Additive Models*. Chapman and Hall. London.

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

[12] Hoerl, A. & Kennard, R. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**, 55–67.

[13] Lin, Y. (2002) Support vector machine and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6**, 259–275.

[14] Mallat, S. & Zhang, Z. (1993) Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing* **41**, 3397–3415.

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

[16] Ng, A. (2004) Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance. In *International Conference on Machine Learning*, Morgan Kaufmann, Banff, Canada.

[17] Rosset, S. & Zhu, J. (2004) Piecewise linear regularized solution paths. Technical Report #431, Department of Statistics, University of Michigan.

[18] Rosset, S., Zhu, J. & Hastie, T. (2004) Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research* **5**, 941–973.

[19] Song, M., Breneman, C., Bi, J., Sukumar, N., Bennett, K., Cramer, S. & Tugcu, N. (2002) Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of Chemical Information and Computer Sciences* **42**, 1347–1357.

[20] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* **58**, 267–288.

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

[22] Wang, L., Zhu, J. & Zou, H. (2006) The doubly regularized support vector machine. *Statistica Sinica*: Special issue on machine learning and data mining. In press.

[23] Zhu, J., Rosset, S., Hastie, T. & Tibshirani, R. (2004) 1-norm SVMs. In *Neural Information Processing Systems* **16**.

[24] Zou, H. & Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* **67**, 301–320.

[25] Zou, H. & Yuan, M. (2005) The F_∞-norm Support Vector Machine. Technical Report #646, School of Statisics, University of Minnesota.

http://www.springer.com/978-3-540-36121-3

Trends in Neural Computation

Chen, K.; Wang, L. (Eds.)
2007, X, 512 p. 159 illus., 18 illus. in color., Hardcover
ISBN: 978-3-540-36121-3