Statistica Sinica **18**(2008), 379-398

THE F_∞-NORM SUPPORT VECTOR MACHINE

Hui Zou and Ming Yuan

University of Minnesota and Georgia Institute of Technology

Abstract: In this paper we propose a new support vector machine (SVM), the F_∞-norm SVM, to perform automatic factor selection in classification. The F_∞-norm SVM methodology is motivated by the feature selection problem in cases where the input features are generated by factors, and the model is best interpreted in terms of significant factors. This type of problem arises naturally when a set of dummy variables is used to represent a categorical factor and/or a set of basis functions of a continuous variable is included in the predictor set. In problems without such obvious group information, we propose to first create groups among features by clustering, and then apply the F_∞-norm SVM. We show that the F_∞-norm SVM is equivalent to a linear programming problem and can be efficiently solved using standard techniques. Analysis on simulated and real data shows that the F_∞-norm SVM enjoys competitive performance when compared with the 1-norm and 2-norm SVMs.

Key words and phrases: F_∞ penalty, factor selection; feature selection, linear programming, L_∞ penalty, support vector machine.

1. Introduction

In the standard binary classification problem, one wants to predict the class labels based on a given vector of features. Let x denote the feature vector. The class labels, y, are coded as $\{1, -1\}$. A classification rule δ is a mapping from x to $\{1, -1\}$ such that a label $\delta(x)$ is assigned to the datum at x. Under the 0-1 loss, the misclassification error of δ is $R(\delta) = P(y \neq \delta(x))$. The smallest classification error is the Bayes error achieved by

$$\underset{c \in \{1,-1\}}{\operatorname{argmax}}\, p(y = c|x),$$

which is referred to as the Bayes rule.

The standard 2-norm support vector machine (SVM) is a widely used classification tool (Vapnik (1995) and Schölkopf and Smola (2002)). The popularity of the SVM is largely due to its elegant margin interpretation and highly competitive performance in practice. Let us first briefly describe the linear SVM. Suppose we have a set of training data $\{(x_i, y_i)\}_{i=1}^n$, where x_i is a vector with p

features, and the output $y_i \in \{1, -1\}$ denotes the class label. The 2-norm SVM finds a hyperplane $(x^T\beta + \beta_0)$ that creates the biggest margin between the training points for class 1 and -1 (Vapnik (1995) and Hastie, Tibshirani and Friedman (2001)):

$$\begin{aligned}
&\max_{\beta,\beta_0} \frac{1}{\|\beta\|_2} \qquad (1.1)\\
&\text{subject to } y_i(\beta_0 + x_i^T\beta) \geq 1 - \xi_i, \forall i\\
&\xi_i \geq 0, \quad \sum \xi_i \leq B,
\end{aligned}$$

where ξ_i are slack variables, and B is a pre-specified positive number that controls the overlap between the two classes. It can be shown that the linear SVM has an equivalent *loss* + *penalty* formulation (Wahba, Lin and Zhang (2000) and Hastie, Tibshirani and Friedman (2001)):

$$(\hat{\beta}, \hat{\beta}_0) = \arg\min_{\beta,\beta_0} \sum_{i=1}^{n} \Big[1 - y_i(x_i^T\beta + \beta_0)\Big]_+ + \lambda\|\beta\|_2^2, \qquad (1.2)$$

where the subscript "+" means the positive part ($z_+ = \max(z, 0)$). The loss function $(1-t)_+$ is called the hinge or SVM loss. Thus the 2-norm SVM is expressed as a quadratically regularized model fitting problem. Lin (2002) showed that, due to the unique property of the hinge loss, the SVM directly approximates the Bayes rule without estimating the conditional class probability, and the quadratic penalty helps control the model complexity to prevent over-fitting the training data.

Another important task in classification is to identify a subset of features which contribute most to classification. The benefit of feature selection is two-fold. It leads to parsimonious models that are often preferred in many scientific problems, and it is also crucial for achieving good classification accuracy in the presence of redundant features (Friedman, Hastie, Rosset, Tibshirani and Zhu (2004) and Zhu, Rosset, Hastie and Tibshirani (2004)). However, the 2-norm SVM classifier cannot automatically select input features, for all elements of $\hat{\beta}$ are typically non-zero. In the machine learning literature, there are several proposals for feature selection in the SVM. Guyon, Weston, Barnhill and Vapnik (2002) proposed the recursive feature elimination (RFE) method; Weston, Mukherjee, Chapelle, Pontil, Poggio and Vapnik (2001) and Grandvalet and Canu (2003) considered some adaptive scaling methods for feature selection in SVMs; Bradley and Mangasarian (1998), Song, Breneman, Bi, Sukumar, Bennett, Cramer and Tugcu (2002) and Zhu, Rosset, Hastie and Tibshirani (2004) considered the 1-norm SVM to accomplish the goal of automatic feature selection in the SVM.

In particular, the 1-norm SVM penalizes the empirical hinge loss by the lasso penalty (Tibshirani (1996)), thus the 1-norm SVM can be formulated in the same fashion as the 2-norm SVM:

$$(\hat{\beta}, \hat{\beta}_0) = \underset{\beta,\beta_0}{\arg\min} \sum_{i=1}^{n} \Big[1 - y_i(x_i^T\beta + \beta_0)\Big]_+ + \lambda\|\beta\|_1. \tag{1.3}$$

The 1-norm SVM shares many of the nice properties of the lasso. The L_1 (lasso) penalty encourages some of the coefficients to be zero if λ is appropriately chosen. Hence the 1-norm SVM performs feature selection through regularization. The 1-norm SVM has significant advantages over the 2-norm SVM when there are many noise variables (Zhu, Rosset, Hastie and Tibshirani (2004)). A study comparing the L_2 and L_1 penalties (Friedman, Hastie, Rosset, Tibshirani and Zhu (2004)) shows that the L_1 norm is preferred if the underlying true model is sparse, while the L_2 norm performs better if most of the predictors contribute to the response. Friedman, Hastie, Rosset, Tibshirani and Zhu (2004) further advocate the *bet-on-sparsity principle*; that is, procedures that do well in sparse problems should be favored.

Although the bet-on-sparsity principle often leads to successful models, the L_1 penalty may not always be the way to achieve this goal. Consider, for example, the cases of categorical predictors. A common practice is to represent the categorical predictor by a set of dummy variables. A similar situation occurs when we express the effect of a continuous factor as a linear combination of a set of basis functions, e.g., univariate splines in generalized additive models (Hastie and Tibshirani (1990)). In such problems it is of more interest to select the important factors than to understand how the individual derived variables explain the response. With the presence of the factor-feature hierarchy, a factor is considered as relevant if any one of its child features is active. Therefore all of a factor's child features have to be excluded in order to exclude the factor from the model. We call this *simultaneous elimination*. Although the 1-norm SVM can annihilate individual features, it oftentimes cannot perform the simultaneous elimination needed to discard a factor. This is largely due to the fact that no factor-feature information is used in (1.3). Generally speaking, if the features are penalized independently, simultaneous elimination is not guaranteed.

In this paper we propose a natural extension of the 1-norm SVM to account for such grouping information. We call the proposal an F_∞-norm SVM because it penalizes the empirical SVM loss by the sum of the factor-wise L_∞ norm. Owing to the nature of the L_∞ norm, the F_∞-norm SVM is able to simultaneously eliminate a given set of features, hence it is a more appropriate tool for factor selection than the 1-norm SVM.

Although our methodology is motivated by problems in which the predictors are naturally grouped, it can also be applied in other settings where the groupings are more loosely defined. We suggest first clustering the input features into groups, and then applying the F_∞-norm SVM. This strategy can be very useful when the predictors are a mixture of true and noise variables, quite common in applications. Clustering takes advantage of the mutual information among the input features, and the F_∞-norm SVM has the ability to perform group-wise variable selection. Hence the F_∞-norm SVM is able to outperform the 1-norm SVM in that it is more efficient in removing the noise features and keeping the true variables.

The rest of the paper is organized as follows. The F_∞-norm SVM methodology is introduced in Section 2. In Section 3 we show that the F_∞-norm SVM can be cast as a linear programming (LP) problem, and efficiently solved using the standard linear programming technique. In Sections 4 and 5 we demonstrate the utility of the F_∞-norm SVM using both simulation and real examples. Section 6 contains some concluding remarks.

2. Methodology

Before delving into the technical details, we define some notation. Consider the vector of input features $x = (\cdots, x^{(j)}, \cdots)$ where $x^{(j)}$ is the j-th input feature $1 \leq j \leq p$. Now suppose that the features are generated by G factors, $F_1, \ldots, F_G$. Let $S_g = \{j : x^{(j)} \text{ is generated by } F_g\}$. Clearly, $\cup_{g=1}^{G} S_g = \{1, \ldots, p\}$ and $S_g \cap S_{g'} = \emptyset$, $\forall g \neq g'$. Write $x_{(g)} = (\cdots x^{(j)} \cdots)^T_{j \in S_g}$ and $\beta_{(g)} = (\cdots \beta_j \cdots)^T_{j \in S_g}$, where β is the coefficient vector in the classifier $(x^T\beta + \beta_0)$ for separating class 1 and class -1. With such notation,

$$x^T\beta + \beta_0 = \sum_{g=1}^{G} x^T_{(g)}\beta_{(g)} + \beta_0. \tag{2.1}$$

Now define the infinity norm of F_g as

$$\|F_g\|_\infty = \|\beta_{(g)}\|_\infty = \max_{j \in S_g}\{|\beta_j|\}. \tag{2.2}$$

Given n training samples $\{(x_i, y_i)\}_{i=1}^n$, the F_∞-norm SVM solves

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \left[1 - y_i\left(\sum_{g=1}^{G} x^T_{i,(g)}\beta_{(g)} + \beta_0\right)\right]_+ + \lambda \sum_{g=1}^{G} \|\beta_{(g)}\|_\infty. \tag{2.3}$$

Note that the empirical hinge loss is penalized by the sum of the infinity norm of factors with a regularization parameter λ. The solution to (2.3) is denoted by

$\hat{\beta}$ and $\hat{\beta}_0$. The fitted classifier is $\hat{f}(x) = \hat{\beta}_0 + x^T\hat{\beta}$, and the classification rule is $sign(\hat{f}(x))$.

The F_∞-norm SVM has the ability to do automatic factor selection. If the regularization parameter λ is appropriately chosen, some $\hat{\beta}_{(g)}$ will be exact zero. Thus the goal of simultaneous elimination of grouped features is achieved via regularization. This nice property is due to the singular nature of the infinity norm: $\|\beta_{(g)}\|_\infty$ is not differentiable at $\beta_{(g)} = 0$. As pointed out in Fan and Li (2001), singularity (at the origin) of the penalty function plays a central role in automatic feature selection. This property of the L_∞ norm has previously been exploited by Turlach, Venables and Wright (2004) to select a *common* subset of predictors to model multiple regression responses.

When each individual feature is considered as a group, the F_∞-norm SVM reduces to the 1-norm SVM, but (2.3) differs from (1.3) because the L_1 norm contains no group information. Therefore, we consider the F_∞-norm SVM as a generalization of the 1-norm SVM by incorporating the factor-feature hierarchy in the SVM machinery.

The L_∞-norm is a special case of the F_∞-norm if we put all predictors into a single group. Then we can consider the L_∞-norm SVM

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \Big[1 - y_i(x_i^T\beta + \beta_0)\Big]_+ + \lambda\Big(\max_j |\beta_j|\Big). \tag{2.4}$$

The L_∞-norm penalty is a direct approach to controlling the variability of the estimated coefficients. Our experience with the L_∞-norm SVM indicates that it may perform quite well in terms of classification accuracy, but all the β_js are typically nonzero. The F_∞-norm penalty mitigates this problem by dividing the predictors into several smaller groups. In later sections, we present some empirical results suggesting that the F_∞ oftentimes outperforms 1-norm and 2-norm SVMs in the presence of factors.

In the following theorem we show that the F_∞ SVM enjoys the so-called margin maximizing property.

Theorem 1. *Assume the data* $\{(x_i, y_i)\}_{i=1}^n$ *are separable. Let* $\hat{\beta}(\lambda)$ *be the solution to* (2.3).

(a) $\lim_{\lambda\to 0} \min_i y_i x_i^T \hat{\beta}(\lambda) = 1$.

(b) *The limit of any converging subsequence of* $(\hat{\beta}(\lambda))/(\|\hat{\beta}(\lambda)\|_{F_\infty})$ *as* $\lambda \to 0$ *is an* F_∞ *margin maximizer. If the margin maximizer is unique, then*

$$\lim_{\lambda\to 0} \frac{\hat{\beta}(\lambda)}{\|\hat{\beta}(\lambda)\|_{F_\infty}} = \underset{\beta:\|\beta\|_{F_\infty}=1}{\operatorname{argmax}} \Big\{ \min_i y_i x_i^T \beta \Big\}.$$

Theorem 1 considers the limiting case of the F_∞-norm SVM classifier when the regularization parameter approaches zero. It extends a similar result for the 2-norm SVM (Rosset and Zhu (2003)). The proof of Theorem 1 is in the appendix. The margin maximization property is theoretically interesting because it is related to the generalization error analysis based on the margin. Generally speaking, the larger the margin, the smaller the upper bound on the generalization error. Theorem 1 also prohibits any potential radical behavior of the F_∞-norm SVM even for $\lambda \to 0$ (no regularization), which helps to prevent severe over-fitting. Of course, similar to the case of the 1-norm and 2-norm SVMs, the regularized F_∞-norm SVM often performs better than its non-regularized version.

3. Algorithm

In this section we show that the optimization problem (2.3) is equivalent to a linear programming (LP) problem, and can therefore be solved using standard LP techniques. The computational efficiency makes the F_∞-norm SVM an attractive choice in many applications.

Note that (2.3) can be viewed as the Lagrange formulation of the constrained optimization problem

$$\arg\min_{\beta,\beta_0} \sum_{g=1}^{G} \|\beta_{(g)}\|_\infty \tag{3.1}$$

subject to

$$\sum_{i=1}^{n} \left[1 - y_i\left(\sum_{g=1}^{G} x_{i,(g)}^T \beta_{(g)} + \beta_0\right)\right]_+ \leq B \tag{3.2}$$

for some B. There is a one-one mapping between λ and B such that the problem at (3.1) and (3.2) and the one at (2.3) are equivalent. To solve (3.1) and (3.2) for a given B, we introduce a set of slack variables

$$\xi_i = \left[1 - y_i\left(\sum_{g=1}^{G} x_{i,(g)}^T \beta_{(g)} + \beta_0\right)\right]_+ \quad i = 1, 2, \ldots, n. \tag{3.3}$$

With such notation, the constraint in (3.2) can be rewritten as

$$y_i(\beta_0 + x_i^T \beta) \geq 1 - \xi_i \text{ and } \quad \xi_i \geq 0 \quad \forall i, \tag{3.4}$$

$$\sum_{i=1}^{n} \xi_i \leq B. \tag{3.5}$$

To further simplify the above formulation, we introduce a second set of slack variables

$$M_g = \|\beta_{(g)}\|_\infty = \max_{j \in S_g}\{|\beta_j|\}. \tag{3.6}$$

Now the objective function in (3.1) becomes $\sum_{g=1}^{G} M_g$, and we need a set of new constraints

$$|\beta_j| \le M_g \quad \forall j \in S_g \quad \text{and} \quad g = 1, \ldots, G. \tag{3.7}$$

Finally, write $\beta_j = \beta_j^+ - \beta_j^-$ where β_j^+ and β_j^- denote the positive and negative parts of β_j, respectively. Then (3.1) and (3.2) can be equivalently expressed

$$\min_{\beta,\beta_0} \sum_{g=1}^{G} M_g \tag{3.8}$$

subject to

$$\begin{aligned} y_i(\beta_0^+ - \beta_0^- + x_i^T(\beta^+ - \beta^-)) \ge 1 - \xi_i, &\quad \xi_i \ge 0 \ \forall i \\ \sum_{i=1}^{n} \xi_i \le B, & \\ \beta_j^+ + \beta_j^- \le M_g &\quad \forall j \in S_g \quad g = 1, \ldots, G, \\ \beta_j^+ \ge 0, \quad \beta_j^- \ge 0 &\quad \forall j = 0, 1, \ldots, p. \end{aligned}$$

This LP formulation of the F_∞-norm SVM is similar to the margin-maximization formulation of the 2-norm SVM.

It is worth pointing out that the above derivation also leads to an alternative LP formulation of the F_∞-norm SVM:

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \xi_i + \lambda \sum_{g=1}^{G} M_g \tag{3.9}$$

subject to

$$\begin{aligned} y_i(\beta_0^+ - \beta_0^- + x_i^T(\beta^+ - \beta^-)) \ge 1 - \xi_i &\quad \xi_i \ge 0 \ \ \forall i, \\ \beta_j^+ + \beta_j^- \le M_g &\quad \forall j \in S_g \quad g = 1, \ldots, G, \\ \beta_j^+ \ge 0, \quad \beta_j^- \ge 0 &\quad \forall j = 0, 1, \ldots, p. \end{aligned}$$

Note that (2.3), (3.8) and (3.9) are three equivalent formulations of the F_∞-norm SVM.

For any given tuning parameter (B or λ), we can efficiently solve the F_∞-norm SVM using the standard LP technique. In applications, it is often important to select a good tuning parameter such that the generalization error of the fitted F_∞-norm SVM is minimized. For this purpose, we can run the F_∞-norm SVM for a grid of tuning parameters, and choose the one that minimizes the K-fold cross-validation score or the test error on an independent validation data set.

4. Simulation

In this section we report on simulation experiments to compare the F_∞-norm SVM with the standard 2-norm SVM and the 1-norm SVM.

In the first set of simulations, we focused on the cases where the predictors are naturally grouped. This situation arises when some of the predictors are latent variables describing the same categorical factor or polynomial effects of the same continuous variable. We considered three simulation models described below.

Model I. Fifteen latent variables $Z_1, \ldots, Z_{15}$ were first simulated according to a centered multivariate normal distribution with covariance between Z_i and Z_j being $0.5^{|i-j|}$. Then Z_i is trichotomized as $0, 1, 2$ if it is smaller than $\Phi^{-1}(1/3)$, larger than $\Phi^{-1}(2/3)$ or in between. The response Y was then simulated from a logisitic regression model with the probability of success being the logit of

$$7.2I(Z_1\!=\!1)–4.8I(Z_1\!=\!0)+4I(Z_3\!=\!1)+2I(Z_3\!=\!0)+4I(Z_5\!=\!1)+4I(Z_5\!=\!0)–4,$$

where $I(\cdot)$ is the indicator function. This model has 30 predictors and 15 groups. The true features are six predictors in three groups (Z_1,Z_3 and Z_5). The Bayes error is 0.095.

Model II. In this example, both main effects and second order interactions were considered. Four categorical factors Z_1, Z_2, Z_3 and Z_4 were first generated as in (I). The response Y was again simulated from a logisitic regression model with the probability of success being the logit of

$$\begin{aligned}&3I(Z_1\!=\!1) + 2I(Z_1\!=\!0) + 3I(Z_2\!=\!1) + 2I(Z_2\!=\!0) + I(Z_1\!=\!1, Z_2\!=\!1)\\&\quad+1.5I(Z_1\!=\!1, Z_2\!=\!0) + 2I(Z_1\!=\!0, Z_2\!=\!1) + 2.5I(Z_1\!=\!0, Z_2\!=\!0) - 4.\end{aligned}$$

In this model there are 32 predictors and 10 groups. The ground truth uses eights predictors in three groups (Z_1, Z_2 and Z_1Z_2 interaction). The Bayes error is 0.116.

Model III. This example concerns additive models with polynomial components. Eight random variables $Z_1, \ldots, Z_8$ and W were independently generated from a standard normal distribution. The covariates were $X_i = (Z_i + W)/\sqrt{2}$. The response followed a logistic regression model with the probability of success being the logit of

$$(X_3^3 + X_3^2 + X_3) + \Big(\frac{1}{3}X_6^3 - X_6^2 + \frac{2}{3}X_6\Big).$$

In this model we have 24 predictors in eight groups. The ground truth involves six predictors in two groups (Z_1 and Z_2). The Bayes error is 0.188.

For each of the above three models, 100 observations were simulated as the training data, and another 100 observations were collected for tuning the regularization parameter for each of the three SVMs. To test the accuracy of the classification rules, we also independently generated $10,000$ observations as a test set. Since the Bayes error is the lower bound for the classification accuracy of any classifier, when evaluating a classifier δ it is reasonable to use its relative misclassification error

$$\mathrm{RME}(\delta) = \frac{\mathrm{Err}(\delta)}{\text{Bayes Error}}.$$

Table 4.1 reports the mean classification error and its standard error (in parentheses) for each method and each model, averaged over 100 runs. Several observations can be made from Table 4.1. In all examples, the F_∞ SVM outperforms the other two methods in terms of classification error. We also see that the F_∞ SVM tends to be more stable than the the other two. Table 4.2 documents the number of factors selected by the F_∞-norm and 1-norm SVMs. It indicates that the F_∞-norm SVM tends to select fewer factors than the 1-norm SVM.

As mentioned in the introduction, the F_∞ SVM can also be applied to problems where the natural grouping information is either hidden or not available. For example, the sonar data considered in Section 5.2 contains 60 continuous predictors, but it is not clear how these 60 predictors are grouped. To tackle this issue, we suggest first grouping the features by clustering and then applying the

Table 4.1. Simulation models I, II and III: compare the accuracy of different SVMs.

	Model I	Model II	Model III
Bayes rule	0.095	0.116	0.188
F_∞-norm	0.120 (0.002)	0.119 (0.010)	0.215 (0.002)
1-norm	0.133 (0.026)	0.142 (0.034)	0.223 (0.003)
2-norm	0.151 (0.019)	0.130 (0.025)	0.228 (0.002)
RME(F_∞)	1.263 (0.021)	1.026 (0.086)	1.144 (0.011)
RME($L1$)	1.400 (0.274)	1.224 (0.293)	1.186 (0.016)
RME($L2$)	1.589 (0.200)	1.121 (0.216)	1.213 (0.011)

Table 4.2. Simulation models I, II and III: the number of factors selected by the F_∞-norm and 1-norm SVMs.

	Model I	Model II	Model III
True	3	3	2
F_∞-norm	11.46 (0.35)	3.66 (0.29)	6.70 (0.16)
1-norm	11.94 (0.34)	4.33 (0.22)	6.67 (0.13)

F_∞ SVM. To demonstrate this strategy, we considered a fourth simulation model.

Model IV. Two random variables Z_1 and Z_2 were independently generated from a standard normal distribution. In addition, 60 standard normal variables $\{\epsilon_i\}$ were generated. The predictors X were

$$\begin{aligned} X_i &= Z_1 + 0.5\epsilon_i,\ i = 1, \ldots, 20, \\ X_i &= Z_2 + 0.5\epsilon_i, i = 21, \ldots, 40, \\ X_i &= \epsilon_i, \qquad\quad i = 41, \ldots, 60. \end{aligned}$$

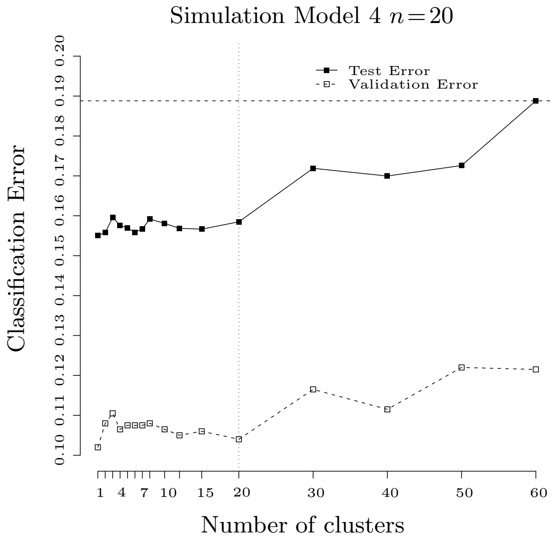
The response followed a logistic regression model with the probability of success being the logit of $4Z_1 + 3Z_2 + 1$. The Bayes error is 0.109.

We simulated 20 (100) observations as the training data, and another 20 (100) observations as the validation data for tuning the three SVMs. An independent set of 10,000 observations were simulated to compute the test error. We repeated the simulation 100 times.

As the oracle who designed the above model, we know that there are 22 groups of predictors. The first 20 predictors form a first group in which the pairwise correlation within the group is 0.8. Likewise, predictors 20-40 form a second group in which the pairwise correlation is also 0.8. The first 40 predictors are considered relevant. The remaining 20 predictors form 20 individual groups of size one, for they are independent noise features. We could fit a F_∞ SVM using the oracle group information, this is not available in applications. A practical strategy is to use the observed data to find the groups on which the F_∞ SVM is to be built. In this work we employed hierarchical clustering to cluster the predictors into k clusters (groups), where the sample correlations were used to measure the closeness of predictors. For given k clusters (groups) we can fit a F_∞ SVM. Thus in this procedure we actually have two tuning parameters: the number of clusters, and B. The validation set was used to find a good choice of (k, B).

Figure 4.1 displays the classification error of the F_∞ SVM using different numbers of clusters (k). Based on the validation error curve we see that the optimal k is 20 and 12 for $n = 20$ and $n = 100$, respectively. It is interesting to see that for any value of k, the classification accuracy of the corresponding F_∞

SVM is better than that of the 1-norm SVM. As shown in Table 4.3, the F_∞-norm SVM via clustering performs almost identically to the F_∞-norm SVM using the oracle group information. In terms of classification accuracy, the F_∞-norm SVM dominates the 1-norm SVM and the 2-norm SVM by a good margin.



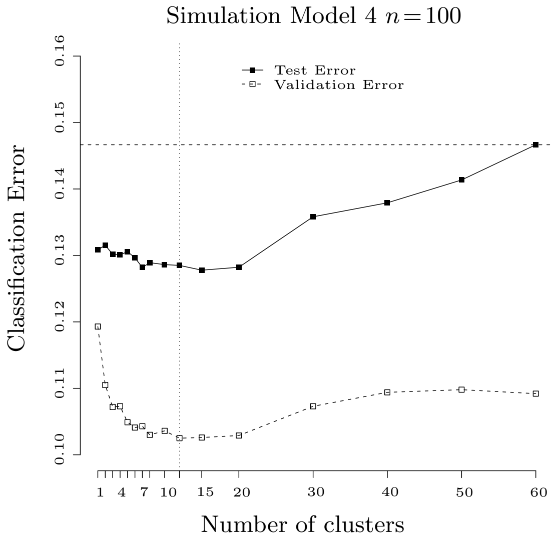


Figure 4.1. Simulation model IV: the validation error and test error vs. the number of clusters (k). For each k we found the value of $B(k)$ giving the smallest validation error. Then the pair of $(k, B(k))$ was used in computing the test error. The broken horizontal lines indicate the test error of the 1-norm SVM. Note that in both plots the F_∞ SVM uniformly dominates the 1-norm SVM regardless the value of k.

Table 4.3. Simulation model IV: compare different SVMs. F_∞-norm (oracle) is the F_∞-norm SVM using the oracle group information. NSG=Number of Selected Groups, and NSP=Number of Selected Predictors. The F_∞-norm SVM is significantly more accurate than both the 1-norm and 2-norm SVMs. The ground truth is that 40 predictors in two groups are true features. The 1-norm SVM severely under-selected the model. In contrast, the F_∞-norm SVM can almost identify the ground truth even when $n = 20$.

Model IV: Bayes Error = 0.109			
Method	Test Error	NSG	NSP
	$n = 20$		
F_∞-norm (k=20)	0.158 (0.004)	2.01 (0.03)	37.99 (0.48)
1-norm	0.189 (0.004)	7.51 (0.25)	7.51 (0.25)
2-norm	0.164 (0.004)		
F_∞-norm (oracle)	0.160 (0.004)	1.97 (0.02)	39.67 (0.33)
RME(F_∞-norm)	1.450 (0.037)		
RME(1-norm)	1.734 (0.037)		
RME(2-norm)	1.505 (0.037)		
	$n = 100$		
F_∞-norm (k=12)	0.129 (0.001)	2.01 (0.01)	40.64 (0.093)
1-norm	0.147 (0.001)	12.21 (0.45)	12.21 (0.45)
2-norm	0.140 (0.001)		
F_∞-norm (oracle)	0.125 (0.001)	2.01 (0.01)	40.09 (0.057)
RME(F_∞-norm)	1.174 (0.009)		
RME(1-norm)	1.349 (0.009)		
RME(2-norm)	1.284 (0.009)		

Furthermore, the F_∞-norm SVM almost identified the ground truth, while the 1-norm SVM severely under-selected the model. Consider the $n = 20$ case. Note that the sample size is even less than the number of true predictors. The F_∞-norm SVM can still select about 40 predictors. In none of the 100 simulations did the 1-norm SVM select all the relevant features. The 1-norm SVM also selected a few noise variables. The probability that the 1-norm SVM discarded all the noise predictors is about 0.42 when $n = 20$ and 0.62 when $n = 100$. Figure 4.2 depicts the probability of perfect variable selection by the F_∞-norm SVM as a function of the number of clusters. Perfect variable selection means that all the true features are selected and all the noise features are eliminated. It is interesting to see that the F_∞-norm SVM can have pretty high probabilities of perfect selection, even when the sample size is less than the number of true predictors. Note that the 1-norm SVM can never select all the true predictors whenever the sample size is less than the number of true predictors, a fundamental difference between the F_∞ penalty and the L_1 penalty.

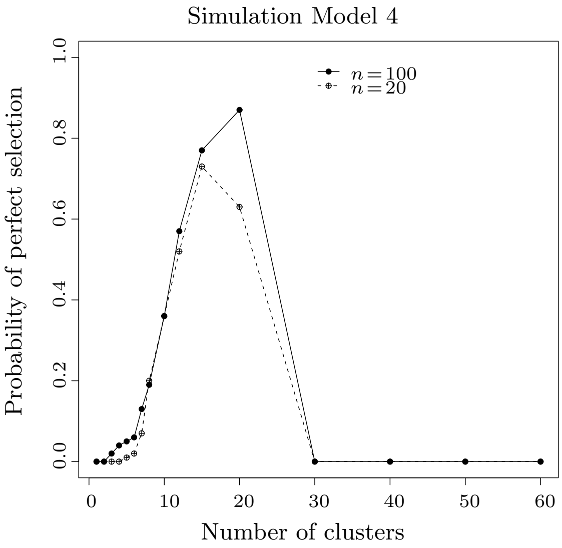


Figure 4.2. Simulation model IV. The probability of perfect selection by the F_∞-norm SVM as functions of the number of clusters.

5. Examples

The simulation study has demonstrated the promising advantages of the F_∞-norm SVM. We now examine the performance of the F_∞-norm SVM and the 1-norm and 2-norm SVMs on two benchmark data sets, obtained from UCI Machine Learning Repository (Newman and Merz (1998)).

5.1. Credit approval data

The credit approval data contains 690 observations with 15 attributes. There are 307 observations in class "+" and 383 observations in class "-". This dataset is interesting because there is a good mix of attributes – six continuous and nine categorical. Some categorical attributes have a large number of values and some have a small number of values. Thus, when they are coded by dummy variables, we have some large groups as well as some small groups. Using the dummy variables to represent the categorical attributes, we end up with 37 predictors which naturally form 10 groups, as displayed in Table 5.4.

We randomly selected 1/2 of the data for training, 1/4 data for tuning, and the remaining 1/4 as the test set. We repeated the randomization 10 times and now report the average test error of each method and its standard error. Table 5.5 summarizes the results. The F_∞-norm SVM appears to be the most accurate classifier. The variable/factor selection results look very interesting. The F_∞ and 1-norm SVMs selected similar numbers of predictors (about 20). However, in this example, model sparsity is best interpreted in terms of the selected factors, for we wish to know which categorical attributes are effective. When considering factor selection, we see that the F_∞-norm SVM provided a much sparser model than the 1-norm SVM.

Table 5.4. The natural groups in the credit approval data. The first group includes the six numeric predictors. The other nine groups represent the nine categorical factors, where the predictors are defined using dummy variables.

group	predictors in the group
1	$(1, 2, 3, 4, 5, 6)$
2	(7)
3	$(8, 9)$
4	$(10, 11)$
5	$(12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24)$
6	$(25, 26, 27, 28, 29, 30, 31, 32)$
7	(33)
8	(34)
9	(35)
10	$(36, 37)$

Table 5.5. Credit approval data: compare different SVMs. NSG=Number of Selected Groups, and NSP=Number of Selected Predictors.

	Test Error	NSP	NSG
F_∞-norm	0.128 (0.008)	19.70 (0.99)	3.00 (0.16)
1-norm	0.132 (0.007)	20.40 (1.35)	7.70 (0.45)
2-norm	0.135 (0.008)		

We rebuilt the F_∞-norm SVM classifier using the entire data set. The selected factors are 1,5, and 7; the selected predictors are $\{1, 2, 3, 4, 5, 6, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 33\}$. The data file concerns credit card applications. So all attribute names and values have been changed to symbols to protect confidentiality. Thus we do not know the exact interpretation of the selected factors and predictors.

5.2. Sonar data

The sonar data has 208 observations with 60 continuous predictors. The task is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. We randomly selected half of the data for training and tuning, and the remaining half of the data were used as a test set. We used 10-fold cross-validation on the training data to find good tuning parameters for the three SVMs. The whole procedure was repeated ten times.

There is no obvious grouping information in this data set. Thus we first applied hierarchical clustering to find the "groups", then we used the clustered groups to fit the F_∞-norm SVM. Figure 5.3 shows the cross-validation errors and the test errors of the F_∞-norm SVM using different number of clusters (k). We see that $k = 6$ yields the smallest cross-validation error. It is worth mentioning that in this example that the 1-norm SVM is uniformly dominated by the F_∞-norm SVM using any value of k. This example and the simulation model IV imply that the mutual information among the predictors could be used to improve the prediction performance of an L_1 procedure.

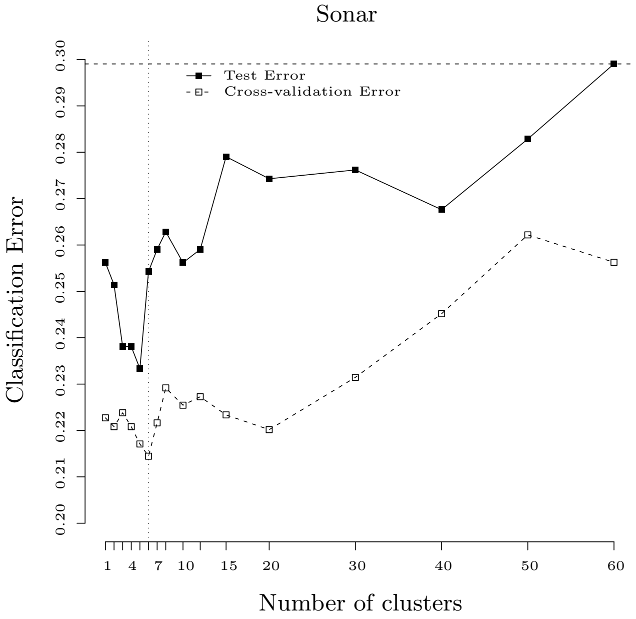


Figure 5.3. Sonar data: the cross-validation error and test error vs. the number of clusters (k). For each k we found the value of $B(k)$ giving the smallest validation error. Then the pair of $(k, B(k))$ was used in computing the test error. The broken horizontal lines indicate the test error of the 1-norm SVM. Note that the F_∞-norm SVM uniformly dominates the 1-norm SVM regardless the value of k. The dotted vertical lines show the chosen optimal k.

Table 5.6 compares the three SVMs. In this example the 2-norm SVM has the best classification performance, closely followed by the F_∞-norm SVM. Although the 1-norm SVM selects a very sparse model, its classification accuracy is significantly worse than that of the F_∞-norm SVM. If jointly considering the classification accuracy and the sparsity of the model, we think the F_∞-norm SVM is the best among the three competitors.

Table 5.6. Sonar data: compare different SVMs.

	Test Error	NSV
F_∞-norm	0.254 (0.009)	46.8 (3.92)
1-norm	0.291 (0.011)	20.4 (1.69)
2-norm	0.237 (0.011)	

We used the entire sonar data set to fit the F_∞-norm SVM. The twelve variables $\{1, 2, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60\}$ were discarded. The 1-norm SVM selected 23 variables which are all included in the set of 48 selected variables by the F_∞-norm SVM. We see that predictors 51-60, representing energy within high frequency bands, do not contribute to the classification of sonar signals.

6. Discussion

In this article we have proposed the F_∞-norm SVM for simultaneous classification and feature selection. When the input features are generated by known factors, the F_∞-norm SVM is able to eliminate a group of features if the corresponding factor is irrelevant to the response. Empirical results show that the F_∞-norm SVM often outperforms the 1-norm SVM and the standard 2-norm SVM. Similar to the 1-norm SVM, the F_∞-norm SVM often enjoys better performance than the 2-norm SVM in the presence of noise variables. When compared with 1-norm SVM, the F_∞-norm SVM is most powerful for factor selection.

With pre-defined groups, the F_∞-norm SVM and the 1-norm SVM have about the same order of computational cost. When there is no obvious group information, the F_∞-norm SVM can be used in combination with clustering among features. Note that with the freedom to select the number of clusters, the F_∞-norm SVM has the 1-norm SVM as a special case and can potentially achieve higher accuracy in classification if both are optimally tuned. Extra computations are required in clustering and selecting the optimal number of clusters. But the extra cost is worthwhile because the gain in accuracy can be substantial, as shown in Sections 4 and 5. We have used hierarchical clustering in our numerical study, because it is very fast to compute.

Clustering itself is a classical yet challenging problem in statistics. To fix ideas, we used hierarchical clustering in the examples. Although this strategy works reasonably well according to our experience, it is certainly worth investigating alternative choices. For example, in projection pursuit, linear combinations of the predictors are used as input features in nonparametric fitting. The important question is how to identify the optimal linear combinations. Zhang, Yu and Shi (2003) proposed a method based on linear discriminant analysis for identifying linear directions in nonparametric regression models (e.g., multivariate additive splines (MARS) models). Suppose that we can safely assume that the clusters/groups can be clearly defined in the space of linear combinations of the predictors. Then a good grouping method seems to be obtainable by combining Zhang's method with clustering. This is an interesting topic for future research.

There are other approaches to automatic factor selection. Consider a penalty function $p_\lambda(\cdot)$ and a norm function $s(\beta)$ such that $0 < C_1 \leq |s(\beta)|/\|\beta\|_\infty \leq C_2 < \infty$, C_1 and C_2 constants. Suppose $p_\lambda(\cdot)$ is singular at zero and consider

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \left[1 - y_i\biggl(\sum_{g=1}^{G} x_{i,(g)}^T \beta_{(g)} + \beta_0\biggr)\right]_+ + \sum_{g=1}^{G} p_\lambda\Bigl(|s(\beta_{(g)})|\Bigr). \tag{6.1}$$

By the analysis in Fan and Li (2001) we know that with a proper choice of λ, some $|s(\beta_{(g)})|$ will be zero. Thus all the variables in group g are eliminated. A good combination of $(p_\lambda(\cdot), s(\cdot))$ can be $p_\lambda(\cdot) = \lambda|\cdot|$ and $s(\beta) = \|\beta\|_q$. The F_∞-norm SVM amounts to using $p_\lambda = \lambda|\cdot|$ and $q = \infty$ in (6.1). The SCAD function (Fan and Li (2001)) gives another popular penalty function. Yuan and Lin (2006) proposed the so-called *group lasso* for factor selection in linear regression. The group lasso strategy can be easily extended to the SVM paradigm as

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \left[1 - y_i\biggl(\sum_{g=1}^{G} x_{i,(g)}^T \beta_{(g)} + \beta_0\biggr)\right]_+ + \lambda \sum_{g=1}^{G} \frac{\sqrt{\beta_{(g)}^T \beta_{(g)}}}{\sqrt{|S_g|}}. \tag{6.2}$$

Hence the group lasso is equivalent to using $p_\lambda(\cdot) = \lambda|\cdot|$ and $s(\beta) = \|\beta\|_2/\sqrt{|S_g|}$ in (6.1). In general, (6.1) (also (6.2)) is a nonlinear optimization problem and can be expensive to solve. We favor the F_∞-norm SVM because of the great computational advantages it brings about.

We have focused on the application of the F_∞-norm in binary classification problems. But the methodology can be easily extended to the case of more than two classes. Lee, Lin and Wahba (2004) proposed the multi-category SVM by utilizing a new multi-category hinge loss. A multi-category F_∞-norm SVM can be defined by replacing the L_2 penalty in the multi-category SVM with the F_∞-norm penalty.

Appendix: proof of theorem 1

We make a note that the proof is in the spirit of Rosset and Zhu (2003). Write

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

Then $\hat{\beta}(\lambda) = \arg\min_{\beta} L(\beta, \lambda)$. Let $m_0 = \min_i y_i x_i^T \beta_0 > 0$ and let $\beta_* = \beta_0 / m_0$.

Part (a). We first show that $\liminf_{\lambda \to 0}\{\min_i y_i x_i^T \hat{\beta}(\lambda)\} \geq 1$. Suppose this is not true, then there is a decreasing sequence of $\{\lambda_k\} \to 0$ and some $\epsilon > 0$ such that, for all k, $\min_i y_i x_i^T \hat{\beta}(\lambda_k) \leq 1 - \epsilon$. Then $L(\beta_*, \lambda_k) \geq L(\hat{\beta}(\lambda_k), \lambda_k) \geq [1 - (1 - \epsilon)]_+ = \epsilon$. However, note that $\min_i y_i x_i^T \beta_* = 1$, therefore

$$\epsilon \leq L(\beta_*, \lambda_k) = \lambda_k \sum_{g=1}^{G} \|\beta_{*(g)}\|_\infty \to 0 \quad \text{as } k \to \infty.$$

This is a contradiction. Now we show $\limsup_{\lambda \to 0}\{\min_i y_i x_i^T \hat{\beta}(\lambda)\} \leq 1$. Assume the contrary, then there is a decreasing sequence of $\{\lambda_k\} \to 0$ and some $\epsilon > 0$ such that, for all k, $\min_i y_i x_i^T \hat{\beta}(\lambda_k) \geq 1 + \epsilon$. Note that

$$L(\hat{\beta}(\lambda_k), \lambda_k) = \lambda_k \sum_{g=1}^{G} \|\hat{\beta}(\lambda_k)\|_\infty,$$
$$L(\frac{\hat{\beta}(\lambda_k)}{1+\epsilon}, \lambda_k) = \lambda_k \sum_{g=1}^{G} \|\hat{\beta}(\lambda_k)\|_\infty \frac{1}{1+\epsilon}.$$

Thus we have $L(\hat{\beta}(\lambda_k)/(1+\epsilon), \lambda_k) < L(\hat{\beta}(\lambda_k), \lambda_k)$, which contradicts the definition of $\hat{\beta}(\lambda_k)$. Thus we claim $\lim_{\lambda \to 0} \min_i y_i x_i^T \hat{\beta}(\lambda) = 1$.

Part (b). Suppose a subsequence of $\hat{\beta}(\lambda_k)/\|\hat{\beta}(\lambda_k)\|_{F_\infty}$ converges to β^* as $\lambda_k \to 0$. Then $\|\beta^*\|_{F_\infty} = 1$. Also denote $\min_i y_i x_i^T \beta$ by $m(\beta)$. We need to show $m(\beta^*) = \max_{\beta: \|\beta\|_{F_\infty} = 1} m(\beta)$. Assume the contrary, then there is some β^{**} such that $\|\beta^{**}\|_{F_\infty} = 1$ and $m(\beta^{**}) > m(\beta^*)$. From part (a),

$$\lim_{\lambda_k \to 0} \min_i y_i x_i^T \frac{\hat{\beta}(\lambda_k)}{\|\hat{\beta}(\lambda_k)\|_{F_\infty}} \cdot \|\hat{\beta}(\lambda_k)\|_{F_\infty} = 1,$$

which implies that $\lim_{\lambda_k \to 0} m(\beta^*)\|\hat{\beta}(\lambda_k)\|_{F_\infty} = 1$. On the other hand, we observe that

$$L\Big(\frac{\beta^{**}}{m(\beta^{**})}, \lambda_k\Big) = \lambda_k \Big\|\frac{\beta^{**}}{m(\beta^{**})}\Big\|_{F_\infty} = \lambda_k \frac{1}{m(\beta^{**})}.$$
$$L(\hat{\beta}(\lambda_k), \lambda_k) \geq \lambda_k \|\hat{\beta}(\lambda_k)\|_{F_\infty}.$$

So we have

$$\frac{L\Big(\frac{\beta^{**}}{m(\beta^{**})}, \lambda_k\Big)}{L(\hat{\beta}(\lambda_k), \lambda_k)} \leq \frac{m(\beta^*)}{m(\beta^{**})} \frac{1}{m(\beta^*)\|\hat{\beta}(\lambda_k)\|_{F_\infty}}.$$

Hence

$$\limsup_{\lambda_k \to 0} \frac{L\left(\frac{\beta^{**}}{m(\beta^{**})}, \lambda_k\right)}{L(\hat{\beta}(\lambda_k), \lambda_k)} \leq \frac{m(\beta^*)}{m(\beta^{**})} < 1,$$

which contradicts the definition of $\hat{\beta}(\lambda_k)$.

Acknowledgement

We would like to thank an associate editor and two referees for their helpful comments.

References

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

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**, 1348-1360.

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**, 102-107.

Grandvalet, Y. and Canu, S. (2003). Adaptive scaling for feature selection in svms. and class prediction by gene expression monitoring. *Advances in Neural Information Processing Systems* **15**.

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. and Tibshirani, R. (1990). *Generalized Additive Models.* Chapman and Hall, London.

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

Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.* **99**, 67-81.

Lin, Y. (2002), Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6**, 259-275.

Newton, D. J. and Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html, Department of Information and Computer Science, University of California, Irvine, CA.

Rosset, S. and Zhu, J. (2003). Margin maximizing loss functions. *Advances in Neural Information Processing Systems* **16**.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels–Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, Cambridge.

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.

Turlach, B., Venables, W. and Wright, S. (2004). Simultaneous variable selection. Technical Report, School of Mathematics and Statistics, The University of Western Australia.

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

Wahba, G., Lin, Y. and Zhang, H. (2000). GACV for support vector machines. In *Advances in Large Margin Classifiers* (Edited by A. Smola, P. Bartlett, B. SchÖlkopf and D. Schuurmans), 297-311. MIT Press, Cambridge, MA.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2001). Feature selection for svms. *Advances in Neural Information Processing Systems* **13**.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B* **68** , 49-67.

Zhang, H., Yu, C.-Y. and Shi, J. (2003). Identification of linear directions in multivariate adaptive spline models. *J. Amer. Statist. Assoc.* **98**, 369-376.

Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2004). 1-norm support vector machines. *Advances in Neural Information Processing Systems* **16**.

School of Statistics, 313 Ford Hall, 224 Church Street S.E., University of Minnesota, Minneapolis, MN 55455, USA.

E-mail: hzou@stat.umn.edu

School of Industrial and Systems Engineering, 427 Groseclose Building, 765 Ferst Drive NW, Georgia Institute of Technology, Atlanta, GA 30332, USA.

E-mail: myuan@isye.gatech.edu

(Received November 2005; accepted June 2006)