

The F_∞ -norm Support Vector Machine

HUI ZOU * MING YUAN †

November 6, 2005

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 are used to represent a categorical factor and/or a set of basis functions of a continuous variable are 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 the standard linear programming technique. Analysis on simulated and real world data shows that the F_∞ -norm SVM enjoys competitive performance when compared with the 1-norm and 2-norm SVMs.

Keywords: Support vector machine; Feature selection; Factor selection; Linear programming; L_1 penalty; F_∞ penalty.

Running title: The F_∞ -norm SVM.

*Address for correspondence: Hui Zou, Assistant Professor. School of Statistics, 313 Ford Hall, 224 Church Street S.E., University of Minnesota, Minneapolis, MN 55455, USA. Email: hzhou@stat.umn.edu.

†Ming Yuan is Assistant Professor, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332. Email: myuan@isye.gatech.edu.

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

$$\arg \max_{c \in \{1, -1\}} p(y = c|x),$$

which is referred to as the Bayes rule.

The standard 2-norm support vector machine (SVM) is a widely used classification tool (Vapnik 1996, Cristianini & Shawe-Taylor 2000, Schölkopf & 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 1996, Hastie et al. 2001):

$$\max_{\beta, \beta_0} \frac{1}{\|\beta\|_2} \tag{1}$$

$$\text{subject to } y_i(\beta_0 + x_i^T \beta) \geq 1 - \xi_i, \quad \forall i \tag{2}$$

$$\xi_i \geq 0, \sum \xi_i \leq B, \tag{3}$$

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 et al. 2000, Hastie et al. 2001)

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{\beta, \beta_0} \sum_{i=1}^n [1 - y_i(x_i^T \beta + \beta_0)]_+ + \lambda \|\beta\|_2^2, \tag{4}$$

where the subscript "+" means the positive part ($z_+ = \max(z, 0)$). The loss function $(1 - t)_+$ is called the hinge loss or SVM loss. Thus the 2-norm SVM is expressed as a quadratically regularized model fitting problem. It has been shown in Lin (2002) 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 attribute most in 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 et al. 2004, Zhu et al. 2003). 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 et al. (2002) proposed the recursive feature elimination (RFE) method. Weston et al. (2001) and Grandvalet & Canu (2003) considered some adaptive scaling methods for feature selection in SVMs. Bradley & Mangasarian (1998), Song et al. (2002) and Zhu et al. (2003) 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:

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{\beta, \beta_0} \sum_{i=1}^n [1 - y_i(x_i^T \beta + \beta_0)]_+ + \lambda \|\beta\|_1. \quad (5)$$

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 shrunken to exact 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 et al. 2003). A study comparing the L_2 and L_1 penalties (Friedman et al. 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 et al. (2004) further advocate the *bet-on-sparsity principle*; namely, 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 perfect choice 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. 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 & Tibshirani 1990). In such problems it is of interest to select the important factors rather than the individual derived variables to 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 that the factor is excluded from the model, which we call *simultaneous elimination*. Although the 1-norm SVM can annihilate individual features, it oftentimes cannot perform the simultaneous elimination, if the factor is meant to be discarded. This is largely due to the fact that no factor-feature information is used in (5). Generally speaking, if the features are penalized independently, the 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 proposed method 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 the problems where the predictors are naturally grouped, it can also be applied in other settings where the grouping is more loosely defined. We suggest to first cluster the input features into groups and then apply the F_∞ SVM. This strategy can be very useful when the predictors are a good mixture of true and noise variables, which is quite common in many real world 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 intro-

duced 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 world examples. Section 6 contains some concluding remarks.

2 The F_∞ -norm SVM Methodology

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

$$x^T \beta + \beta_0 = \sum_{g=1}^G x_{(g)}^T \beta_{(g)} + \beta_0. \quad (6)$$

Now define the infinity norm of F_g as

$$\|F_g\|_\infty = \|\beta_{(g)}\|_\infty = \max_{j \in S_g} \{|\beta_j|\}. \quad (7)$$

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

$$\min_{\beta, \beta_0} \sum_{i=1}^n \left[1 - y_i \left(\sum_{g=1}^G x_{i,(g)}^T \beta_{(g)} + \beta_0 \right) \right]_+ + \lambda \sum_{g=1}^G \|\beta_{(g)}\|_\infty. \quad (8)$$

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 (8) 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 encourages sparsity at the factor level. 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 & 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 et al. (2004) to select a *common* subset of predictors to model multiple regression responses.

The F_∞ -norm SVM is closely connected with the 1-norm SVM. When each individual feature is considered as a group, the F_∞ -norm SVM reduces to the 1-norm SVM. However, except for this trivial case, the criterion (8) is very different to (5) 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.

A fundamental difference between the F_∞ -norm and the 1-norm SVMs is that if necessary, the former can select more features than the sample size, whereas the latter cannot. An intuitive explanation is that if a factor is selected, then the F_∞ -norm SVM model tends to include all the predictors in that group. Hence the number of selected predictors by the F_∞ -norm SVM can exceed the sample size. An illustrative example will be presented in Section 4 to show this difference.

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

$$\min_{\beta, \beta_0} \sum_{i=1}^n [1 - y_i (x_i^T \beta + \beta_0)]_+ + \lambda \left(\max_j |\beta_j| \right). \quad (9)$$

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_j s are typically nonzero. The F_∞ -norm penalty mitigates this problem by dividing the predictors into several smaller groups.

Yuan & Lin (2005) proposed an alternative strategy for factor selection in linear regression. They considered the so-called *group lasso* estimate defined as the minimizer to the following penalized least squares

$$\min_{\beta} \text{RSS} + \lambda \sum_{g=1}^G \frac{\sqrt{\beta_{(g)}^T \beta_{(g)}}}{\sqrt{|S_g|}}, \quad (10)$$

where RSS stands for the usual residual-sum-squares and $|S_g|$ is the size of the g -th group. Similar approaches have also been discussed by Bakin (1999) and Grandvalet & Canu (1998). This strategy can be easily extended to the SVM paradigm as follows:

$$\min_{\beta, \beta_0} \sum_{i=1}^n \left[1 - y_i \left(\sum_{g=1}^G x_{i,(g)}^T \beta_{(g)} + \beta_0 \right) \right]_+ + \lambda \sum_{g=1}^G \frac{\sqrt{\beta_{(g)}^T \beta_{(g)}}}{\sqrt{|S_g|}}. \quad (11)$$

In general, (11) is a nonlinear optimization problem and can be expensive to solve. In contrast, as we will show in the next section, the F_∞ -norm SVM can be cast as a linear programming problem. We favor the F_∞ -norm SVM over (11) because of the great computational advantage this formulation brings about.

3 Algorithm

In this section we show that the optimization problem (8) 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 real applications.

Note that (8) can be viewed as the equivalent Lagrange formulations of the following constrained optimization problem

$$\arg \min_{\beta, \beta_0} \sum_{g=1}^G \|\beta_{(g)}\|_\infty \quad (12)$$

subject to

$$\sum_{i=1}^n \left[1 - y_i \left(\sum_{g=1}^G x_{i,(g)}^T \beta_{(g)} + \beta_0 \right) \right]_+ \leq B \quad (13)$$

for some B . There is a one-one mapping between λ and B such that the new optimization problem (12) and (13) and the primal problem (8) are equivalent. To solve (12) and (13) for a given B , we introduce a set of slack variables

$$\xi_i = \left[1 - y_i \left(\sum_{g=1}^G x_{i,(g)}^T \beta_{(g)} + \beta_0 \right) \right]_+ \quad i = 1, 2, \dots, n. \quad (14)$$

With such notation, the constraint in (13) can be rewritten as

$$y_i(\beta_0 + x_i^T \beta) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \quad \forall i, \quad (15)$$

$$\sum_{i=1}^n \xi_i \leq B. \quad (16)$$

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

$$M_g = \|\beta_{(g)}\|_\infty = \max_{j \in S_g} \{|\beta_j|\}. \quad (17)$$

Now the objective function in (12) becomes $\sum_{g=1}^G M_g$, and we need a set of new constraints

$$|\beta_j| \leq M_g \quad \forall j \in S_g \quad \text{and} \quad g = 1, \dots, G. \quad (18)$$

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 (12) and (13) can be equivalently expressed

$$\min_{\beta, \beta_0} \sum_{g=1}^G M_g \quad (19)$$

subject to

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

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:

$$\min_{\beta, \beta_0} \sum_{i=1}^n \xi_i + \lambda \sum_{g=1}^G M_g \quad (20)$$

subject to

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

Note that (8), (19) and (20) 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 real world 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. 10-fold cross-validation is often recommended in practice (Hastie et al. 2001).

4 Simulation

We conducted simulation experiments to compare the F_∞ -norm SVM with the standard 2-norm SVM and the 1-norm SVM. In what follows, we use the F_∞ SVM, L_2 SVM and L_1 SVM to denote the F_∞ -norm, 2-norm and 1-norm SVMs, respectively.

In the first set of simulation, 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, \dots, 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 logistic 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 logistic regression model with the probability of success being the logit of

$$3I(Z_1 = 1) + 2I(Z_1 = 0) + 3I(Z_2 = 1) + 2I(Z_2 = 0) + I(Z_1 = 1, Z_2 = 1) \\ + 1.5I(Z_1 = 1, Z_2 = 0) + 2I(Z_1 = 0, Z_2 = 1) + 2.5I(Z_1 = 0, Z_2 = 0) - 4.$$

In this model there are 32 predictors and 10 groups. The ground truth uses eight 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, \dots, Z_8 and W were independently generated from a standard normal distribution. The covariates is then defined as $X_i = (Z_i + W)/\sqrt{2}$. The response follows a logistic regression model with the probability of success being the logit of

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

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 10000 observations as a test set. Since the Bayes error is the lower bound for the classification accuracy of any classifier, when comparing two classifiers δ_1 and δ_2 , it is reasonable to define the relative efficiency (RE) as

$$\text{RE}(\delta_1, \delta_2) = \frac{\text{Err}(\delta_2) - \text{Bayes Error}}{\text{Err}(\delta_1) - \text{Bayes Error}}.$$

Table 1 reports the mean classification error and its standard error (in parentheses) for each of the method and each of the model averaged over 100 runs. Several observations can be made from Table 1. In all examples, the F_∞ SVM outperforms the other two methods in

terms of classification error. The relative efficiency over the L_1 SVM and the L_2 SVM can be as much as 8.67 and 4.67 (model II). We also see that the F_∞ SVM tends to be more stable than the the other two SVMs. Table 2 documents the number of factors selected by F_∞ and L_1 SVMs. It indicates that the F_∞ SVM tends to select fewer factors than the L_1 SVM.

	Model I	Model II	Model III
Bayes Error	0.095	0.116	0.188
F_∞	0.120 (0.002)	0.119 (0.010)	0.215 (0.002)
L_1	0.133 (0.026)	0.142 (0.034)	0.223 (0.003)
L_2	0.151 (0.019)	0.130 (0.025)	0.228 (0.002)
RE($F_\infty, L1$)	1.52	8.67	1.29
RE($F_\infty, L2$)	2.24	4.67	1.48

Table 1: Simulation models I, II and III: compare the accuracy of different SVMs.

	Model I	Model II	Model III
True	3	3	2
F_∞	11.46 (0.35)	3.66 (0.29)	6.70 (0.16)
L_1	11.94 (0.34)	4.33 (0.22)	6.67 (0.13)

Table 2: Simulation models I, II and III: the number of factors selected by the F_∞ and L_1 SVMs.

As mentioned in the introduction, the F_∞ SVM can also 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 to first group the features by clustering and then apply the F_∞ SVM. To demonstrate this strategy, we considered the 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 created as the follows:

$$X_i = Z_1 + 0.5\epsilon_i, \quad i = 1, \dots, 20,$$

$$X_i = Z_2 + 0.5\epsilon_i, \quad i = 21, \dots, 40,$$

$$X_i = \epsilon_i, \quad i = 41, \dots, 60.$$

The response follows 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 10000 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 the group one, and the pairwise correlation within the group is 0.8. Likewise, predictors 20-40 form the group two and the pairwise correlation is also 0.8. The first 40 predictors are considered relevant. The rest 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. On the other hand, the oracle group information is not available in real world 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 pair of (k, B) .

Figure 1 displays the classification error of the F_∞ SVM using different number of clusters (k). Based on the validation error curve we see that the optimal k is 20 and 15 for $n = 20$ and $n = 100$, respectively. It is interesting to see that for any value of k , the classification

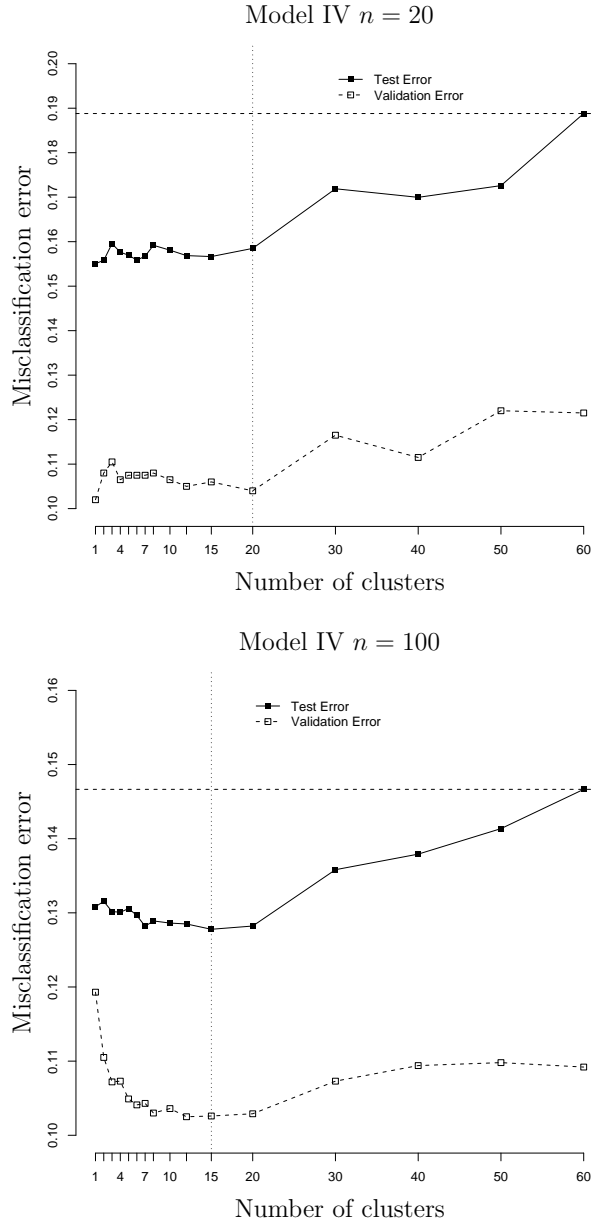


Figure 1: Simulation model IV: the validation error and test error vs. the number of clusters (k). For each k we found the value of $B(k)$ giving the smallest validation error. Then the pair of $(k, B(k))$ was used in computing the test error. The broken horizontal lines indicate the test error of the L_1 SVM. Note that in both plots the F_∞ SVM uniformly dominates the L_1 SVM regardless the value of k . The dotted vertical lines show the chosen optimal k .

accuracy of the corresponding F_∞ SVM is better than that of the L_1 SVM. As shown in Table 3, the F_∞ SVM via clustering performs almost identically to the F_∞ SVM using the oracle group information. In terms of classification accuracy the F_∞ SVM dominates the L_1 SVM and the L_2 SVM by a good margin. The relative efficiency can be as high as 2.

Furthermore, the F_∞ SVM almost identified the ground truth, while the L_1 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_∞ SVM can still select about 40 predictors. In none of the 100 simulations did the L_1 SVM select all the relevant features. The L_1 SVM also selected a few noise variables. The probability that the L_1 SVM discarded all the noise predictors is about 0.42 when $n = 20$ and 0.62 when $n = 100$. In contrast, with a probability more than 0.80, the F_∞ SVM eliminated all the noise features. Figure 2 depicts the probability of perfect variable selection by the F_∞ SVM as a function of the number of clusters. The perfect variable selection means that all the true features are selected and all the noise features are eliminated. We see that the chosen F_∞ SVMs via cross-validation have pretty high probabilities of perfect selection, even when the sample size is less than the number of (true) predictors. This simulation shows the fundamental difference between the F_∞ penalty and the L_1 penalty.

5 Real Data Examples

The simulation study has demonstrated the promising advantages of the F_∞ SVM. We now examine the performance of the F_∞ SVM and the L_1 and L_2 SVMs on two benchmark data sets, obtained from UCI Machine Learning Repository (D.J. Newman & Merz 1998).

5.1 Credit approval data

The first example is the credit approval data containing 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.

Model IV: Bayes Error = 0.109			
Method	Test Error	NSG	NSP
$n = 20$			
$F_\infty(k = 20)$	0.158 (0.004)	2.01 (0.03)	37.99 (0.48)
L_1	0.189 (0.004)	7.51 (0.25)	7.51 (0.25)
L_2	0.164 (0.004)		
$F_\infty(\text{oracle})$	0.160 (0.004)	1.97 (0.02)	39.67 (0.33)
RE($F_\infty, L1$)	1.63		
RE($F_\infty, L2$)	1.12		
$n = 100$			
$F_\infty(k = 15)$	0.128 (0.001)	2.01 (0.01)	40.32 (0.075)
L_1	0.147 (0.001)	12.21 (0.45)	12.21 (0.45)
L_2	0.140 (0.001)		
$F_\infty(\text{oracle})$	0.125 (0.001)	2.01 (0.01)	40.09 (0.057)
RE($F_\infty, L1$)	2.00		
RE($F_\infty, L2$)	1.63		

Table 3: Simulation model IV: compare different SVMs. $F_\infty(\text{oracle})$ is the F_∞ SVM using the oracle group information. NSG=Number of Selected Groups, and NSP=Number of Selected Predictors. The ground truth is that 40 predictors in two groups are true features. Note that the F_∞ SVM can select more than n predictors even when $n < p$. The F_∞ SVM is significantly more accurate than both the L_1 and L_2 SVMs.

Model IV

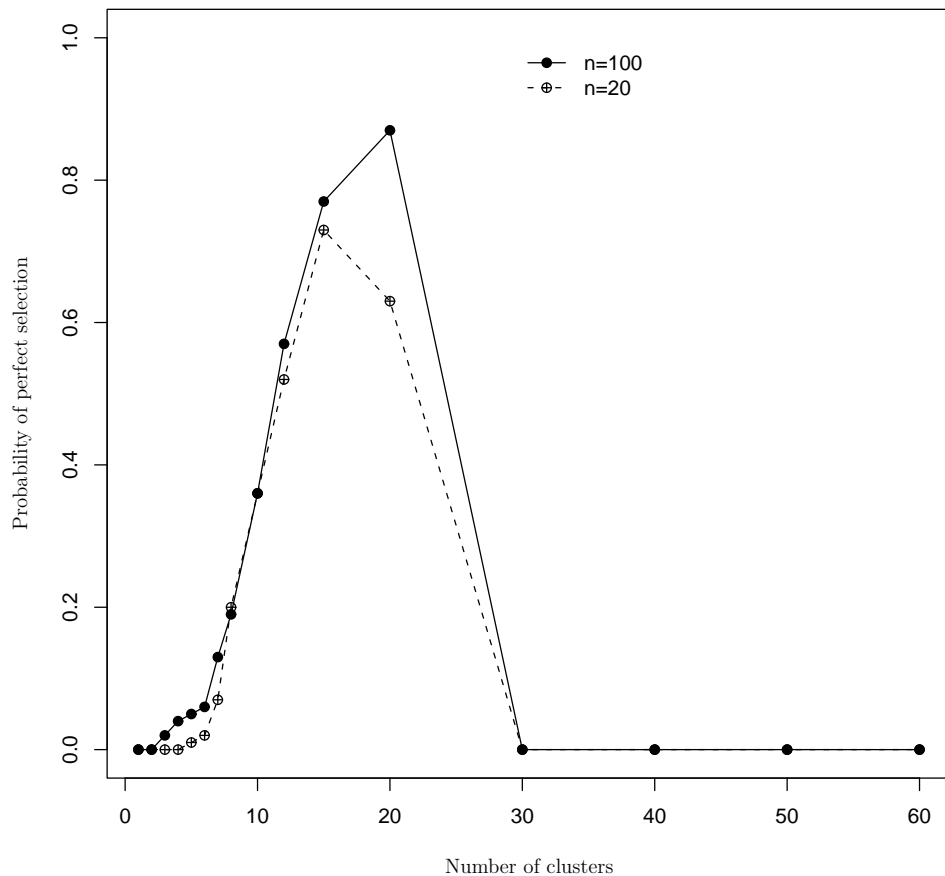


Figure 2: Simulation model IV. The probability of perfect selection by the F_∞ SVM as functions of the number of clusters. Among the 60 predictors, 40 of them are true features. Even in the case when $n = 20 < p = 60$, the chosen F_∞ SVM based on the validation error can exactly select the true model with a high probability.

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

Some categorical attributes have large number of values and some have small number of values. Thus when they are coded by dummy variables, we have some large groups as well as some small size 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 4.

We randomly selected 1/2 data for training, 1/4 data for tuning, and the remaining 1/4 as the test set. We repeated the randomization 10 times and reported the average test error of each method and its standard error. Table 5 summarizes the results. The F_∞ SVM appears to be the most accurate classifier. The variable/factor selection results look very interesting. The F_∞ and L_1 SVMs selected similar number of predictors (about 20). However, in this example, the model sparsity is best interpreted in terms of the selected factors, for we wish to know which categorical attributes are effective. When considering the factor selection, we see that the F_∞ SVM provided a much sparser model than the L_1 SVM.

We rebuilt the F_∞ SVM classifier using the entire data set. The selected factors are 1,5,

	Test Error	NSP	NSG
F_∞	0.128 (0.008)	19.70 (0.99)	3.00 (0.16)
L_1	0.132 (0.007)	20.40 (1.35)	7.70 (0.45)
L_2	0.135 (0.008)		

Table 5: Credit approval data: compare different SVMs. NSG=Number of Selected Groups, and NSP=Number of Selected Predictors.

and 7; and the selected predictors are

$$\{1, 2, 3, 4, 5, 6, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 33\}.$$

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 the good tuning parameters of 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 a F_∞ SVM. Figure 3 shows the cross-validation errors and the test errors of the F_∞ 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 the L_1 SVM is uniformly dominated by the F_∞ SVM using any value of k . This example together with 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.

Table 6 compares the three SVMs. In this example the L_2 SVM has the best classification performance, closely followed by the F_∞ SVM. Although the L_1 SVM selects a very sparse model, its classification accuracy is significantly worse than that of the F_∞ SVM. If jointly

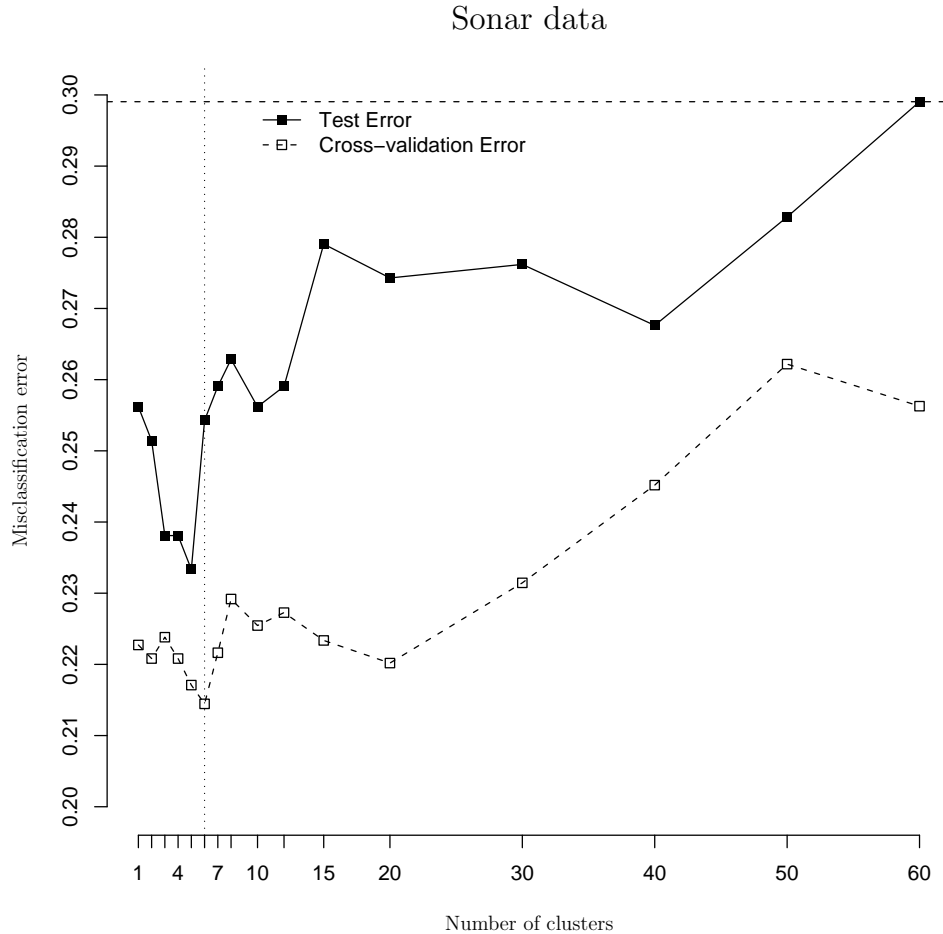


Figure 3: Sonar data: the cross-validation error and test error vs. the number of clusters (k). For each k we found the value of $B(k)$ giving the smallest validation error. Then the pair of $(k, B(k))$ was used in computing the test error. The broken horizontal lines indicate the test error of the L_1 SVM. Note that the F_∞ SVM uniformly dominates the L_1 SVM regardless the value of k . The dotted vertical lines show the chosen optimal k .

	Test Error	NSV
F_∞	0.254 (0.009)	46.8 (3.92)
L_1	0.291 (0.011)	20.4 (1.69)
L_2	0.237 (0.011)	

Table 6: Sonar data: compare different SVMs.

considering the classification accuracy and the sparsity of the model, we think the F_∞ SVM is the best among the three competitors.

We used the entire sonar data set to fit the F_∞ SVM. Twelve variables,

$$\{1, 2, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60\}$$

were discarded. The L_1 SVM selected 23 variables which are all included in the set of 48 selected variables by the F_∞ SVM.

6 Summary

In this article we have proposed the F_∞ -norm SVM for simultaneous classification and feature selection. When the inputs 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. We also proposed an efficient linear programming algorithm for solving the F_∞ -norm SVM. We believe the F_∞ -norm SVM is a practically useful tool.

In this paper, we have focused on the application of the F_∞ -norm in the binary classification problem. The methodology developed in this article, however, can be easily extended to the case of more than two classes. Lee et al. (2004) proposed the multi-category SVM, a direct multi-category generalization of the binary SVM. The key idea in their approach is a new multi-category hinge loss. By replacing the L_2 penalty in the multi-category SVM with the F_∞ -norm penalty, we can develop a multi-category F_∞ -norm SVM. This is an interesting direction of further investigation.

References

- Bakin, S. (1999), Adaptive Regression and Model Selection in Data Mining Problems, PhD thesis, Australia National University, Canberra ACT 0200, Australia.
- Bradley, P. & Mangasarian, O. (1998), Feature selection via concave minimization and support vector machines, *in* J. Shavlik, ed., ‘ICML’98’, Morgan Kaufmann.
- Cristianini, N. & Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines*, Cambridge University Press.
- D.J. Newman, S. Hettich, C. B. & Merz, C. (1998), ‘UCI repository of machine learning databases’.
- Fan, J. & Li, R. (2001), ‘Variable selection via nonconcave penalized likelihood and its oracle properties’, *Journal of the American Statistical Association* **96**, 1348–1360.
- Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004), ‘Discussion of boosting papers’, *Annals of Statistics* **32**, 102–107.
- Grandvalet, Y. & Canu, S. (1998), ‘Outcomes of the equivalence of adaptive ridge with least absolute shrinkage’, *Advances in Neural Information Processing Systems 11*.
- Grandvalet, Y. & Canu, S. (2003), ‘Adaptive scaling for feature selection in svms’, *Advances in Neural Information Processing Systems 15*.
- Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002), ‘Gene selection for cancer classification using support vector machines’, **46**, 389–422.
- Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall, London.
- Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning; Data mining, Inference and Prediction*, Springer Verlag, New York.

- Lee, Y., Lin, Y. & Wahba, G. (2004), ‘Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data’, *Journal of the American Statistical Association* **99**, 67–81.
- Lin, Y. (2002), ‘Support vector machines and the bayes rule in classification’, *Data Mining and Knowledge Discovery* **6**, 259–275.
- Schölkopf, B. & 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. & Tugcu, N. (2002), ‘Prediction of protein retention times in anion-exchange chromatography systems using support vector regression’, *Journal of Chemical Information and Computer Sciences* p. September.
- Tibshirani, R. (1996), ‘Regression shrinkage and selection via the lasso’, *Journal of the Royal Statistical Society, series B* **58**, 267.
- Turlach, B., Venables, W. & Wright, S. (2004), Simultaneous variable selection, Technical report, School of Mathematics and Statistics, The University of Western Australia.
- Vapnik, V. (1996), *The Nature of Statistical Learning*, Springer Verlag, New York.
- Wahba, G., Lin, Y. & Zhang, H. (2000), GACV for support vector machines, in A. Smola, P. Bartlett, B. Schölkopf & D. Schuurmans, eds, ‘Advances in Large Margin Classifiers’, MIT Press, Cambridge, MA., pp. 297–311.
- Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. & Vapnik, V. (2001), ‘Feature selection for svms’, *Advances in Neural Information Processing Systems* **13**.
- Yuan, M. & Lin, Y. (2005), ‘Model selection and estimation in regression with grouped variables’, *Journal of the Royal Statistical Society, series B, To Appear*.
- Zhu, J., Rosset, S., Hastie, T. & Tibshirani, R. (2003), ‘1-norm svms’, *Advances in Neural Information Processing Systems* **16**.