DEPARTMENT OF STATISTICS
North Carolina State University
2501 Founders Drive, Campus Box 8203
Raleigh, NC 27695-8203

Institute of Statistics Mimeo Series No. 2596

July 13, 2006

Variable Selection for Multicategory SVM via Sup-Norm Regularization

Hao Helen Zhang
Department of Statistics, North Carolina State University, Raleigh, NC 27695

Yufeng Liu
Department of Statistics and Operations Research, UNC-Chapel Hill, NC 27599

Yichao Wu
Department of Statistics and Operations Research, UNC-Chapel Hill, NC 27599

Ji Zhu
Department of Statistics, University of Michigan, Ann Arbor, MI 48109

Supported in part by National Science Foundation grants.

Abstract

The Support Vector Machine (SVM) has been a popular classification method in machine learning and has enjoyed great successes for many applications. However, the standard SVM cannot select variables automatically and consequently its solution typically utilizes all input variables. This makes it difficult to identify important variables which are predictive of the response and can be a concern for many problems. In this paper, we propose a novel type of regularization for the multicategory SVM (MSVM), which automates the process of variable selection and results in a classifier with enhanced interpretability and improved accuracy, especially for high dimensional low sample size data. The MSVM generally requires estimation of multiple discriminating functions and applies the argmax rule for prediction. For each individual variable, we propose to characterize its importance by the supnorm of its coefficient vector associated with different functions, and then minimize the MSVM hinge loss function subject to a penalty on the sum of supnorms. The adaptive regularization, which imposes different penalties on different variables, is studied as well. Moreover, we develop an algorithm to compute the proposed supnorm MSVM effectively. Finally, the performance of the proposed method is demonstrated through simulation studies and an application to microarray gene expression data.

Key words: L_1-norm penalty, multicategory, solution path, sup-norm, SVM.

1 Introduction

In supervised learning problems, we are given a training set of n examples from K different populations. For each example in the training set, we observe its covariate $\mathbf{x}_i \in \mathbb{R}^d$ and the corresponding label y_i indicating the membership. Our ultimate goal is to learn a classification rule which can accurately predict the class label of a future example based on its covariate. Among many classification methods, the Support Vector Machine (SVM) has gained much popularity in both machine learning and statistics. For references on the binary SVM, see Vapnik (1998), Christianini and Shawe-Taylor (2000), Schölkopf and Smola (2002), and references therein. Recently a few attempts have been made to generalize SVM to multiclass problems, such as Vapnik (1998), Weston and Watkins (1999), Crammer and Singer (2001), Lee, Lin and Wahba (2004), and Liu and Shen (2006).

While the SVM outperforms many other methods in terms of classification accuracy in numerous real problems, the implicit nature of its solutions makes it less attractive in providing insights into the predictive ability of individual variables. Oftentimes, selecting relevant variables is the primary goal of data mining. For the binary SVM, Bradley and Mangasarian (1998) demonstrated the utility of the L_1 penalty, which can effectively select variables by shrinking small or redundant coefficients to zero. Zhu et al. (2003) provides an efficient algorithm to compute the entire solution path for the L_1-norm SVM. Other forms of penalty have been also studied in the context of binary SVMs, such as the L_0 penalty (Weston et al., 2003), the SCAD penalty (Zhang et al., 2006), the combination of L_0 and L_1 penalty (Liu and Wu, 2006), the combination of L_1 and L_2 penalty (Wang et al. 2006), and others (Zou, 2006).

For multiclass problems, variable selection becomes more complex than the binary case, since the MSVM requires estimation of multiple discriminating functions, among which each function has its own subset of important predictors. One natural idea is to extend the L_1 SVM to L_1 MSVM, as done in the recent work of Lee et al. (2005) and Wang and Shen (2006). However, the L_1 penalty does not distinguish the source of coefficients. It treats different coefficients equally, no matter they correspond to the same variable or different variables. In this paper, we propose a new regularized MSVM for effective variable selection. In contrast to the L_1 MSVM, which imposes a penalty on the sum of absolute values of all coefficients, we penalize the sup-norm of the coefficients associated with each variable. The proposed method is shown to be able to achieve a higher degree of model parsimony than the L_1 MSVM without compromising classification accuracy.

This paper is organized as follows. Section 2 formulates the sup-norm regularization for the MSVM. Section 3 proposes an efficient algorithm to implement the MSVM. Section 4 discusses an adaptive approach to improve performance of the sup-norm MSVM by allowing different penalties for different covariates. Numerical results on simulated and gene expression data are given in Sections 5 and 6, followed by a summary.

2 Methodology

In K-category classification problems, we code y as $\{1, \ldots, K\}$ and define $\mathbf{f} = (f_1, \ldots, f_K)$ as a decision function vector. Here f_j, a mapping from the input domain $\mathbb{R}^d$ to $\mathbb{R}$, represents the class j; $j = 1, \ldots, K$. A classifier induced by $\mathbf{f}$,

$$\phi(\mathbf{x}) = \arg \max_{k=1,\ldots,K} f_k(\mathbf{x}),$$

assigns a new input vector $\mathbf{x}$ to the class with the largest $f_k(\mathbf{x})$. To ensure uniqueness of the solution, the sum-to-zero constraint $\sum_{k=1}^{K} f_k = 0$ is enforced. Given a classifier $\mathbf{f}$, its generalization performance is measured by the generalization error, $\text{GE}(\mathbf{f}) = P(Y \neq \arg\max_k f_k(\mathbf{x}))$.

We assume the n training pairs $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ are independently and identically distributed according to an unknown probability distribution $P(\mathbf{x}, y)$, with $p_k(\mathbf{x}) = \Pr(Y = k|\mathbf{X} = \mathbf{x})$ the conditional probability of class k given $\mathbf{X} = \mathbf{x}$. The Bayes rule which minimizes the GE is then given by

$$\phi_B(\mathbf{x}) = \arg \min_{k=1,\ldots,K} [1 - p_k(\mathbf{x})] = \arg \max_{k=1,\ldots,K} p_k(\mathbf{x}). \tag{2.1}$$

For linear classification rules, we start with $f_k(\mathbf{x}) = b_k + \sum_{j=1}^{d} w_{kj} x_j$, $k = 1, \ldots, K$. The sum-to-zero constraint then becomes

$$\sum_{k=1}^{K} b_k = 0, \quad \sum_{k=1}^{K} w_{kj} = 0, \quad j = 1, \ldots, d. \tag{2.2}$$

For nonlinear problems, we assume $f_k(\mathbf{x}) = b_k + \sum_{j=1}^{q} w_{kj} h_j(\mathbf{x})$ using a set of basis functions $\{h_j(\mathbf{x})\}$. This linear representation of a nonlinear classifier through basis functions will greatly facilitate the formulation of the proposed method. Alternatively nonlinear classifiers can also be achieved by applying the kernel trick (Boser et al., 1992). However, the kernel classifier is often given as a black box function, where the contribution of each individual covariate to the decision rule is too implicit to characterize. Therefore we will use the basis expansion to construct nonlinear classifiers in the paper.

The standard multicategory SVM (MSVM; Lee et al. 2004) solves

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[f_k(\mathbf{x}_i)+1]_+ + \lambda \sum_{k=1}^{K}\sum_{j=1}^{d} w_{kj}^2, \tag{2.3}$$

under the sum-to-zero constraint. To achieve variable selection, Wang and Shen (2006) proposed to impose the L_1 penalty on the coefficients and the corresponding L_1 MSVM then solves

$$\min_{\mathbf{b},\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[b_k + \mathbf{w}_k^T\mathbf{x}_i + 1]_+ + \lambda \sum_{k=1}^{K}\sum_{j=1}^{d} |w_{kj}| \tag{2.4}$$

under the sum-to-zero constraint. The L_1 MSVM treats all w_{kj}'s equally without distinction. As opposed to this, we take into account the fact that some of the coefficients are associated with the same covariate, therefore it is more natural to treat them as a group rather than separately.

Define the weight matrix W of size $K \times d$ such that its (k, j) entry is w_{kj}. The structure of W is shown as follows.

$$\begin{array}{c|c|c|c|c|c|}
 & x_1 & \cdots & x_j & \cdots & x_d \\
\text{Class 1} & w_{11} & \cdots & w_{1j} & \cdots & w_{1d} \\
 & \cdots & \cdots & \cdots & \cdots & \cdots \\
\text{Class k} & w_{k1} & \cdots & w_{kj} & \cdots & w_{kd} \\
 & \cdots & \cdots & \cdots & \cdots & \cdots \\
\text{Class K} & w_{K1} & \cdots & w_{Kj} & \cdots & w_{Kd}
\end{array}$$

Throughout the paper, we use $\mathbf{w}_k = (w_{k1}, \ldots, w_{kd})^{\mathrm{T}}$ to represent the kth row vector of W, and $\mathbf{w}_{(j)} = (w_{1j}, \ldots, w_{Kj})^{\mathrm{T}}$ for the jth column vector of W. According to Crammer and Singer (2001), the value $b_k + \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$ defines the similarity score of the class k, and the predicted label is the index of the row attaining the highest similarity score with $\mathbf{x}$. We define the sup-norm for the coefficient vector $\mathbf{w}_{(j)}$ as

$$\|\mathbf{w}_{(j)}\|_\infty = \max_{k=1,\cdots,K} |w_{kj}|. \tag{2.5}$$

In this way, the importance of each covariate x_j is directly controlled by its largest absolute coefficient. We propose the sup-norm regularization for MSVM:

$$\begin{aligned} \min_{\mathbf{b},\mathbf{w}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[b_k + \mathbf{w}_k^T\mathbf{x}_i + 1]_+ + \lambda \sum_{j=1}^{d} \|\mathbf{w}_{(j)}\|_\infty, \\ \text{subject to } & \mathbf{1}^T\mathbf{b} = 0, \quad \mathbf{1}^T\mathbf{w}_{(j)} = 0, \qquad \text{for } \; j = 1, \ldots, d, \end{aligned} \tag{2.6}$$

where $\mathbf{b} = (b_1, \ldots, b_K)^{\mathrm{T}}$. For three-class problems, we can show that the L_1 MSVM and the new proposed sup-norm MSVM give identical solutions after adjusting the tuning parameters, which is due to the sum-to-zero constraints on $\mathbf{w}_{(j)}$'s. This equivalence, however, does not hold for the adaptive procedures introduced in Section 4.

Lemma 2.1. *When $K = 3$, the L_1 MSVM* (2.4) *and the sup-norm MSVM* (2.6) *are equivalent.*

When $K > 3$, our empirical experience shows that the sup-norm MSVM generally performs well in terms of classification accuracy. More importantly, the sup-norm MSVM tends to make the solution more sparse than the L_1 MSVM, and identifies important variables more precisely. To further see the difference between the L_1 penalty and the sup-norm penalty, we note that the Kd coefficients fall into d groups, each of size K. A noise variable is removed if and only if all corresponding K estimated coefficients are 0. On the other hand, if a variable is important with a positive sup-norm, the sup-norm penalty, unlike the L_1 penalty, does not put any additional penalties on the other $K-1$ coefficients. This is desirable since a variable will be kept in the model as long as the sup-norm of the K coefficient is positive. No further shrinkage is needed for the remaining coefficients in terms of variable selection. For illustration, we plot the region $0 \leq t_1 + t_2 \leq C$ in

Figure 1, where $t_1 = \max(w_{11}, w_{21}, w_{31}, w_{41})$ and $t_2 = \max(w_{12}, w_{22}, w_{32}, w_{42})$. Clearly, the sup-norm penalty shrinks sum of two maximums corresponding to two variables. This helps to lead to more parsimonious models. In short, in contrast to the L_1 penalty, the sup-norm utilizes the group information of the decision function vector and consequently the sup-norm MSVM can deliver better variable selection.

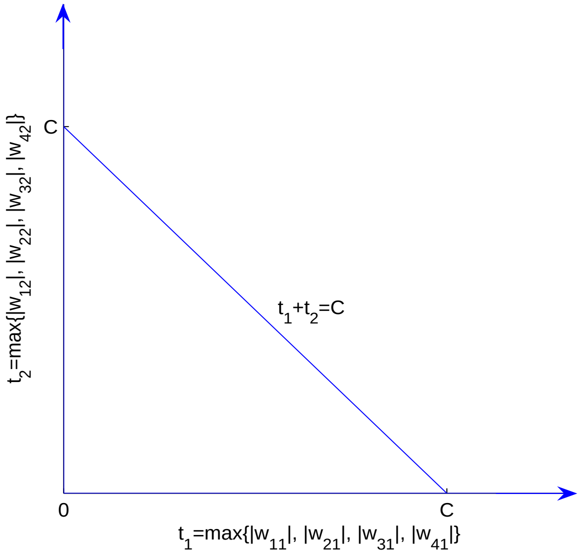
C
t_1+t_2=C
t_2=max{|w_12|, |w_22|, |w_32|, |w_42|}
0
C
t_1=max{|w_11|, |w_21|, |w_31|, |w_41|}

Figure 1: Illustrative plot of the shrinkage property of the sup-norm.

The tuning parameter λ in (2.6) balances the tradeoff between data fit and model parsimony. A proper choice of λ is important to assure good performance of the resulting classifier. If λ chosen is too small, the procedure tends to overfit the training data and gives a less sparse solution; on the other hand, if λ is too large, the solution can become very sparse but possibly with a low prediction power. The choice of parameters is typically done by minimizing either an estimate of generalization error or other related performance measure. For simulations, we generate an extra independent tuning set to choose the best λ. For real data, we use leave-one-out cross validation of the misclassification rate to select λ.

3 Computational Algorithms

In this section we show that the optimization problem (2.6) can be converted to a linear programming (LP) problem, and can therefore be solved using standard LP techniques in polynomial time. This great computational advantage is very important in real applications, especially for large data sets.

Define the $n \times K$ matching matrix A by $a_{ik} = I(y_i \neq k)$ for $i = 1, \ldots, n$ and $k = 1, \ldots, K$. First we introduce slack variables ξ_{ik} such that

$$\xi_{ik} = \left[b_k + \mathbf{w}_k^T \mathbf{x}_i + 1\right]_+ \quad \text{for} \quad i = 1, \ldots, n; \quad k = 1, \ldots, K. \tag{3.1}$$

The optimization problem (2.6) is then equivalent to

$$\begin{aligned} \min_{\mathbf{b}, \mathbf{w}, \boldsymbol{\xi}} \quad & \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} a_{ik} \xi_{ik} + \lambda \sum_{j=1}^{d} \|\mathbf{w}_{(j)}\|_\infty, \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{b} = 0, \quad \mathbf{1}^T \mathbf{w}_{(j)} = 0, \qquad\qquad j = 1, \ldots, d, \\ & \xi_{ik} \geq b_k + \mathbf{w}_k^T \mathbf{x}_i + 1, \quad \xi_{ik} \geq 0, \quad i = 1, \ldots, n; \quad k = 1, \ldots, K. \end{aligned} \tag{3.2}$$

To further simplify (3.2), we introduce a second set of slack variables

$$\eta_j = \|\mathbf{w}_{(j)}\|_\infty = \max_{k=1,\ldots,K} |w_{kj}|,$$

which add some new constraints to the problem:

$$|w_{kj}| \leq \eta_j, \quad \text{for} \quad k = 1, \ldots, K; \quad j = 1, \ldots, d.$$

Finally write $w_{kj} = w_{kj}^+ - w_{kj}^-$, where w_{kj}^+ and w_{kj}^- denote the positive and negative parts of w_{kj}, respectively. Similarly, $\mathbf{w}_j^+$ and $\mathbf{w}_j^-$ respectively consist of the positive and negative parts of components in $\mathbf{w}_j$. Denote $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)^T$; then (3.2) becomes

$$\begin{aligned} \min_{\mathbf{b}, \mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\eta}} \quad & \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \sum_{k=1}^{K} a_{ik} \xi_{ik} + \lambda \sum_{j=1}^{d} \eta_j, \\ \text{subject to} \quad & \mathbf{1}^T \mathbf{b} = 0, \quad \mathbf{1}^T [\mathbf{w}_{(j)}^+ - \mathbf{w}_{(j)}^-] = 0, \qquad j = 1, \ldots, d, \\ & \xi_{ik} \geq b_k + [\mathbf{w}_k^+ - \mathbf{w}_k^-]^T \mathbf{x}_i + 1, \quad \xi_{ik} \geq 0, \qquad i = 1, \ldots, n; \quad k = 1, \ldots, K, \\ & \mathbf{w}_{(j)}^+ + \mathbf{w}_{(j)}^- \leq \boldsymbol{\eta}, \quad \mathbf{w}_{(j)}^+ \geq \mathbf{0}, \quad \mathbf{w}_{(j)}^- \geq \mathbf{0}, \qquad j = 1, \ldots, d. \end{aligned} \tag{3.3}$$

4 Adaptive Penalty

In (2.4) and (2.6), same weights are used for different variables in the penalty terms, which may be too restrictive. In this section, we suggest that different variables should be penalized differently according to their relative importance. Ideally, large penalties should be imposed on redundant variables in order to eliminate them from models more easily; and small penalties should be used on important variables in order to retain them in the final classifier. Motivated by this, we consider the following adaptive L_1 MSVM:

$$\begin{aligned} \min_{\mathbf{b},\mathbf{w}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[b_k + \mathbf{w}_k^T\mathbf{x}_i + 1]_+ + \lambda \sum_{k=1}^{K}\sum_{j=1}^{d} \tau_{kj}|w_{kj}|, \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{b} = 0, \quad \mathbf{1}^T\mathbf{w}_{(j)} = 0, \qquad \text{for } \ j = 1, \ldots, d, \end{aligned} \tag{4.1}$$

where $\tau_{kj} > 0$ represents the weight for coefficient w_{kj}.

Due to the special form of the sup-norm SVM, we consider the following two ways to employ the adaptive penalties:

[I]

$$\begin{aligned} \min_{\mathbf{b},\mathbf{w}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[b_k + \mathbf{w}_k^T\mathbf{x}_i + 1]_+ + \lambda \sum_{j=1}^{d} \tau_j \|\mathbf{w}_{(j)}\|_\infty, \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{b} = 0, \quad \mathbf{1}^T\mathbf{w}_{(j)} = 0, \qquad \text{for } \ j = 1, \ldots, d, \end{aligned} \tag{4.2}$$

[II]

$$\begin{aligned} \min_{\mathbf{b},\mathbf{w}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)[b_k + \mathbf{w}_k^T\mathbf{x}_i + 1]_+ + \lambda \sum_{j=1}^{d} \|(\boldsymbol{\tau}\mathbf{w})_{(j)}\|_\infty, \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{b} = 0, \quad \mathbf{1}^T\mathbf{w}_{(j)} = 0, \qquad \text{for } \ j = 1, \ldots, d, \end{aligned} \tag{4.3}$$

where the vector $(\boldsymbol{\tau}\mathbf{w})_{(j)} = (\tau_{1j}w_{1j}, \ldots, \tau_{Kj}w_{Kj})^{\mathrm{T}}$ for $j = 1, ..., d$.

In (4.1), (4.2), and (4.3), the weights can be regarded as leverage factors, which are adaptively chosen such that large penalties are imposed on coefficients of unimportant covariates and small penalties on coefficients of important ones. Let $\tilde{\mathbf{w}}$ be the solution to standard MSVM (2.3) with the L_2 penalty. Our empirical experience suggests that

$$\tau_{kj} = \frac{1}{|\tilde{w}_{kj}|}$$

is a good choice for (4.1) and (4.3), and

$$\tau_j = \frac{1}{\|\tilde{\mathbf{w}}_{(j)}\|_\infty}$$

is a good choice for (4.2). If $\tilde{w}_{kj} = 0$, which implies the infinite penalty on w_{kj}, we set the corresponding coefficient solution $\hat{w}_{kj}$ to be zero.

In terms of computational issues, all three problems (4.1), (4.2), and (4.3) can be solved as LP problems. Their entire solution paths can be easily obtained by minor modifications of the algorithms in Wang and Shen (2006) and in Section 3.

5 Simulation

In this section, we demonstrate the performance of six MSVM methods: the standard L_2 MSVM, L_1 MSVM, sup-norm SVM, adaptive L_1 MSVM, and the two adaptive sup-norm MSVMs. Three simulation models are considered: (1) a linear example with four classes; (2) a linear example with five classes; (3) a nonlinear example with three classes. In each simulation setting, n observations are simulated as the training data, and another n observations are generated for tuning the regularization parameter λ for each procedure. To test the accuracy of the classification rules, we also independently generate n' observations as a test set. The tuning parameter λ is selected via a grid search over the grid: $\log_2(\lambda) = -14 : 1 : 15$. When a tie occurs, we choose the larger value of λ.

We conduct 100 simulations for each classification method under all settings. Each fitted classifier is then evaluated in terms of its classification accuracy and variable selection performance. For each method, we report its average testing error, the number of correct and incorrect zero coefficients, and the model size. We also summarize the frequency of each variable being selected over 100 runs. All simulations are done using the optimization software CPLEX.

5.1 Four-Class Linear Example

Consider a four-class example, with the input vector $\mathbf{x}$ in a 20-dimensional space. The first two components of the input vector are generated from a mixture Gaussian in the following way: for each class $k = 1, 2, 3, 4$, generate (x_1, x_2) independently from $N(\boldsymbol{\mu}_k, \sigma_1^2 I_2)$, with $\boldsymbol{\mu}_1 = (\sqrt{2}, \sqrt{2})$, $\boldsymbol{\mu}_2 = (-\sqrt{2}, \sqrt{2})$, $\boldsymbol{\mu}_3 = (-\sqrt{2}, -\sqrt{2})$, $\boldsymbol{\mu}_4 = (\sqrt{2}, -\sqrt{2})$, and the remaining eighteen components are i.i.d. generated from $N(0, \sigma_2^2)$. We generate the same number of observations in each class. Here $\sigma_1 = \sqrt{2}, \sigma_2 = 1, n = 200$, and $n' = 40,000$.

Table 1: Classification and variable selection results for the four-class linear example in Section 5.1.

Method	Testing Error	Correct Zeros	Incorrect Zeros	Model Size
L2	0.346 (0.029)	0.00	0.00	20.00
L1	0.418 (0.036)	18.11	0.10	17.92
Adapt-L1	0.411(0.038)	29.34	0.13	15.69
Supnorm	0.296 (0.006)	70.00	0.00	2.50
Adapt-supI	0.296 (0.006)	71.96	0.00	2.01
Adapt-supII	0.327 (0.029)	69.00	0.00	2.75
Bayes	0.292 (—)	72	0	2

Table 1 summarizes the performance of various procedures. In terms of classification accuracy, the sup-norm MSVM and its type I adaptive variant are the best; the corresponding testing error 0.296 is very close to the Bayes error. Over totally 100 runs, the L_2 SVM never selects the correct model, the L_1 MSVM and the adaptive L_1 MSVM both select the correct model 4 times, the sup-norm SVM selects the correct model 80 times, type I adaptive supnorm MSVM selects the correct model 99 times, and the type II selects the correct model 87 times.

Table 2 shows the frequency of each variable being selected by each procedure in 100 runs. The type II sup-norm MSVM performs the best among all. Overall the adaptive

MSVMs show significant improvement over the non-adaptive classifiers in terms of both classification accuracy and variable selection.

5.2 Five-Class Example

The setting of this example is similar to the four-class example, except that the five centers are

$$\boldsymbol{\mu}_i = 2\left(\cos([2k-1]\pi/5), \sin([2k-1]\pi/5)\right), \quad k = 1, 2, 3, 4, 5,$$

and $\mathbf{x}$ is 10-dimensional. The variances of X_1 and X_2 are both $\sigma_1^2 = 2$ and those of the other eight X's are $\sigma_2^2 = 1$. We have $n = 250$ and $n' = 50,000$ respectively.

Table 3: Classification and variable selection results for the five-class example in Section 5.2.

Method	Testing Error	Correct Zeros	Incorrect Zeros	Model Size
L2	0.454 (0.034)	0.00	0.00	10.00
L1	0.558 (0.022)	24.88	2.81	6.60
Adapt-L1	0.553 (0.020)	30.23	2.84	5.14
Supnorm	0.453 (0.020)	33.90	0.01	3.39
Adapt-supI	0.455 (0.024)	39.92	0.01	2.08
Adapt-supII	0.457 (0.046)	39.40	0.09	2.17
Bayes	0.387 (—)	41	0	2

Table 3 shows that, in terms of classification accuracy, the L_2 MSVM, supnorm MSVM, and two adaptive supnorm MSVMs are more accurate than the L_1 SVM and the adaptive L_1 SVM. In term of identifying correct models in the 100 runs, the L_2 SVM never selects the correct model, the L_1 MSVM selects the correct model 21 times, the adaptive L_1 MSVM selects the correct model 40 times, the sup-norm MSVM selects the correct model 68 times, and both adaptive supnorm MSVMs select the correct model at least 97 times.

Table 2: Variable selection frequency results of the four-class linear example in Section 5.1.

	Selection Frequency																			
Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
L2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
L1	100	100	90	86	90	85	90	90	85	86	91	92	89	93	90	86	92	81	89	87
Adapt-L1	100	100	82	78	79	78	79	72	75	82	75	76	76	74	73	77	77	72	72	72
Supnorm	100	100	4	3	3	5	0	2	1	3	4	5	2	3	2	2	3	3	1	4
Adapt-supI	100	100	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Adapt-supII	100	100	4	6	5	3	5	4	4	6	3	2	2	4	4	4	6	6	3	4

Table 4 shows the frequency of each variable being selected by all six procedures in 100 runs. Overall, the adaptive sup-norm MSVMs outperform other procedures.

Table 4: Variable selection frequency results of the five-class example in Section 5.2.

	Selection Frequency									
Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
L2	100	100	100	100	100	100	100	100	100	100
L1	100	100	59	55	60	58	56	61	57	54
Adapt-L1	100	100	44	40	43	37	39	41	35	35
Supnorm	100	100	15	17	20	17	14	20	17	19
Adapt-supI	100	100	1	1	0	2	1	1	1	1
Adapt-supII	100	100	2	2	2	2	2	2	3	2

5.3 Nonlinear Example

In this nonlinear 3-class example, we first generate $x_1 \sim \text{Unif}[-3, 3]$ and $x_2 \sim \text{Unif}[-6, 6]$. Define the functions

$$\begin{aligned} f_1 &= -2x_1 + 0.2x_1^2 - 0.1x_2^2 + 0.2, \\ f_2 &= -0.4x_1^2 + 0.2x_2^2 - 0.4, \\ f_3 &= 2x_1 + 0.2x_1^2 - 0.1x_2^2 + 0.2, \end{aligned}$$

and set $p_k(\mathbf{x}) = P(Y = k|X = \mathbf{x}) \propto \exp(f_k(\mathbf{x})), k = 1, 2, 3$. The Bayes boundary is plotted in Figure 2. We also generate three noise variables $x_i \sim N(0, \sigma^2)$, $i = 3, 4, 5$. To achieve nonlinear classification, we fit the nonlinear MSVM by including the five main effects, their square terms, and their cross products as the basis functions. In this example, we set $\sigma = 2$, $n = 200$, and $n' = 40,000$.

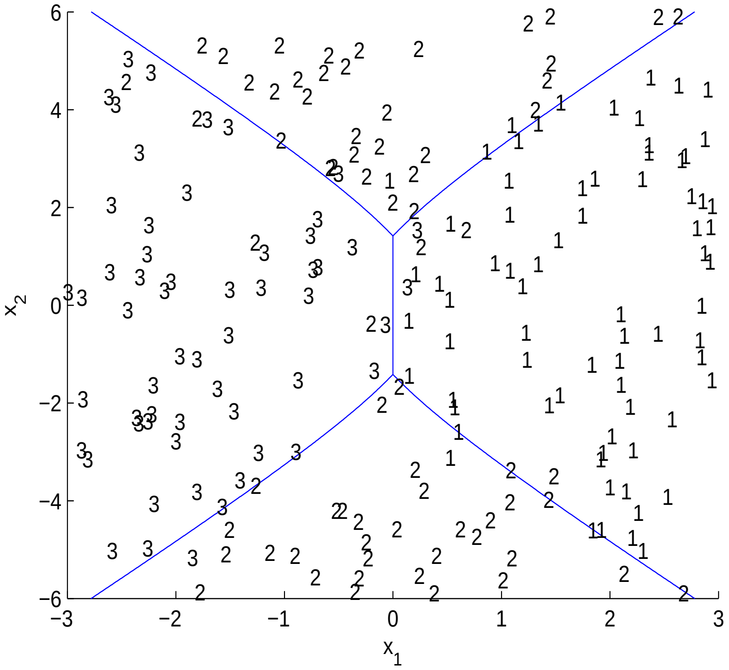

Figure 2: Bayes boundary for the nonlinear 3-class example in Section 5.3.

Table 5: Classification and variable selection results for the nonlinear example in Section 5.3.

Method	Testing Error	Correct Zeros	Incorrect Zero	Model Size
L2	0.167 (0.013)	0.00	0.00	20.00
L1	0.151 (0.012)	21.33	0.00	14.91
Adapt-L1	0.140 (0.010)	43.06	0.00	6.92
Supnorm	0.150 (0.012)	22.54	0.00	14.43
Adapt-supI	0.140 (0.010)	40.75	0.00	7.21
Adapt-supII	0.140 (0.0105)	41.37	0.00	6.21
Bayes	0.120 (—)	52	0	3

The results are summarized in Tables 5 and 6. Clearly, the adaptive L_1 SVM and the two adaptive sup-norm SVMs deliver better accurate classification and yield more spare classifiers than the other methods. In this example, there are correlations among

Table 6: Variable selection frequency results of nonlinear example.

										Selection Frequency										
Method	x_1	x_1^2	x_2^2	x_2	x_3	x_4	x_5	x_3^2	x_4^2	x_5^2	x_1x_2	x_1x_3	x_1x_4	x_1x_5	x_2x_3	x_2x_4	x_2x_5	x_3x_4	x_3x_5	x_4x_5
L2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
L1	100	100	100	69	44	50	43	80	84	89	80	55	57	65	86	88	90	69	72	70
Adapt-L1	100	100	100	33	21	21	20	24	18	22	31	20	18	20	28	26	31	20	17	22
Supnorm	100	100	100	67	37	42	34	84	80	75	79	62	58	55	87	89	91	62	68	73
Adapt-supI	100	100	100	31	21	21	26	21	25	24	31	22	17	28	30	29	30	24	16	25
Adapt-supII	100	100	100	22	18	12	19	18	16	18	25	15	14	19	30	23	22	16	17	17

Table 7: Class distribution of the microarray exmaple.

Data set	NB	RMS	BL	EWS	Total
Training	12	20	8	23	63
Test	6	5	3	6	20

covariates and consequently the variable selection task becomes more challenging. This difficulty is reflected in the variable selection frequency reported in Table 6. Despite the difficulty, the adaptive procedures are able to remove noise variables reasonably well.

6 Real Example

DNA microarray technology has made it possible to monitor mRNA expressions of thousands of genes simultaneously. In this section, we apply our six different MSVMs on the children cancer data set in Khan et al. (2001). Khan et al. (2001) classified the small round blue cell tumors (SRBCTs) of childhood into 4 classes; namely neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS) using cDNA gene expression profiles. After filtering, 2308 gene profiles out of 6567 genes are given in the data set, available at http://research.nhgri.nih.gov/microarray/Supplement/. The data set is consisted of a training set of size 63 and a test set of size 20. The distribution of the four distinct tumor categories in the training and test sets is given in Table 6. Note that Burkitt lymphoma (BL) is a subset of NHL.

To analyze the data, we first standardize the data sets by applying a simple linear transformation based on the training data. To be specific, we standardize the expression $\tilde{x}_{gi}$ of the g-th gene of subject i to obtain x_{gi} by the following formula:

$$x_{gi} = \frac{\tilde{x}_{gi} - \frac{1}{n}\sum_{j=1}^{n} \tilde{x}_{gj}}{sd(\tilde{x}_{g1}, \cdots, \tilde{x}_{gn})}.$$

Then we rank all genes using their marginal relevance in class separation by adopting

Table 8: Classification results of the microarray data using 200 genes.

		Selected genes		
Penalty	Testing Error	Top 100	Bottom 100	LOOCV error
L2	0	100	100	0
L1	1/20	62	1	0
Adp-L1	0	53	1	0
Supnorm	1/20	53	0	0
Adp-supI	1/20	50	0	0
Adp-supII	1/20	47	0	0

a simple criterion used in Dudoit et al. (2002). Specifically, the relevance measure for gene g is defined to be the ratio of between classes sum of squares to within class sum of squares as follows:

$$R(g) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(\bar{x}_{\cdot g}^{(j)} - \bar{x}_{\cdot g})^2}{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(x_{ig} - \bar{x}_{\cdot g}^{(j)})^2}, \tag{6.1}$$

where n is the size of the training set, $\bar{x}_{\cdot g}^{(j)}$ denotes the average expression level of gene g for class j observations, and $\bar{x}_{\cdot g}$ is the overall mean expression level of gene g in the training set. To examine the performance of variable selection of all different methods, we select the top 100 and bottom 100 genes as covariates according the relevance measure R.

All six MSVMs with different penalties are applied to the training set with leave-one-out cross validation. The results are tabulated in Table 8. All methods have 0 leave-one-out cross validation errors and 0 or 1 misclassification on the testing set. In terms of gene selection, three sup-norm MSVMs are able eliminate all bottom 100 genes and they use around 50 genes out of the top 100 genes to achieve comparable classification performance to other methods.

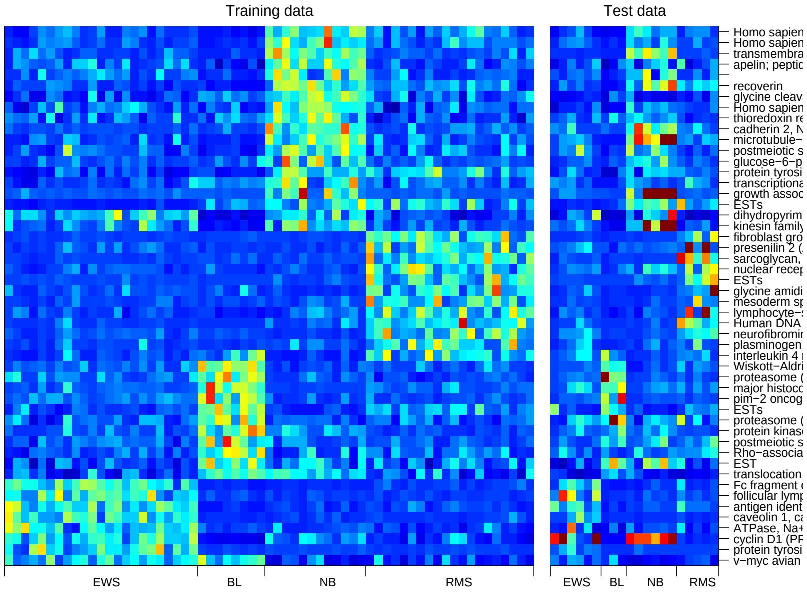

Figure 3: Heat maps of the microarray data. The left and right panels represent the training and testing sets respectively.

In Figure 3, we plot heat maps of both training and testing sets on the left and right panels respectively. In these heat maps, rows represent 50 genes selected by the Type I sup-norm MSVM and columns represent patients. The gene expression values are reflected by colors on the plots, with red representing the highest expression level and blue the lowest expression level. For visualization, we group columns within each class together and use hierarchical clustering with correlation distance on the training set to order the genes so that genes close to each other have similar expressions. From the left panel on Figure 3, we can observe four block structures associated with four classes. This implies that the 50 genes selected are highly informative in predicting the tumor types. For the testing set shown on the right panel, even though the block structure is not as clean as the training set partially due to the small testing size, we can still identify the blocks for all classes. Therefore, the proposed sup-norm MSVMs are indeed effective in performing simultaneous classification and variable selection.

7 Discussion

In this paper, we propose a new regularization method using the sup-norm to MSVM to achieve variable selection. Through the new penalty, the natural group effect among different coefficients of the same variable is embedded in the regularization framework. As a result, the sup-norm MSVMs can perform better variable selection and deliver more parsimonious classifiers than the L_1 MSVMs.

In some problems, it is possible to form groups among covariates. As argued in Yuan and Lin (2006) and Zou and Yuan (2006), it is advisable to use such group information in the model building process to improve accuracy of the prediction. If such kind of information is available for multicategory classification, there will be two kinds of group information available for model building, one type of group formed by the same covariate corresponding to different classes as considered in the paper and the other kind formed among covariates. A future research direction is to combine both group information to construct a new multicategory classification method. We believe that such potential classifiers can outperform those without using the additional information.

Appendix

Proof of Lemma 2.1: Without loss of generality, assume that $\{w_{1j}, w_{2j}, w_{3j}\}$ are all nonzero. Because of the sum-to-zero constraint $w_{1j} + w_{2j} + w_{3j} = 0$, there must be one component out of $\{w_{1j}, w_{2j}, w_{3j}\}$ has a different sign from the other two. Suppose the sign of w_{1j} differs from the other two and then $|w_{1j}| = |w_{2j}| + |w_{3j}|$ by the sum-to-zero constraint. Consequently, we have $|w_{1j}| = \max\{|w_{1j}|, |w_{2j}|, |w_{3j}|\}$. Therefore, $\sum_{k=1}^{3} |w_{kj}| = 2\|\mathbf{w}_{(j)}\|_\infty$. The equivalence of problem (2.3) with the tuning parameter λ and problem (2.6) the tuning parameter 2λ can be then established. This completes the proof.

References

Boser, B. E. and Guyon, I. M. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Fifth Annual ACM Workshop on Computational Learning Theory.* Pittsburgh, PA, ACM Press, 144–152.

Bradley, P. and Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines, In J Shavlik(eds), *ICML'98.* Morgan Kaufmann.

Crammer, K., and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, **2**, 265-292.

Christianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods.* Cambridge University Press.

Dudoit, S., Fridlyand, J. and Speed, T. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of American Statistical Association*, **97**, 77-87.

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**, 465: 67-81.

Lee, Y., Kim, Y., Lee, S., and Koo, J.-Y. (2005). Structured Multicategory Support Vector Machine with ANOVA decomposition. *Biometrika*, in press.

Liu, Y. and Wu, Y. (2006). Variable selection via a combination of the L_0 and L_1 penalties. Submitted.

Liu, Y. and Shen, X. (2006). Multicategory ψ-learning. *Journal of the American Statistical Association*, 101, 474, 500-509.

Schölkopf, B. and Smola, A. J. (2002) *Learning with Kernels.* MIT Press.

Vapnik, V. (1998). *Statistical learning theory.* Wiley.

Wang, L., Zhu, J. and Zou, H. (2006). The Doubly Regularized Support Vector Machine. *Statistica Sinica.* **16**.

Weston, J. and Watkins, C. (1999) Multiclass support vector machines. In M. Verleysen, editor, *Proceedings of ESANN99*, D. Facto Press.

Weston, J., Elisseeff, A., Schölkopf, B. and Tipping, M. (2003) Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, **3**, 1439-1461.

Yuan, M. and Lin, Y. (2006), Model Selection and Estimation in Regression with Grouped Variables, Journal of the Royal Statistical Society, Series B, **68**(1), 49-67.

Zhang, H. H., Ahn, J., Lin, X., and Park, C. (2006) Gene Selection Using Support Vector Machines With Nonconvex Penalty. *Bioinformatics*, **22**, 88-95.

Zhu, J., Hastie, T., Rosset, S., and Tibshirani, R. (2003). 1-norm support vector machines. *Neural Information Processing Systems*, **16.**

Zou, H. and Yuan, M. (2005). The F_∞-norm Support Vector Machine. School of Statistics Technical Report #646.

Zou, H. (2006). Feature Selection and Classification via a Hybrid Support Vector Machine. School of Statistics Technical Report #652.