Statistics and Its Interface Volume 4 (2011) 19–26

# Class-specific variable selection for multicategory support vector machines

Jian Guo

This paper proposes a class-specific variable selection method for multicategory support vector machines (MSVMs). Different from existing variable selection methods for MSVMs, the proposed method not only captures the important variables for classification, but also identifies the discriminable and nondiscriminable classes so as to enhance the interpretation for multicategory classification problems. Specifically, it minimizes the hinge loss of MSVMs coupled with a pairwise fusion penalty. For each variable, this penalty identifies nondiscriminable classes by imposing their associated coefficients in the decision functions to some identical value. Several simulated and real examples demonstrate that the proposed method provides better interpretation through class-specific variable selection while preserving comparable prediction performance with other MSVM methods.

AMS 2000 subject classifications: 62F07, Ranking and selection.

Keywords and phrases: Fusion penalty, Lasso, Support vector machine, Variable selection.

## 1. INTRODUCTION

Classification, regression and density function estimation are three canonical problems in machine learning. In a classification problem, all samples in the training data set are accompanied with the class labels indicating their class membership. The class labels of the samples in the test data set are unobserved. The task of the classification problem is to learn a discrimination rule from the training data and use it to predict the class labels of the test data. In the past decade, support vector machines (SVMs) gained a high degree of attention due to their outstanding prediction performance in real data analysis. The original SVM was proposed by Vapnik [14] based on the statistical learning theory. Vapnik [14] also introduced the kernel trick to make SVM have good performance for nonlinear classification problems. The original SVM was designed for a binary classification problem and it was extended to multicategory classification problems in different ways [3, 9, 10, 15, 19, 20]. The class-specific variable selection method proposed in this paper is based on the MSVM framework proposed by Lee et al. [9]. To clarify the notation, we use SVMs to represent binary SVMs and use MSVMs to represent multicategory SVMs.

Besides the emphasis of prediction performance, in recent years researchers have paid more and more attention to the interpretability of SVMs. One challenging task of interpretation is how to select the most informative variables for SVM classification. Bradley and Mangasarian [1] reformulated the standard binary SVM problem into a "loss+penalty" form and demonstrated that the utility of the $\ell_1$ penalty can effectively select significant variables by shrinking the small and redundant coefficients to zero. Zhu et al. [23] provided an efficient algorithm to compute the entire solution path for the $\ell_1$ SVM. Under the same framework, other forms of penalties were also studied, such as the $\ell_0$ penalty [18], the $\ell_q$ penalty [12], the combination of $\ell_0$ and $\ell_1$ penalties [11], the elasticnet penalty [17], the SCAD penalty [21] and the $F_\infty$-norm penalty [24]. Variable selection for MSVMs is more complex since we need to estimate multiple decision functions each of which has its own important variables. Wang and Shen [16] selected informative variables by replacing the $\ell_2$ penalty in the standard MSVM [9] with an $\ell_1$ penalty. Thereafter, Zhang et al. [22] proposed a supnorm penalty for MSVM. This penalty shrinks all coefficients associated with the same variable simultaneously and hence it tends to produce more sparse solutions than the $\ell_1$ MSVM.

All existing variable selection methods for MSVMs select informative variables in a "one-in-all-out" manner; that is, a variable is selected if it is important for at least one pair of classes and removed only if it is unimportant for all classes. However, in many practical situations, one may be interested in identifying which variables are important (discriminative) for which specific classes, or in other words, which classes are discriminable for which variables. For example, let's imagine a three-class problem with two variables. The first variable may be important for discriminating classes 1 and 2, but unimportant for classes 2 and 3; on the other hand, the second variable may be important for discriminating classes 2 and 3, but unimportant for classes 1 and 2. We believe that such situations arise often in high-dimensional data, for example, in data obtained from high-throughput expression technologies.

To address this problem, this paper proposes a *class-specific* variable selection method for MSVMs. Specifically, a *pairwise fusion* penalty is introduced to penalize the difference between (all) pairs of coefficients for each variable and shrink the coefficients of nondiscriminable classes to some identical value. If all coefficients associated with a variable are "fused," this variable is regarded as noninformative and removed from the model. Otherwise, the pairwise fusion penalty has the flexibility of only fusing the coefficients of nondiscriminable classes for this variable.

The rest of this article is organized as follows. Section 2 proposes the class-specific variable selection method for MSVMs and introduces a linear programming algorithm to solve the consequent optimization problem; Sections 3 and 4 evaluate the performance of the proposed method by two simulated examples and one real example, respectively; and we conclude in Section 5.

## 2. METHODOLOGY

Suppose we observed $n$ sample pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,p})$ is a vector composed of $p$ variables and $y_i$ is the label of $\boldsymbol{x}_i$. For a $K$-category classification problem, $y_i \in \{1, 2, \dots, K\}$. Without loss of generality, we assume $\sum_{i=1}^n x_{i,j} = 0$ for all $1 \le j \le p$. A multicategory support vector machine aims to learn $K$ decision functions $\mathrm{f} = (\mathrm{f}_1, \dots, \mathrm{f}_K)$ from the data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where each $\mathrm{f}_k(\boldsymbol{x}_i)$, a mapping from the input domain $\mathbb{R}^p$ to $\mathbb{R}$, represents the strength of the evidence that an example with input $\boldsymbol{x}_i$ belonging to class $k$. Given an estimate of the decision functions $\widehat{\mathrm{f}}$, MSVM assigns a new data point $\boldsymbol{x}^*$ to the class $k^* = \arg\max_{1 \le k \le K} \widehat{\mathrm{f}}_k(\boldsymbol{x}^*)$.

In linear classification cases, we assume $\mathrm{f}_k(\boldsymbol{x}_i) = \boldsymbol{w}_k \boldsymbol{x}_i^{\intercal} + b_k$, where $b_k$ is the intercept and $\boldsymbol{w}_k = (w_{k,1}, \dots, w_{k,p})$ is a $p$-dimensional row vector, where each component $w_{k,j}$ captures the contribution of the $j$-th variable to the $k$-th class. All $w'_{k,j}s$ $(1 \le k \le K, 1 \le j \le p)$ are referred as (decision) coefficients and they can be collected in a $K \times p$ matrix (as follows) with rows corresponding to classes and columns to variables

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,j} & \cdots & w_{1,p} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,j} & \cdots & w_{2,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{K,1} & w_{K,2} & \cdots & w_{K,j} & \cdots & w_{K,p} \end{bmatrix} .$$

Throughout the paper, we use $\boldsymbol{w}_k$ to represent the coefficients associated with the $k$-th class ($k$-th row vector of $\boldsymbol{W}$) and use $\boldsymbol{w}_{(j)} = (w_{1,j}, \dots, w_{K,j})^{\intercal}$ to represent the coefficients for the $j$-th variable ($j$-th column vector of $\boldsymbol{W}$).

In this paper, we focus on a family of MSVM methods based on the following "Loss+Penalty" framework

$$(1) \qquad \min_{\boldsymbol{b}, \boldsymbol{W}} \quad \sum_{i=1}^n \sum_{k=1}^K \mathbb{I}(y_i \ne k)[b_k + \boldsymbol{w}_k^{\intercal} \boldsymbol{x}_i + 1]_+ + \mathrm{J}_\lambda(\boldsymbol{W})$$

$$\text{subject to} \quad \sum_{k=1}^K b_k = 0, \quad \sum_{k=1}^K w_{k,j} = 0, \quad 1 \le j \le p ,$$

where $\boldsymbol{b} = (b_1, \dots, b_K)^{\intercal}$ is the vector of intercepts and $\mathbb{I}(y_i \ne k)$ is an indicator function with value 1 if $y_i \ne k$ and 0 otherwise. The sum-to-zeros constraints $\sum_{k=1}^K b_k = 0$ and $\sum_{k=1}^K w_{k,j} = 0$ impose the identifiability of the solution and they are also the necessary conditions for the Fisher consistency of the MSVM [9]. $\mathrm{J}_\lambda(\boldsymbol{W})$ is a penalty function with tuning parameter $\lambda$. It involves some prior information to help estimate the coefficients in $\boldsymbol{W}$. For example, the standard MSVM [9] employs an $\ell_2$ penalty as follows

$$(2) \qquad \mathrm{J}_\lambda^{L2}(\boldsymbol{W}) = \lambda \sum_{j=1}^p \sum_{k=1}^K w_{k,j}^2 .$$

For the purpose of variable selection, Wang and Shen [16] proposed to use the $\ell_1$ penalty as follows

$$(3) \qquad \mathrm{J}_\lambda^{L1}(\boldsymbol{W}) = \lambda \sum_{j=1}^p \sum_{k=1}^K \tau_{k,j} |w_{k,j}| ,$$

where $\tau_{k,j}$ is the adaptive weight defined as $\tau_{k,j} = 1/|\tilde{w}_{k,j}|^\gamma$ for some $\gamma > 0$. Due to its singularity property, the $\ell_1$ penalty shrinks some $w_{k,j}$'s to be exactly zero and removes the $j$-th variable from the model if all coefficients associated with the $j$-th variable (i.e., $w_{k,j}$ for all $1 \le k \le K$) are shrunken to zero (in this case, the $j$-th variable does not contribute to discriminating between the decision functions $\mathrm{f}_1, \dots, \mathrm{f}_K$ and thus it is a *noninformative* variable). Zhang et al. [22] proposed a supnorm ($\ell_\infty$) penalty (as follows) to remove the insignificant variables more efficiently.

$$(4) \qquad \mathrm{J}_\lambda^{SN}(\boldsymbol{W}) = \lambda \sum_{j=1}^p \|\boldsymbol{w}_{(j)}\|_\infty = \lambda \sum_{j=1}^p \max_{1 \le k \le K} \tau_{k,j} |w_{k,j}| .$$

This penalty treats all coefficients associated with the same variable as a natural group and shrinks them to zero simultaneously. It should be noted that the $\ell_1$ penalty usually tends to shrink only some $w_{k,j}$'s to zero, thus being more flexible but less efficient in removing noninformative variables. In Zhang et al. [22], the adaptive weights $\tau_{k,j}$, $1 \le k \le K, 1 \le j \le p$ are defined in two ways: (1) $\tau_{k,j} = 1/|\tilde{w}_{k,j}|^\gamma$, $1 \le k \le K, 1 \le j \le p$; (2) $\tau_{1,j} = \cdots = \tau_{K,j} = 1/\max\{|\tilde{w}_{1,j}|^\gamma, \dots, |\tilde{w}_{K,j}|^\gamma\}$, $1 \le j \le p$.

### 2.1 Class-specific variable selection

Given our focus on class-specific variable selection introduced in Section 1, we propose the following *pairwise fusion penalty* for MSVM

$$(5) \qquad \mathrm{J}^{PF}(\boldsymbol{W}) = \sum_{j=1}^p \sum_{1 \le k < k' \le K} \tau_{k,k'}^{(j)} |w_{k,j} - w_{k',j}| .$$

For each variable, this penalty aims at shrinking the differences between the coefficients associated with every pair of classes. Due to the singularity of the absolute value function, some terms in the sum are shrunken to exactly zero, resulting in some coefficients $w_{k,j}$'s having identical values. For example, if coefficients $w_{k,j} = w_{k',j}$, then $f_k(\boldsymbol{x}) - f_{k'}(\boldsymbol{x})$ doesn't depend on the $j$-th variable. Consequently, this variable is considered to be unimportant for discriminating between class $k$ and $k'$, though it may be important for separating other classes. Moreover, if all coefficients for the same variable are shrunken to the same value, then this variable doesn't help discriminate between the decision functions $\mathrm{f}_1, \ldots, \mathrm{f}_K$ and can be removed from the model. We refer this variable as a noninformative variable. For a two-class problem, the pairwise fusion penalty proposed here is equivalent to the $\ell_1$ penalty under the constraint $\sum_{k=1}^K w_{k,j} = 0$ $(1 \le j \le p)$. Here we set the adaptive weights $\tau_{k,k'}^{(j)}$'s based on the intuition: if variable $j$ is important for separating classes $k$ and $k'$, we would like the corresponding $\tau_{k,k'}^{(j)}$ to be small, thus the difference between $w_{k,j}$ and $w_{k',j}$ is lightly penalized. On the other hand, if variable $j$ is unimportant for separating clusters $k$ and $k'$, we would like the corresponding $\tau_{k,k'}^{(j)}$ to be large, hence the difference between $w_{k,j}$ and $w_{k',j}$ is heavily penalized. In our implementation, we compute the weights using the estimates from the standard MSVM, i.e., $\tau_{k,k'}^{(j)} = 1/|\tilde{w}_{k,j} - \tilde{w}_{k',j}|^\gamma$ where $\tilde{w}_{k,j}$ is the estimate of $w_{k,j}$ by solving (1) with penalty (2). Note that this decomposition has also been used by Guo et al. [7] for clustering purpose.

## 2.2 Algorithm

Here we discuss how to minimize objective function (1) with penalty (5), i.e., the optimization problem as follows

$$\begin{aligned}(6)\quad \min_{b,W}\quad & \sum_{i=1}^n \sum_{k=1}^K \mathbb{I}(y_i \neq k)[b_k + w_k^T x_i + 1]_+ \\ & + \lambda \sum_{j=1}^p \sum_{1 \le k < k' \le K} \tau_{k,k'}^{(j)} |w_{k,j} - w_{k',j}| \\ \text{subject to}\quad & \sum_{k=1}^K b_k = 0, \quad \sum_{k=1}^K w_{k,j} = 0, \quad 1 \le j \le p\ .\end{aligned}$$

Objective function (6) can be converted to a standard linear programming (LP) problem and solved by most linear programming software. Specifically, denote $a_{i,k} = \mathbb{I}(y_i \neq k)$, $\xi_{i,k} = [b_k + w_k^T x_i + 1]_+$ and $\theta_{k,k',j} = w_{k,j} - w_{k',j}$. To deal with the absolute value in (6), let $\theta_{k,k',j}^+ = \max\{0, \theta_{k,k',j}\}$ be the positive part of $\theta_{k,k',j}$ and $\theta_{k,k',j}^- = \max\{0, -\theta_{k,k',j}\}$ be the negative part of $\theta_{k,k',j}$. Consequently, $\theta_{k,k',j} = \theta_{k,k',j}^+ - \theta_{k,k',j}^-$ and $|\theta_{k,k',j}| = \theta_{k,k',j}^+ + \theta_{k,k',j}^-$. Thus, (6) can be written as

$$\begin{aligned}(7)\quad \min_{\boldsymbol{b},\boldsymbol{W},\boldsymbol{\Theta},\boldsymbol{\xi}}\quad & \sum_{i=1}^n \sum_{k=1}^K a_{i,k}\xi_{i,k} \\ & + \lambda \sum_{j=1}^p \sum_{1 \le k < k' \le K} \tau_{k,k'}^{(j)} (\theta_{k,k',j}^+ + \theta_{k,k',j}^-) \\ \text{subject to}\quad & \xi_{i,k} \ge b_k + w_k^T x_i + 1, \quad \xi_{i,k} \ge 0, \\ & \quad \text{for all} \ \ 1 \le i \le n, \quad 1 \le k \le K; \\ & \sum_{k=1}^K b_k = 0, \quad \sum_{k=1}^K w_{k,j} = 0, \\ & \quad \text{for all} \ \ 1 \le j \le p; \\ & \theta_{k,k',j}^+ - \theta_{k,k',j}^- = w_{k,j} - w_{k',j}, \\ & \theta_{k,k',j}^+ \ge 0, \quad \theta_{k,k',j}^- \ge 0, \\ & \quad \text{for all} \ \ 1 \le k < k' \le K, \quad 1 \le j \le p\end{aligned}$$

where $\boldsymbol{\Theta} = \{\theta_{k,k',j}^+, \theta_{k,k',j}^- : 1 \le k < k' \le K,\ 1 \le j \le p\}$ and $\boldsymbol{\xi} = (\xi_{i,k})_{n \times K}$. In this article, objective function (7) was solved by the mathematical programming language AMPL with linear programming package CPLEX.

# 3. SIMULATION STUDY

In this section, we illustrate the performance of the proposed class-specific variable selection method on two synthetic examples with four and five classes, respectively. We compare eight different MSVM methods, coupled with: the $\ell_2$ penalty ("L2", equation (2)), the $\ell_1$ penalty ("L1", equation (3) but setting all $\tau_{k,j} = 1$) and its adaptive counterpart ("AL1", equation (3)), the supnorm penalty ("SN", equation (4) but setting all $\tau_{k,j} = 1$) and its two adaptive counterparts ("ASN-I" and "ASN-II", equation (4) with two types of adaptive weights), the proposed pairwise fusion penalty ("PF", equation (5) but setting all $\tau_{k,k'}^{(j)} = 1$) and its adaptive counterpart ("APF", equation (5)). In each simulation, 200 training observations, 200 validation observations and 10,000 test observations are generated. The tuning parameter $\lambda$ is selected on the validation set via a grid $\{2^{-15}, 2^{-14}, \ldots, 2^{15}\}$. We repeat this procedure 100 times for each simulation and record the average test error rates as compared to the true class labels, and average selection rate for both informative and noninformative variables.

### Example 1

In this simulation, there are four classes and $p = 102$ variables, with the first two variables being informative and the remaining ones noninformative. The variables were generated according to the following mechanism: the two informative variables $x_1$ and $x_2$ are independently uniformly distributed in $[-1, 1]$, whereas the remaining 100 noninformative variables *i.i.d.* follow $N(0, 8^2)$. Denote $\boldsymbol{x} = (x_1, \ldots, x_{102})$, then we define the decision function for the

$k$-th class as follows

$$\mathrm{f}_k(\boldsymbol{x}) = \begin{cases} 10x_1 + 5x_2, & \text{if } k = 1; \\ 5x_2, & \text{if } k = 2; \\ -5x_2, & \text{if } k = 3; \\ -10x_1 - 5x_2, & \text{if } k = 4. \end{cases}$$

and we assign $\boldsymbol{x}$ to class k with a probability proportional to $\exp\{f_k(\boldsymbol{x})\}$. In this example, $x_1$ is unimportant for discriminating between classes 2 and 3 and $x_2$ is unimportant for discriminating between classes 1 and 2, as well as classes 3 and 4.

### Example 2

In this example, a five-class scenario is considered. There are a total of $p = 103$ variables with the first three informative and the other 100 noninformative. Similar to Example 1, the informative variables are independently uniformly distributed in $[-1, 1]$, whereas the 100 noninformative variables *i.i.d.* follow $N(0, 8^2)$. We define the decision function for the $k$-th class as

$$\mathrm{f}_k(\boldsymbol{x}) = \begin{cases} 4x_1 - 10x_2 + 6x_3, & \text{if } k = 1; \\ 4x_1 + x_3, & \text{if } k = 2; \\ -x_1 + x_3, & \text{if } k = 3; \\ -x_1 - 4x_3, & \text{if } k = 4; \\ -6x_1 + 10x_2 - 4x_3, & \text{if } k = 5. \end{cases}$$

and assign $\boldsymbol{x}$ to class k with a probability proportional to $\exp\{f_k(\boldsymbol{x})\}$. Notice that $x_1$ is unimportant for discriminating between classes 1 and 2 and classes 3 and 4; $x_2$ is unimportant for classes 2, 3 and 4; and $x_3$ is unimportant for classes 2 and 3 and classes 4 and 5.

For each example, the Bayes error of each replicated data is computed as a benchmark for all competing models. The prediction and variable selection results are summarized in Table 1. The results were averaged over 100 replication, where the corresponding standard deviations are in the parentheses. We can see that, for every method, the adaptive penalty has significant effect to reduce the error rate and to reduce the number of incorrectly selected noninformative variables. Comparing all methods with adaptive penalties, the proposed AFP method achieves the lowest error rate and the least number of incorrectly selected noninformative variables in both examples, although the advantage is not significant.

If a variable can not discriminate a pair of classes, and the corresponding estimated coefficients are also the same, we consider this as a correct "fusion". Table 2 summarizes these results. Specifically, each row in the table gives the proportion of correctly fused variables (averaged over 100 replications) that are noninformative for separating the corresponding pair of classes (indicated in column "Pair"). For example, the second row shows that on average 62% of the first two informative variables are correctly fused for classes 1 and 2 by the AFP method. It is clear that APF dominates other methods in terms of correctly fusing the coefficients of nondiscriminable classes. It should also be pointed out that although AL1 and ASN-I correctly fuse some coefficients of nondiscriminable classes, e.g., in the first row (AL1) as well as in the second and third rows (ASN-I), the result is an artifact. In Example 1, the coefficients of classes 2 and 3 for variable 1 are all equal to zero, which happens to be the value that the $\ell_1$ penalty shrinks to. The same reasoning applies to classes 2, 3 and 4 for variable 2 in Example 2 (rows 6–8, column "AL1"). On the other hand, in Example 1, although classes 1 and 2 (as well as classes 3 and 4) have the same coefficient for variable 2, the L1 method fails to fuse them, since their coefficients are different from zero. In contrast to AL1, ASN-I tends to encourage the coefficients with large magnitudes to have some identical values. In Example 1, for instance, the coefficients of classes 1 and 2 (as well as those of classes 3 and 4) for variable 2 have the same value with large magnitude, thus they are identified by ASN-I. On the other hand, it fails to fuse classes 2 and 3 for variable 1, since their coefficients are close to zero. However, the APF method identifies the structure correctly in both situations.

*Table 1. Simulation results for Example 1. "Error rate" is the proportion of mis-classified samples in the test data set. "Info" is the number of selected informative variables (out of 2 for Example 1 and out of 3 for Example 2). "Noninfo" is the number of noninformative variables (out of 100). All results are averaged over 100 replications and their corresponding standard deviations are recorded in the parentheses*

| Example | Method | Error rate (%) | Info | Noninfo |
|---|---|---|---|---|
| 1 | Bayes error | 13.0 (–) | – | – |
| | L2 | 54.5 (3.99) | 2.0 (0.00) | 100.0 (0.00) |
| | L1 | 51.1 (2.86) | 2.0 (0.00) | 99.9 (0.36) |
| | SN | 50.1 (2.72) | 2.0 (0.00) | 99.8 (0.55) |
| | PF | 51.3 (2.85) | 2.0 (0.00) | 99.9 (0.48) |
| | AL1 | 17.9 (3.27) | 2.0 (0.00) | 4.4 (5.56) |
| | ASN-I | 15.0 (1.59) | 2.0 (0.00) | 1.2 (1.95) |
| | ASN-II | 16.8 (3.15) | 2.0 (0.00) | 0.9 (2.44) |
| | APF | 14.0 (1.00) | 2.0 (0.00) | 0.1 (0.31) |
| 2 | Bayes error | 13.8 (–) | – | – |
| | L2 | 56.6 (3.68) | 3.0 (0.00) | 100.0 (0.00) |
| | L1 | 54.5 (4.11) | 2.8 (0.72) | 94.1 (23.80) |
| | SN | 54.8 (4.13) | 2.8 (0.72) | 94.0 (24.00) |
| | PF | 54.7 (3.89) | 2.8 (0.72) | 94.0 (24.00) |
| | AL1 | 20.9 (3.10) | 3.0 (0.28) | 7.4 (8.27) |
| | ASN-I | 21.7 (2.23) | 3.0 (0.00) | 4.1 (6.36) |
| | ASN-II | 19.3 (3.01) | 3.0 (0.14) | 3.4 (5.00) |
| | APF | 18.8 (4.00) | 3.0 (0.14) | 0.2 (0.72) |

## 4. REAL DATA ANALYSIS

### 4.1 Microarray example

In this example, we apply the proposed pairwise fusion SVM method to conduct class-specific variable selection on a

*Table 2. Results of class-specific variable selection for nondiscriminable class pairs*

| Example | Variable | Pair | L2 | L1 | AL1 | SN | ASN-I | ASN-II | PF | APF |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2/3 | 0.00 | 0.02 | 0.28 | 0.00 | 0.00 | 0.00 | 0.02 | 0.28 |
|  | 2 | 1/2 | 0.00 | 0.00 | 0.00 | 0.38 | 0.92 | 0.00 | 0.24 | 0.62 |
|  |  | 3/4 | 0.00 | 0.00 | 0.00 | 0.38 | 0.92 | 0.00 | 0.32 | 0.74 |
| 2 | 1 | 1/2 | 0.00 | 0.06 | 0.02 | 0.14 | 0.48 | 0.02 | 0.16 | 0.72 |
|  |  | 3/4 | 0.00 | 0.14 | 0.78 | 0.06 | 0.00 | 0.02 | 0.14 | 0.88 |
|  | 2 | 2/3 | 0.00 | 0.12 | 0.72 | 0.06 | 0.00 | 0.00 | 0.08 | 0.72 |
|  |  | 2/4 | 0.00 | 0.08 | 0.36 | 0.06 | 0.00 | 0.00 | 0.06 | 0.34 |
|  |  | 3/4 | 0.00 | 0.08 | 0.54 | 0.06 | 0.00 | 0.00 | 0.08 | 0.52 |
|  | 3 | 2/3 | 0.00 | 0.20 | 1.00 | 0.06 | 0.00 | 0.00 | 0.14 | 0.86 |
|  |  | 4/5 | 0.00 | 0.06 | 0.02 | 0.12 | 0.44 | 0.00 | 0.18 | 0.74 |

microarray dataset of small round blue cell tumors (SRBCT) of childhood cancer [8]. This dataset contains the expression profiles of 2,308 genes obtained and 83 tissue samples. These subjects are classified into four tumor subtypes: Ewing family of tumors (EWS), rhabdomyosarcoma (RMS), neuroblastoma (NB) and Burkitt lymphoma (BL). We preprocessed the data by selecting a subset of 500 genes according to their marginal relevance criterion [4, 22]:

$$R_j = \frac{\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{I}(y_i = k)(\mu_{k,j} - \mu_j)^2}{\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{I}(y_i = k)(x_{i,j} - \mu_{k,j})^2} \qquad (8)$$

where $\mu_{k,j}$ is the mean of all samples in class $k$ and variable $j$ and $\mu_j$ is the mean of all samples in variable $j$. The term on the numerator reflects the between-class distance and the term on the denominator reflects the within-class distance. Therefore, this criterion gives large values to those genes expressing heterogeneously across the classes and homogenously within the classes. The top 500 genes are collected and the new data are centered and scaled along each variable before classification. The total 83 samples are randomly split into a training set (2/3 of all samples) and a test set (1/3 of the samples) and this procedure was repeated 100 times.

All MSVM methods with different penalties are applied to the 100 training/test sets. The tuning parameter $\lambda$ of each MSVM method is selected by five-fold cross validation on each training set. The test errors and the number of selected genes over 100 replications are averaged (by median) and listed in Table 3. All the compared methods produce zero error rates and select similar number of genes.

To evaluate the class-specific variable selection, we pick one out of 100 replications and illustrate the estimation of the APF method using a heatmap in Figure 1. In this figure, the rows correspond to the genes selected by the APF method and the column to the six pairs formed from the four subtypes. A black (white) spot indicates that the estimated coefficients of the corresponding gene for the two subtypes are different (identical). For example, gene "1358266" can not discriminate subtypes EWS, NB and BL, but it can discriminate them from subtype RMS. We can see that APF provides a more informative way for describing the functions of a gene with respect to discriminating different tumor subtypes.

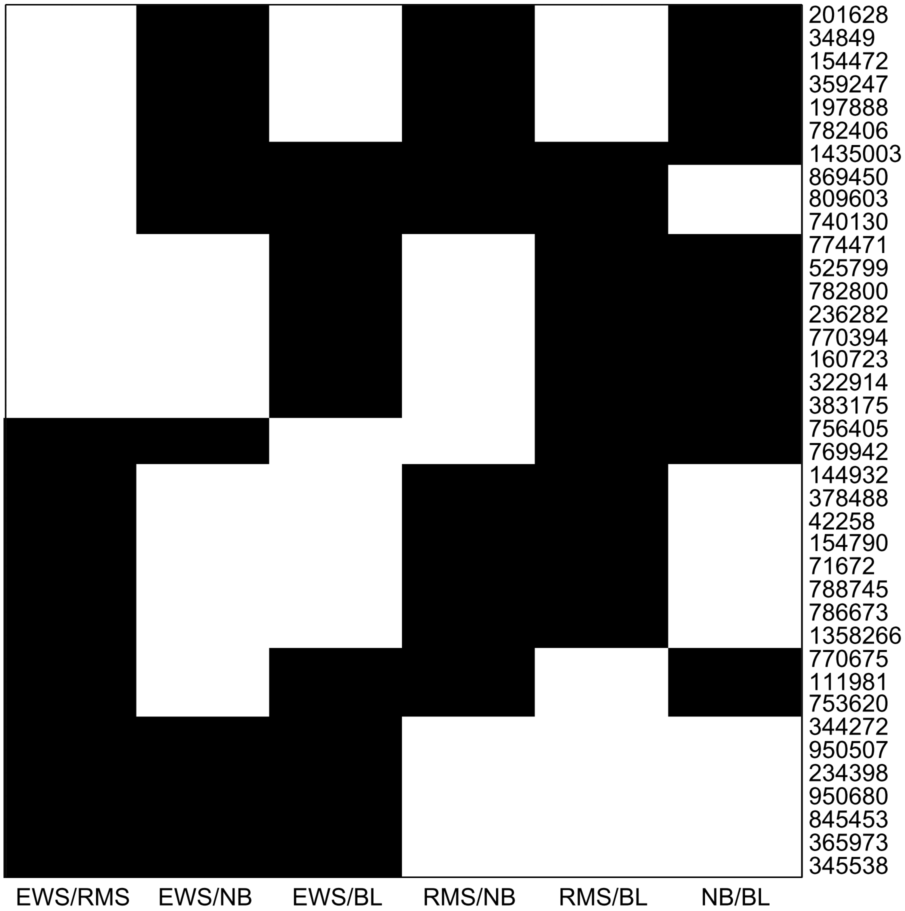
*Table 3. Classification and variable selection results of SRBCT data set. The numbers in the table are the medians of 100 rounds of random splits and the numbers in the parentheses are the corresponding median absolute deviations*

| Method | Test error (%) | Selected genes (#) |
|---|---|---|
| AL1 | 0 (0) | 46.5 (13.46) |
| ASN-I | 0 (0) | 51.5 (12.11) |
| ASN-II | 0 (0) | 42.0 (11.59) |
| APF | 0 (0) | 38.0 (10.07) |

## 4.2 Web mining example

The data set comes from the World Wide Knowledge Base project at Carnegie Mellon University. It was collected in 1997 and includes webpages from websites at computer science departments in the following four universities: Cornell, Texas, Washington, and Wisconsin. The webpages were manually classified into seven categories, but in this example, only 1,396 webpages corresponding to the four largest categories were used: student (544 webpages), faculty (374 webpages), course (310 webpages) and project (168 webpages). The original data set was preprocessed by Cardoso-Cachopo [2] following the following steps: (1) Substituting space for tab, newline, and return characters; (2) Keeping only letters (that is, turning punctuation, numbers, etc. into spaces) and turning all letters to lowercase; (3) Removing words less than 3 characters long and removing the 524 smart stopwords; (4) Substituting a single space for multiple spaces; (5) Stemming the documents by applying a stemmer algorithm [13] to the remaining text.

The log-entropy weighting method [5] was used to calculate the term-document matrix $\boldsymbol{X} = (x_{i,j})_{n\times p}$, with $n$ and $p$ denoting the number of webpages and distinct terms, respectively. Let $f_{i,j}$, $1 \le i \le n, 1 \le j \le p$ be the number of times the $j$-th term appears in the $i$-th webpage and let $p_{i,j} = f_{i,j}/\sum_{i=1}^{n} f_{i,j}$. Then, the log-entropy weight of the



| | EWS/RMS | EWS/NB | EWS/BL | RMS/NB | RMS/BL | NB/BL |
|---|---|---|---|---|---|---|
| 201628 | □ | ■ | □ | ■ | □ | ■ |
| 34849 | □ | ■ | □ | ■ | □ | ■ |
| 154472 | □ | ■ | □ | ■ | □ | ■ |
| 359247 | □ | ■ | □ | ■ | □ | ■ |
| 197888 | □ | ■ | □ | ■ | □ | ■ |
| 782406 | □ | ■ | □ | ■ | □ | ■ |
| 1435003 | □ | ■ | ■ | ■ | ■ | ■ |
| 869450 | □ | ■ | ■ | ■ | ■ | □ |
| 809603 | □ | ■ | ■ | ■ | ■ | □ |
| 740130 | □ | ■ | ■ | ■ | ■ | □ |
| 774471 | □ | □ | ■ | □ | ■ | ■ |
| 525799 | □ | □ | ■ | □ | ■ | ■ |
| 782800 | □ | □ | ■ | □ | ■ | ■ |
| 236282 | □ | □ | ■ | □ | ■ | ■ |
| 770394 | □ | □ | ■ | □ | ■ | ■ |
| 160723 | □ | □ | ■ | □ | ■ | ■ |
| 322914 | □ | □ | ■ | □ | ■ | ■ |
| 383175 | □ | □ | ■ | □ | ■ | ■ |
| 756405 | ■ | ■ | □ | □ | ■ | ■ |
| 769942 | ■ | ■ | □ | □ | ■ | ■ |
| 144932 | ■ | □ | □ | ■ | ■ | □ |
| 378488 | ■ | □ | □ | ■ | ■ | □ |
| 42258 | ■ | □ | □ | ■ | ■ | □ |
| 154790 | ■ | □ | □ | ■ | ■ | □ |
| 71672 | ■ | □ | □ | ■ | ■ | □ |
| 788745 | ■ | □ | □ | ■ | ■ | □ |
| 786673 | ■ | □ | □ | ■ | ■ | □ |
| 1358266 | ■ | □ | □ | ■ | ■ | □ |
| 770675 | ■ | □ | ■ | ■ | □ | ■ |
| 111981 | ■ | □ | ■ | ■ | □ | ■ |
| 753620 | ■ | □ | ■ | ■ | □ | ■ |
| 344272 | ■ | ■ | ■ | □ | □ | □ |
| 950507 | ■ | ■ | ■ | □ | □ | □ |
| 234398 | ■ | ■ | ■ | □ | □ | □ |
| 950680 | ■ | ■ | ■ | □ | □ | □ |
| 845453 | ■ | ■ | ■ | □ | □ | □ |
| 365973 | ■ | ■ | ■ | □ | □ | □ |
| 345538 | ■ | ■ | ■ | □ | □ | □ |

*Figure 1. Results of class-specific variable selection for the APF method on the SRBCT data. Each row corresponds to a gene (denoted by its ID). Each column corresponds to a pair of tumor subtypes; for example, "EWS/NB" indicates subtypes EWS and NB. A black (white) spot indicates that the estimated coefficients of the corresponding gene for the two subtypes are different (identical).*

$j$-th term is defined as

$$e_j = 1 + \sum_{i=1}^{n} p_{i,j} \log(p_{i,j}) / \log(n) \ .$$

Finally, the term-document matrix $\boldsymbol{X}$ is defined as

$$x_{i,j} = e_j \log(1 + f_{i,j}) \ , 1 \le i \le n \ , \ 1 \le j \le p \ ,$$

and it is normalized along each column. The data are further cleaned by extracting $n = 1,396$ documents from the four largest categories and keep only top $p = 100$ terms with the highest log-entropy weights out of a total of 4,800 terms. Before doing the analysis for this paper, the cleaned data has been used to explore the word networks using graphical models [6]. Similar to the SRBCT example, all documents in the cleaned web mining data set were randomly split into a training set and a test set and this procedure was repeated 100 times.

*Table 4. Classification and variable selection results of web mining data set using top 100 terms. The numbers in the table are the medians of 100 rounds of random splits and the numbers in the parentheses are the corresponding median absolute deviations*

| Method | Test error (%) | Selected terms (#) |
|---|---|---|
| AL1 | 21.0 (1.65) | 90.5 (7.06) |
| ASN-I | 20.8 (1.47) | 89.0 (7.24) |
| ASN-II | 20.9 (1.58) | 87.0 (6.62) |
| APF | 20.8 (1.54) | 93.5 (7.41) |

Table 4 shows the prediction and variable selection results from different MSVM methods. The results are averaged (by median) over 100 replications. Again, all these methods produce similar test error rates and select similar number of variables (terms). Figure 2 illustrates the results of class-specific variable selection from one out of 100 replications. We can see that many terms only contribute to discriminate a set of topics. For example, the term "system" can discriminate the topics "faculty", "student" and "project", but it

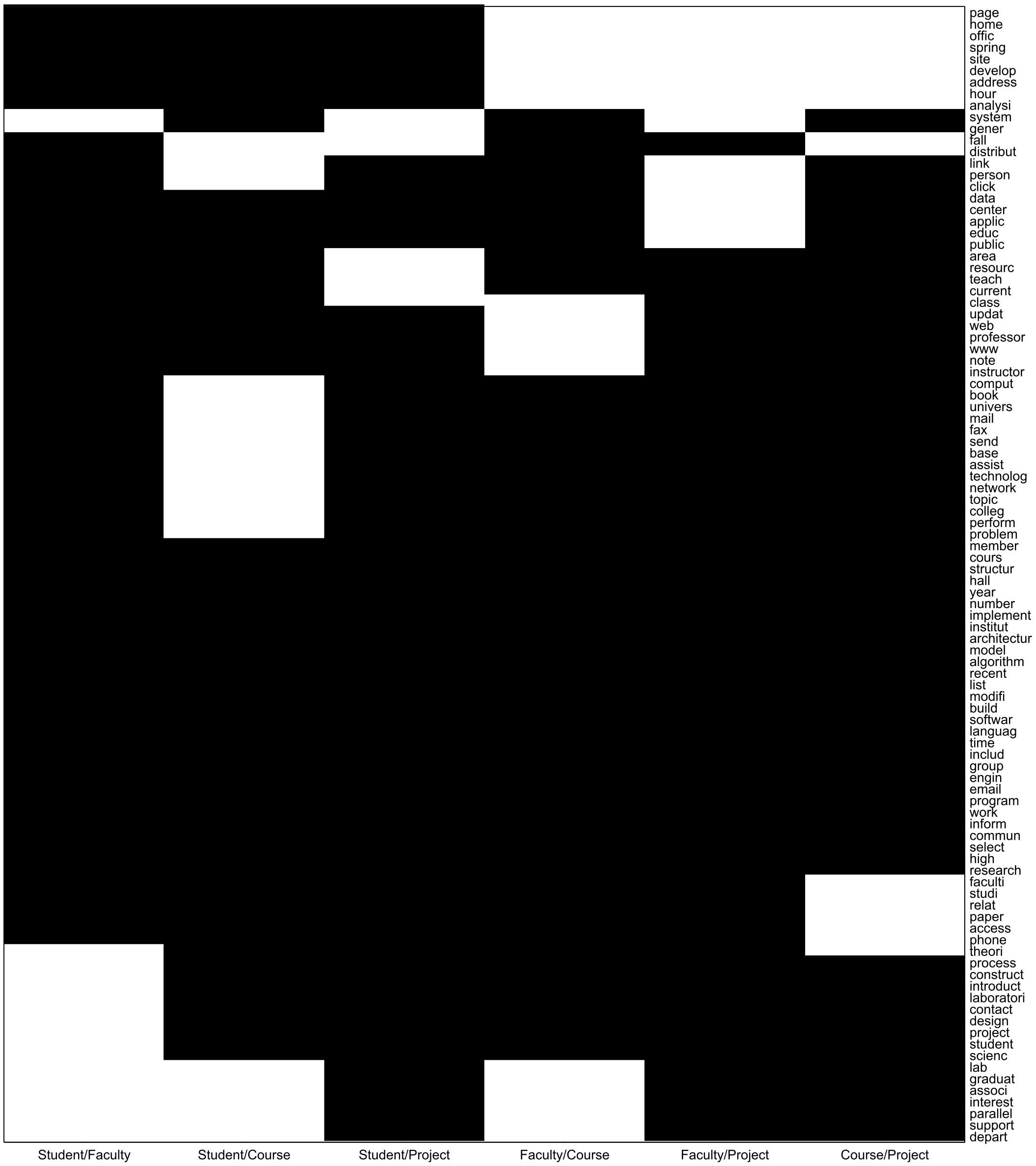


*Figure 2. Results of class-specific variable selection for the APF method on the web mining data. Each row corresponds to a term and each column corresponds to a pair of topics. A black (white) spot indicates that the the corresponding terms is discriminable (nondiscriminable) for the associated two topics.*

can discriminate them from the topic “course”. Therefore, class-specific variable selection provides better interpretation and it helps deeper understand the structure of the data.

## 5. CONCLUSION

This paper develops a method for simultaneously classifying high-dimensional data and selecting informative variables, by employing a penalized multicategory support vector machine framework. In particular, the proposed method penalizes the difference between the coefficients for each pair of classes and for each variable, which allows one to identify and remove unimportant variables for selected subsets of classes. This allows one to gain more insight into the function of particular variables and potentially discover heterogeneous structures that other available methods are unable to capture.

## ACKNOWLEDGEMENTS

The author thanks the associate editor and the reviewer for their helpful comments. The author also thanks Elizaveta Levina, George Michailidis and Ji Zhu for their discussion and suggestions. I also thank Yichao Wu and Yufeng Liu for providing the code to implement supnorm SVM.

*Received 17 June 2010*

## REFERENCES

[1] Bradley, P. and Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines. In *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA.

[2] Cardoso-Cachopo, A. (2009). http://web.ist.utl.pt/~acardoso/datasets/.

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

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

[5] Dumais, S. (1991). Improving the retrieval of information from external source. *Behavior Research Methods, Instruments and Computers*, **23** 229–236.

[6] Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2010). Joint estimation of multiple graphical models. *Biometrika* To appear.

[7] Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2010). Pairwise variable selection for high-dimensional model-based clustering. *Biometrics*, **66** 793–804.

[8] Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C. and Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, **7** 673–679.

[9] 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. *Journal of the American Statistical Association*, **99** 67–81.

[10] Liu, Y. and Shen, X. (2006). Multicategory $\psi$-learning. *Journal of the American Statistical Association*, **101** 500–509.

[11] Liu, Y. and Wu, Y. (2007). Variable selection via a combination of the $\ell_0$ and $\ell_1$ penalties. *Journal of Computational and Graphical Statistics*, **16** 782–798.

[12] Liu, Y., Zhang, H. H., Park, C. and Ahn, J. (2007). Support vector machines with adaptive $\ell_q$ penalties. *Computational Statistics and Data Analysis*, **51** 6380–6394.

[13] Porter, M. (1980). An algorithm for suffix stripping. *Program*, **14** 130–137.

[14] Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag, New York. MR1367965

[15] Vapnik, V. (1998). *Statistical learning theory*. Wiley. MR1641250

[16] Wang, L. and Shen, X. (2007). On $\ell_1$-norm multi-class support vector machines: methodology and theory. *Journal of the American Statistical Association*, **102** 583–594.

[17] Wang, L., Zhu, J. and Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica*, **16** 589–615.

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

[19] Weston, J. and Watkins, C. (1999). Multiclass support vector machines. In *Proceedings of ESANN99*. D. Facto Press.

[20] Wu, Y. and Liu, Y. (2007). Robust truncated-hinge-loss support vector machines. *Journal of the American Statistical Association*, **102** 974–983.

[21] Zhang, H., Ahn, J., Lin, X. and Park, C. (2006). Gene selection using support vector machines with nonconvex penalty. *Bioinformatics*, **22** 88–95.

[22] Zhang, H., Liu, Y., Wu, Y. and Zhu, J. (2008). Variable selection for the multicategory SVM via adaptive sup-norm regularization. *Electronic Journal of Statistics*, **2** 149–167.

[23] Zhu, J., Hastie, T., Rosset, S. and Tibshirani, R. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, **5** 1391–1415.

[24] Zou, H. and Yuan, M. (2008). The $F_\infty$-norm support vector machine. *Statistica Sinica*, **18** 379–398.

Jian Guo
269 West Hall, 1085 South University
Department of Statistics
University of Michigan
Ann Arbor, MI 48109-1107
USA
E-mail address: guojian@umich.edu