Cancer Informatics

ORIGINAL RESEARCH

Improved Sparse Multi-Class SVM and Its Application for Gene Selection in Cancer Classification

Lingkang Huang[1,3,4], Hao Helen Zhang[2,5], Zhao-Bang Zeng[3] and Pierre R. Bushel[4]

[1]GlaxoSmithKline, Research and Development, Division of Quantitative Sciences, Research Triangle Park, NC 27709, USA. [2]Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA. [3]Bioinformatics Research Center, North Carolina State University, Raleigh, NC 27695, USA. [4]Biostatistics Branch, National Institute of Environmental Health Sciences, Research Triangle Park, NC 27709, USA. [5]Department of Mathematics, University of Arizona, Tucson, AZ 85718. Corresponding author email: lingkang.huang@gmail.com

Abstract

Background: Microarray techniques provide promising tools for cancer diagnosis using gene expression profiles. However, molecular diagnosis based on high-throughput platforms presents great challenges due to the overwhelming number of variables versus the small sample size and the complex nature of multi-type tumors. Support vector machines (SVMs) have shown superior performance in cancer classification due to their ability to handle high dimensional low sample size data. The multi-class SVM algorithm of Crammer and Singer provides a natural framework for multi-class learning. Despite its effective performance, the procedure utilizes all variables without selection. In this paper, we propose to improve the procedure by imposing shrinkage penalties in learning to enforce solution sparsity.

Results: The original multi-class SVM of Crammer and Singer is effective for multi-class classification but does not conduct variable selection. We improved the method by introducing soft-thresholding type penalties to incorporate variable selection into multi-class classification for high dimensional data. The new methods were applied to simulated data and two cancer gene expression data sets. The results demonstrate that the new methods can select a small number of genes for building accurate multi-class classification rules. Furthermore, the important genes selected by the methods overlap significantly, suggesting general agreement among different variable selection schemes.

Conclusions: High accuracy and sparsity make the new methods attractive for cancer diagnostics with gene expression data and defining targets of therapeutic intervention.

Availability: The source MATLAB code are available from http://math.arizona.edu/~hzhang/software.html.

Keywords: support vector machine (SVM), multi-class SVM, variable selection, shrinkage methods, classification, microarray, cancer classification

Cancer Informatics 2013:12 143–153

doi: 10.4137/CIN.S10212

This article is available from http://www.la-press.com.

Introduction

With the boost of modern techniques such as microarrays and next-generation sequencing in biological sciences, more and more high-throughput data are generated and utilized for basic science and for translational medicine. A typical gene expression data set contains tens or hundreds of thousands (p) input variables, which greatly exceeds the sample size n, i.e., $p >> n$. Many classical multivariate analysis methods have difficulties in handling such data because of the curse of dimensionality. However, the support vector machine (SVM),[1,2] originally designed for binary classification, has shown success in learning large p small n data and is useful for cancer classification.[3,4]

Cancer classification using gene expression data often results in multi-class problems, classifying tumor cells to multiple subtypes. In previous studies, samples were defined as (x_i, y_i), $i = 1, \ldots, n$, where x_i is the gene expression profile of the ith sample and $y_i \in \{1, \ldots, K\}$ is the cancer type. There are several methods available to extend the binary SVM ($K = 2$) to $K \geq 3$. One common approach is to decompose the multi-class problem into multiple binary problems,[5,6] using one-versus-rest or one-versus-one schemes, and combine learned multiple binary rules by a voting method. These approaches are useful in practice but have some limitations. First, the one-versus-rest approach may fail if no class dominates the union of the others.[7] Second, the one-versus-rest approach tends to yield unbalanced classification problems, especially if one class is much smaller than the union of remaining classes. Third, the one-versus-one approach trains each classifier based on only a portion of samples, which may increase the solution variablity. Fourth, these procedures do not effectively capture the correlation between different classes.[8] For example, tumor sub-types are more correlated to each other than to normal samples.

A better method for handling multi-class problems is to separate all the classes by estimating K discriminating functions $(f_1(x), f_2(x), \ldots, f_K(x))$ simultaneously. The decision rule is then defined as $\Phi(X) = \arg\max_{k=1}^{K} f_k(X)$, assigning the label r to an input x if $f_r(x)$ gives the highest value. Several generalized loss functions have been proposed for multi-class SVMs (MSVMs), including Weston and Watkins (1999),[9] Crammer and Singer (2001),[8] Lee et al. (2004),[7] and Liu and Shen (2006).[10] Among those available, the loss function used by Crammer and Singer[8] and Liu and Shen[10] gives a natural extension of the hinge loss from binary to multi-class problems, which is our main focus in this paper.

Besides classification, another question of primary interest to biologists is to identify important genes for tumor classification. Since including too many redundant variables in a model may negatively impact its prediction performance,[3] variable selection is important and necessary for accurate cancer classification. The redundant variables include both noise variables and variables which are highly correlated with the predictor variables. Furthermore, building a sparse and more interpretable model can reduce the number of follow-up experiments to a manageable size. One common approach of variable selection is gene-ranking: first, rank genes using univariate measurements such as p-values from hypothesis tests or correlation coefficients between individual inputs and the response, then sequentially add/remove genes to/from the model, and finally select the model based on cross-validation or the validation error. Despite their popularity in practice, gene-ranking methods have some drawbacks. First, genes are pre-selected based on individual effects, so their combined effects cannot be taken into account. This can be an issue since many genes tend to be highly correlated. In addition, these procedures separate variable selection and classification in two stages, and hence selected variables are not guaranteed to contribute significantly to the final classifier. This may result in suboptimal performance of classification.

The standard SVMs are equipped with L_2 penalty for regularization; see Guyon et al. (2002)[11] for the binary SVM and Lee et al. (2004)[7] for the MSVM. Since L_2 penalty shrinks the fitted coefficients towards zero, it effectively controls the model variability and improves prediction performance especially when many variables are highly correlated.[3] However, L_2 penalty can not set small coefficients to exactly zeros, so all variables are utilized in the learned model. For the purpose of variable selection, Bradley and Mangasarian[12] introduced L_1 penalty to the binary SVM for achieving sparsity in the solution. By shrinking small coefficients to exact zeros, L_1 SVM can build a parsimonious model with more accuracy than the standard L_2 SVM when many redundant variables

exist. A large p and small n data set can be directly fed into the L_1 model without pre-screening.

In this paper, we consider variable selection for the multi-class SVM, which is more challenging than the binary case because of the increased complexity in multi-class learning. The work on the MSVM variable selection in literature is limited but includes Wang et al. (2007),[13] Lee et al. (2006),[14] and Zhang et al. (2008).[15] In particular, Wang et al.[13] studied the L_1-norm MSVM and developed the solution path algorithm, while Zhang et al.[15] proposed a new penalty form, called the sup-norm penalty, which was shown to lead to more sparse models than L_1 penalty. Lee et al.[14] proposed to first make a functional ANOVA decomposition for the decision function and then conduct variable selection by imposing a soft-thresholding penalty on the functional components. All of these methods are based on the loss function of Lee et al.[7]

In this work, we suggest several new variable selection procedures for MSVM based on the loss function of Crammer and Singer.[8] Compared to other loss functions, this particular function provides a direct generalization of the hinge loss in binary SVMs and has a natural interpretation in terms of the functional margin. In practice, the resulting classifiers have shown competitive performance. We first considered linear classification problems. A group of regularization problems are proposed for sparse multi-class learning, and the computational algorithms are discussed as well. We then extended the methods to nonlinear cases. Our methods are particularly useful for analyzing large p and small n data, for example, high dimensional microarray or other "-omics" data. We applied the methods to two microarray data sets, acute leukemia study[16] and small round blue cell tumors.[17] The results suggest promising performance of the new methods in terms of accurately predicting the classes using a minimal number of genes.

Methods

Given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$, where $x_i \in R^p$ and $y_i \in \{1,2, \ldots, K\}$, the goal of multiclass classification is to learn the optimal decision rule $\Phi : R^p \rightarrow \{1,2, \ldots, K\}$ which can accurately predict labels for future observations. For the MSVM, we need to learn multiple discriminant functions $f(x) = (f_1(x), \ldots, f_K(x))$, where $f_k(x)$ represents the strength of evidence that a sample x belongs to class k. The decision rule is $\Phi(X) = \arg\max_{k=1,\ldots,K} f_k(X)$, and the classification boundary between any two classes k and l is $\{x : f_k(x) = f_l(x)\}$ for $k \neq l$.

When $K = 2$, the label y is coded as $\{+1, -1\}$ by convention. Consider the linear classifier $f(\mathbf{x}) = \beta_0 + x^T\beta$. The binary SVM minimizes $\|\beta\|^2 + \lambda \sum_{i=1}^n \xi_i$ subject to the following constraint, depending on whether the data are separable:

Binary SVM

$$\begin{cases} \text{separable case: } y_i f(x_i) \geq 1, \forall i, & (1) \\ \text{non-separable: } y_i f(x_i) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i & (2) \end{cases}$$

In the binary SVM objective function, the term $\|\beta\|^2 = \sum_{j=1}^p \beta_j^2$ controls the width of the margin, the quantity $\sum_{i=1}^n \xi_i$ is an upper bound for the misclassification error on the training set when data are non-separable and $\lambda > 0$ is the tuning parameter. Equivalently, the binary SVM can be formulated as a regularization problem using the hinge loss function as: $\min_f \sum_{i=1}^n [1 - y_i f(x_i)]_+ + \lambda \|\beta\|^2$.

Crammer and Singer[8] extended the hinge loss from the binary SVM to multi-class problems. In the separable case, the discriminating functions are required to satisfy constraint (3) for all observations: if x belongs to class y, then $f_y(x)$ is greater than any $f_k(x)$ with $k \neq y$ by at least margin 1. In the non-separable case, $\xi_i \geq 0$ are introduced to get the relaxed constraint (4):

MSVM

$$\begin{cases} \text{separable case: } f_{yi}(x_i) - \max\limits_{\substack{k \neq y_i \\ k=1,\ldots,K}} f_k(x_i) \geq 1, & (3) \\ \text{no-separable: } f_{yi}(x_i) - \max\limits_{\substack{k \neq y_i \\ k=1,\ldots,K}} f_k(x_i) \geq 1 - \xi_i & (4) \end{cases}$$

For linear classification, we assume $f_k(x) = \beta_{k0} + x^T\beta_k$ for $k = 1, \ldots, K$. The MSVM of Crammer and Singer[8] solves:

$$\min_{\beta, \beta_0, \xi} \sum_{i=1}^n \xi_i + \lambda \sum_{k=1}^K \|\beta_k\|^2 \qquad (5)$$

$$\text{subject to: } f_{yi}(x_i) - \max_{k \neq y_i} f_k(x_i) \geq 1 - \xi_i,$$
$$\xi \geq 0, \quad \forall i.$$

To avoid estimation redundancy, the constraint $\sum_{k=1}^{K} f_k = 0$ is often invoked. In (5), $\sum_{i=1}^{n} \xi_i$ bounds the training error, and $\sum_{k=1}^{K} \| \beta_k \|^2 = \sum_{k=1}^{K} \sum_{j=1}^{p} \beta_{kj}^2$ controls model complexity. The problem can be solved by quadratic programming (QP). As Liu and Shen[10] shows, this formulation has a natural interpretation of minimizing a generalized hinge loss $[1 - \min_{k \neq y} g_k(f(\mathrm{x}), y)]+$, where $g_k = f_y(\mathrm{x}) - f_k(\mathrm{x})$. The generalized function margin of f is defined as the vector $\mathrm{g} = (\mathrm{g}_1, \ldots, g_{y-1}, \mathrm{g}_{y+1}, \ldots, g_K)$.

Crammer and Singer[8] imposed L_2 penalty on the coefficients β in (5). The resulting solution utilizes all variables, which may diminish the prediction accuracy when there are many redundant noise variables. In the following sections, we utilize the same loss function but suggest different penalty forms to control model complexity and achieve sparse solutions. In particular, we investigate four different penalties: L_1 penalty, adaptive L_1 penalty, sup-norm penalty and adaptive sup-norm penalty, and discuss computational algorithms for each type of regularization.

L_1 Penalty: The L_1 penalty is also known as LASSO penalty.[18] The MSVM learning with L_1 penalty solves:

$$\begin{aligned} \min_{\beta, \beta_0, \xi} \quad & \sum_{i=1}^{n} \xi_i + \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} | \beta_{kj} | \\ \text{subject to:} \quad & f_{yi}(\mathrm{x}_i) - \max_{k \neq yi} f_k(\mathrm{x}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ \forall i. \end{aligned} \tag{6}$$

To eliminate the absolute operation in (6), define $| \beta_{kj} | = \beta_{kj}^+ + \beta_{kj}^-$ and $\beta_{kj} = \beta_{kj}^+ - \beta_{kj}^-$, where $\beta_{kj}^+ = \beta_{kj}$ if $\beta_{kj} \geq 0$, or 0, otherwise; $\beta_{kj}^- = -\beta_{kj}$ if $\beta_{kj} \leq 0$, or 0, otherwise. Then, the L_1 MSVM can be expressed as the following linear programming (LP) equation:

$$\min_{\beta, \beta_0, \xi} \quad \sum_{i=1}^{n} \xi_i + \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} (\beta_{kj}^+ + \beta_{kj}^-) \tag{7}$$

$$\begin{aligned} \text{s.t.:} \quad & \sum_{j=1}^{p} (\beta_{yj}^+ - \beta_{yj}^-) x_{ij} - \sum_{j=1}^{p} (\beta_{kj}^+ - \beta_{kj}^-) x_{ij} \geq 1 - \xi_i, \\ & \text{for } i = 1, \ldots, n; \quad k = 1, \ldots, K, \ k \neq y_i \\ & \sum_{k=1}^{K} \beta_{k,0} = 0; \sum_{k=1}^{K} (\beta_{kj}^+ - \beta_{kj}^-) = 0, \ j = 1, \ldots, p \\ & \beta_{kj}^+ \geq 0, \ \beta_{kj}^- \geq 0, \ \xi_i \geq 0, \ \forall k, j, i. \end{aligned}$$

Adaptive L_1 Penalty: The adaptive L_1 penalty, also known as the adaptive LASSO, was first studied in various regression models.[19–21] Instead of applying the same penalty to coefficients, the adaptive L_1 penalty assigns different penalties to coefficients adaptively: large coefficients receive small penalties, while small coefficients receive large penalties. In this way, large coefficients can be protectively preserved during the selection process and small coefficients are decreased to zero more, resulting more sparse models. We propose the adaptive L_1 MSVM by solving the following optimization problem:

$$\begin{aligned} \min_{\beta, \beta_0, \xi} \quad & \sum_{i=1}^{n} \xi_i + \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} W_{kj} | \beta_{kj} | \\ \text{subject to:} \quad & f_{yi}(\mathrm{x}_i) - \max_{k \neq yi} f_k(\mathrm{x}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ \forall i. \end{aligned} \tag{8}$$

Choices of weights in (8) are essential to adaptive procedures. We propose the construction of weights as $W_{kj} = | \tilde{\beta}_{kj} |^{-1}$, where $\tilde{\beta}_{kj}$'s are the solution to the standard L_2 MSVM (5), as the ridge penalty generally produces stable and robust estimates even when collinearity exists among covariates. The optimization problem of adaptive L_1 MSVM has the same constraints as L_1 MSVM, with the objective function (7) replaced by the following function:

$$\min_{\beta, \beta_0, \xi} \quad \sum_{i=1}^{n} \xi_i + \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} \frac{\beta_{kj}^+ + \beta_{kj}^-}{| \tilde{\beta}_{kj} |}. \tag{9}$$

Sup-norm Penalty: In K-class learning problems, we need to fit K functions $(f_1(\mathrm{x}), \ldots, f_K(\mathrm{x}))$. These functions are associated with a $K \times p$ coefficients matrix (β_{kj}), $1 \leq k \leq K$, $1 \leq j \leq p$. In theory, if the jth variable is unimportant, then all the coefficients $\{\beta_{kj}, k = 1, \ldots, K\}$ should be zero. Motivated by this, Zhang et al.[15] suggested to penalize the maximum absolute value of K coefficients associated with each variable, i.e., $\eta_j = \max_{k=1,\ldots,K} | \beta_{kj} |$ for $j = 1, \ldots, p$. It is clear that if $\hat{\eta}_j = 0$, then $\hat{\beta}_{kj} = 0$ for all $1 \leq k \leq K$. We propose to solve:

$$\min_{\beta,\beta_0,\xi} \quad \sum_{i=1}^{n}\xi_i + \lambda\sum_{j=1}^{p}\eta_j \tag{10}$$

$$\text{s.t.:} \quad \sum_{j=1}^{p}(\beta_{yj}^{+}-\beta_{yj}^{-})x_{ij} - \sum_{j=1}^{p}(\beta_{kj}^{+}-\beta_{kj}^{-})x_{ij} \geq 1-\xi_i,$$
$$\text{for } i=1,\ldots,n;\ k=1,\ldots,K,\ k\neq y_i,$$
$$\eta_j \geq \beta_{kj}^{+}+\beta_{kj}^{-}, k=1,\ldots,K;\ j=1,\ldots,p,$$
$$\sum_{k=1}^{K}\beta_{k,0}=0;\ \sum_{k=1}^{K}(\beta_{kj}^{+}-\beta_{kj}^{-})=0,\ j=1,\ldots,p$$
$$\beta_{kj}^{+}\geq 0, \beta_{kj}^{-}\geq 0, \eta_j \geq 0, \xi_i \geq 0, \forall k,j,i.$$

Adaptive Sup-norm Penalty: The adaptive sup-norm penalty shares the same motivation as the adaptive L_1 penalty: important variables are given small penalties and noise variables are given large penalties. In particular, we replace the second term in (10) by $\lambda\sum_{j=1}^{p} w_j\eta_j$. To Construct the weights, we propose to use $w_j = \tilde{\eta}_j^{-1}$ for all j, where $\tilde{\eta}_j = \max_{k=1,\ldots,K} |\tilde{\beta}_{kj}|$ and $\tilde{\beta}_{kj}$'s are the solution to L_2 MSVM (5). If $\tilde{\eta}_j$ is large, then w_j is small and η_j is given a small penalty and vice-versa. The resulting optimization problem has the same constraints as the sup-norm MSVM, with the objective function in (10) replaced as the following:

$$\min_{\beta,\beta_0,\xi} \quad \sum_{i=1}^{n}\xi_i + \lambda\sum_{j=1}^{p}\frac{\eta_j}{\tilde{\eta}_j}. \tag{11}$$

Nonlinear extension

We have given four new regularization forms of MSVM for variable selection in linear classification. Next, we show that these methods can be easily extended to the non-linear case by using the basis expansion approach. Let $\mathbf{h}(\mathbf{x}) = \{h_1(\mathrm{x}), h_2(\mathrm{x}), \ldots, h_q(\mathrm{x})\}$ be a dictionary of basis functions transformed from $\mathbf{x}$. We construct the decision function as $f_k(\mathrm{x}) = \beta_{k0} + \sum_{j=1}^{q}(h_k(\mathrm{x}))_j\beta_{kj}$, which is linear in the transformed space but nonlinear in terms of the original $\mathbf{x}$. The new design matrix is $\mathbf{H} = (h_k(\mathrm{x}_i))_{n\times q}$. For implementation, we simply treat $\mathrm{h}_i = (\mathrm{h}(\mathrm{x}_i))_{1\times q}$ as x_i and replace x_{ij} with h_{ij} in the above four regularization forms. With this approach, note that variable selection is conducted for the transformed features $\{h_1(\mathrm{x}), h_2(\mathrm{x}), \ldots, h_q(\mathrm{x})\}$. Therefore, we suggest to use the nonlinear transformations which are interpretable, such as the polynomial transformation.

Model tuning

The choice of tuning parameter λ is crucial in the above regularization problems, since it controls the trade-off between the training error and generalization performance of classifiers. It also has an impact on sparsity of the solution. To select the optimal λ, we use a validation set in simulated examples and use five-fold cross validation in real data analysis. A fine grid search is conducted over a wide range of values of λ, and the best λ is identified as the one which gives the least tuning error or cross validation error.

Results and Discussion

Simulation study

We illustrate the performance of new methods for prediction and variable selection in both linear and nonlinear settings using simulated data sets. The Bayes rule and L_2 MSVM of Crammer and Singer[8] (denoted as "L2 MSVM (C&S)") are also included. The Bayes rule is the optimal classification rule if the underlying distribution of the data is known. It serves as the golden standard to evaluate the performance of the trained classifiers. We conducted 100 simulations for each classification method and report the average performance of the methods, including test error on test samples, model size, and the total selection frequency of individual inputs in 100 runs.

Linear example

This is a linear classification problem with p = 20 and k = 4. The first two components of x from class k are from $N(\mu_k, \sigma^2 I_2)$, with μ's values being $(\sqrt{2},\sqrt{3}),(-\sqrt{3},\sqrt{2}),(-\sqrt{2},-\sqrt{3}),(\sqrt{3},-\sqrt{2})$. Here $\sigma=\sqrt{2}$ and I_2 is the identity matrix of size 2. Thus, the x_1 and x_2 marginally both follow a mixture of normal distributions with $E(\mathrm{x}_i)=0$ and $\mathrm{Var}(\mathrm{x}_i)=4.5, i=1,2$. The rest of the 18 components of $\boldsymbol{x}$ are i.i.d. from $N(0, 1)$. To introduce some informative but redundant variables, two new variables x_3' and x_4', which are highly correlated with x_1 and x_2, were generated to replace the noise variables x_3 and x_4. Let correlation parameters $\rho_1 = 0.8$ and $\rho_2 = 0.9$, $x_3' = \rho_1 * x_1/\sqrt{4.5} + \sqrt{(1-\rho_1^2)} * x_3$ and $x_4' = \rho_2 * x_2/\sqrt{4.5} + \sqrt{(1-\rho_2^2)} * x_4$. So, only x_1 and x_2 are important; the x_3' and x_4' are two variables

highly correlated with x_1 and x_2; x_5~x_{20} are noise variables. Two hundred training and 200 tuning samples, with equal samples from each class, were generated to learn and tune the model. 40,000 test samples were generated to evaluate the model performance.

Table 1 reports the selection frequency of each variable over 100 runs. Important variables x_1 and x_2 are never missed by any method. The rest of the variables, either noise variables or informative but redundant variables, are selected with different frequencies by different methods. The adaptive sup-norm MSVM selects noise or informative but redundant variables with fewer than 10 times in 100 runs, which is a much lower selection frequency than that of L_1 MSVM. Furthermore, all methods except L_2 MSVM can handle informative but redundant variables very well. The x_3' and x_4', which are correlated to important variables x_1 and x_2 with $\rho_1 = 0.8$ and $\rho_2 = 0.9$, are selected fewer than 15 times in 100 runs using any of four proposed methods, which is fewer most noise variables.

Table 2 summarizes the average test error and model size of 100 runs. The numbers in the parentheses are standard errors (SE) of the mean of test errors from 100 simulations. The Bayes error (i.e., the optimum classification error) is 0.246 and L_2 MSVM has test error 0.296. All new methods are statistically better than L_2 MSVM, with adaptive sup-norm MSVM giving the smallest test error 0.255. Adaptive penalties tend to enhance model sparsity, and the adaptive sup-norm yields the most compact model of size 3.25 on average. Overall, adaptive sup-norm MSVM is the best for both variable selection and prediction accuracy.

Nonlinear example

Consider a nonlinear three-class example in a large p small n setting. Generate $x \in R^{20}$ as following: (x_1, x_2) are uniformly distributed in the square $[-3,3] \times [-3,3]$, and the remaining 18 components $x_3, \ldots, x_{20}$ are i.i.d. from $N(0, 2)$. Define the three functions:

$$f_1(x) = -2x_1 + 0.2x_1^2 - 0.4x_2^2 + 0.2,$$
$$f_2(x) = -0.4x_1^2 + 0.8x_2^2 - 0.4,$$
$$f_3(x) = 2x_1 + 0.2x_1^2 - 0.4x_2^2 + 0.2.$$

For $\mathbf{x}$, its class label is assigned using the multinomial sampling $(p_1(x), p_2(x), p_3(x))$ with $p_k(x) \propto f_k(x)$.

Table 1. Selection frequency of individual variables over 100 runs for the linear example.

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 MSVM (C&S)	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
L1 MSVM	100	100	13	4	26	21	31	28	29	20	22	22	28	24	22	20	25	32	26	23
Sup MSVM	100	100	10	5	15	16	14	14	21	13	13	16	13	16	9	9	14	18	16	17
Adapt-L1 MSVM	100	100	11	13	14	8	12	13	14	13	11	10	16	8	9	12	11	11	11	11
Adapt-Sup MSVM	100	100	6	4	7	8	8	6	9	10	9	4	9	5	6	5	6	8	7	8

Table 2. Average test error and model size for the linear example.

Method	Test error (SE)	Selected variables		Model size
		Important 2 var.	Noise or informative but redundant 18 var.	
L2 MSVM (C&S)	0.296 (1.4×10^{-3})	2	18	20
L1 MSVM	0.263 (1.4×10^{-3})	2	4.16	6.16
Adapt-L1 MSVM	0.262 (1.0×10^{-3})	2	2.08	4.08
Sup MSVM	0.258 (1.2×10^{-3})	2	2.49	4.49
Adapt-Sup MSVM	0.255 (6.1×10^{-4})	2	1.25	3.25
Bayes	0.246 (–)	2	0	2

Thus, the classification boundary is nonlinear and determined only by x_1, x_1^2 and x_2^2. We fit the nonlinear MSVM by including 20 main effects, all quadratic effects and their 2-way interaction effects as basis functions, which results in totally $P = 230$ terms in the model. Let $n = 120$, thus, $p > n$. Additional 120 tuning samples were generated for tuning the optimal λ and 30,000 test samples were generated to evaluate the model performance.

Table 3 reports the average test error and model size over 100 runs for each method. Note that L_1 MSVM and sup-norm MSVM are equivalent for three-class problems.[15] The Bayes error is 0.120, L_2 MSVM has the test error 0.441, and all the new methods show a significant improvement over L_2 MSVM. Adaptive sup-norm MSVM gives the smallest error 0.147, very close to the Bayes error. L_2 MSVM does not perform well here, mainly due to a large number of noise variables contained in data. With regard to variable selection, L_2 MSVM includes almost all variables in the fitted model, and the average model size is 221.87. The new MSVMs produce much smaller models while identifying the three important variables correctly. Adaptive sup-norm MSVM yields the most parsimonious model of size 8.58 on average. Adaptive L_1 MSVM works similarly, with test error 0.152 and on average, selecting nine variables. Again, adaptively-weighted penalties are shown to produce more sparsity than equally-weighted penalties.

Table 4 summarizes the selection frequency of each term in the adaptive sup-norm MSVM model: those of main effects given in the first row, those of quadratic terms given on the main diagonal line, and those of 190 two-way interaction terms given in intersections of the corresponding rows and columns. We observe that the three important effects (x_1, x_1^2, x_2^2) are always selected, and noise variables are selected with a low frequency (fewer than 10 times in 100 runs).

Real data

One important application of our new methods is classification and variable selection of microarray or other "-omics" data. We analyze two cancer gene expression data sets: leukemia data[16] and small round blue cell tumor data.[17] In addition to distinguishing multi-type tumors, another primary goal is to identify signature genes which are responsible for classification and helpful for understanding the underlying mechanism of cancer. Since microarray data typically represent a large number of genes ($p >> n$), one common practice is selecting relevant genes before building a classifier. A popular approach of

Table 3. Average test error and model size for the nonlinear example.

Method	Test error (SE)	Selected variables		Model size
		Important 3 var.	Noise 227 var.	
L2 MSVM (C&S)	0.441 (2.1×10^{-3})	3	218.87	221.87
L1/Sup MSVM	0.160 (2.4×10^{-3})	3	18.34	21.34
Adapt-L1 MSVM	0.152 (2.1×10^{-3})	2.98	6.08	9.06
Adapt-Sup MSVM	0.147 (1.9×10^{-3})	3	5.58	8.58
Bayes	0.120 (–)	3	0	3

Table 4. The variable selection frequencies of adaptive sup-norm MSVM over 100 runs for the nonlinear example.

List	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}
	100	1	0	2	0	3	0	1	0	0	1	0	2	2	2	0	1	0	0	1
x_1	**100**	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_2	9	**100**	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_3	0	5	6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_4	1	4	3	8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_5	4	3	3	1	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_6	1	1	1	2	1	7	.	.	.	.	.	.	.	.	.	.	.	.	.	.
x_7	3	1	1	4	2	3	5	.	.	.	.	.	.	.	.	.	.	.	.	.
x_8	3	4	5	2	2	0	0	5	.	.	.	.	.	.	.	.	.	.	.	.
x_9	2	6	2	2	1	2	2	2	6	.	.	.	.	.	.	.	.	.	.	.
x_{10}	1	4	4	2	4	2	1	3	2	7	.	.	.	.	.	.	.	.	.	.
x_{11}	1	8	1	4	2	2	1	4	1	1	7	.	.	.	.	.	.	.	.	.
x_{12}	2	3	3	1	3	0	4	2	2	2	1	9	.	.	.	.	.	.	.	.
x_{13}	2	5	2	0	3	2	2	4	2	3	0	4	9	.	.	.	.	.	.	.
x_{14}	2	3	3	2	1	5	1	4	1	4	3	3	3	5	.	.	.	.	.	.
x_{15}	2	2	3	0	6	3	1	4	3	3	3	1	2	1	5	.	.	.	.	.
x_{16}	0	6	0	5	4	2	0	4	2	4	0	2	4	0	3	4	.	.	.	.
x_{17}	1	5	0	2	1	4	2	2	2	1	3	1	1	1	3	2	4	.	.	.
x_{18}	0	4	3	2	1	0	2	3	2	2	4	3	3	1	2	0	2	4	.	.
x_{19}	0	4	3	4	6	3	3	0	1	3	1	2	2	3	0	1	3	2	9	.
x_{20}	1	3	2	4	3	1	3	3	3	0	0	0	3	1	2	0	2	1	1	6

gene selection is gene ranking based on univariate statistics such as F-statistic and *p*-value. The weaknesses of ranking methods include: (1) classification and variable selection are performed separately and (2) the correlation and interaction among genes cannot be fully taken into account. However, rank-based screening has been found useful at an initial step by filtering irrelevant features and therefore beneficial to the refined variable selection process that follows, as in Lee et al.,[7] Wang and Shen,[13] Zhang et al.,[15] and so on. Pre-screening is commonly used in microarray data analysis to remove genes which do not contribute expression changes across the samples (i.e., those that are considered flat), as uninformative genes add noise to the downstream analysis. In practice, it is recommended to conduct two-stage modeling: feature screening (based on simple tests) followed by formal model building (based on more sophisticated variable selection procedures) to enhance the final variable selection results. We adopted the two-stage modeling in our real data analysis. Compared to univariate analysis done in most gene-ranking approaches, our new classification methods conduct joint selection and can account for gene-gene interactions naturally. The following results show that the methods effectively select important genes and achieve high accuracy at the same time. Therefore, they provide alternative promising tools for cancer classification using gene expression data.

Leukemia study

The leukemia study[16] analyzed human bone marrow samples using oligonucleotide microarrays produced by Affymetrix. The data consist of 7129 probe sets, which represent 6817 human genes and 72 samples in three classes: acute myeloid leukemia (AML), T-cell, and B-cell acute lymphoblastic leukemia (ALL_T and ALL_B). There are 38 training samples (19 ALL_B, 8 ALL_T, 11 AML) and 34 test samples (19 ALL_B, 1 ALL_T, 14 AML). We preprocessed the data following Dudiot et al.[22] and selected the subset of 742 genes by F-ratio test for memory and computational efficiency. Then, L_2 MSVM and four new approaches were applied for gene selection as well as model building. Variable selection and parameter choice during model building were done strictly on the training data set.

Table 5 shows that L_2 MSVM only misclassifies 1 out of 34 test samples, but its solution depends on a large number of genes (429 genes). In contrast, our new methods select a very small set of genes (14, 9, 4 genes for L_1 MSVM, adaptive L_1 MSVM, and

Table 5. Classification and selection results for the leukemia study.

Method	Test error	No. of genes
L2 MSVM (C&S)	1/34	429
L1/Sup MSVM	2/34	14
Adapt-L1 MSVM	3/34	9
Adapt-Sup MSVM	3/34	4

adaptive sup-norm MSVM respectively) while giving comparable accuracy. Table 6 shows a significant overlap in the selection: all four genes selected by adaptive sup-norm MSVM are also selected by others, and 8 of 9 genes selected by adaptive L_1 MSVM are selected by L_1 MSVM. Not all these genes are top-ranked by F-test, which does not take into account gene interactions.

To interpret the role of selected genes in classification, we now examine the three discriminant functions given by adaptive sup-norm MSVM:

$$\begin{aligned}\hat{f}_{\text{ALL\_B}} &= -0.037*\text{TCRB} - 0.330*\text{MAL} \\ &\quad -0.640*\text{CST3} + 0.091*\text{TCL1}, \\ \hat{f}_{\text{ALL\_T}} &= 0.162*\text{TCRB} + 0.450*\text{MAL}, \\ \hat{f}_{\text{AML}} &= -0.124*\text{TCRB} - 0.121*\text{MAL} \\ &\quad + 0.640*\text{CST3} - 0.091*\text{TCL1}.\end{aligned}$$

Each test sample has three predicted decision values $(\hat{f}_{\text{ALL\_B}}, \hat{f}_{\text{ALL\_T}}, \hat{f}_{\text{AML}})$ and assigned to a class with the largest value. T-cell receptor, beta cluster (*TCRB*), and *MAL* genes have positive coefficients in $\hat{f}_{\text{ALL\_T}}$ and negative coefficients in $\hat{f}_{\text{ALL\_B}}$ and $\hat{f}_{\text{AML}}$, and are useful to separate ALL_T from the other two classes. This pattern is also confirmed by Figure 1, which illustrates the hierarchical clustering structure of the data corresponding to the four selected probe sets (i.e., four genes). *TCRB* (X00437_s_at) and *MAL* (X76223_s_at) have high expression values (in red) for most ALL_T samples and low expression (in green) for most of the ALL_B and AML samples. The relevance of the *MAL* gene with T-cell ALL was reported in the literature. For example, the *MAL* gene shows significant higher expression level in acute T-cell leukemia/lymphoma than in chronic T-cell leukemia.[23] Gene Cystatin C (*CST3*) is helpful in distinguishing all three classes, since its coefficient is zero in $\hat{f}_{\text{ALL\_T}}$, is negative in $\hat{f}_{\text{ALL\_B}}$, and is positive in $\hat{f}_{\text{AML}}$. Correspondingly, gene *CST3* (M27891_at) has low values in most ALL_B samples but high values in most AML samples in Figure 1. *CST3* is one of the genes reported by Golub,[16] which can differentiate the ALL vs. AML. Gene T-cell leukemia/lymphoma 1 (*TCL1*; X82240_rna1_at) reveals the opposite patterns, which has high values in most ALL_B samples but low values in most AML and ALL_T samples (Fig. 1). It is reported that *TCL1* shows significant higher expression during pre-B-cell acute lymphoblastic leukemia progression.[24] All four genes have been individually or jointly identified as one of the predictor genes to

Table 6. Selected genes by various methods for the leukemia study.

Probe set ID	Adapt-sup	Adapt-L1	L1/Sup	Rank of F-test	Gene name/description
X00437_s_at	1	1	1	1	TCRB (T-cell receptor, beta cluster)
X76223_s_at	1	1	1	3	MAL gene
M27891_at	1	1	1	12	CST3 (Cystatin C)
X82240_rna1_at	1	1	1	19	TCL1 (T-cell leukemia/lymphoma)
X59871_at	–	1	1	8	TCF7 (Transcription factor 7; T-cell specific)
M11722_at	–	1	1	157	Terminal transferase mRNA
U89922_s_at	–	1	1	324	LTB (Lymphotoxin-beta)
Z14982_rna1_at	–	1	1	527	MHC-encoded proteasome subunit gene LAMP 7-E1 gene
M21624_at	–	1	–	462	TCRD (T-cell receptor, delta)
U05259_rna1_at	–	–	1	10	MB-1 gene
X58529_at	–	–	1	27	IGHM Immunoglobulin mu
M74719_at	–	–	1	46	SEF2-1A mRNA, 5′ end
Y00787_s_at	–	–	1	58	Interleukin-8 precursor
M19507_at	–	–	1	112	MPO (Myeloperoxidase)
U01317_cds4_at	–	–	1	390	Delta-globin gene

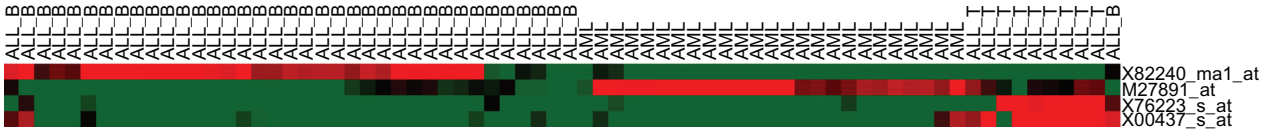

Figure 1. Hierarchical clustering of all training and test samples based on 4 selected genes in leukemia study.
Notes: All samples, including 38 training and 34 test samples, are plotted. Each column represents a sample from one of classes: AML, ALL_T or ALL_B. Each row represents the expression profile of a gene (labeled as a probe set ID) across all samples. The color scale ranges from green for an expression value less than the mean to red for an expression value greater than the mean. The hierarchical clustering result is generated using the public software Cluster (http://rana.lbl.gov/EisenSoftware.htm) and viewed by the Java TreeView (http://jtreeview.sourceforge.net/). The hierarchical clustering used Pearson correlation for gene similarity metric and average-linkage algorithm for clustering correlation matrixes.

differentiate between AML and ALL or among AML, ALL_T and ALL_B in the leukemia study using various analysis methods.[25–29] In particular, a penalized likelihood method,[29] called structured polychotomous machine, selected the exactly same four genes with the same prediction accuracy obtained in this study.

Small Round Blue Cell Tumor (SRBCT) study

The SRBCT data are from cDNA microarrays using standard protocols of the National Human Genome Research Institute (NHGRI).[17] There are 63 training and 20 test samples, categorized into 4 classes: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS). We began with 2308 genes available at http://research.nhgri.nih.gov/microarray/Supplement/, and conducted gene screening with F-ratio tests. We include the top 333 and bottom 300 genes for analysis and show results in Table 7. Variable selection and parameter choice during model building were done strictly on the training data set.

We observe that all the new methods have test error 0 except L1-norm SVM, which misclassifies 1 out of 20 test samples. With regard to gene selection, all the new methods successfully exclude the bottom 300 genes from the final model. The number of selected genes ranges between 28–36, with adaptive sup-norm MSVM selecting the smallest number of genes. Compared to other MSVM methods applied by Lee et al.[14] and Zhang et al.[15] on the same data set, our new methods give better or comparable prediction accuracy overall and they select a smaller number of genes. When examining the genes selected by the four new methods, we observe a large overlap across the final lists. In particular, 10 genes are commonly selected by all four methods, and 13 genes are selected by three methods, demonstrating general agreement among different variable selection schemes.

Table 7. Classification and selection results for the SRBCT study.

Method	Test error	Selected genes	
		Top 333 genes	Bottom 300 genes
L2 MSVM (C&S)	0/20	194	124
L1 MSVM	1/20	31	0
Sup MSVM	0/20	36	0
Adapt-L1 MSVM	0/20	31	0
Adapt-Sup MSVM	0/20	28	0

Conclusions

We proposed to improve the standard MSVM of Crammer and Singer[8] by constructing a new class of regularization methods which incorporates variable selection in the model learning. Performance of the new methods is demonstrated via numerical studies. Compared to the standard L_2 MSVM, the new methods are shown to achieve high prediction accuracy and are able to build sparse and more interpretable models. In both simulations and real data analyses, adaptive sup-norm MSVM shows the best performance among all the methods with regard to either variable selection or prediction accuracy. The combination of high accuracy and effective selection makes the new methods attractive for high-dimensional data analysis and powerful tools for cancer biomarker discovery based on gene expression data.

Authors Contributions

HZ and LH designed the penalized MSVMs. LH developed and implemented the method. HZ supervised the study. ZZ and PB provided valuable suggestions and evaluated the results. All authors contributed to writing this paper; proofread and approved the final manuscript.

Funding

This work was supported by National Science Foundation [DMS0645293 to HZ], National Institute of Health [R01CA085848 to HZ; R24GM078233, RC2GM092729 to ZZ], and National Institute of Environmental Health Sciences [ES102345-04 to PB].

Competing Interests

Author(s) disclose no potential conflicts of interest.

Disclosures and Ethics

As a requirement of publication the authors have provided signed confirmation of their compliance with ethical and legal obligations including but not limited to compliance with ICMJE authorship and competing interests guidelines, that the article is neither under consideration for publication nor published elsewhere, of their compliance with legal and ethical guidelines concerning human and animal research participants (if applicable), and that permission has been obtained for reproduction of any copyrighted material.This article was subject to blind, independent, expert peer review.The reviewers reported no competing interests.

References

1. Boser E, Guyon M, Vapnik V. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Conference on Computational Learning Theory; Pittsburgh, PA*. 1992:144–52.
2. Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995;20: 273–9.
3. Zhu J, Hastie T, Rosset S, Tibshirani R. 1-norm support vector machines. *Neural Information Processing Systems*. 2003;16:49–56.
4. Zhang HH, Ahn J, Lin X, Park C. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*. 2006;22:88–95.
5. Dietterich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*. 1995;2: 263–86.
6. Allwein EL, Schapire RE, Singer Y. Reducing multi-class to binary: A unifying approach for margin classifiers. In *Machine Learning: Proceedings of the Seventeenth International Conference*; 2000.
7. Lee Y, Lin Y, Wahba G. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association.* 2004;99:465: 67–81.
8. Crammer K, Singer Y. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research.* 2001;2:265–92.
9. Weston J, Watkins C. Support vector machines for multi-class pattern recognition. 1999:219–24.
10. Liu Y, Shen X. Multicategory psi-learning. *Journal of the American Statistical Association*. 2006;101:474–509.
11. Guyon I, Weston J, Barnhill S, Vapnik V. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*. 2002; 46:389–422.
12. Bradley PS, Mangasarian OL. Feature selection via concave minimization and support vector machines. In *Proceedings of the 13th International Conference on Machine Learning; CA*. 1998:82–90.
13. Wang L, Shen X. On L1-norm multi-class support vector machines: methodology and theory. *Journal of the American Statistical Association*. 2007;102:583–94.
14. Lee Y, Kim Y, Lee S, Koo J. Structured multicategory support vector machines with analysis of variance decomposition. *Biometrika*. 2006;93: 555–71.
15. Zhang HH, Liu Y, Wu Y, Zhu J. Variable selection for multicategory SVM via sup-norm regularization. *Electronic Journal of Statistics*. 2008;2:149–67.
16. Golub TR, Slonim DK, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*. 1999;286:531–7.
17. Khan J, Wei JS, Ringner M, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine.* 2001;7:673–9.
18. Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society, B*. 1996;58:267–88.
19. Zou H. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association.* 2006;101:1418–29.
20. Zhang HH, Lu W. Adaptive-LASSO for Cox's proportional hazard model. *Biometrika*. 2007;94:691–703.
21. Wang H, Li G, Jiang G. Robust regression shrinkage and consistent variable selection via the LAD-LASSO. *Journal of Business & Economics Statistics.* 2007;25:347–55.
22. Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association.* 2002;97:77–87.
23. Kohno T, Moriuchi R, Katamine S, Yamada Y, Tomonaga M, T M. Identification of genes associated with the progression of adult T cell leukemia (ATL). *Jpn J Cancer Res*. 2000;91(11):1103–10.
24. Fears S, Chakrabarti SR, Nucifora G, Rowley JD. Difierential expression of TCL1 during pre-B-cell acute lymphoblastic leukemia progression. *Cancer Genet Cytogenet*. 2002;135(2):110–9.
25. Huang L. An integrated method for cancer classification and rule extraction from microarray data. *J Biomed Sci.* 24 2009;16:25.
26. Krishnapuram B, Carin L, Hartemink A. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *J Comput Biol.* 2004;11(2–3):227–42.
27. Reverter F, Vegas E, Sanchez P. Mining gene expression profiles: an integrated implementation of kernel principal component analysis and singular value decomposition. *Genomics Proteomics Bioinformatics.* 2010;8(3):200–10.
28. Wang H, Huang D. A gene selection algorithm based on the gene regulation probability using maximal likelihood estimation. *Biotechnol Lett.* 2005;27(8):597–603.
29. Koo JY, Sohn I, Kim S, Lee JW. Structured polychotomous machine diagnosis of multiple cancer types using gene expression. *Bioinformatics*. 2006;22(8):950–8.