DEPARTMENT OF STATISTICS

North Carolina State University

2501 Founders Drive, Campus Box 8203

Raleigh, NC 27695-8203

Institute of Statistics Mimeo Series No. 2573

August 2004

Multiclass Proximal Support Vector Machines

Yongqiang Tang Hao Helen Zhang

Department of Statistics, North Carolina State University, Raleigh, NC

`ytang,hzhang2@stat.ncsu.edu`

Supported in part by National Science Foundation grants DMS-0405913.

MULTICLASS PROXIMAL SUPPORT VECTOR MACHINES

YONGQIANG TANG, HAO HELEN ZHANG

ABSTRACT. We propose an extension of proximal support vector machines (PSVM) to the multi-class case. Unlike the one-versus-rest approach that constructs the decision rule based on multiple binary classification tasks, the multiclass PSVM (MPSVM) considers all classes simultaneously and provides a unifying framework when there are either equal or unequal misclassification costs. The MPSVM is built in a regularization framework of reproducing kernel Hilbert space (RKHS) and implements the Bayes rule asymptotically. With regard to computation, the MPSVM simply solves a system of linear equations and demands much less computational effort than the SVM, which can be slow due to optimizing a large-scaled quadratic programming under linear constraints. Some efficient algorithm is suggested and one stable computation strategy is also provided for ill-posed cases. The effectiveness of the MPSVM was demonstrated by both simulation studies and applications to cancer classification using microarray data.

1. INTRODUCTION

Consider the multiclass classification problem with class labels $\{1, \dots, k\}$. Given a training set $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the measurement vector and y_i is the class label, our task is to learn a classification rule from the training set to predict the true class of a future object by only observing its measurement vector. Support vector machine (SVM) (Boser et al. (1992), Vapnik (1998), Burges (1998), and Cristianini and Shawe-Taylor (2000)) has shown successful performances in various studies. What makes the SVM attractive is its ability to condense the information contained in the training set and to find a decision surface determined by certain

Key words and phrases. Bayes optimal classification rules, Nonstandard classifications, Reproducing kernel Hilbert space.

Dr. Hao Helen Zhang's research is partially supported by NSF Grant DMS-0405913.

points of the training set. Lee et al. (2004) further generalized the SVM to multicategory SVM (MSVM).

The implementation of the SVM or MSVM demands solving a quadratic programming under linear constraints. For large datasets, solving a constrained quadratic problem is generally computational expensive. In the case of a multiclass classification, the computation can be challenging even for a moderate-size dataset with many classes. Proximal support vector machines (PSVM) was introduced recently as a variant of SVM for binary classifications in Suykens and Vandewalle (1999) and Fung and Mangasarian (2001b). In theory, both PSVM and SVM asymptotically target on the optimal Bayes rule, which explains their comparable prediction accuracies in most empirical studies. More importantly, the PSVM solves a system of linear equations using an extremely fast and simple algorithm and thus demands much less computational effort than the SVM. The least squares SVM (Suykens and Vandewalle (1999), Suykens et al. (2002) and Van Gestel et al. (2002a)) has similar loss functions as the PSVM except that the latter penalizes the constant term as well. Several proposals for multiclass PSVM and least squares SVM are based on the one-versus-rest scheme, which essentially transforms the multiclass separation into k binary classifications by artificially relabeling the data and constructing decision rules according to the maximal output. See Van Gestel et al. (2002b) and Fung and Mangasarian (2001a) for details.

The multiclass PSVM (MPSVM) proposed in this paper targets directly on the boundary among k classes by estimating some functions of k conditional probabilities simultaneously. The MPSVM is constructed in a regularization framework of reproducing kernel Hilbert space (RKHS) and it is optimal in terms of implementing Bayes rule asymptotically. Compared to the approaches based on the one-versus-rest scheme, the MPSVM is more flexible to handle nonstandard situations where there are unequal misclassification costs or nonrepresentative training sets. With regard to computation, the MPSVM can be solved more efficiently than the MSVM.

The paper is organized as follows. We briefly review the Bayes classification rule in Section 2 and the binary SVM and PSVM in Section 3. In Section 4, we introduce the MPSVM, study its statistical properties and present two computation algorithms. Section 5 illustrates the performance of MPSVM with simulation examples. In Section 6, the MPSVM is applied to cancer classifications using gene expression data.

2. BAYES CLASSIFICATION RULES

The Bayes rule is the optimal classification rule if the underlying distribution of the data is known. They serve as golden standards for any reasonable classifiers to approximate. In practice, the underlying distribution is rarely known. One common way to approximate the Bayes rules is to estimate the distribution or related classification functions from the training set.

For a k -class classification problem, we need to learn a classification rule $\phi(\mathbf{x}) : \mathfrak{R}^d \rightarrow \{1, \dots, k\}$. Assume that observations in the training data are i.i.d. and drawn from some distribution $P(\mathbf{x}, y)$. Let $p_j(\mathbf{x}) = \Pr(Y = j | \mathbf{X} = \mathbf{x})$ be the conditional probability of class j given $\mathbf{X} = \mathbf{x}$ for $j = 1, \dots, k$. Let C be a $k \times k$ cost matrix with entry C_{jl} meaning the cost paid for classifying an observation from class j to class l . All C_{jj} ($j = 1, \dots, k$) should equal 0 since the correct decision should not be penalized. The Bayes rule, minimizing the expected cost of misclassifying an observation

$$E[C_{Y\phi(\mathbf{x})}] = E_{\mathbf{X}} \left[\sum_{l=1}^k C_{l\phi(\mathbf{x})} \Pr(Y = l | \mathbf{X} = \mathbf{x}) \right] = E_{\mathbf{X}} \left[\sum_{l=1}^k C_{l\phi(\mathbf{x})} p_l(\mathbf{x}) \right],$$

is given by

$$(1) \quad \phi_B(\mathbf{x}) = \arg \min_{j=1, \dots, k} \left[\sum_{l=1}^k C_{lj} p_l(\mathbf{x}) \right].$$

When the misclassification costs are all equal, that is, $C_{lj} = 1$ if $j \neq l$, the Bayes rule simplifies to

$$(2) \quad \phi_B(\mathbf{x}) = \arg \min_{j=1, \dots, k} [1 - p_j(\mathbf{x})] = \arg \max_{j=1, \dots, k} p_j(\mathbf{x}),$$

which could be interpreted as minimizing the expected misclassification rates $E\{Y \neq \phi(\mathbf{X})\}$.

The majority of classification methods is “standard” in the sense that the samples in the training set represent some target population and the costs of different misclassifications are the same. However, nonstandard classifications may arise in many real situations. See Lin et al. (2002). One typical situation is that some type of misclassifications may be more serious than others and unequal misclassification costs should be assumed. For example, in the diagnosis of disease, considering a healthy person classified as sick or a sick patient as healthy may have different consequences. Sampling bias is another issue that needs special attention (Lin et al. (2002) and Lee et al. (2004)). In many studies where the proportions of each class in the true population were extremely unbalance, the minor classes were usually over-sampled and major classes were usually down-sampled. If the sampling scheme depends only on the class labels Y , and not on the measurements $\mathbf{X}$, the Bayes rule is

$$\phi_B(\mathbf{x}) = \arg \min_{j=1,\dots,k} \left[\sum_{l=1}^k \frac{\pi_l}{\pi_l^s} C_{lj} p_l^s(\mathbf{x}) \right],$$

where π_l is the proportion of class l in the target population, π_l^s is the proportion of class l in the training set, and $p_l^s(\mathbf{x}) = \Pr(Y^s = l | \mathbf{X}^s = \mathbf{x})$ is the conditional probability that a sample randomly drawn from the training set belongs to class l given $\mathbf{X}^s = \mathbf{x}$ (Lee et al. (2004)). A remedy for sampling bias is to define the new misclassification costs according to

$$(3) \quad C_{jl}^{new} = C_{jl} \frac{\pi_l}{\pi_l^s}.$$

In this paper, we call a classification problem “nonstandard” if misclassification costs are not all equal or there is a sampling bias. We are not going to differentiate the two nonstandard cases since the issue of sampling bias could be handled by defining new costs according to (3).

3. BINARY SVM AND PSVM

In binary classification problems, the class labels y_i are often coded as $\{+1, -1\}$. The SVM is motivated by the geometric interpretation of maximizing the margin. It has been shown that SVM

methodology can be cast as a regularization problem in a reproducing kernel Hilbert space (RKHS). See Wahba (1990), Girosi (1998), and Poggio and Girosi (1998). Lin et al. (2002) extended SVM to the nonstandard case. Let $\mathcal{H}_K$ be an RKHS with reproducing kernel K . The SVM for both standard and nonstandard classifications solves

$$(4) \quad \min_f \frac{1}{n} \sum_{i=1}^n L(y_i)(1 - y_i f(\mathbf{x}_i))_+ + \lambda \|h\|_{\mathcal{H}_K}^2,$$

over all the functions of the form $f(\mathbf{x}) = h(\mathbf{x}) + \beta_0$, where $h \in \mathcal{H}_K$, $(v)_+ = \max\{v, 0\}$, and $L(y_i)$ is the cost for misclassifying the i -th observation. The classification rule is $\text{sign}[f(\mathbf{x})]$.

Suykens and Vandewalle (1999) and Fung and Mangasarian (2001b) proposed PSVM, which classifies the points by assigning them to the closest of two parallel planes in input or feature space that are pushed apart as far as possible. In an RKHS, the PSVM could be formulated as

$$(5) \quad \min_f \frac{1}{n} \sum_{i=1}^n L(y_i)[1 - y_i f(\mathbf{x}_i)]^2 + \lambda(\|h\|_{\mathcal{H}_K}^2 + \beta_0^2).$$

Since each $y_i = 1$ or -1 , formulation (5) is equivalent to

$$(6) \quad \min_f \frac{1}{n} \sum_{i=1}^n L(y_i)[y_i - f(\mathbf{x}_i)]^2 + \lambda(\|h\|_{\mathcal{H}_K}^2 + \beta_0^2).$$

Thus PSVM could be interpreted as the Ridge regression model (Agarwal (2002)).

By the representer theorem (Kimeldorf and Wahba (1971) and Wahba (1999)), the solutions to (4) and (5) have a representation of the form

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \beta_i K(\mathbf{x}_i, \mathbf{x}),$$

where $\{\beta_i\}_{i=0}^n \in \mathfrak{R}$. The RKHS framework is closely related to the well-known kernel methodology. The fact that for any positive definite kernel, there exists a unique RKHS is well established by the *Moore Aronszjan* theorem (Aronszajn (1950)).

When $K(\mathbf{x}_i, \mathbf{x}_l) = \langle \mathbf{x}_i, \mathbf{x}_l \rangle$ or $h(\mathbf{x}) = \tilde{\beta}\mathbf{x}$ and misclassification costs are all equal, formulation (5) is equivalent to

$$\min_f \frac{1}{n} \sum_{i=1}^n [1 - y_i(\beta_0 + \tilde{\beta}\mathbf{x})]^2 + \lambda(\|\tilde{\beta}\|^2 + \beta_0^2).$$

This is the linear PSVM studied in Suykens and Vandewalle (1999) and Fung and Mangasarian (2001b). In general, the formulation (5) under equal misclassification costs reduces to

$$(7) \quad \min_f \frac{1}{n} \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]^2 + \lambda \left(\sum_{j=1}^n \sum_{l=1}^n \beta_j \beta_l K(\mathbf{x}_j, \mathbf{x}_l) + \beta_0^2 \right).$$

Formulation (7) is slightly different from the original kernel PSVM (equation (19), Fung and Mangasarian (2001b)), which is equivalent to

$$\min_f \frac{1}{n} \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]^2 + \lambda \left(\sum_{j=1}^n \beta_j^2 + \beta_0^2 \right).$$

One strategy for constructing multiclass classifiers is to transform the multiclass classification problem into a series of binary subproblems. For SVM-type classifiers, one-vs-rest scheme has been widely used to handle the multiclass classification problems (Van Gestel et al. (2002b) and Fung and Mangasarian (2001a)). The general idea of this scheme is to solve k binary subproblems, each trained to separate one class from the rest, and to construct decision rules according to the maximal output. One disadvantage of one-versus-rest scheme is that the resulting two-class subproblems are often very unbalanced, leading to poor performances in some cases (Fung and Mangasarian (2001a)). In addition, it is not easy to take into account unequal misclassification costs in the one-versus-rest scheme. One alternative strategy is to separate all the k classes simultaneously instead of classifying one from the rest. In fact, it has been used to formulate multiclass SVM classifiers such as in Vapnik (1998), Weston and Watkins (1999), and Lee et al. (2004). These methods generally need to solve a quadratic programming with linear constraints. For large datasets, the computation effort can be very expensive.

4. MULTICLASS PSVM

We propose an extension of PSVM to the multiclass case. The MPSVM embraces the binary PSVM as a special case. In Section 4.1, we present the formulation of MPSVM, investigate the properties of its solution and study its asymptotic classification performance. Section 4.2 introduces two efficient computational strategies for implementing the MPSVM.

4.1. The formulation and theoretical properties. In the multiclass problem, the class label is coded as a k -dimensional vector

$$\mathbf{y}_i = \begin{cases} \left(1, \frac{-1}{k-1}, \dots, \frac{-1}{k-1}\right)' & \text{if observation } i \text{ is in category 1,} \\ \left(\frac{-1}{k-1}, 1, \dots, \frac{-1}{k-1}\right)' & \text{if observation } i \text{ is in category 2,} \\ \dots & \\ \left(\frac{-1}{k-1}, \frac{-1}{k-1}, \dots, 1\right)' & \text{if observation } i \text{ is in category } k. \end{cases}$$

We define a k -tuple of separating function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))'$. Since the sum of components in each $\mathbf{y}$ is 0, we let $\mathbf{f}$ satisfy the sum-to-zero constraint, $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ for any $\mathbf{x} \in \mathfrak{R}^d$. Analogous to the binary case, we consider $\mathbf{f}(\mathbf{x}) \in \prod_{j=1}^k (\{1\} + \mathcal{H}_K)$, the product space of k reproducing kernel Hilbert spaces $\mathcal{H}_K$. In other words, each component $f_j(\mathbf{x})$ can be expressed as $h_j(\mathbf{x}) + \beta_{j0}$, where $h_j \in \mathcal{H}_K$ and $\beta_{j0} \in \mathfrak{R}$. Define the diagonal matrix $W(\mathbf{y}_i) = \text{diag}\{C_{j1}, C_{j2}, \dots, C_{jk}\}$ where C_{jl} ($l = 1, \dots, k$) are the costs of classifying the i -th observation from class j to class l if $\mathbf{y}_i$ indicates class j . The MPSVM is proposed as minimizing

$$(8) \quad \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i))' W(\mathbf{y}_i) (\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)) + \frac{1}{2} \lambda \sum_{j=1}^k (\|h_j\|_{\mathcal{H}_K}^2 + b_j^2),$$

subject to $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ for any $\mathbf{x}$. The classification rule induced by $\mathbf{f}(\mathbf{x})$ is $\phi(\mathbf{x}) = \arg \max_j f_j(\mathbf{x})$.

This formulation handles the standard and nonstandard classifications in a unified way.

The functional form of the solutions to (8) is given by the following representer theorem. Proof of this theorem and all other proofs are given in the appendix.

Theorem 1. Representer Theorem *If the reproducing kernel K is positive definite, minimizing (8) under the sum-to-zero constraint is equivalent to finding $(f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$ of the form*

$$(9) \quad f_j(\mathbf{x}) = \beta_{j0} + \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \mathbf{x}),$$

subject to $\sum_{j=1}^k \beta_{jl} = 0$ for $l = 0, \dots, n$.

The formulation (8) embraces the binary PSVM as a special case. To verify this, we will show that the data fit functionals and penalties of model (6) and (8) are identical when $k = 2$. When $\mathbf{y}_i = (1, -1)$ (1 in the binary notation), then $W(\mathbf{y}_i) = \text{diag}\{0, L(1)\}$ and $(\mathbf{y}_i - f(\mathbf{x}_i))'W(\mathbf{y}_i)(\mathbf{y}_i - f(\mathbf{x}_i)) = L(1)(f_2(\mathbf{x}_i) + 1)^2 = L(1)(1 - f_1(\mathbf{x}_i))^2$, where $L(1)$ is the cost of misclassifying a positive sample. Similarly, when $\mathbf{y}_i = (-1, 1)$ (-1 in the binary notation), we have $W(\mathbf{y}_i) = \text{diag}\{L(-1), 0\}$ and $(\mathbf{y}_i - f(\mathbf{x}_i))'W(\mathbf{y}_i)(\mathbf{y}_i - f(\mathbf{x}_i)) = L(-1)(-1 - f_1(\mathbf{x}_i))^2$, where $L(-1)$ is the cost of misclassifying a negative sample. Thereby, the data fit functionals in (6) and (8) are identical, $f_1(\mathbf{x})$ playing the same role as $f(\mathbf{x})$ in (6). Also, note that $\frac{\lambda}{2} \sum_{j=1}^2 (\|h_j\|_{\mathcal{H}_K}^2 + \beta_{j0}^2) = \frac{\lambda}{2} (\|h_1\|_{\mathcal{H}_K}^2 + \beta_{10}^2 + \|h_2\|_{\mathcal{H}_K}^2 + \beta_{20}^2) = \lambda (\|h_1\|_{\mathcal{H}_K}^2 + \beta_{10}^2)$ since $\beta_{10} + \beta_{20} = 0$ and $h_1(\mathbf{x}) + h_2(\mathbf{x}) = 0$ for any $\mathbf{x}$ by Theorem 1. So the penalties to the model (6) and (8) are also identical.

Theorem 2 says that the proposed MPSVM implements Bayes rule asymptotically under certain regularity conditions. We use the theoretical framework Cox and O'Sullivan (1990) for analyzing the asymptotics of penalized methods, which had been used to study the SVM and MSVM (Lin (2002) and Lee et al. (2004)). In the formulation (8), the data fit functional component indicates that the estimate should follow the pattern in the data, whereas the penalty component imposes smoothing conditions. The limit of the data fit functional as the sample size goes to infinity, $E[(\mathbf{Y} - f(\mathbf{X}))'W(\mathbf{Y})(\mathbf{Y} - f(\mathbf{X}))]$, could be used to identify the target function, which is the minimizer of the limiting functional. Under the assumption that the target function can be approximated by the elements in the RKHS and certain other regularity conditions, the solution to (8) will approach the target function as $n \rightarrow \infty$.

Theorem 2. *The minimizer of $E[(\mathbf{Y} - f(\mathbf{X}))'W(\mathbf{Y})(\mathbf{Y} - f(\mathbf{X}))]$ under the sum-to-zero constraint is implements the Bayes rule, that is*

$$\arg \max_{j=1,\dots,k} f_j(\mathbf{x}) = \arg \min_{j=1,\dots,k} \sum_{l=1}^k c_{lj} p_l(\mathbf{x}).$$

In particular, under equal misclassification costs, $\arg \max_{j=1,\dots,k} f_j(\mathbf{x}) = \arg \max_{j=1,\dots,k} p_j(\mathbf{x})$.

Remark 3. Inversing (23) in the appendix yields an estimate of the class probabilities. When misclassification costs are all equal, $\tau_j(\mathbf{x})$ reduces to $1 - p_j(\mathbf{x})$ and

$$(10) \quad \hat{p}_j(\mathbf{x}) = 1 - (k-1) \frac{1/(1 - \hat{f}_j(\mathbf{x}))}{\sum_{l=1}^k 1/(1 - \hat{f}_l(\mathbf{x}))},$$

for each $j = 1, \dots, k$.

4.2. Computation Algorithms. Compared to SVM and MSVM, the MPSVM simply solves a system of linear equations and demands much less computational effort. Most classification methods involve parameters that are often predetermined by some tuning strategies such as cross-validation. The computational advantage of the MPSVM makes it possible to perform a finer search of these parameters, which may lead to better results. The speed of the MPSVM makes it also very promising for analysis of large data sets. In this section, two computation strategies are presented and summarized in Algorithm 1 and 2. We assume the kernel K is strictly positive definite. Algorithm 1 runs fast, however it may fail when K is close to singularity since it is impossible or difficult to inverse K numerically. Algorithm 2 is specifically designed to prevent the ill-posed situations.

Let $\hat{K}$ be the $n \times n$ matrix with il entry $K(\mathbf{x}_i, \mathbf{x}_l)$. Let $Z = [\mathbf{1}_n \quad \hat{K}]$ and $G = \begin{pmatrix} 1 & \mathbf{0}'_n \\ \mathbf{0}_n & \hat{K} \end{pmatrix}$, where $\mathbf{1}_n = (1, \dots, 1)'$ and $\mathbf{0}_n = (0, \dots, 0)'$. For each $j = 1, \dots, k$, define vectors $\mathbf{y}_j^* = (y_{1j}, \dots, y_{nj})'$, $\boldsymbol{\beta}_j = (\beta_{j0}, \beta_{j1}, \dots, \beta_{jn})'$ and the $n \times n$ diagonal matrix $W_j^* = \text{diag}\{C_{\text{cat}\{1\}j}, \dots, C_{\text{cat}\{n\}j}\}$, where

$\text{cat}\{l\}$ is the class the l -th observation belongs to, and $C_{\text{cat}\{l\}j}$ is the cost of classifying the l -th observation as class j . Let $\lambda^* = n\lambda/2$, $\beta_j^* = G^{1/2}\beta_j$, $Z^* = ZG^{-1/2}$, $Z_j = W_j^{*1/2}Z^*$ and $\mathbf{y}_j^{**} = W_j^{*1/2}\mathbf{y}_j^*$ for $j = 1, \dots, k$. By Theorem 1, minimizing (8) is equivalent to minimizing

$$(11) \quad \sum_{j=1}^k [(\mathbf{y}_j^* - Z\beta_j)' W_j^* (\mathbf{y}_j^* - Z\beta_j) + \lambda^* \beta_j' G \beta_j]$$

$$(12) \quad = \sum_{j=1}^k [(\mathbf{y}_j^{**} - Z_j \beta_j^*)' (\mathbf{y}_j^{**} - Z_j \beta_j^*) + \lambda^* \beta_j^{*'} \beta_j^*]$$

subject to $\sum_{j=1}^k \beta_j = \sum_{j=1}^k \beta_j^* = \mathbf{0}_{n+1}$.

To solve (12), we consider its Wolfe dual problem

$$L_D = \sum_{j=1}^k [(\mathbf{y}_j^{**} - Z_j \beta_j^*)' (\mathbf{y}_j^{**} - Z_j \beta_j^*) + \lambda^* \beta_j^{*'} \beta_j^*] - 2\mathbf{u}^{*'} \sum_{j=1}^k \beta_j^*,$$

where $\mathbf{u} = 2\mathbf{u}^* \in \Re^{n+1}$ is the Lagrange multiplier.

Setting equal to zero the gradient of L_D with respect to β_j^* yields

$$\frac{\partial L_D}{\partial \beta_j^*} = 2Z_j' (Z_j \beta_j^* - \mathbf{y}_j^{**}) + 2\lambda^* \beta_j^* - 2\mathbf{u}^* = 0.$$

Thus

$$(13) \quad \beta_j^* = (Z_j' Z_j + \lambda^* I)^{-1} (Z_j' \mathbf{y}_j^{**} + \mathbf{u}^*) = B_j (Z_j' \mathbf{y}_j^{**} + \mathbf{u}^*),$$

where $B_j = (Z_j' Z_j + \lambda^* I)^{-1}$ for $j = 1, \dots, k$.

Since $\mathbf{0}_{n+1} = \sum_{j=1}^k \beta_j^* = \sum_{j=1}^k B_j (Z_j' \mathbf{y}_j^{**} + \mathbf{u}^*)$, we get

$$(14) \quad \mathbf{u}^* = - \left(\sum_{j=1}^k B_j \right)^{-1} \left(\sum_{j=1}^k B_j Z_j' \mathbf{y}_j^{**} \right).$$

The solution to (11) is given by

$$(15) \quad \beta_j = G^{-1/2} \beta_j^* = A_j (Z_j' W_j^* \mathbf{y}_j^* + \mathbf{v}),$$

where $A_j = G^{-1/2} B_j G^{-1/2} = (Z_j' W_j^* Z_j + \lambda^* G)^{-1}$ and $\mathbf{v} = - \left(\sum_{j=1}^k A_j \right)^{-1} \left(\sum_{j=1}^k A_j Z_j' W_j^* \mathbf{y}_j^* \right)$.

For any (new) observation $\mathbf{x}$, define the vector $K_{\mathbf{x}} = [K(\mathbf{x}, \mathbf{x}_1), \dots, K(\mathbf{x}, \mathbf{x}_n)]'$. Algorithm 1 could be summarized as follows.

ALGORITHM 1. (1) Solve (11) using (15).

(2) To predict a (new) sample, compute $\hat{f}_j(\mathbf{x}) = [1, K'_{\mathbf{x}}]\beta_j$ for $j = 1, \dots, k$ and classify the sample to the class giving the largest $\hat{f}_j(\mathbf{x})$.

In theory, the matrix $\hat{K}$ is strictly positive definite. However, in practice it can be numerically ill-posed. Algorithm 1 may fail when $\hat{K}$, and hence G and A_j ($j = 1, \dots, k$), are close to singularities. In Algorithm 2, the inversion of G is avoided by using $ZG^{-1} = [\mathbf{1}_n, I]$ and $ZG^{-1}Z' = \mathbf{1}_n\mathbf{1}_n' + \hat{K}$, which is achieved by some strategic matrix transformation of the solution given in (13) and (14) via the Sherman-Morrison-Woodbury formula. Our experience shows that, when $\hat{K}$ is good-conditioned, we should use Algorithm 1 since it is much more efficient than Algorithm 2; otherwise Algorithm 2 provides an alternative way to get the solution.

By the use of Sherman-Morrison-Woodbury formula,

$$(16) \quad B_j = (Z_j'Z_j + \lambda^*I)^{-1} = 1/\lambda^* [I - Z_j'(\lambda^*I + Z_jZ_j')^{-1}Z_j] = 1/\lambda^* (I - Z^{*'}C_jZ^*),$$

where

$$(17) \quad C_j = W_j^{*1/2} \left[\lambda^*I + W_j^{*1/2}Z^*Z^{*'}W_j^{*1/2} \right]^{-1} W_j^{*1/2} = \underline{l}_j \odot [\lambda^*I + \underline{l}_j \odot Z^*Z^{*'}]^{-1},$$

$\bar{l}_j$ be the column vector containing the diagonal elements of $W_j^{*1/2}$, $\underline{l}_j = \bar{l}_j\bar{l}_j'$, and $\odot$ is the elementwise products of two matrices.

It is easy to verify that

$$(18) \quad B_jZ_j'\mathbf{y}_j^{**} = 1/\lambda^* [I - Z_j'(\lambda^*I + Z_jZ_j')^{-1}Z_j] Z_j'\mathbf{y}_j^{**} = Z_j'(\lambda^*I + Z_jZ_j')^{-1}\mathbf{y}_j^{**} = Z^{*'}C_j\mathbf{y}_j^*.$$

Let $C = 1/k \sum_{j=1}^k C_j$. Then

$$(19) \quad B = \sum_{j=1}^k B_j = k/\lambda^* (I - Z^{*'} C Z^*).$$

By the Sherman-Morrison-Woodbury formula,

$$(20) \quad B^{-1} = \lambda^*/k [I - Z^{*'} (Z^* Z^{*'} - C^{-1})^{-1} Z^*].$$

Combining equations (14), (20) and (18), we get

$$(21) \quad \begin{aligned} \mathbf{u}^* &= -B^{-1} \left(\sum_{j=1}^k B_j Z_j' \mathbf{y}_j^{**} \right) \\ &= -\lambda^*/k (I - Z^{*'} D Z^*) \left[Z^{*'} \left(\sum_{j=1}^k C_j \mathbf{y}_j^* \right) \right] \\ &= -\lambda^* Z^{*'} Q, \end{aligned}$$

where $D = (Z^* Z^{*'} - C^{-1})^{-1}$, $H = 1/k \sum_{j=1}^k C_j \mathbf{y}_j^*$, and $Q = H - D(Z^* Z^{*'})H = -DC^{-1}H$.

By equations (14), (21), (18) and (16),

$$\beta_j^* = B_j Z_j' \mathbf{y}_j^{**} + B_j \mathbf{u}^* = Z^{*'} C_j \mathbf{y}_j^* - (I - Z^{*'} C_j Z^*) Z^{*'} Q = Z^{*'} [C_j (\mathbf{y}_j^* + Z^* Z^{*'} Q) - Q].$$

Hence

$$\beta_j = G^{-1/2} \beta_j^* = G^{-1/2} Z^{*'} [C_j (\mathbf{y}_j^* + Z^* Z^{*'} Q) - Q] = [\mathbf{1}_n, I]' [C_j (\mathbf{y}_j^* + Z^* Z^{*'} Q) - Q].$$

The prediction of $\hat{f}_j$ for $\mathbf{x}$ is given by

$$(22) \quad \hat{f}_j(\mathbf{x}) = [1, K_{\mathbf{x}}'] \beta_j = (\mathbf{1}_n + K_{\mathbf{x}}) [C_j (\mathbf{y}_j^* + Z^* Z^{*'} Q) - Q].$$

ALGORITHM 2. (1) Compute $Z^* Z^{*'} = ZG^{-1}Z' = \mathbf{1}_n \mathbf{1}_n' + \hat{K}$

(2) Compute C_j ($j = 1, \dots, k$), C , D , H and Q as defined in (17) and (21).

(3) To predict a (new) sample, compute $\hat{f}_j(\mathbf{x})$ for $j = 1, \dots, k$ using (22) and classify the sample to the class giving the largest $\hat{f}_j(\mathbf{x})$.

5. NUMERICAL EXAMPLES

We illustrate the MPSVM by simulation studies in this section and analysis of microarray data in the next section. Throughout, the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_l) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_l\|^2}{\sigma^2}\right)$ was used and (λ^*, σ^2) was determined by the use of 10-fold cross-validation. Section 5.2 illustrated an example on nonstandard classifications. In all other examples, equal misclassification costs were assumed.

5.1. Comparison with MSVM using simulation. Similar to Lee et al. (2004), a training set of sample size 200 was simulated according to the three-class model on the unit interval $[0, 1]$ with the conditional probabilities $p_1(x) = 0.97 \exp(-3x)$, $p_3(x) = \exp(-2.5(x - 1.2)^2)$, and $p_2(x) = 1 - p_1(x) - p_3(x)$. We applied the MPSVM to the simulated data and used (10) to estimate the conditional probabilities. Figure 1(a) and 1(b) depicted the true and estimated conditional probabilities, respectively. The MPSVM recovered the shape of the true probability functions to some extent. According to (2), the Bayes class boundaries were at $x = 0.260$ where $p_1(x) = p_2(x) > p_3(x)$ and $x = 0.613$ where $p_1(x) < p_2(x) = p_3(x)$ while the two critical points estimated by the MPSVM were around 0.262 and 0.615. So the MPSVM classifier approximate the Bayes optimal rule reasonably well.

(a) The true conditional probabilities

(b) The probabilities estimated by MPSVM

FIGURE 1. The true and estimated conditional probability functions

To compare with the MSVM, we simulated 100 replicate training datasets of sample size 200 and a testing dataset of sample size 10000 and applied MPSVM and one-versus-rest PSVM to each training set. The trained classification rules were then used to evaluate the testing error rates over the testing data. The Bayes optimal classification error rates was 0.3841, whereas the average testing error rate ($\pm$ standard deviation) over 100 replicates was 0.4068 ± 0.0101 by the MPSVM, and 0.4081 ± 0.0124 by the one-versus-rest PSVM. The same simulation strategy was used to evaluate the MSVM and one-versus-rest SVM in Lee et al. (2004), where the average testing error rate was 0.3951 ± 0.0099 by the MSVM, and 0.4307 ± 0.0132 by the one-versus-rest SVM. The performances of PMSVM and one-versus-rest PSVM were almost as good as that of MSVM. The error rate by the one-versus-rest SVM was a little higher. One possible reason was that the one-versus-rest SVM may not approximate the Bayes rule well even when the sample size gets large (Lee et al. (2004)), while all other three classifiers are asymptotically optimal.

5.2. Sampling bias. To illustrate the effective of MPSVM in handling sampling bias, we firstly simulated a sample of size 500 from the three-class model on the square $A = [0, \sqrt{10}] \times [0, \sqrt{10}]$ with the conditional probabilities satisfying $Pr(Y = j|\mathbf{x} \in A_l) = 0.9$ when $j = l$ and $Pr(Y = j|\mathbf{x} \in A_l) = 0.05$ when $j \neq l$ for any $1 \leq j, l \leq 3$, where $A_1 = \{\mathbf{X} : X_1^2 + X_2^2 \leq 2, X_1 > X_2\}$, $A_2 = \{\mathbf{X} : X_1^2 + X_2^2 \leq 2, X_1 \leq X_2\}$, and $A_3 = \{\mathbf{X} : X_1^2 + X_2^2 > 2\}$ partition the square A . Since the area ratio of A_1, A_2 and A_3 was $1 : 1 : 8$ and about 400 observations belonged to class 3, any observation in class 3 were retained with probability $1/8$. So the final sample was biased and was not completely random from the true model. Figure 2(a) plots the data and Bayes decision boundaries A_1, A_2 and A_3 . We fitted the MPSVM model using equal misclassification costs and the costs adjusted according to (3). The corresponding decision boundaries were plotted in Figure 2(b) and 2(c). The decision boundaries based on adjusted costs in Figure 2(b) looked more similar

to the Bayes boundaries in Figure 2(a). Adjusting the misclassification costs greatly improved the performance of the classifications.

(a) the data and Bayes class boundaries

(b) The decision boundaries by MPSVM with adjusted misclassification costs

(c) The decision boundaries by MPSVM with equal misclassification costs

FIGURE 2. Sampling Bias

6. APPLICATIONS

The MSPVM was applied to cancer classifications using two benchmark microarray datasets. One challenge in microarray analysis is that the the number of genes (p) is usually much bigger than the sample size (n), which results in the “large p , small n ” problem. One typical approach is to reduce the dimension of feature by the use of principle component analysis or factor analysis (Khan et al. (2001), West (2003), etc) before applying classification methods. As regularized regression models, the SVM-type methods offer an alternative way to handle the case where the dimension of the input is bigger than the sample size and may be well-suited to the analysis of microarray data.

6.1. Leukemia data. The leukemia dataset was published by Golub et al. (1999) for the classification of two leukemia, ALL (acute myeloid leukemia) and AML (acute lymphoblastic leukemia). The two cancer types were identified based on their origins, lymphoid (lymph or lymphatic tissue

related) and myeloid (bone marrow related), respectively. ALL could be further divided into B-cell and T-cell ALLs. There are 38 training samples (ALL B-cell: 19, ALL T-cell 8 and AML: 11) in the training set and 34 samples (ALL B-cell: 19, ALL T-cell 1 and AML: 14) in the testing set. For each sample, the expression values of 7129 genes were available. In the literature, it has been treated either as a two-class (ALL/AML) problem (Golub et al. (1999), Dudoit et al. (2002), Tibshirani et al. (2002), Nguyen and Rocke (2002b), etc) or as a three-class (AML/ALL B-cell/ALL T-cell) problem (Lee and Lee (2003), Nguyen and Rocke (2002a), Dudoit et al. (2002), etc).

In this paper, we considered the leukemia data as a three-class problem. To compare the classification performance of MPSVM with that of MSVM, the same procedure in Lee and Lee (2003) was used to preprocess the data: (i) thresholding (floor of 100 and ceiling of 16,000), (ii) filtering (exclusion of genes with $\max/\min \leq 5$ and $\max - \min \leq 500$ across the samples), (iii) base 10 logarithmic transformation. The filtering resulted in 3571 genes. After these initial steps, each array was then standardized and F-tests were used to select 40 genes with the largest F-statistics.

Figure 6.1 plots the heat maps of the training set and testing set, in which the horizontal rows represent genes, whereas the columns represent samples. The colors depict gene expression values from green (large negative) to red (large positive). For the training set, samples from each class were put together, and hierarchical clustering was used to order the genes so that genes close to each other had similar expression values across the training samples. From Figure 3(a), we can see that for each class, some subset of genes were highly expressed so that the selected 40 genes were very informative in discriminating the three classes. In the heat map (Figure 3(b)) of the testing set, the order of genes was the same as that in the training set, and hierarchical clustering was used to order the testing samples. The expression pattern of testing samples in Figure 3(b) seems to match very well with that of the training samples in Figure 3(a).

Since the sample size was small, the performance of MPSVM and one-versus-rest PSVM was quite robust to the choice of tuning parameters. Using various equally good tuning parameters, no

(a) Heat map of the training set

(b) Heat map of the testing set

FIGURE 3. Heat maps of leukemia data.

or one testing sample was misclassified by the MPSVM and the one-versus-rest PSVM. One testing sample was misclassified using the GACV tuning and 0.8 testing sample on average was misclassified using leave-one-out cross-validation tuning by the MSVM (table 1, Lee and Lee (2003)). This demonstrates that the performance of MPSVM is comparable to that the MSVM while MPSVM is much quicker to implement.

6.2. Small round blue cell tumors data. The small round blue cell tumors of children data set was published in Khan et al. (2001). There are four tumor types: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (BL/NHL), and Ewing family of tumors (EWS). The data set contains 2308 genes out of 6567 after some filtering procedure. There are 63 samples (BL: 8, EWS: 23, NB 12, EWS: 23) in the training set and 20 samples (BL: 3, EWS: 6, NB 5, EWS: 20) in the testing set. Similar to the approach for the analysis of the leukemia data, the data were logarithm-transformed and F -tests were used to pre-select 40 genes with the smallest p-values. The testing errors by both MPSVM and one-versus-rest PSVM were 0. This dataset has

been analyzed by the use of many methods (Khan et al. (2001), Lee and Lee (2003), Dudoit et al. (2002), Tibshirani et al. (2002), etc). Nearly all classification methods yield 0 testing errors.

APPENDIX

Proof of Theorem 1: Let $f_j(\mathbf{x}) = \beta_{j0} + h_j(\mathbf{x})$ with $h_j \in \mathcal{H}_K$ for $j = 1, \dots, k$. Consider the decomposition $h_j(\cdot) = \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) + \rho_j(\cdot)$, where $\rho_j(\cdot)$ is the element in the RKHS orthogonal to the span of $\{K(\mathbf{x}_i, \cdot), i = 1, \dots, n\}$. By the sum-to-zero constraint, there is

$$f_k(\cdot) = - \sum_{j=1}^{k-1} \beta_{j0} - \sum_{j=1}^{k-1} \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) - \sum_{j=1}^{k-1} \rho_j(\cdot).$$

By the definition of K , we have $\langle h_j(\cdot), K(\mathbf{x}_i, \cdot) \rangle_{\mathcal{H}_K} = h_j(\mathbf{x}_i)$ for $i = 1, \dots, n$. Thus

$$\begin{aligned} f_j(\mathbf{x}_i) &= \beta_{j0} + h_j(\mathbf{x}_i) \\ &= \beta_{j0} + \langle h_j, K(\mathbf{x}_i, \cdot) \rangle_{\mathcal{H}_K} \\ &= \beta_{j0} + \langle \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) + \rho_j(\cdot), K(\mathbf{x}_i, \cdot) \rangle_{\mathcal{H}_K} \\ &= \beta_{j0} + \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Therefore the data fit functional in (8) does not depend on $\rho_j(\cdot)$ for $j = 1, \dots, k$. Furthermore,

$$\|h_j\|^2 = \left\| \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) \right\|^2 + \|\rho_j\|^2 \quad \text{for } 1 \leq j \leq k-1$$

and

$$\|h_k\|^2 = \left\| \sum_{j=1}^{k-1} \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) \right\|^2 + \left\| \sum_{j=1}^{k-1} \rho_j \right\|^2.$$

Obviously, the minimizer of (8) does not depend on ρ_j . This implies equation (9).

It remains to show that minimizing (8) subject to $\sum_{j=1}^k \beta_{ji} = 0$ for $i = 0, \dots, n$ is equivalent to minimizing (8) subject to $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ for any $\mathbf{x}$. Let $\hat{K}$ be the n by n matrix with il entry $K(\mathbf{x}_i, \mathbf{x}_l)$. Let $f_j(\cdot) = \beta_{j0} + \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot)$ be the minimizer of (8) under the sum-to-zero

constraint for any $\mathbf{x}$. Let $\beta_j^* = (\beta_{j1}, \dots, \beta_{jn})'$ for $j = 1, \dots, k$. Define $\bar{\beta}_l = 1/k \sum_{j=1}^k \beta_{jl}$ for $l = 0, \dots, n$ and

$$f_j^*(\cdot) = \beta_{j0}^* + \sum_{i=1}^n \beta_{ji}^* K(\mathbf{x}_i, \cdot) = (\beta_{j0} - \bar{\beta}_0) + \sum_{i=1}^n (\beta_{ji} - \bar{\beta}_i) K(\mathbf{x}_i, \cdot).$$

Then $f_j^*(\cdot) = f_j(\cdot)$ since $\sum_{j=1}^k f_j(\cdot) \equiv 0$. However,

$$\begin{aligned} \sum_{j=1}^k (\|h_j^*\|^2 + \beta_{j0}^{*2}) &= \sum_{j=1}^k (\beta_j^{*'} \hat{K} \beta_j^* + \beta_{j0}^2) - k\bar{\beta}_0^2 - 1/k \left(\sum_{j=1}^k \beta_j^* \right)' \hat{K} \left(\sum_{j=1}^k \beta_j^* \right) \\ &\leq \sum_{j=1}^k (\beta_j^{*'} \hat{K} \beta_j^* + \beta_{j0}^2) = \sum_{j=1}^k (\|h_j\|^2 + \beta_{j0}^2). \end{aligned}$$

The equality has to hold since f_j is the global minimizer. This implies $\bar{\beta}_0 = 0$ and

$\left(\sum_{j=1}^k \beta_j^* \right)' \hat{K} \left(\sum_{j=1}^k \beta_j^* \right) = 0$. When $\hat{K}$ is positive definite, $\sum_{j=1}^k \beta_{ji} = 0$ for $i = 0, \dots, n$. On the contrary, let $f_j(\cdot) = \beta_{j0} + \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot)$ be the minimizer of (8) that satisfies $\sum_{j=1}^k \beta_{ji} = 0$ for $i = 0, \dots, n$. Let $\beta_j^* = (\beta_{j1}, \dots, \beta_{jn})'$ for $j = 1, \dots, k$. Thus $\hat{K} \left(\sum_{j=1}^k \beta_j^* \right) = 0$ and

$$0 = \left(\sum_{j=1}^k \beta_j^* \right)' \hat{K} \left(\sum_{j=1}^k \beta_j^* \right) = \left\| \sum_{j=1}^k \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \cdot) \right\|^2.$$

It means that $\sum_{j=1}^k \sum_{i=1}^n \beta_{ji} K(\mathbf{x}_i, \mathbf{x}) = 0$ and hence $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ for any $\mathbf{x}$.

Proof of theorem 2: Since $E[(f(\mathbf{X}) - \mathbf{Y})' W(\mathbf{Y})(f(\mathbf{X}) - \mathbf{Y})] = E(E[(f(\mathbf{X}) - \mathbf{Y})' W(\mathbf{Y})(f(\mathbf{X}) - \mathbf{Y}) | \mathbf{X}])$, we can minimize $E[(f(\mathbf{X}) - \mathbf{Y})' W(\mathbf{Y})(f(\mathbf{X}) - \mathbf{Y})]$ by minimizing $E[(f(\mathbf{X}) - \mathbf{Y})' W(\mathbf{Y})(f(\mathbf{X}) - \mathbf{Y}) | \mathbf{X} = \mathbf{x}]$ for every $\mathbf{x}$. Note that $c_{jl} = 0$ if $j = l$, then

$$\begin{aligned} Q_{\mathbf{x}}(f) &= E[(f(\mathbf{X}) - \mathbf{Y})' W(\mathbf{Y})(f(\mathbf{X}) - \mathbf{Y}) | \mathbf{X} = \mathbf{x}] \\ &= \sum_{j=1}^k \left[\sum_{l=1}^k c_{jl} \left(f_l(\mathbf{x}) + \frac{1}{k-1} \right)^2 \right] p_j(\mathbf{x}) \\ &= \sum_{j=1}^k \left[\sum_{l=1}^k c_{lj} p_l(\mathbf{x}) \right] \left(f_j(\mathbf{x}) + \frac{1}{k-1} \right)^2 \\ &= \sum_{j=1}^k \tau_j(\mathbf{x}) \left(f_j(\mathbf{x}) + \frac{1}{k-1} \right)^2, \end{aligned}$$

where $\tau_j(\mathbf{x}) = \sum_{l=1}^k c_{lj} p_l(\mathbf{x})$. Consider minimizing $Q_{\mathbf{x}}(f)$ subject to $\sum_{j=1}^k f_j(\mathbf{x}) = 0$. Using the Lagrange multiplier $\gamma > 0$, we get

$$A_{\mathbf{x}}(f) = \sum_{j=1}^k \tau_j(\mathbf{x}) \left(f_j(\mathbf{x}) + \frac{1}{k-1} \right)^2 - \gamma \sum_{j=1}^k f_j(\mathbf{x}).$$

Then for each $j = 1, \dots, k$, we have

$$\frac{\partial A_{\mathbf{x}}}{\partial f_j} = 2\tau_j \left(f_j + \frac{1}{k-1} \right) - \gamma = 0 \quad \rightarrow f_j = \frac{\gamma}{2\tau_j} - \frac{1}{k-1}.$$

Since $\sum_{j=1}^k f_j = 0$, we have

$$\gamma = \frac{2k}{k-1} \cdot \frac{1}{\sum_{j=1}^k \frac{1}{\tau_j}}.$$

The solution is given by

$$(23) \quad f_j = \frac{k}{k-1} \frac{\frac{1}{\tau_j}}{\sum_{l=1}^k \frac{1}{\tau_l}} - \frac{1}{k-1}.$$

Thus $\arg \max_{j=1, \dots, k} f_j(\mathbf{x}) = \arg \min_{j=1, \dots, k} \tau_j(\mathbf{x}) = \arg \min_{j=1, \dots, k} \sum_{l=1}^k c_{lj} p_l(\mathbf{x})$.

REFERENCES

- AGARWAL, D. K. (2002). Shrinkage estimator generalizations of proximal support vector machines. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.
- ARONSAJN, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society* **68** 337–404.
- BOSER, B. E., GUYON, I. M. and VAPNIK, V. (1992). A training algorithm for optimal margin classifiers. In *5th Annual ACM Workshop on Computational Learning Theory* (D. Haussler, ed.). ACM Press, Pittsburgh, PA.
- BURGES, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2(2)** 121–167.

- COX, D. and O’SULLIVAN, F. (1990). Asymptotic analysis of penalized likelihood type estimators. *Annals of statistics* **18** 1676–1695.
- CRISTIANINI, N. and SHAWE-TAYLOR, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.
- DUDOIT, S., FRIDLAND, 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.
- FUNG, G. and MANGASARIAN, O. L. (2001a). Multicategory proximal support vector machine classifiers. *Machine Learning*, in press.
- FUNG, G. and MANGASARIAN, O. L. (2001b). Proximal support vector machine classifiers. In *Proceedings of the 7th ACM Conference on Knowledge Discovery and Data Mining*. ACM press.
- GIROSI, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation* **10** 1455–1480.
- GOLUB, T., SLONIM, D., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J., COLLIER, H., LOH, M., DOWNING, J., CALIGIURI, M., BLOOMFIELD, C. and LANDER, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286** 531–537.
- KHAN, J., WEI, J., RINGNER, M., SAAL, L., LADANYI, M., WESTERMANN, F., BERTHOLD, F., SCHWAB, M., ANTONESCU, C., PETERSON, C. and MELTZER, P. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7** 673–679.
- KIMELDORF, G. and WAHBA, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications* **33** 82–95.
- LEE, Y. and LEE, C. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics* **19** 1132–1139.

- 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.
- LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6** 259–275.
- LIN, Y., LEE, Y. and WAHBA, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning* **46** 191–202.
- NGUYEN, D. V. and ROCKE, D. M. (2002a). Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics* **18** 1216–1226.
- NGUYEN, D. V. and ROCKE, D. M. (2002b). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18** 39–50.
- POGGIO, T. and GIROSI, F. (1998). A sparse representation for functional approximation. *Neural Computation* **10** 1445–1454.
- SUYKENS, J. A. K., VAN GESTEL, T., DE BRABANTER, J., DE MOOR, B. and VANDEWALLE, J. (2002). *Least Squares Support Vector Machines*. World Scientific, Singapore.
- SUYKENS, J. A. K. and VANDEWALLE, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters* **9(3)** 293–300.
- TIBSHIRANI, R., HASTIE, T., NARASIMHAN, B. and CHU, G. (2002). Diagnosis of multiple cancer types by shrunk centroids of gene expression. *Proceedings of the National Academy of Sciences USA* **99** 6567–6572.
- VAN GESTEL, T., SUYKENS, J. A. K., LANCKRIET, G., LAMBRECHTS, A., DE MOOR, B. and VANDEWALLE, J. (2002a). Bayesian framework for least-squares support vector machine classifiers, Gaussian processes and kernel Fisher discriminant analysis. *Neural Computation* **15(5)** 1115–1148.

- VAN GESTEL, T., SUYKENS, J. A. K., LANCKRIET, G., LAMBRECHTS, A., DE MOOR, B. and VANDEWALLE, J. (2002b). Multiclass LS-SVMs: moderated outputs and coding-decoding schemes. *Neural Processing Letters* **15**(1) 45–58.
- VAPNIK, V. N. (1998). *Statistical Learning Theory*. John Wiley, New-York.
- WAHBA, G. (1990). *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics, 59, SIAM, Philadelphia, Pennsylvania.
- WAHBA, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In *Advances in Kernel Methods: Support Vector Learning* (B. Scholkopt, C. Burges and A. Smola, eds.). MIT Press.
- WEST, M. (2003). Bayesian factor regression models in the “large p, small n” paradigm. *Bayesian Statistics* **7** 723–732.
- WESTON, J. and WATKINS, C. (1999). Support vector machines for multiclass pattern recognition. In *Proceedings of the 7th European Symposium On Artificial Neural Networks*. Bruges, Belgium.
- DEPARTMENT OF STATISTICS, NORTH CAROLINA STATE UNIVERSITY, CAMPUS BOX 8203, RALEIGH, NC 27695-8203, USA
- E-mail address:* hzhang2@stat.ncsu.edu