Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data

Yoonkyung LEE, Yi LIN, and Grace WAHBA

Two-category support vector machines (SVM) have been very popular in the machine learning community for classification problems. Solving multicategory problems by a series of binary classifiers is quite common in the SVM paradigm; however, this approach may fail under various circumstances. We propose the multicategory support vector machine (MSVM), which extends the binary SVM to the multicategory case and has good theoretical properties. The proposed method provides a unifying framework when there are either equal or unequal misclassification costs. As a tuning criterion for the MSVM, an approximate leave-one-out cross-validation function, called Generalized Approximate Cross Validation, is derived, analogous to the binary case. The effectiveness of the MSVM is demonstrated through the applications to cancer classification using microarray data and cloud classification with satellite radiance profiles.

KEY WORDS: Generalized approximate cross-validation; Nonparametric classification method; Quadratic programming; Regularization method; Reproducing kernel Hilbert space.

1. INTRODUCTION

The support vector machine (SVM) has exploded in popularity within the machine learning literature and, more recently, has received increasing attention from the statistics community as well. (For a comprehensive list of references, see *http://www.kernel-machines.org.*) This article concerns SVM's for classification problems, particularly those involving more than two classes. The SVM paradigm, originally designed for the binary classification problem, has a nice geometrical interpretation of discriminating one class from another by a hyperplane with the maximum margin (for an overview, see Vapnik 1998). It is commonly known that the SVM paradigm can sit comfortably in the regularization framework, where we have a data fit component ensuring the model's fidelity to the data and a penalty component enforcing the model simplicity (see Wahba 1998; Evgeniou, Pontil, and Poggio 2000, for more details). Considering that regularized methods, such as the penalized likelihood method and smoothing splines, have long been studied in the statistics literature, it appears quite natural to shed fresh light on the SVM and illuminate its properties in a similar fashion.

From this statistical stand point, Lin (2002) argued that the empirical success of the SVM can be attributed to the fact that for appropriately chosen tuning parameters, the SVM implements the optimal classification rule asymptotically in a very efficient manner. To be precise, let $\mathbf{X} \in \mathbb{R}^d$ be covariates used for classification and let Y be the class label, either 1 or -1 in the binary case. We define $(\mathbf{X}, Y)$ as a random pair from the underlying distribution $\Pr(\mathbf{x}, y)$. The theoretically optimal classification rule, the so-called "Bayes decision rule," minimizes the misclassification error rate; it is given by $\operatorname{sign}(p_1(\mathbf{x}) - 1/2)$, where $p_1(\mathbf{x}) = \Pr(Y = 1|\mathbf{X} = \mathbf{x})$, the conditional probability of the positive class given $\mathbf{X} = \mathbf{x}$. Lin (2002) showed that the solution of SVM's, denoted by $f(\mathbf{x})$, directly targets the Bayes decision rule $\operatorname{sign}(p_1(\mathbf{x}) - 1/2)$ without estimating the conditional probability function $p_1(\mathbf{x})$.

We turn our attention to the multicategory classification problem. We assume the class label $Y \in \{1, \ldots, k\}$ without loss of generality, where k is the number of classes. Define $p_j(\mathbf{x}) = \Pr(Y = j|\mathbf{X} = \mathbf{x})$. In this case the Bayes decision rule assigns a new $\mathbf{x}$ to the class with the largest $p_j(\mathbf{x})$. Two general strategies are used to tackle the multicategory problem. One strategy is to solve the multicategory problem by solving a series of binary problems; the other is to consider all of the classes at once. (See Dietterich and Bakiri 1995 for a general scheme to use binary classifiers to solve multiclass problems.) Allwein, Schapire, and Singer (2000) proposed a unifying framework to study the solution of multiclass problems obtained by multiple binary classifiers of certain types (see also Crammer and Singer 2000). Constructing pairwise classifiers or one-versus-rest classifiers is a popular approach in the first strategy. The pairwise approach has the disadvantage of potential variance increase, because smaller observations are used to learn each classifier. Moreover, it allows only a simple cost structure when different misclassification costs are concerned (see Friedman 1996 for details). For SVM's, the one-versus-rest approach has been widely used to handle the multicategory problem. The conventional recipe using the SVM scheme is to train k one-versus-rest classifiers and assign a new $\mathbf{x}$ to the class giving the largest $f_j(\mathbf{x})$ for $j = 1, \ldots, k$, where $f_j(\mathbf{x})$ is the SVM solution from training class j versus the rest. Even though the method inherits the optimal property of SVM's for discriminating one class from the rest, it does not necessarily imply the best rule for the original k-category classification problem. Leaning on the insight that we have from the two-category SVM, $f_j(\mathbf{x})$ will approximate $\operatorname{sign}(p_j(\mathbf{x}) - 1/2)$. If there is a class j with $p_j(\mathbf{x}) > 1/2$ given $\mathbf{x}$, then we can easily pick the majority class j by comparing $f_\ell(\mathbf{x})$'s for $\ell = 1, \ldots, k$, because $f_j(\mathbf{x})$ would be near 1 and all of the other $f_\ell(\mathbf{x})$ would be close to -1, creating a big contrast. However, if there is no dominating class, then all $f_j(\mathbf{x})$'s would be close to -1, making

Yoonkyung Lee is Assistant Professor, Department of Statistics, The Ohio State University, Columbus, OH 43210 (E-mail: *yklee@stat.ohio-state.edu*). Yi Lin is Associate Professor (E-mail: *yilin@stat.wisc.edu*), Grace Wahba is Bascom and I. J. Schoenberg Professor (E-mail: *wahba@stat.wisc.edu*), Department of Statistics, University of Wisconsin, Madison, WI 53706. Lee's research was supported in part by National Science Foundation (NSF) grant DMS 0072292 and National Aeronautic and Space Administration (NASA) grant NAG5 10273. Lin's research was supported in part by NSF grant DMS 0134987. Wahba's research was supported in part by National Institutes of Health grant EY09946, NSF grant DMS 0072292, and NASA grant NAG5 10273. The authors thank the editor, an associate editor, and referees for their helpful comments and suggestions.

Journal of the American Statistical Association
March 2004, Vol. 99, No. 465, Theory and Methods
DOI 10.1198/016214504000000098

the class prediction based on them very obscure. Apparently, this is different from the Bayes decision rule. Thus there is a demand for a true extension of SVM's to the multicategory case, which would inherit the optimal property of the binary case and treat the problem in a simultaneous fashion. In fact, some authors have proposed alternative multiclass formulations of the SVM considering all of the classes at once (Vapnik 1998; Weston and Watkins 1999; Bredensteiner and Bennett 1999). However, the relation of these formulations (which have been shown to be equivalent) to the Bayes decision rule is not clear from the literature, and we show that they do not always implement the Bayes decision rule. So the motive is to design an optimal MSVM that continues to deliver the efficiency of the binary SVM. With this intent, we devise a loss function with suitable class codes for the multicategory classification problem and extend the SVM paradigm to the multiclass case. We show that this extension ensures that the solution directly targets the Bayes decision rule in the same fashion as for the binary case. Its generalization to handle unequal misclassification costs is quite straightforward and is carried out in a unified way, thereby encompassing the version of the binary SVM modification for unequal costs of Lin, Lee, and Wahba (2002).

Section 2 briefly states the Bayes decision rule for either equal or unequal misclassification costs. Section 3 reviews the binary SVM. Section 4, the main part of the article, presents a formulation of the MSVM, deriving the dual problem for the proposed method as well as a data-adaptive tuning method analogous to the binary case. Section 5 presents a numerical study for illustration. Then, Section 6 explores cancer diagnosis using gene expression profiles and cloud classification using satellite radiance profiles. Finally, Section 7 presents concluding remarks and discussion of future directions.

2. THE CLASSIFICATION PROBLEM AND THE BAYES RULE

In this section we state the theoretically best classification rules derived under a decision-theoretic formulation of classification problems. Their derivations are fairly straightforward and can be found in any general reference to classification problems. In the classification problem, we are given a training dataset comprising n observations $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, n$. Here $\mathbf{x}_i \in \mathbb{R}^d$ represents covariates, and $y_i \in \{1, \ldots, k\}$ denotes a class label. The task is to learn a classification rule, $\phi(\mathbf{x}) : \mathbb{R}^d \to \{1, \ldots, k\}$, that closely matches attributes, $\mathbf{x}_i$, to the class label, y_i. We assume that each $(\mathbf{x}_i, y_i)$ is an independent random observation from a target population with probability distribution $\Pr(\mathbf{x}, y)$. Let $(\mathbf{X}, Y)$ denote a generic pair of a random realization from $\Pr(\mathbf{x}, y)$, and let $p_j(\mathbf{x}) = \Pr(Y = j | \mathbf{X} = \mathbf{x})$ for $j = 1, \ldots, k$. If the misclassification costs are all equal, then the loss by the classification rule ϕ at $(\mathbf{x}, y)$ is defined as

$$l\big(y, \phi(\mathbf{x})\big) = I\big(y \neq \phi(\mathbf{x})\big), \tag{1}$$

where $I(\cdot)$ is the indicator function, which is 1 if its argument is true and 0 otherwise. The Bayes decision rule minimizing the expected misclassification rate is

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

When the misclassification costs are not equal, (as is commonly the case when solving real-world problems), we change the loss (1) to reflect the cost structure. First, define $C_{j\ell}$ for $j, \ell = 1, \ldots, k$ as the cost of misclassifying an example from class j to class ℓ. C_{jj} for $j = 1, \ldots, k$ are all 0. The loss function for the unequal costs is then

$$l\big(y, \phi(\mathbf{x})\big) = C_{y\phi(\mathbf{x})}. \tag{3}$$

Analogous to the equal cost case, the best classification rule is given by

$$\phi_B(\mathbf{x}) = \arg \min_{j=1,\ldots,k} \sum_{\ell=1}^{k} C_{\ell j} p_\ell(\mathbf{x}). \tag{4}$$

Along with different misclassification costs, sampling bias that leads to distortion of the class proportions merits special attention in the classification problem. So far, we have assumed that the training data are truly from the general population that would generate future observations. However, it is often the case that while collecting data, we tend to balance each class by oversampling minor class examples and downsampling major class examples. Let π_j be the prior proportion of class j in the general population, and let π_j^s be the prespecified proportion of class j examples in a training dataset; π_j^s may be different from π_j if sampling bias has occurred. Let $(\mathbf{X}^s, Y^s)$ be a random pair obtained by the sampling mechanism used in the data collection stage, and let $p_j^s(\mathbf{x}) = \Pr(Y^s = j | \mathbf{X}^s = \mathbf{x})$. Then (4) can be rewritten in terms of the quantities for $(\mathbf{X}^s, Y^s)$ and π_j's, which we assume are known a priori:

$$\begin{aligned} \phi_B(\mathbf{x}) &= \arg \min_{j=1,\ldots,k} \sum_{\ell=1}^{k} \frac{\pi_\ell}{\pi_\ell^s} C_{\ell j} p_\ell^s(\mathbf{x}) \\ &= \arg \min_{j=1,\ldots,k} \sum_{\ell=1}^{k} l_{\ell j} p_\ell^s(\mathbf{x}), \end{aligned} \tag{5}$$

where $l_{\ell j}$ is defined as $(\pi_\ell / \pi_\ell^s) C_{\ell j}$, which is a modified cost that takes into account the sampling bias together with the original misclassification cost. Following the usage of Lin et al. (2002), we call the case when misclassification costs are not equal or a sampling bias exists *nonstandard*, as opposed to the standard case, when misclassification costs are equal and no sampling bias exists.

3. SUPPORT VECTOR MACHINES

We briefly review the standard SVM's for the binary case. SVM's have their roots in a geometrical interpretation of the classification problem as a problem of finding a separating hyperplane in a multidimensional input space (see Boser, Guyon, and Vapnik 1992; Vapnik 1998; Burges 1998; Cristianini and Shawe-Taylor 2000; Schölkopf and Smola 2002; references therein). The class labels y_i are either 1 or -1 in the SVM setting. Generalizing SVM classifiers from hyperplanes to nonlinear functions, the following SVM formulation has a tight link to regularization methods. The SVM methodology seeks a function $f(\mathbf{x}) = h(\mathbf{x}) + b$ with $h \in H_K$, a reproducing kernel Hilbert space (RKHS), and b, a constant minimizing

$$\frac{1}{n} \sum_{i=1}^{n} \big(1 - y_i f(\mathbf{x}_i)\big)_+ + \lambda \|h\|_{H_K}^2, \tag{6}$$

where $(x)_+ = \max(x, 0)$, and $\|h\|^2_{H_K}$ denotes the square norm of the function h defined in the RKHS with the reproducing kernel function $K(\cdot, \cdot)$. If H_K is the d-dimensional space of homogeneous linear functions $h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ with $\|h\|^2_{H_K} = \|\mathbf{w}\|^2$, then (6) reduces to the linear SVM. (For more information on RKHS, see Wahba 1990.) Here λ is a tuning parameter. The classification rule $\phi(\mathbf{x})$ induced by $f(\mathbf{x})$ is $\phi(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$. Note that the hinge loss function, $(1 - y_i f(\mathbf{x}_i))_+$, is closely related to the misclassification loss function, which can be reexpressed as $[-y_i\phi(\mathbf{x}_i)]_* = [-y_i f(\mathbf{x}_i)]_*$, where $[x]_* = I(x \geq 0)$. Indeed, the hinge loss is the tightest upper bound to the misclassification loss from the class of convex upper bounds, and when the resulting $f(\mathbf{x}_i)$ is close to either 1 or -1, the hinge loss function is close to two times the misclassification loss.

Two types of theoretical explanations are available for the observed good behavior of SVM's. The first, and the original, explanation is represented by theoretical justification of the SVM in Vapnik's structural risk minimization approach (Vapnik 1998). Vapnik's arguments are based on upper bounds of the generalization error in terms of the Vapnik–Chervonenkis dimension. The second type of explanation was provided by Lin (2002), who identified the asymptotic target function of the SVM formulation and associated it with the Bayes decision rule. With the class label Y either 1 or -1, one can verify that the Bayes decision rule in (2) is $\phi_B(\mathbf{x}) = \text{sign}(p_1(\mathbf{x}) - 1/2)$. Lin showed that if the RKHS is rich enough, then the decision rule implemented by $\text{sign}(f(\mathbf{x}))$ approaches the Bayes decision rule as the sample size n goes to ∞ for appropriately chosen λ. For example, the Gaussian kernel is one of typically used kernels for SVM's, the RKHS induced by which is flexible enough to approximate $\text{sign}(p_1(\mathbf{x}) - 1/2)$. Later, Zhang (2001) also noted that the SVM is estimating the sign of $p_1(\mathbf{x}) - 1/2$, not the probability itself.

Implementing the Bayes decision rule is not going to be the unique property of the SVM of course. (See, e.g., Wahba 2002, where penalized likelihood estimates of probabilities, which could be used to generate a classifier, are discussed in parallel with SVM's. See also Lin 2001 and Zhang 2001, which provide general treatments of various convex loss functions in relation to the Bayes decision rule.) However, the efficiency of the SVM's in going straight for the classification rule is valuable in a broad class of practical applications, including those discussed in this article. It is worth noting that due to its efficient mechanism, the SVM estimates the most likely class code, not the posterior probability for classification, and thus recovering a real probability from the SVM function is inevitably limited. As referee stated, "it would clearly be useful to output posterior probabilities based on SVM outputs," but we note here that the SVM does not carry probability information. Illustrative examples have been given by Lin (2002) and Wahba (2002).

4. MULTICATEGORY SUPPORT VECTOR MACHINES

In this section we present the extension of the SVM's to the multicategory case. Beginning with the standard case, we generalize the hinge loss function and show that the generalized formulation encompasses that of the two-category SVM, retaining desirable properties of the binary SVM. After we state the standard part of our new extension, we note its relationship to some other MSVM's that have been proposed. Then, straightforward modification follows for the nonstandard case. Finally, we derive the dual formulation through which we obtain the solution, and address how to tune the model-controlling parameter(s) involved in the MSVM.

4.1 Standard Case

Assuming that all of the misclassification costs are equal and no sampling bias exists in the training dataset, consider the k-category classification problem. To carry over the symmetry of class label representation in the binary case, we use the following vector-valued class codes, denoted by $\mathbf{y}_i$. For notational convenience, we define $\mathbf{v}_j$ for $j = 1, \ldots, k$ as a k-dimensional vector with 1 in the jth coordinate and $-1/(k-1)$ elsewhere. Then $\mathbf{y}_i$ is coded as $\mathbf{v}_j$ if example i belongs to class j. For instance, if example i falls into class 1, then $\mathbf{y}_i = \mathbf{v}_1 = (1, -1/(k-1), \ldots, -1/(k-1))$; similarly, if it falls into class k, then $\mathbf{y}_i = \mathbf{v}_k = (-1/(k-1), \ldots, -1/(k-1), 1)$. Accordingly, we define a k-tuple of separating functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ with the sum-to-0 constraint, $\sum_{j=1}^k f_j(\mathbf{x}) = 0$, for any $\mathbf{x} \in \mathbb{R}^d$. The k functions are constrained by the sum-to-0 constraint, $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ in this particular setting, for the same reason that the $p_j(\mathbf{x})$'s, the conditional probabilities of k classes, are constrained by the sum-to-1 condition, $\sum_{j=1}^k p_j(\mathbf{x}) = 1$. These constraints reflect the implicit nature of the response Y in classification problems that each y_i takes one and only one class label from $\{1, \ldots, k\}$. We justify the utility of the sum-to-0 constraint later as we illuminate properties of the proposed method. Note that the constraint holds implicitly for coded class labels $\mathbf{y}_i$. Analogous to the two-category case, we consider $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \in \prod_{j=1}^k (\{1\} + H_{K_j})$, the product space of k RKHS's H_{K_j} for $j = 1, \ldots, k$. In other words, each component $f_j(\mathbf{x})$ can be expressed as $h_j(\mathbf{x}) + b_j$ with $h_j \in H_{K_j}$. Unless there is compelling reason to believe that H_{K_j} should be different for $j = 1, \ldots, k$, we assume that they are the same RKHS denoted by H_K. Define $\mathbf{Q}$ as the $k \times k$ matrix with 0 on the diagonal and 1 elsewhere. This represents the cost matrix when all of the misclassification costs are equal. Let $\mathbf{L}(\cdot)$ be a function that maps a class label $\mathbf{y}_i$ to the jth row of the matrix $\mathbf{Q}$ if $\mathbf{y}_i$ indicates class j. So if $\mathbf{y}_i$ represents class j, then $\mathbf{L}(\mathbf{y}_i)$ is a k-dimensional vector with 0 in the jth coordinate and 1 elsewhere. Now, we propose that to find $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \in \prod_1^k (\{1\} + H_K)$ with the sum-to-0 constraint, minimizing the following quantity is a natural extension of SVM methodology:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{L}(\mathbf{y}_i) \cdot \big(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i\big)_+ + \frac{1}{2}\lambda \sum_{j=1}^{k} \|h_j\|^2_{H_K}, \qquad (7)$$

where $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ is defined as $[(f_1(\mathbf{x}_i) - y_{i1})_+, \ldots, (f_k(\mathbf{x}_i) - y_{ik})_+]$ by taking the truncate function "$(\cdot)_+$" componentwise; and the "$\cdot$" operation in the data fit functional indicates the Euclidean inner product. The classification rule induced by $\mathbf{f}(\mathbf{x})$ is naturally $\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x})$.

As with the hinge loss function in the binary case, the proposed loss function has an analogous relation to the misclassification loss (1). If $\mathbf{f}(\mathbf{x}_i)$ itself is one of the class codes, then $\mathbf{L}(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ is $k/(k-1)$ times the misclassification

loss. When $k = 2$, the generalized hinge loss function reduces to the binary hinge loss. If $\mathbf{y}_i = (1, -1)$ (1 in the binary SVM notation), then $\mathbf{L}(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ = (0, 1) \cdot [(f_1(\mathbf{x}_i) - 1)_+, (f_2(\mathbf{x}_i) + 1)_+] = (f_2(\mathbf{x}_i) + 1)_+ = (1 - f_1(\mathbf{x}_i))_+$. Likewise, if $\mathbf{y}_i = (-1, 1)$ (-1 in the binary SVM notation), then $\mathbf{L}(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ = (f_1(\mathbf{x}_i) + 1)_+$. Thereby the data fit functionals in (6) and (7) are identical, with f_1 playing the same role as f in (6). Also, note that $(\lambda/2) \sum_{j=1}^{2} \|h_j\|_{H_K}^2 = (\lambda/2) \times (\|h_1\|_{H_K}^2 + \|-h_1\|_{H_K}^2) = \lambda \|h_1\|_{H_K}^2$, by the fact that $h_1(\mathbf{x}) + h_2(\mathbf{x}) = 0$ for any $\mathbf{x}$, discussed later. So the penalties to the model complexity in (6) and (7) are identical. These identities verify that the binary SVM formulation (6) is a special case of (7) when $k = 2$. An immediate justification for this new formulation is that it carries over the efficiency of implementing the Bayes decision rule in the same fashion. We first identify the asymptotic target function of (7) in this direction. The limit of the data fit functional in (7) is $E[\mathbf{L}(\mathbf{Y}) \cdot (\mathbf{f}(\mathbf{X}) - \mathbf{Y})_+]$.

Lemma 1. The minimizer of $E[\mathbf{L}(\mathbf{Y}) \cdot (\mathbf{f}(\mathbf{X}) - \mathbf{Y})_+]$ under the sum-to-0 constraint is $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ with

$$f_j(\mathbf{x}) = \begin{cases} 1 & \text{if } j = \arg\max_{l=1,\ldots,k} p_l(\mathbf{x}) \\ -\dfrac{1}{k-1} & \text{otherwise.} \end{cases} \tag{8}$$

Proof of this lemma and other proofs are given in Appendix A. The minimizer is exactly the code of the most probable class. The classification rule induced by $\mathbf{f}(\mathbf{x})$ in Lemma 1 is $\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x}) = \arg\max_j p_j(\mathbf{x}) = \phi_B(\mathbf{x})$, the Bayes decision rule (2) for the standard multicategory case.

Other extensions to the k class case have been given by Vapnik (1998), Weston and Watkins (1999), and Bredensteiner and Bennett (1999). Guermeur (2000) showed that these are essentially equivalent and amount to using the following loss function with the same regularization terms as in (7):

$$l\big(y_i, \mathbf{f}(\mathbf{x}_i)\big) = \sum_{j=1, j \neq y_i}^{k} \big(f_j(\mathbf{x}_i) - f_{y_i}(\mathbf{x}_i) + 2\big)_+, \tag{9}$$

where the induced classifier is $\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x})$. Note that the minimizer is not unique, because adding a constant to each of the f_j, $j = 1, 2, \ldots, k$ does not change the loss function. Guermeur (2000) proposed adding sum-to-0 constraints to ensure the uniqueness of the optimal solution. The population version of the loss at $\mathbf{x}$ is given by

$$E\big[l\big(Y, \mathbf{f}(\mathbf{X})\big)\big|\mathbf{X} = \mathbf{x}\big] = \sum_{j=1}^{k} \left[\sum_{m \neq j} \big(f_m(\mathbf{x}) - f_j(\mathbf{x}) + 2\big)_+ \right] p_j(\mathbf{x}). \tag{10}$$

The following lemma shows that the minimizer of (10) does not always implement the Bayes decision rule through $\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x})$.

Lemma 2. Consider the case of $k = 3$ classes with $p_1 < 1/3 < p_2 < p_3 < 1/2$ at a given point $\mathbf{x}$. To ensure uniqueness, without loss of generality we can fix $f_1(\mathbf{x}) = -1$. Then the unique minimizer of (10), (f_1, f_2, f_3) at $\mathbf{x}$ is $(-1, 1, 1)$.

4.2 Nonstandard Case

First, we consider different misclassification costs only, assuming no sampling bias. Instead of the equal cost matrix $\mathbf{Q}$ used in the definition of $\mathbf{L}(\mathbf{y}_i)$, define a $k \times k$ cost matrix $\mathbf{C}$ with entry $C_{j\ell}$, the cost of misclassifying an example from class j to class ℓ. Modify $\mathbf{L}(\mathbf{y}_i)$ in (7) to the jth row of the cost matrix $\mathbf{C}$ if $\mathbf{y}_i$ indicates class j. When all of the misclassification costs, $C_{j\ell}$, are equal to 1, the cost matrix $\mathbf{C}$ becomes $\mathbf{Q}$. So the modified map $\mathbf{L}(\cdot)$ subsumes that for the standard case.

Now we consider the sampling bias concern together with unequal costs. As illustrated in Section 2, we need a transition from $(\mathbf{X}, Y)$ to $(\mathbf{X}^s, Y^s)$, to differentiate a "training example" population from the general population. In this case, with little abuse of notation we redefine a generalized cost matrix $\mathbf{L}$ whose entry $l_{j\ell}$ is given by $(\pi_j/\pi_j^s) C_{j\ell}$ for $j, \ell = 1, \ldots, k$. Accordingly, define $\mathbf{L}(\mathbf{y}_i)$ to be the jth row of the matrix $\mathbf{L}$ if $\mathbf{y}_i$ indicates class j. When there is no sampling bias (i.e., $\pi_j = \pi_j^s$ for all j), the generalized cost matrix $\mathbf{L}$ reduces to the ordinary cost matrix $\mathbf{C}$. With the finalized version of the cost matrix $\mathbf{L}$ and the map $\mathbf{L}(\mathbf{y}_i)$, the MSVM formulation (7) still holds as the general scheme. The following lemma identifies the minimizer of the limit of the data fit functional, which is $E[\mathbf{L}(\mathbf{Y}^s) \cdot (\mathbf{f}(\mathbf{X}^s) - \mathbf{Y}^s)_+]$.

Lemma 3. The minimizer of $E[\mathbf{L}(\mathbf{Y}^s) \cdot (\mathbf{f}(\mathbf{X}^s) - \mathbf{Y}^s)_+]$ under the sum-to-0 constraint is $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ with

$$f_j(\mathbf{x}) = \begin{cases} 1 & \text{if } j = \arg\min_{\ell=1,\ldots,k} \sum_{m=1}^{k} l_{m\ell} p_m^s(\mathbf{x}) \\ -\dfrac{1}{k-1} & \text{otherwise.} \end{cases} \tag{11}$$

The classification rule derived from the minimizer in Lemma 3 is $\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x}) = \arg\min_{j=1,\ldots,k} \sum_{\ell=1}^{k} l_{\ell j} \times p_\ell^s(\mathbf{x}) = \phi_B(\mathbf{x})$, the Bayes decision rule (5) for the nonstandard multicategory case.

4.3 The Representer Theorem and Dual Formulation

Here we explain the computations to find the minimizer of (7). The problem of finding constrained functions $(f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ minimizing (7) is turned into that of finding finite-dimensional coefficients with the aid of a variant of the representer theorem. (For the representer theorem in a regularization framework involving RKHS, see Kimeldorf and Wahba 1971, Wahba 1998.) Theorem 1 says that we can still apply the representer theorem to each component $f_j(\mathbf{x})$, but with some restrictions on the coefficients due to the sum-to-0 constraint.

Theorem 1. To find $(f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \in \prod_1^k (\{1\} + H_K)$ with the sum-to-0 constraint, minimizing (7) is equivalent to finding $(f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ of the form

$$f_j(\mathbf{x}) = b_j + \sum_{i=1}^{n} c_{ij} K(\mathbf{x}_i, \mathbf{x}), \qquad \text{for } j = 1, \ldots, k, \tag{12}$$

with the sum-to-0 constraint only at $\mathbf{x}_i$ for $i = 1, \ldots, n$, minimizing (7).

Switching to a Lagrangian formulation of the problem (7), we introduce a vector of nonnegative slack variables, $\boldsymbol{\xi}_i \in \mathbb{R}^k$,

to take care of $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$. By Theorem 1, we can write the primal problem in terms of b_j and c_{ij} only. Let $\mathbf{L}_j \in \mathbb{R}^n$ for $j = 1, \ldots, k$ be the jth column of the $n \times k$ matrix with the ith row $\mathbf{L}(\mathbf{y}_i) \equiv (L_{i1}, \ldots, L_{ik})$. Let $\boldsymbol{\xi}_{\cdot j} \in \mathbb{R}^n$ for $j = 1, \ldots, k$ be the jth column of the $n \times k$ matrix with the ith row $\boldsymbol{\xi}_i$. Similarly, let $\mathbf{y}_{\cdot j}$ denote the jth column of the $n \times k$ matrix with the ith row $\mathbf{y}_i$. With some abuse of notation, let $\mathbf{K}$ be the $n \times n$ matrix with ijth entry $K(\mathbf{x}_i, \mathbf{x}_j)$. Then the primal problem in vector notation is

$$\min L_P(\boldsymbol{\xi}, \mathbf{c}, \mathbf{b}) = \sum_{j=1}^{k} \mathbf{L}_j^t \boldsymbol{\xi}_{\cdot j} + \frac{1}{2} n\lambda \sum_{j=1}^{k} \mathbf{c}_{\cdot j}^t \mathbf{K} \mathbf{c}_{\cdot j}, \tag{13}$$

subject to

$$b_j \mathbf{e} + \mathbf{K}\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} \le \boldsymbol{\xi}_{\cdot j} \qquad \text{for } j = 1, \ldots, k, \tag{14}$$

$$\boldsymbol{\xi}_{\cdot j} \ge \mathbf{0} \qquad \text{for } j = 1, \ldots, k, \tag{15}$$

and

$$\left(\sum_{j=1}^{k} b_j\right)\mathbf{e} + \mathbf{K}\left(\sum_{j=1}^{k} \mathbf{c}_{\cdot j}\right) = \mathbf{0}. \tag{16}$$

This is a quadratic optimization problem with some equality and inequality constraints. We derive its Wolfe dual problem by introducing nonnegative Lagrange multipliers $\boldsymbol{\alpha}_{\cdot j} = (\alpha_{1j}, \ldots, \alpha_{nj})^t \in \mathbb{R}^n$ for (14), nonnegative Lagrange multipliers $\boldsymbol{\gamma}_j \in \mathbb{R}^n$ for (15), and unconstrained Lagrange multipliers $\boldsymbol{\delta}_f \in \mathbb{R}^n$ for (16), the equality constraints. Then the dual problem becomes a problem of maximizing

$$\begin{aligned} L_D = & \sum_{j=1}^{k} \mathbf{L}_j^t \boldsymbol{\xi}_{\cdot j} + \frac{1}{2} n\lambda \sum_{j=1}^{k} \mathbf{c}_{\cdot j}^t \mathbf{K} \mathbf{c}_{\cdot j} \\ & + \sum_{j=1}^{k} \boldsymbol{\alpha}_{\cdot j}^t (b_j \mathbf{e} + \mathbf{K}\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} - \boldsymbol{\xi}_{\cdot j}) \\ & - \sum_{j=1}^{k} \boldsymbol{\gamma}_j^t \boldsymbol{\xi}_{\cdot j} + \boldsymbol{\delta}_f^t \left(\left(\sum_{j=1}^{k} b_j \right) \mathbf{e} + \mathbf{K} \left(\sum_{j=1}^{k} \mathbf{c}_{\cdot j} \right) \right) \end{aligned} \tag{17}$$

subject to, for $j = 1, \ldots, k$,

$$\frac{\partial L_D}{\partial \boldsymbol{\xi}_{\cdot j}} = \mathbf{L}_j - \boldsymbol{\alpha}_{\cdot j} - \boldsymbol{\gamma}_j = \mathbf{0}, \tag{18}$$

$$\frac{\partial L_D}{\partial \mathbf{c}_{\cdot j}} = n\lambda \mathbf{K}\mathbf{c}_{\cdot j} + \mathbf{K}\boldsymbol{\alpha}_{\cdot j} + \mathbf{K}\boldsymbol{\delta}_f = \mathbf{0}, \tag{19}$$

$$\frac{\partial L_D}{\partial b_j} = (\boldsymbol{\alpha}_{\cdot j} + \boldsymbol{\delta}_f)^t \mathbf{e} = 0, \tag{20}$$

$$\boldsymbol{\alpha}_{\cdot j} \ge \mathbf{0}, \tag{21}$$

and

$$\boldsymbol{\gamma}_j \ge \mathbf{0}. \tag{22}$$

Let $\bar{\boldsymbol{\alpha}}$ be $(\sum_{j=1}^{k} \boldsymbol{\alpha}_{\cdot j})/k$. Because $\boldsymbol{\delta}_f$ is unconstrained, we may take $\boldsymbol{\delta}_f = -\bar{\boldsymbol{\alpha}}$ from (20). Accordingly, (20) becomes $(\boldsymbol{\alpha}_{\cdot j} - \bar{\boldsymbol{\alpha}})^t \mathbf{e} = 0$. Eliminating all of the primal variables in L_D by the equality constraint (18) and using relations from (19) and (20), we have the following dual problem:

$$\min L_D(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{j=1}^{k} (\boldsymbol{\alpha}_{\cdot j} - \bar{\boldsymbol{\alpha}})^t \mathbf{K} (\boldsymbol{\alpha}_{\cdot j} - \bar{\boldsymbol{\alpha}}) + n\lambda \sum_{j=1}^{k} \boldsymbol{\alpha}_{\cdot j}^t \mathbf{y}_{\cdot j} \tag{23}$$

subject to

$$\mathbf{0} \le \boldsymbol{\alpha}_{\cdot j} \le \mathbf{L}_j \qquad \text{for } j = 1, \ldots, k \tag{24}$$

and

$$(\boldsymbol{\alpha}_{\cdot j} - \bar{\boldsymbol{\alpha}})^t \mathbf{e} = 0 \qquad \text{for } j = 1, \ldots, k. \tag{25}$$

Once the quadratic programming problem is solved, the coefficients can be determined by the relation $\mathbf{c}_{\cdot j} = -(\boldsymbol{\alpha}_{\cdot j} - \bar{\boldsymbol{\alpha}})/(n\lambda)$ from (19). Note that if the matrix $\mathbf{K}$ is not strictly positive definite, then $\mathbf{c}_{\cdot j}$ is not uniquely determined. b_j can be found from any of the examples with $0 < \alpha_{ij} < L_{ij}$. By the Karush–Kuhn–Tucker complementarity conditions, the solution satisfies

$$\boldsymbol{\alpha}_{\cdot j} \perp (b_j \mathbf{e} + \mathbf{K}\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} - \boldsymbol{\xi}_{\cdot j}) \qquad \text{for } j = 1, \ldots, k \tag{26}$$

and

$$\boldsymbol{\gamma}_j = (\mathbf{L}_j - \boldsymbol{\alpha}_{\cdot j}) \perp \boldsymbol{\xi}_{\cdot j} \qquad \text{for } j = 1, \ldots, k, \tag{27}$$

where "$\perp$" means that the componentwise products are all 0. If $0 < \alpha_{ij} < L_{ij}$ for some i, then ξ_{ij} should be 0 from (27), and this implies that $b_j + \sum_{l=1}^{n} c_{lj} K(\mathbf{x}_l, \mathbf{x}_i) - y_{ij} = 0$ from (26). It is worth noting that if $(\alpha_{i1}, \ldots, \alpha_{ik}) = \mathbf{0}$ for the ith example, then $(c_{i1}, \ldots, c_{ik}) = \mathbf{0}$. Removing such an example $(\mathbf{x}_i, \mathbf{y}_i)$ would have no effect on the solution. Carrying over the notion of support vectors to the multicategory case, we define support vectors as examples with $\mathbf{c}_i = (c_{i1}, \ldots, c_{ik}) \ne \mathbf{0}$. Hence, depending on the number of support vectors, the MSVM solution may have a sparse representation, which is also one of the main characteristics of the binary SVM. In practice, solving the quadratic programming problem can be done via available optimization packages for moderate-sized problems. All of the examples presented in this article were done via MATLAB 6.1 with an interface to PATH 3.0, an optimization package implemented by Ferris and Munson (1999).

4.4 Data-Adaptive Tuning Criterion

As with other regularization methods, the effectiveness of the proposed method depends on tuning parameters. Various tuning methods have been proposed for the binary SVM's (see, e.g., Vapnik 1995; Jaakkola and Haussler 1999; Joachims 2000; Wahba, Lin, and Zhang 2000; Wahba, Lin, Lee, and Zhang 2002). We derive an approximate leave-one-out cross-validation function, called *generalized approximate cross-validation* (GACV), for the MSVM. This is based on the leave-one-out arguments reminiscent of GACV derivations for penalized likelihood methods.

For concise notation, let $J_\lambda(\mathbf{f}) = (\lambda/2) \sum_{j=1}^{k} \|h_j\|_{H_K}^2$ and $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$. Denote the objective function of the MSVM (7) by $I_\lambda(\mathbf{f}, \mathbf{y})$; that is, $I_\lambda(\mathbf{f}, \mathbf{y}) = (1/n) \sum_{i=1}^{n} g(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i)) + J_\lambda(\mathbf{f})$, where $g(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i)) \equiv \mathbf{L}(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$. Let $\mathbf{f}_\lambda$ be the minimizer of $I_\lambda(\mathbf{f}, \mathbf{y})$. It would be ideal, but is only theoretically possible, to choose tuning parameters that minimize the generalized comparative Kullback–Leibler distance (GCKL) with

respect to the loss function, $g(\mathbf{y}, \mathbf{f}(\mathbf{x}))$, averaged over a dataset with the same covariates $\mathbf{x}_i$ and unobserved $\mathbf{Y}_i$, $i = 1, \ldots, n$,

$$GCKL(\lambda) = E_{true}\frac{1}{n}\sum_{i=1}^{n} g\big(\mathbf{Y}_i, \mathbf{f}_\lambda(\mathbf{x}_i)\big)$$

$$= E_{true}\frac{1}{n}\sum_{i=1}^{n} \mathbf{L}(\mathbf{Y}_i)\cdot\big(\mathbf{f}_\lambda(\mathbf{x}_i) - \mathbf{Y}_i\big)_+.$$

To the extent that the estimate tends to the correct class code, the convex multiclass loss function tends to $k/(k-1)$ times the misclassification loss, as discussed earlier. This also justifies using GCKL as an ideal tuning measure, and thus our strategy is to develop a data-dependent computable proxy of GCKL and choose tuning parameters that minimize the proxy of GCKL.

We use the leave-one-out cross-validation arguments to derive a data-dependent proxy of the GCKL as follows. Let $\mathbf{f}_\lambda^{[-i]}$ be the solution to the variational problem when the ith observation is left out, minimizing $(1/n)\sum_{l=1,l\neq i}^{n} g(\mathbf{y}_l, \mathbf{f}_l) + J_\lambda(\mathbf{f})$. Further, $\mathbf{f}_\lambda(\mathbf{x}_i)$ and $\mathbf{f}_\lambda^{[-i]}(\mathbf{x}_i)$ are abbreviated by $\mathbf{f}_{\lambda i}$ and $\mathbf{f}_{\lambda i}^{[-i]}$. Let $f_{\lambda j}(\mathbf{x}_i)$ and $f_{\lambda j}^{[-i]}(\mathbf{x}_i)$ denote the jth components of $\mathbf{f}_\lambda(\mathbf{x}_i)$ and $\mathbf{f}_\lambda^{[-i]}(\mathbf{x}_i)$, respectively. Now, we define the leave-one-out cross-validation function that would be a reasonable proxy of $GCKL(\lambda)$: $V_0(\lambda) = (1/n)\sum_{i=1}^{n} g(\mathbf{y}_i, \mathbf{f}_{\lambda i}^{[-i]})$. $V_0(\lambda)$ can be reexpressed as the sum of $OBS(\lambda)$, the observed fit to the data measured as the average loss and $D(\lambda)$, where $OBS(\lambda) = (1/n)\sum_{i=1}^{n} g(\mathbf{y}_i, \mathbf{f}_{\lambda i})$ and $D(\lambda) = (1/n)\sum_{i=1}^{n}(g(\mathbf{y}_i, \mathbf{f}_{\lambda i}^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_{\lambda i}))$. For an approximation of $V_0(\lambda)$ without actually doing the leave-one-out procedure, which may be prohibitive for large datasets, we approximate $D(\lambda)$ further using the leave-one-out lemma. As a necessary ingredient for this lemma, we extend the domain of the function $\mathbf{L}(\cdot)$ from a set of k distinct class codes to allow argument $\mathbf{y}$ not necessarily a class code. For any $\mathbf{y} \in \mathbb{R}^k$ satisfying the sum-to-0 constraint, we define $\mathbf{L}: \mathbb{R}^k \to \mathbb{R}^k$ as $\mathbf{L}(\mathbf{y}) = (w_1(\mathbf{y})[-y_1 - 1/(k-1)]_*, \ldots, w_k(\mathbf{y})[-y_k - 1/(k-1)]_*)$, where $[\tau]_* = I(\tau \geq 0)$ and $(w_1(\mathbf{y}), \ldots, w_k(\mathbf{y}))$ is the jth row of the extended misclassification cost matrix $\mathbf{L}$ with the jl entry $(\pi_j/\pi_j^s)C_{jl}$ if $\arg\max_{l=1,\ldots,k} y_l = j$. If there are ties, then $(w_1(\mathbf{y}), \ldots, w_k(\mathbf{y}))$ is defined as the average of the rows of the cost matrix $\mathbf{L}$ corresponding to the maximal arguments. We can easily verify that $\mathbf{L}(0, \ldots, 0) = (0, \ldots, 0)$ and that the extended $\mathbf{L}(\cdot)$ coincides with the original $\mathbf{L}(\cdot)$ over the domain of class codes. We define a class prediction, $\boldsymbol{\mu}(\mathbf{f})$, given the SVM output $\mathbf{f}$ as a function truncating any component $f_j < -1/(k-1)$ to $-1/(k-1)$ and replacing the rest by

$$\frac{\sum_{j=1}^{k} I(f_j < -1/(k-1))}{k - \sum_{j=1}^{k} I(f_j < -1/(k-1))}\left(\frac{1}{k-1}\right)$$

to satisfy the sum-to-0 constraint. If $\mathbf{f}$ has a maximum component greater than 1, and all of the others less than $-1/(k-1)$, then $\boldsymbol{\mu}(\mathbf{f})$ is a k-tuple with 1 on the maximum coordinate and $-1/(k-1)$ elsewhere. So the function $\boldsymbol{\mu}$ maps $\mathbf{f}$ to its most likely class code if a class is strongly predicted by $\mathbf{f}$. In contrast, if none of the coordinates of $\mathbf{f}$ is less than $-1/(k-1)$, then $\boldsymbol{\mu}$ maps $\mathbf{f}$ to $(0, \ldots, 0)$. With this definition of $\boldsymbol{\mu}$, the following can be shown.

Lemma 4 (Leave-one-out lemma). The minimizer of $I_\lambda(\mathbf{f}, \mathbf{y}^{[-i]})$ is $\mathbf{f}_\lambda^{[-i]}$, where $\mathbf{y}^{[-i]} = (\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{y}_{i+1}, \ldots, \mathbf{y}_n)$.

For notational simplicity, we suppress the subscript "λ" from $\mathbf{f}$ and $\mathbf{f}^{[-i]}$. We approximate $g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i)$, the contribution of the ith example to $D(\lambda)$, using the foregoing lemma. Details of this approximation are given in Appendix B. Let $(\mu_{i1}(\mathbf{f}), \ldots, \mu_{ik}(\mathbf{f})) = \boldsymbol{\mu}(\mathbf{f}(\mathbf{x}_i))$. From the approximation

$$g\big(\mathbf{y}_i, \mathbf{f}_i^{[-i]}\big) - g(\mathbf{y}_i, \mathbf{f}_i) \approx (k-1)K(\mathbf{x}_i, \mathbf{x}_i) \times \sum_{j=1}^{k} L_{ij}\left[f_j(\mathbf{x}_i) + \frac{1}{k-1}\right]_* c_{ij}\big(y_{ij} - \mu_{ij}(\mathbf{f})\big),$$

$$D(\lambda) \approx \frac{1}{n}\sum_{i=1}^{n}(k-1)K(\mathbf{x}_i, \mathbf{x}_i) \times \sum_{j=1}^{k} L_{ij}\left[f_j(\mathbf{x}_i) + \frac{1}{(k-1)}\right]_* c_{ij}\big(y_{ij} - \mu_{ij}(\mathbf{f})\big).$$

Finally, we have

$$GACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{L}(\mathbf{y}_i)\cdot\big(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i\big)_+ + \frac{1}{n}\sum_{i=1}^{n}(k-1)K(\mathbf{x}_i, \mathbf{x}_i) \times \sum_{j=1}^{k} L_{ij}\left[f_j(\mathbf{x}_i) + \frac{1}{k-1}\right]_* c_{ij}\big(y_{ij} - \mu_{ij}(\mathbf{f})\big). \tag{28}$$

From a numerical standpoint, the proposed GACV may be vulnerable to small perturbations in the solution, because it involves sensitive computations, such as checking the condition $f_j(\mathbf{x}_i) < -1/(k-1)$ or evaluating the step function $[f_j(\mathbf{x}_i) + 1/(k-1)]_*$. To enhance the stability of the GACV computation, we introduce a tolerance term, ϵ. The nominal condition $f_j(\mathbf{x}_i) < -1/(k-1)$ is implemented as $f_j(\mathbf{x}_i) < -(1+\epsilon)/(k-1)$, and likewise the step function $[f_j(\mathbf{x}_i) + 1/(k-1)]_*$ is replaced by $[f_j(\mathbf{x}_i) + (1+\epsilon)/(k-1)]_*$. The tolerance is set to be 10^{-5}, for which empirical studies show that GACV becomes robust against slight perturbations of the solutions up to a certain precision.

5. NUMERICAL STUDY

In this section we illustrate the MSVM through numerical examples. We consider various tuning criteria, some of which are available only in simulation settings, and compare the performance of GACV with those theoretical criteria. Throughout this section, we use the Gaussian kernel function, $K(\mathbf{s}, \mathbf{t}) = \exp(-\frac{1}{2\sigma^2}\|\mathbf{s} - \mathbf{t}\|^2)$, and we searched λ and σ over a grid.

We considered a simple three-class example on the unit interval $[0, 1]$ with $p_1(x) = .97\exp(-3x)$, $p_3(x) = \exp(-2.5\times(x-1.2)^2)$, and $p_2(x) = 1 - p_1(x) - p_3(x)$. Class 1 is most likely for small x, whereas class 3 is most likely for large x. The in-between interval is a competing zone for three classes,

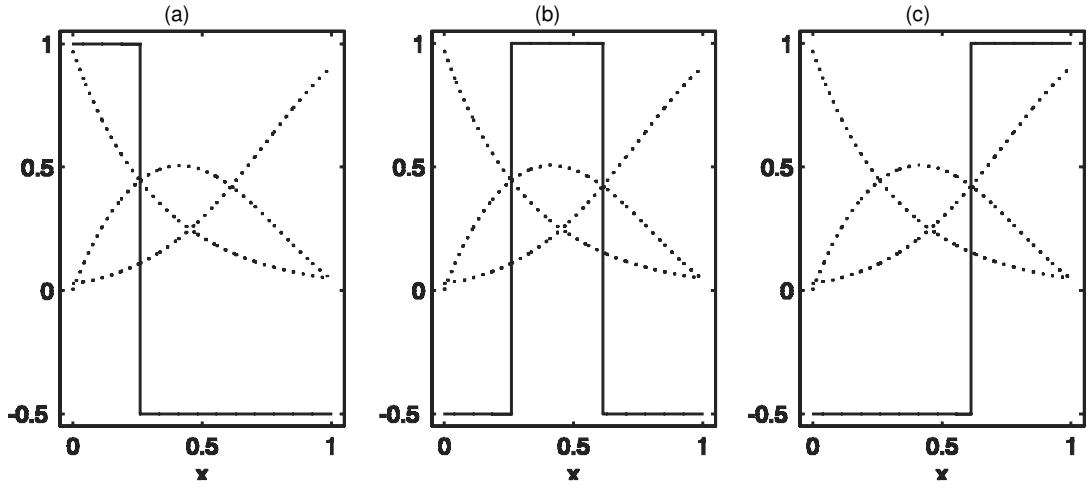

Figure 1. MSVM Target Functions for the Three-Class Example: (a) $f_1(x)$; (b) $f_2(x)$; (c) $f_3(x)$. The dotted lines are the conditional probabilities of three classes.

although class 2 is slightly dominant. Figure 1 depicts the ideal target functions, $f_1(x)$, $f_2(x)$, and $f_3(x)$, defined in Lemma 1, for this example. Here $f_j(x)$ assumes the value 1 when $p_j(x)$ is larger than $p_l(x)$, $l \neq j$, and $-1/2$ otherwise. In contrast, the ordinary one-versus-rest scheme is actually implementing the equivalent of $f_j(x) = 1$ if $p_j(x) > 1/2$ and $f_j(x) = -1$ otherwise; that is, for $f_j(x)$ to be 1, class j must be preferred over the *union of the other classes*. If no class dominates the union of the others for some x, then the $f_j(x)$'s from the one-versus-rest scheme do not carry sufficient information to identify the most probable class at x. In this example, chosen to illustrate how a one-versus-rest scheme may fail in some cases, prediction of class 2 based on $f_2(x)$ of the one-versus-rest scheme would be theoretically difficult, because the maximum of $p_2(x)$ is barely .5 across the interval. To compare the MSVM and the one-versus-rest scheme, we applied both methods to a dataset with sample size $n = 200$. We generated the attribute x_i's from the uniform distribution on [0, 1], and given x_i, randomly assigned the corresponding class label y_i according to the conditional probabilities $p_j(x)$. We jointly tuned the tuning parameters λ and σ, to minimize the GCKL distance of the estimate $\mathbf{f}_{\lambda,\sigma}$ from the true distribution.

Figure 2 shows the estimated functions for both the MSVM and the one-versus-rest methods with both tuned via GCKL. The estimated $f_2(x)$ in the one-versus-rest scheme is almost -1 at any x in the unit interval, meaning that it could not learn a classification rule associating the attribute x with the class distinction (class 2 vs. the rest, 1 or 3). In contrast, the MSVM was able to capture the relative dominance of class 2 for middle values of x. Presence of such an indeterminate region would amplify the effectiveness of the proposed MSVM. Table 1 gives the tuning parameters chosen by other tuning criteria alongside GCKL and highlights their inefficiencies for this example. When we treat all of the misclassifications equally, the true tar-

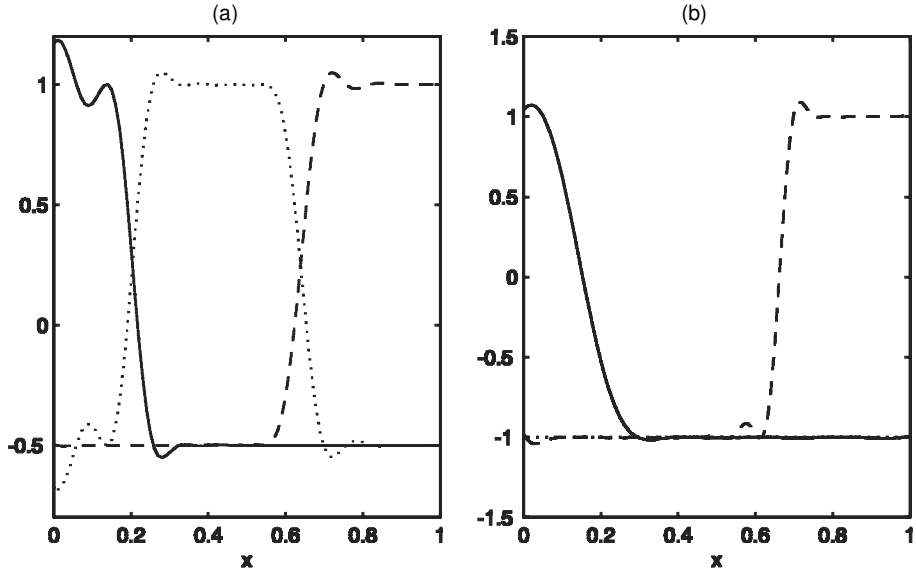

Figure 2. Comparison of the (a) MSVM and (b) One-Versus-Rest Methods. The Gaussian kernel function was used, and the tuning parameters λ and σ were chosen simultaneously via GCKL (——, f_1; · · · · ·, f_2; - - - - -, f_3).

Table 1. Tuning Criteria and Their Inefficiencies

Criterion	*($\log_2\lambda, \log_2\sigma$)*	*Inefficiency*
MISRATE	$(-11,-4)$	$*$
GCKL	$(-9,-4)$	$.4001/.3980 = 1.0051$
TUNE	$(-5,-3)$	$.4038/.3980 = 1.0145$
GACV	$(-4,-3)$	$.4171/.3980 = 1.0480$
Ten-fold cross-validation	$(-10,-1)$	$.4112/.3980 = 1.0331$
	$(-13,0)$	$.4129/.3980 = 1.0374$

NOTE: "$*$" indicates that the inefficiency is defined relative to the minimum MISRATE (.3980) at $(-11,-4)$.

get GCKL is given by

$$E_{true}\frac{1}{n}\sum_{i=1}^{n}\mathbf{L}(\mathbf{Y}_i)\cdot\big(\mathbf{f}(\mathbf{x}_i)-\mathbf{Y}_i\big)_+ = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\left(f_j(\mathbf{x}_i)+\frac{1}{k-1}\right)_+\big(1-p_j(\mathbf{x}_i)\big).$$

More directly, the misclassification rate (MISRATE) is available in simulation settings, which is defined as

$$E_{true}\frac{1}{n}\sum_{i=1}^{n}I\left(Y_i\neq\arg\max_{j=1,\ldots,k}f_{ij}\right) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}I\left(f_{ij}=\max_{l=1,\ldots,k}f_{il}\right)\big(1-p_j(\mathbf{x}_i)\big).$$

In addition, to see what we could expect from data-adaptive tuning procedures, we generated a tuning set of the same size as the training set and used the misclassification rate over the tuning set (TUNE) as a yardstick. The inefficiency of each tuning criterion is defined as the ratio of MISRATE at its minimizer to the minimum MISRATE; thus it suggests how much misclassification would be incurred relative to the smallest possible error rate by the MSVM if we know the underlying probabilities. As it is often observed in the binary case, GACV tends to pick larger λ than does GCKL. However, we observe that TUNE, the other data-adaptive criterion when a tuning set is available, gave a similar outcome. The inefficiency of GACV is 1.048, yielding a misclassification rate of .4171, slightly larger than the optimal rate .3980. As expected, this rate is a little worse than having an extra tuning set, but almost as good as 10-fold cross-validation, which requires about 10 times more computations than GACV. Ten-fold cross-validation has two minimizers, which suggests the compromising role between λ and σ for the Gaussian kernel function.

To demonstrate that the estimated functions indeed affect the test error rate, we generated 100 replicate datasets of sample size 200 and applied the MSVM and one-versus-rest SVM classifiers, combined with GCKL tuning, to each dataset. Based on the estimated classification rules, we evaluated the test error rates for both methods over a test dataset of size 10,000. For the test dataset, the Bayes misclassification rate was .3841, whereas the average test error rate of the MSVM over 100 replicates was .3951 with standard deviation .0099 and that of the one-versus-rest classifiers was .4307 with standard deviation .0132. The MSVM yielded a smaller test error rate than the one-versus-rest scheme across all of the 100 replicates.

Other simulation studies in various settings showed that MSVM outputs approximate coded classes when the tuning parameters are appropriately chosen, and that often GACV and TUNE tend to oversmooth in comparison with the theoretical tuning measures GCKL and MISRATE.

For comparison with the alternative extension using the loss function in (9), three scenarios with $k = 3$ were considered; the domain of x was set to be $[-1, 1]$, and $p_j(x)$ denotes the conditional probability of class j given x:

1. $p_1(x) = .7 - .6x^4$, $p_2(x) = .1 + .6x^4$, and $p_3(x) = .2$. In this case there is a dominant class (class with the conditional probability greater than $1/2$) for most part of the domain. The dominant class is 1 when $x^4 \leq 1/3$ and 2 when $x^4 \geq 2/3$.
2. $p_1(x) = .45 - .4x^4$, $p_2(x) = .3 + .4x^4$, and $p_3(x) = .25$. In this case, there is no dominant class over a large subset of the domain, but one class is clearly more likely than the other two classes.
3. $p_1(x) = .45 - .3x^4$, $p_2(x) = .35 + .3x^4$, and $p_3(x) = .2$. Again, there is no dominant class over a large subset of the domain, and two classes are competitive.

Figure 3 depicts the three scenarios. In each scenario, x_i's

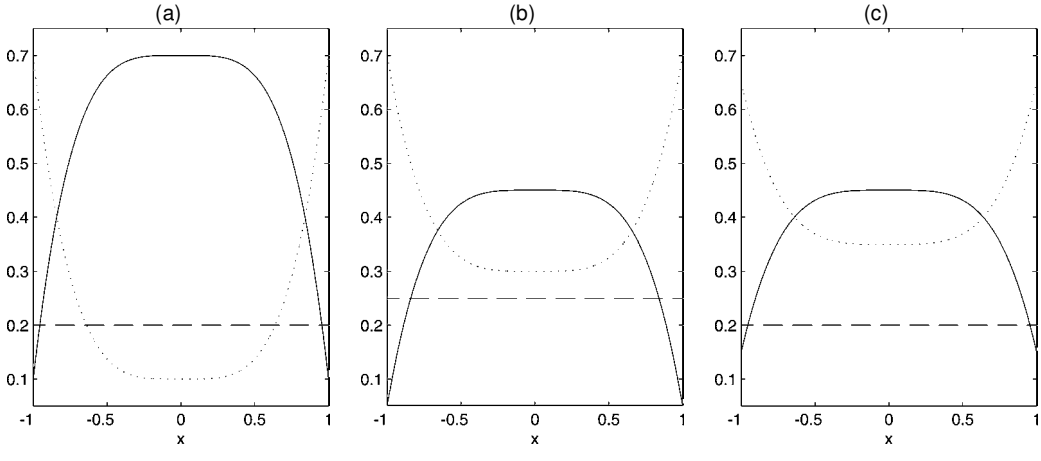

Figure 3. Underlying True Conditional Probabilities in Three Situations. (a) Scenario 1, dominant class; (b) scenario 2, the lowest two classes compete; (c) scenario 3, the highest two classes compete. Class 1, solid; class 2, dotted; class 3, dashed.

Table 2. Approximated Classification Error Rates

Scenario	*Bayes rule*	*MSVM*	*Other extension*	*One-versus-rest*
1	.3763	$.3817_{(.0021)}$	$.3784_{(.0005)}$	$.3811_{(.0017)}$
2	.5408	$.5495_{(.0040)}$	$.5547_{(.0056)}$	$.6133_{(.0072)}$
3	.5387	$.5517_{(.0045)}$	$.5708_{(.0089)}$	$.5972_{(.0071)}$

NOTE: The numbers in parentheses are the standard errors of the estimated classification error rates for each case.

of size 200 were generated from the uniform distribution on $[-1, 1]$, and the tuning parameters were chosen by GCKL. Table 2 gives the misclassification rates of the MSVM and the other extension averaged over 10 replicates. For reference, table also gives the one-versus-rest classification error rates. The error rates were numerically approximated, with the true conditional probabilities and the estimated classification rules evaluated on a fine grid. In scenario 1, the three methods are almost indistinguishabledue to the presence of a dominant class mostly over the region. When the lowest two classes compete without a dominant class in scenario 2, the MSVM and the other extension perform similarly, with clearly lower error rates than the one-versus-rest approach. But when the highest two classes compete, the MSVM gives smaller error rates than the alternative extension, as expected by Lemma 2. The two-sided Wilcoxon test for the equality of test error rates of the two methods (MSVM/other extension) shows a significant difference with the p value of .0137 using the paired 10 replicates in this case.

We carried out a small-scale empirical study over four datasets (wine, waveform, vehicle, and glass) from the UCI data repository. As a tuning method, we compared GACV with 10-fold cross-validation, which is one of the popular choices. When the problem is almost separable, GACV seems to be effective as a tuning criterion with a unique minimizer, which is typically a part of the multiple minima of 10-fold cross-validation. However, with considerable overlap among classes, we empirically observed that GACV tends to oversmooth and result in a little larger error rate compared with 10-fold cross-validation. It is of some research interest to understand why the GACV for the SVM formulation tends to overestimate λ. We compared the performance of MSVM with 10-fold CV with that of the linear discriminant analysis (LDA), the quadratic discriminant analysis (QDA), the nearest-neighbor (NN) method, the one-versus-rest binary SVM (OVR), and the alternative multiclass extension (AltMSVM). For the one-versus-rest SVM and the alternative extension, we used 10-fold cross-validation for tuning. Table 3 summarizes the comparison results in terms of the classification error rates. For wine and glass, the error rates represent the average of the misclassification rates cross-validated over 10 splits. For waveform and vehicle, we evaluated the error rates over test sets of size 4,700 and 346, which were held out. MSVM performed the best over the waveform and vehicle datasets. Over the wine dataset, the performance of MSVM was about the same as that of QDA, OVR, and AltMSVM, slightly worse than LDA, and better than NN. Over the glass data, MSVM was better than LDA but not as good as NN, which performed the best on this dataset. AltMSVM performed better than our MSVM in this case. It is clear that the relative performance of different classification methods depends on the problem at hand, and that no single classification method dominates all other methods. In practice, simple methods, such as LDA, often outperform more sophisticated methods. The MSVM is a general purpose classification method that is a useful new addition to the toolbox of the data analyst.

Table 3. Classification Error Rates

Dataset	*MSVM*	*QDA*	*LDA*	*NN*	*OVR*	*AltMSVM*
Wine	.0169	.0169	.0112	.0506	.0169	.0169
Glass	.3645	NA	.4065	.2991	.3458	.3170
Waveform	.1564	.1917	.1757	.2534	.1753	.1696
Vehicle	.0694	.1185	.1908	.1214	.0809	.0925

NOTE: NA indicates that QDA is not applicable, because one class has fewer observations than the number of variables, so the covariance matrix is not invertible.

6. APPLICATIONS

Here we present two applications to problems arising in oncology, (cancer classification using microarray data) and meteorology (cloud detection and classification via satellite radiance profiles). Complete details of the cancer classification application have been given by Lee and Lee (2003), and details of the cloud detection and classification application by Lee, Wahba, and Ackerman (2004), (see also Lee 2002).

6.1 Cancer Classification With Microarray Data

Gene expression profiles are the measurements of relative abundance of mRNA corresponding to the genes. Under the premise of gene expression patterns as "fingerprints" at the molecular level, systematic methods of classifying tumor types using gene expression data have been studied. Typical microarray training datasets (a set of pairs of a gene expression profile $\mathbf{x}_i$ and the tumor type y_i into which it falls) have a fairly small sample size, usually less than 100, whereas the number of genes involved is on the order of thousands. This poses an unprecedented challenge to some classification methodologies. The SVM is one of the methods that was successfully applied to the cancer diagnosis problems in previous studies. Because in principle SVM can handle input variables much larger than the sample size through its dual formulation, it may be well suited to the microarray data structure.

We revisited the dataset of Khan et al. (2001), who classified the small round blue cell tumors (SRBCT's) of childhood into four classes—neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin's lymphoma (NHL), and the Ewing family of tumors (EWS)—using cDNA gene expression profiles. (The dataset is available from *http://www.nhgri.nih.gov/DIR/Microarray/Supplement/.*) A total of 2,308 gene profiles out of 6,567 genes are given in the dataset after filtering for a minimal level of expression. The training set comprises 63 SRBCT cases (NB, 12; RMS, 20; BL, 8; EWS, 23), and the test set comprises 20 SRBCT cases (NB, 6; RMS, 5; BL, 3; EWS, 6) and five non-SRBCT's. Note that Burkitt's lymphoma (BL) is a subset of NHL. Khan et al. (2001) successfully classified the tumor types into four categories using artificial neural networks. Also, Yeo and Poggio (2001) applied k nearest-neighbor (NN), weighted voting, and linear SVM in one-versus-rest fashion to this four-class problem, and compared the performance of these methods when combined with several feature selection methods for each binary classification problem. Yeo and Poggio reported

that mostly SVM classifiers achieved the smallest test error and leave-one-out cross-validation error when 5 to 100 genes (features) were used. For the best results shown, perfect classification was possible in testing the blind 20 cases as well as in cross-validating 63 training cases.

For comparison, we applied the MSVM to the problem after taking the logarithm base 10 of the expression levels and standardizing arrays. Following a simple criterion of Dudoit, Fridlyand, and Speed (2002), the marginal relevance measure of gene l in class separation is defined as the ratio

$$\frac{BSS(l)}{WSS(l)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(\bar{x}_{\cdot l}^{(j)} - \bar{x}_{\cdot l})^2}{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(x_{il} - \bar{x}_{\cdot l}^{(j)})^2}, \tag{29}$$

where $\bar{x}_{\cdot l}^{(j)}$ indicates the average expression level of gene l for class j and $\bar{x}_{\cdot l}$ is the overall mean expression levels of gene l in the training set of size n. We selected genes with the largest ratios. Table 4 summarizes the classification results by MSVM's with the Gaussian kernel function. The proposed MSVM's were cross-validated for the training set in leave-one-out fashion, with zero error attained for 20, 60, and 100 genes, as shown in the second column. The last column gives the final test results. Using the top-ranked 20, 60, and 100 genes, the MSVM's correctly classify 20 test examples. With all of the genes included, one error occurs in leave-one-out cross-validation. The misclassified example is identified as EWS-T13, which reportedly occurs frequently as a leave-one-out cross-validation error (Khan et al. 2001; Yeo and Poggio 2001). The test error using all genes varies from zero to three, depending on tuning measures used. GACV tuning gave three test errors, leave-one-out cross-validation, zero to three test errors. This range of test errors is due to the fact that multiple pairs of (λ, σ) gave the same minimum in leave-one-out cross-validation tuning, and all were evaluated in the test phase, with varying results. Perfect classification in cross-validation and testing with high-dimensional inputs suggests the possibility of a compact representation of the classifier in a low dimension. (See Lee and Lee 2003, fig. 3, for a principal components analysis of the top 100 genes in the training set.) Together, the three principal components provide 66.5% (individual contributions, 27.52%, 23.12%, and 15.89%) of the variation of the 100 genes in the training set. The fourth component, not included in the analysis, explains only 3.48% of the variation in the training dataset. With the three principal components only, we applied the MSVM. Again, we achieved perfect classification in cross-validation and testing.

Figure 4 shows the predicted decision vectors (f_1, f_2, f_3, f_4) at the test examples. With the class codes and the color scheme described in the caption, we can see that all the 20 test examples from 4 classes are classified correctly. Note that the test examples are rearranged in the order EWS, BL, NB, RMS, and non-SRBCT. The test dataset includes five non-SRBCT cases.

Table 4. Leave-One-Out Cross-Validation Error and Test Error for the SRBCT Dataset

Number of genes	Leave-one-out cross-validation error	Test error
20	0	0
60	0	0
100	0	0
All	1	0–3
3 Principal components (100)	0	0

In medical diagnosis, attaching a confidence statement to each prediction may be useful in identifying such borderline cases. For classification methods whose ultimate output is the estimated conditional probability of each class at $\mathbf{x}$, one can simply set a threshold such that the classification is made only when the estimated probability of the predicted class exceeds the threshold. There have been attempts to map outputs of classifiers to conditional probabilities for various classification methods, including the SVM, in multiclass problems (see Zadrozny and Elkan 2002; Passerini, Pontil, and Frasconi 2002; Price, Knerr, Personnaz, and Dreyfus 1995; Hastie and Tibshirani 1998). However, these attempts treated multiclass problems as a series of binary class problems. Although these previous methods may be sound in producing the class probability estimate based on the outputs of binary classifiers, they do not apply to any method that handles all of the classes at once. Moreover, the SVM in particular is not designed to convey the information of class probabilities. In contrast to the conditional probability estimate of each class based on the SVM outputs, we propose a simple measure that quantifies empirically how close a new covariate vector is to the estimated class boundaries. The measure proves useful in identifying borderline observations in relatively separable cases.

We discuss some heuristics to reject weak predictions using the measure, analogous to the prediction strength for the binary SVM of Mukherjee et al. (1999). The MSVM decision vector $(f_1, \ldots, f_k)$ at $\mathbf{x}$, close to a class code, may mean strong prediction away from the classification boundary. The multiclass hinge loss with the standard cost function $\mathbf{L}(\cdot)$, $g(\mathbf{y}, \mathbf{f}(\mathbf{x})) \equiv \mathbf{L}(\mathbf{y}) \cdot (\mathbf{f}(\mathbf{x}) - \mathbf{y})_+$ sensibly measures the proximity between an MSVM decision vector $\mathbf{f}(\mathbf{x})$ and a coded class $\mathbf{y}$, reflecting how strong their association is in the classification context. For the time being, we use a class label and its vector-valued class code interchangeably as an input argument of the hinge loss g and other occasions; that is, we let $g(j, \mathbf{f}(\mathbf{x}))$ represent $g(\mathbf{v}_j, \mathbf{f}(\mathbf{x}))$. We assume that the probability of a correct prediction given $\mathbf{f}(\mathbf{x})$, $\Pr(Y = \arg\max_j f_j(\mathbf{x}) | \mathbf{f}(\mathbf{x}))$, depends on $\mathbf{f}(\mathbf{x})$ only through $g(\arg\max_j f_j(\mathbf{x}), \mathbf{f}(\mathbf{x}))$, the loss for the predicted class. The smaller the hinge loss, the stronger the prediction. Then the strength of the MSVM prediction, $\Pr(Y = \arg\max_j f_j(\mathbf{x}) | \mathbf{f}(\mathbf{x}))$, can be inferred from the training data by cross-validation. For example, leaving out $(\mathbf{x}_i, y_i)$, we get the MSVM decision vector $\mathbf{f}(\mathbf{x}_i)$ based on the remaining observations. From this, we get a pair of the loss, $g(\arg\max_j f_j(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_i))$, and the indicator of a correct decision, $I(y_i = \arg\max_j f_j(\mathbf{x}_i))$. If we further assume the complete symmetry of k classes, that is, $\Pr(Y = 1) = \cdots = \Pr(Y = k)$ and $\Pr(\mathbf{f}(\mathbf{x}) | Y = y) = \Pr(\pi(\mathbf{f}(\mathbf{x})) | Y = \pi(y))$ for any permutation operator π of $\{1, \ldots, k\}$, then it follows that $\Pr(Y = \arg\max_j f_j(\mathbf{x}) | \mathbf{f}(\mathbf{x})) = \Pr(Y = \pi(\arg\max_j f_j(\mathbf{x})) | \pi(\mathbf{f}(\mathbf{x})))$. Consequently, under these symmetry and invariance assumptions with respect to k classes, we can pool the pairs of the hinge loss and the indicator for all of the classes and estimate the invariant prediction strength function in terms of the loss, regardless of the predicted class. In almost-separable classifica-

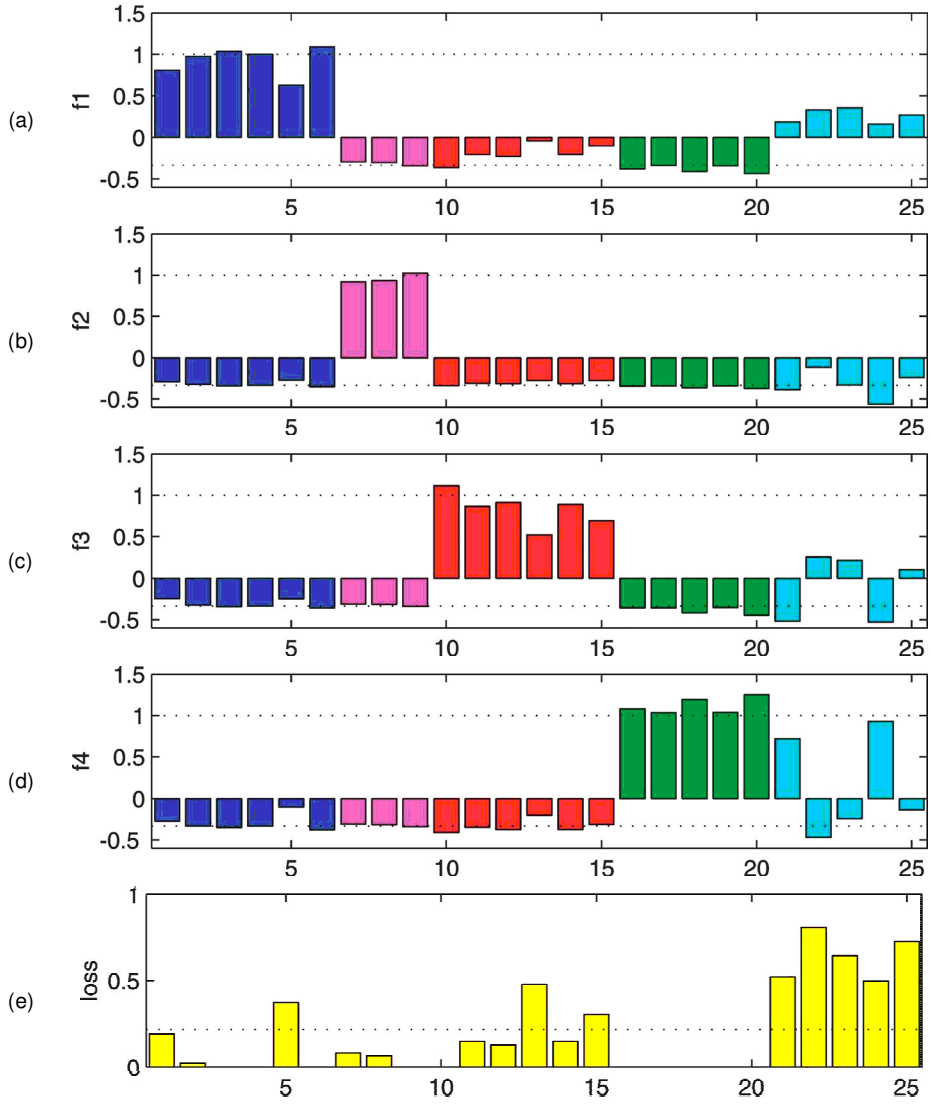

Figure 4. The Predicted Decision Vectors [(a) f_1, (b) f_2, (c) f_3, (d) f_4] for the Test Examples. The four class labels are coded according as EWS in blue: $(1, -1/3, -1/3, -1/3)$, BL in purple: $(-1/3, 1, -1/3, -1/3)$, NB in red: $(-1/3, -1/3, 1, -1/3)$, and RMS in green: $(-1/3, -1/3, -1/3, 1)$. The colors indicate the true class identities of the test examples. All the 20 test examples from four classes are classified correctly and the estimated decision vectors are pretty close to their ideal class representation. The fitted MSVM decision vectors for the five non-SRBCT examples are plotted in cyan. (e) The loss for the predicted decision vector at each test example. The last five losses corresponding to the predictions of non-SRBCT's all exceed the threshold (the dotted line) below which indicates a strong prediction. Three test examples falling into the known four classes cannot be classified confidently by the same threshold. (Reproduced with permission from Lee and Lee 2003. Copyright 2003, Oxford University Press.)

tion problems, we might see the loss values for the correct classifications only, impeding estimation of the prediction strength. We can apply the heuristics of predicting a class only when its corresponding loss is less than, say, the 95th percentile of the empirical loss distribution. This cautious measure was exercised in identifying the five non-SRBCT's. Figure 4(e) depicts the loss for the predicted MSVM decision vector at each test example, including five non-SRBCT's. The dotted line indicates the threshold of rejecting a prediction given the loss; that is, any prediction with loss above the dotted line will be rejected. This threshold was set at .2171, which is a jackknife estimate of the 95th percentile of the loss distribution from 63 correct predictions in the training dataset. The losses corresponding to the predictions of the five non-SRBCT's all exceed the threshold, whereas 3 test examples out of 20 can not be classified confidently by thresholding.

6.2 Cloud Classification With Radiance Profiles

The moderate resolution imaging spectroradiometer (MODIS) is a key instrument of the earth observing system (EOS). It measures radiances at 36 wavelengths including infrared and visible bands every 1 to 2 days with a spatial resolution of 250 m to 1 km. (For more information about the MODIS instrument, see *http://modis.gsfc.nasa.gov/*.) EOS models require knowledge of whether a radiance profile is cloud-free or not. If the profile is not cloud-free, then information concerning the types of clouds is valuable. (For more information on the MODIS cloud mask algorithm with a simple threshold technique, see Ackerman et al. 1998.) We applied the MSVM to simulated MODIS-type channel data to classify the radiance profiles as clear, liquid clouds, or ice clouds. Satellite observations at 12 wavelengths (.66, .86, .46, .55, 1.2, 1.6, 2.1, 6.6, 7.3, 8.6, 11, and 12 microns, or MODIS channels 1, 2, 3, 4, 5,

6, 7, 27, 28, 29, 31, and 32) were simulated using DISORT, driven by STREAMER (Key and Schweiger 1998). Setting atmospheric conditions as simulation parameters, we selected atmospheric temperature and moisture profiles from the 3I thermodynamic initial guess retrieval (TIGR) database, and set the surface to be water. A total of 744 radiance profiles over the ocean (81 clear scenes, 202 liquid clouds, and 461 ice clouds) are included in the dataset. Each simulated radiance profile consists of seven reflectances (R), at .66, .86, .46, .55, 1.2, 1.6, and 2.1 microns, and five brightness temperatures (BT), at 6.6, 7.3, 8.6, 11, and 12 microns. No single channel seemed to give a clear separation of the three categories. The two variables $R_{channel2}$ and $\log_{10}(R_{channel5}/R_{channel6})$ were initially used for classification based on an understanding of the underlying physics and an examination of several other scatterplots. To test how predictive $R_{channel2}$ and $\log_{10}(R_{channel5}/R_{channel6})$ are, we split the dataset into a training set and a test set, and applied the MSVM with two features only to the training data. We randomly selected 370 examples, almost half of the original data, as the training set. We used the Gaussian kernel and tuned the tuning parameters by five-fold cross-validation. The test error rate of the SVM rule over 374 test examples was 11.5% (43 of 374). Figure 5(a) shows the classification boundaries determined by the training dataset in this case. Note that many ice cloud examples are hidden underneath the clear-sky examples in the plot. Most of the misclassifications in testing occurred due to the considerable overlap between ice clouds and clear-sky examples at the lower left corner of the plot. It turned out that adding three more promising variables to the MSVM did not significantly improve the classification accuracy. These variables are given in the second row of Table 5; again the choice was based on knowledge of the underlying physics and pairwise scatterplots. We could classify correctly just five more examples than in the two-features-only case with a misclassification rate of 10.16% (38 of 374). Assuming no such domain knowledge regarding which features to examine, we applied the MSVM to the original 12 radiance channels without any transformations or variable selections. This yielded 12.03% test error rate, slightly larger than the MSVM's with two or five features. Interestingly, when all of the variables were transformed by the logarithm function, the MSVM achieved its minimum error rate. We compared the MSVM with the tree-structured classification method, because it is somewhat similar to, albeit much more sophisticated than, the MODIS cloud mask algorithm. We used the library "tree" in the R package. For each combination of the variables, we determined the size of the fitted tree by 10-fold cross validation of the training set and estimated its error rate over the test set. The results are given in the column "TREE" in Table 5. The MSVM gives smaller test error rates than the tree method over all of the combinations of the variables considered. This suggests the possibility that the proposed MSVM improves the accuracy of the current cloud detection algorithm. To roughly measure the difficulty of the classification problem due to the intrinsic overlap between class distributions, we applied the NN method; the results, given in the last column of Table 5, suggest that the dataset is not trivially separable. It would be interesting to investigate further whether any sophisticated variable (feature) selection method may substantially improve the accuracy.

Table 5. Test Error Rates for the Combinations of Variables and Classifiers

Number of variables	Variable descriptions	Test error rates (%) MSVM	TREE	1-NN
2	(a) R_2, $\log_{10}(R_5/R_6)$	11.50	14.97	16.58
5	(a)+R_1/R_2, BT_{31}, $BT_{32} - BT_{29}$	10.16	15.24	12.30
12	(b) original 12 variables	12.03	16.84	20.86
12	log-transformed (b)	9.89	16.84	18.98

So far, we have treated different types of misclassification equally. However, a misclassification of clouds as clear could be more serious than other kinds of misclassifications in practice, because essentially this cloud detection algorithm will be used

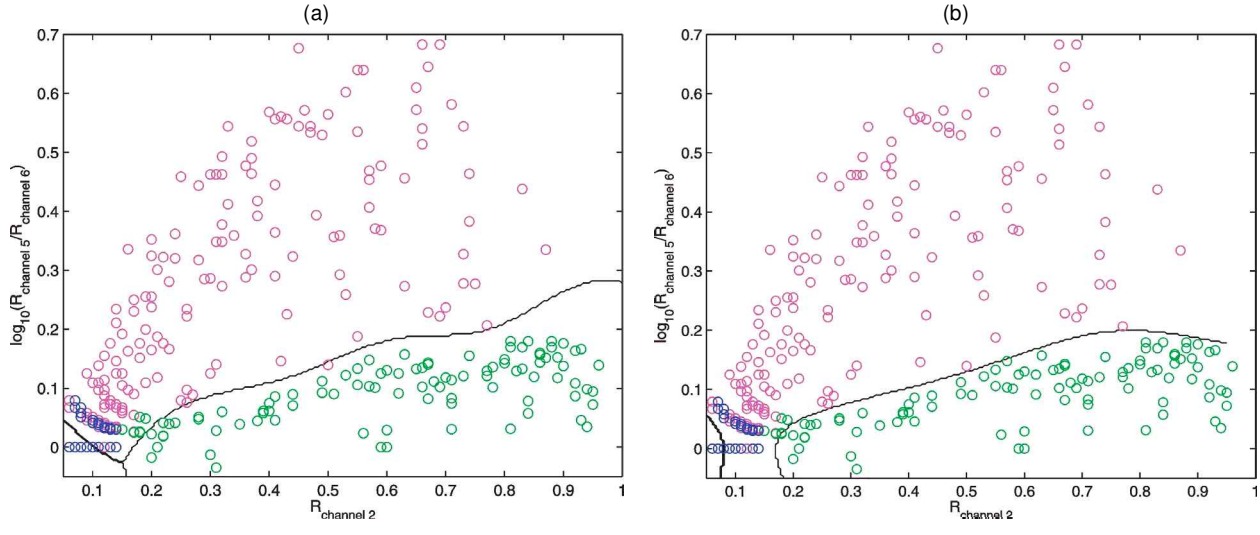

Figure 5. The Classification Boundaries of the MSVM. They are determined by the MSVM using 370 training examples randomly selected from the dataset in (a) the standard case and (b) the nonstandard case, where the cost of misclassifying clouds as clear is 1.5 times higher than the cost of other types of misclassifications. Clear sky, blue; water clouds, green; ice clouds, purple. (Reproduced with permission from Lee et al. 2004. Copyright 2004, American Meteorological Society.)

as cloud mask. We considered a cost structure that penalizes misclassifying clouds as clear 1.5 times more than misclassifications of other kinds; its corresponding classification boundaries are shown in Figure 5(b). We observed that if the cost 1.5 is changed to 2, then no region at all remains for the clear-sky category within the square range of the two features considered here. The approach to estimating the prediction strength given in Section 6.1 can be generalized to the nonstandard case, if desired.

7. CONCLUDING REMARKS

We have proposed a loss function deliberately tailored to target the coded class with the maximum conditional probability for multicategory classification problems. Using the loss function, we have extended the classification paradigm of SVM's to the multicategory case so that the resulting classifier approximates the optimal classification rule. The nonstandard MSVM that we have proposed allows a unifying formulation when there are possibly nonrepresentative training sets and either equal or unequal misclassification costs. We derived an approximate leave-one-out cross-validation function for tuning the method, and compared this with conventional k-fold cross-validation methods. The comparisons, through several numerical examples, suggested that the proposed tuning measure is sharper near its minimizer than the k-fold cross-validation method, but tends to slightly oversmooth. Then we demonstrated the usefulness of the MSVM through applications to a cancer classification problem with microarray data and cloud classification problems with radiance profiles.

Although the high dimensionality of data is tractable in the SVM paradigm, its original formulation does not accommodate variable selection. Rather, it provides observationwise data reduction through support vectors. Depending on applications, it is of great importance not only to achieve the smallest error rate by a classifier, but also to have its compact representation for better interpretation. For instance, classification problems in data mining and bioinformatics often pose a question as to which subsets of the variables are most responsible for the class separation. A valuable exercise would be to further generalize some variable selection methods for binary SVM's to the MSVM. Another direction of future work includes establishing the MSVM's advantages theoretically, such as its convergence rates to the optimal error rate, compared with those indirect methods of classifying via estimation of the conditional probability or density functions.

The MSVM is a generic approach to multiclass problems treating all of the classes simultaneously. We believe that it is a useful addition to the class of nonparametric multicategory classification methods.

APPENDIX A: PROOFS

Proof of Lemma 1

Because $E[\mathbf{L}(\mathbf{Y})\cdot(\mathbf{f}(\mathbf{X})-\mathbf{Y})_+] = E(E[\mathbf{L}(\mathbf{Y})\cdot(\mathbf{f}(\mathbf{X})-\mathbf{Y})_+|\mathbf{X}])$, we can minimize $E[\mathbf{L}(\mathbf{Y})\cdot(\mathbf{f}(\mathbf{X})-\mathbf{Y})_+]$ by minimizing $E[\mathbf{L}(\mathbf{Y})\cdot(\mathbf{f}(\mathbf{X})-\mathbf{Y})_+|\mathbf{X}=\mathbf{x}]$ for every $\mathbf{x}$. If we write out the functional for each $\mathbf{x}$, then we have

$$
\begin{aligned}
&E\big[\mathbf{L}(\mathbf{Y})\cdot\big(\mathbf{f}(\mathbf{X})-\mathbf{Y}\big)_+\big|\mathbf{X}=\mathbf{x}\big] \\
&\quad=\sum_{j=1}^{k}\left(\sum_{l\neq j}\left(f_l(\mathbf{x})+\frac{1}{k-1}\right)_+\right)p_j(\mathbf{x}) \\
&\quad=\sum_{j=1}^{k}\big(1-p_j(\mathbf{x})\big)\left(f_j(\mathbf{x})+\frac{1}{k-1}\right)_+. \qquad \text{(A.1)}
\end{aligned}
$$

Here we claim that it is sufficient to search over $\mathbf{f}(\mathbf{x})$ with $f_j(\mathbf{x})\geq -1/(k-1)$ for all $j=1,\ldots,k$, to minimize (A.1). If any $f_j(\mathbf{x}) < -1/(k-1)$, then we can always find another $\mathbf{f}^*(\mathbf{x})$ that is better than or as good as $\mathbf{f}(\mathbf{x})$ in reducing the expected loss, as follows. Set $f_j^*(\mathbf{x})$ to be $-1/(k-1)$ and subtract the surplus $-1/(k-1)-f_j(\mathbf{x})$ from other component $f_l(\mathbf{x})$'s that are greater than $-1/(k-1)$. The existence of such other components is always guaranteed by the sum-to-0 constraint. Determine $f_i^*(\mathbf{x})$ in accordance with the modifications. By doing so, we get $\mathbf{f}^*(\mathbf{x})$ such that $(f_j^*(\mathbf{x})+1/(k-1))_+ \leq (f_j(\mathbf{x})+1/(k-1))_+$ for each j. Because the expected loss is a nonnegatively weighted sum of $(f_j(\mathbf{x})+1/(k-1))_+$, it is sufficient to consider $\mathbf{f}(\mathbf{x})$ with $f_j(\mathbf{x})\geq -1/(k-1)$ for all $j=1,\ldots,k$. Dropping the truncate functions from (A.1), and rearranging, we get

$$
\begin{aligned}
&E\big[\mathbf{L}(\mathbf{Y})\cdot\big(\mathbf{f}(\mathbf{X})-\mathbf{Y}\big)_+\big|\mathbf{X}=\mathbf{x}\big] \\
&\quad=1+\sum_{j=1}^{k-1}\big(1-p_j(\mathbf{x})\big)f_j(\mathbf{x})+\big(1-p_k(\mathbf{x})\big)\left(-\sum_{j=1}^{k-1}f_j(\mathbf{x})\right) \\
&\quad=1+\sum_{j=1}^{k-1}\big(p_k(\mathbf{x})-p_j(\mathbf{x})\big)f_j(\mathbf{x}).
\end{aligned}
$$

Without loss of generality, we may assume that $k=\arg\max_{j=1,\ldots,k}p_j(\mathbf{x})$ by the symmetry in the class labels. This implies that to minimize the expected loss, $f_j(\mathbf{x})$ should be $-1/(k-1)$ for $j=1,\ldots,k-1$ because of the nonnegativity of $p_k(\mathbf{x})-p_j(\mathbf{x})$. Finally, we have $f_k(\mathbf{x})=1$ by the sum-to-0 constraint.

Proof of Lemma 2

For brevity, we omit the argument $\mathbf{x}$ for f_j and p_j throughout the proof, and refer to (10) as $R(\mathbf{f}(\mathbf{x}))$. Because we fix $f_1=-1$, R can be seen as a function of (f_2, f_3),

$$
\begin{aligned}
R(f_2,f_3) &= (3+f_2)_+p_1+(3+f_3)_+p_1+(1-f_2)_+p_2 \\
&\quad+(2+f_3-f_2)_+p_2+(1-f_3)_+p_3+(2+f_2-f_3)_+p_3.
\end{aligned}
$$

Now consider (f_2, f_3) in the neighborhood of $(1,1)$: $0<f_2<2$ and $0<f_3<2$. In this neighborhood we have $R(f_2,f_3)=4p_1+2+[f_2(1-2p_2)+(1-f_2)_+p_2]+[f_3(1-2p_3)+(1-f_3)_+p_3]$ and $R(1,1)=4p_1+2+(1-2p_2)+(1-2p_3)$. Because $1/3<p_2<1/2$, if $f_2>1$, then $f_2(1-2p_2)+(1-f_2)_+p_2=f_2(1-2p_2)>1-2p_2$, and if $f_2<1$, then $f_2(1-2p_2)+(1-f_2)_+p_2=f_2(1-2p_2)+(1-f_2)p_2=(1-f_2)(3p_2-1)+(1-2p_2)>1-2p_2$. Therefore, $f_2(1-2p_2)+(1-f_2)_+p_2\geq 1-2p_2$, with the equality holding only when $f_2=1$. Similarly, $f_3(1-2p_3)+(1-f_3)_+p_3\geq 1-2p_3$, with the equality holding only when $f_3=1$. Hence, for any $f_2\in(0,2)$ and $f_3\in(0,2)$, we have that $R(f_2,f_3)\geq R(1,1)$, with the equality holding only if $(f_2,f_3)=(1,1)$. Because R is convex, we see that $(1,1)$ is the unique global minimizer of $R(f_2,f_3)$. The lemma is proved.

In the foregoing, we used the constraint $f_1=-1$. Other constraints certainly can be used. For example, if we use the constraint $f_1+f_2+f_3=0$ instead of $f_1=-1$, then the global minimizer under the constraint is $(-4/3, 2/3, 2/3)$. This is easily seen from the fact that $R(f_1,f_2,f_3)=R(f_1+c,f_2+c,f_3+c)$ for any (f_1,f_2,f_3) and any constant c.

Proof of Lemma 3

Parallel to all of the arguments used for the proof of Lemma 1, it can be shown that

$$E\big[\mathbf{L}(\mathbf{Y}^s)\cdot\big(\mathbf{f}(\mathbf{X}^s)-\mathbf{Y}^s\big)_+\big|\mathbf{X}^s=\mathbf{x}\big] = \frac{1}{k-1}\sum_{j=1}^{k}\sum_{\ell=1}^{k} l_{\ell j}\,p_\ell^s(\mathbf{x}) + \sum_{j=1}^{k}\left(\sum_{\ell=1}^{k} l_{\ell j}\,p_\ell^s(\mathbf{x})\right) f_j(\mathbf{x}).$$

We can immediately eliminate from consideration the first term, which does not involve any $f_j(\mathbf{x})$. To make the equation simpler, let $W_j(\mathbf{x})$ be $\sum_{\ell=1}^{k} l_{\ell j}\,p_\ell^s(\mathbf{x})$ for $j=1,\ldots,k$. Then the whole equation reduces to the following up to a constant:

$$\sum_{j=1}^{k} W_j(\mathbf{x})f_j(\mathbf{x}) = \sum_{j=1}^{k-1} W_j(\mathbf{x})f_j(\mathbf{x}) + W_k(\mathbf{x})\left(-\sum_{j=1}^{k-1} f_j(\mathbf{x})\right) = \sum_{j=1}^{k-1}\big(W_j(\mathbf{x})-W_k(\mathbf{x})\big)f_j(\mathbf{x}).$$

Without loss of generality, we may assume that $k=\arg\min_{j=1,\ldots,k} W_j(\mathbf{x})$. To minimize the expected quantity, $f_j(\mathbf{x})$ should be $-1/(k-1)$ for $j=1,\ldots,k-1$ because of the nonnegativity of $W_j(\mathbf{x})-W_k(\mathbf{x})$ and $f_j(\mathbf{x})\ge -1/(k-1)$ for all $j=1,\ldots,k$. Finally, we have $f_k(\mathbf{x})=1$ by the sum-to-0 constraint.

Proof of Theorem 1

Consider $f_j(\mathbf{x})=b_j+h_j(\mathbf{x})$ with $h_j\in H_K$. Decompose $h_j(\cdot)=\sum_{l=1}^{n} c_{lj}K(\mathbf{x}_l,\cdot)+\rho_j(\cdot)$ for $j=1,\ldots,k$, where c_{ij}'s are some constants and $\rho_j(\cdot)$ is the element in the RKHS orthogonal to the span of $\{K(\mathbf{x}_i,\cdot), i=1,\ldots,n\}$. By the sum-to-0 constraint, $f_k(\cdot)=-\sum_{j=1}^{k-1} b_j-\sum_{j=1}^{k-1}\sum_{i=1}^{n} c_{ij}K(\mathbf{x}_i,\cdot)-\sum_{j=1}^{k-1}\rho_j(\cdot)$. By the definition of the reproducing kernel $K(\cdot,\cdot)$, $(h_j, K(\mathbf{x}_i,\cdot))_{H_K}=h_j(\mathbf{x}_i)$ for $i=1,\ldots,n$. Then

$$f_j(\mathbf{x}_i)=b_j+h_j(\mathbf{x}_i)=b_j+\big(h_j,K(\mathbf{x}_i,\cdot)\big)_{H_K} = b_j+\left(\sum_{l=1}^{n} c_{lj}K(\mathbf{x}_l,\cdot)+\rho_j(\cdot),K(\mathbf{x}_i,\cdot)\right)_{H_K} = b_j+\sum_{l=1}^{n} c_{lj}K(\mathbf{x}_l,\mathbf{x}_i).$$

Thus the data fit functional in (7) does not depend on $\rho_j(\cdot)$ at all for $j=1,\ldots,k$. On the other hand, we have $\|h_j\|_{H_K}^2=\sum_{i,l} c_{ij}c_{lj}K(\mathbf{x}_l,\mathbf{x}_i)+\|\rho_j\|_{H_K}^2$ for $j=1,\ldots,k-1$, and $\|h_k\|_{H_K}^2=\|\sum_{j=1}^{k-1}\sum_{i=1}^{n} c_{ij}K(\mathbf{x}_i,\cdot)\|_{H_K}^2+\|\sum_{j=1}^{k-1}\rho_j\|_{H_K}^2$. To minimize (7), obviously $\rho_j(\cdot)$ should vanish. It remains to show that minimizing (7) under the sum-to-0 constraint at the data points only is equivalent to minimizing (7) under the constraint for every $\mathbf{x}$. Now let $\mathbf{K}$ be the $n\times n$ matrix with il entry $K(\mathbf{x}_i,\mathbf{x}_l)$. Let $\mathbf{e}$ be the column vector with n 1's and let $\mathbf{c}_{\cdot j}=(c_{1j},\ldots,c_{nj})^t$. Given the representation (12), consider the problem of minimizing (7) under $(\sum_{j=1}^{k} b_j)\mathbf{e}+\mathbf{K}(\sum_{j=1}^{k}\mathbf{c}_{\cdot j})=0$. For any $f_j(\cdot)=b_j+\sum_{i=1}^{n} c_{ij}K(\mathbf{x}_i,\cdot)$ satisfying $(\sum_{j=1}^{k} b_j)\mathbf{e}+\mathbf{K}(\sum_{j=1}^{k}\mathbf{c}_{\cdot j})=0$, define the centered solution $f_j^*(\cdot)=b_j^*+\sum_{i=1}^{n} c_{ij}^*K(\mathbf{x}_i,\cdot)=(b_j-\bar b)+\sum_{i=1}^{n}(c_{ij}-\bar c_i)K(\mathbf{x}_i,\cdot)$, where $\bar b=(1/k)\sum_{j=1}^{k} b_j$ and $\bar c_i=(1/k)\sum_{j=1}^{k} c_{ij}$. Then $f_j(\mathbf{x}_i)=f_j^*(\mathbf{x}_i)$, and

$$\sum_{j=1}^{k}\|h_j^*\|_{H_K}^2=\sum_{j=1}^{k}\mathbf{c}_{\cdot j}^t\mathbf{K}\mathbf{c}_{\cdot j}-k\bar{\mathbf{c}}^t\mathbf{K}\bar{\mathbf{c}}\le\sum_{j=1}^{k}\mathbf{c}_{\cdot j}^t\mathbf{K}\mathbf{c}_{\cdot j}=\sum_{j=1}^{k}\|h_j\|_{H_K}^2.$$

Because the equality holds only when $\mathbf{K}\bar{\mathbf{c}}=0$ [i.e., $\mathbf{K}(\sum_{j=1}^{k}\mathbf{c}_{\cdot j})=0$], we know that at the minimizer, $\mathbf{K}(\sum_{j=1}^{k}\mathbf{c}_{\cdot j})=0$, and thus $\sum_{j=1}^{k} b_j=0$. Observe that $\mathbf{K}(\sum_{j=1}^{k}\mathbf{c}_{\cdot j})=0$ implies

$$\left(\sum_{j=1}^{k}\mathbf{c}_{\cdot j}\right)^t\mathbf{K}\left(\sum_{j=1}^{k}\mathbf{c}_{\cdot j}\right)=\left\|\sum_{i=1}^{n}\left(\sum_{j=1}^{k}c_{ij}\right)K(\mathbf{x}_i,\cdot)\right\|_{H_K}^2 = \left\|\sum_{j=1}^{k}\sum_{i=1}^{n}c_{ij}K(\mathbf{x}_i,\cdot)\right\|_{H_K}^2=0.$$

This means that $\sum_{j=1}^{k}\sum_{i=1}^{n} c_{ij}K(\mathbf{x}_i,\mathbf{x})=0$ for every $\mathbf{x}$. Hence, minimizing (7) under the sum-to-0 constraint at the data points is equivalent to minimizing (7) under $\sum_{j=1}^{k} b_j+\sum_{j=1}^{k}\sum_{i=1}^{n} c_{ij}K(\mathbf{x}_i,\mathbf{x})=0$ for every $\mathbf{x}$.

Proof of Lemma 4 (Leave-One-Out Lemma)

Observe that

$$I_\lambda\big(\mathbf{f}_\lambda^{[-i]},\mathbf{y}^{[-i]}\big) = \frac{1}{n}g\big(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}),\mathbf{f}_{\lambda i}^{[-i]}\big)+\frac{1}{n}\sum_{l=1,l\ne i}^{n} g\big(\mathbf{y}_l,\mathbf{f}_{\lambda l}^{[-i]}\big)+J_\lambda\big(\mathbf{f}_\lambda^{[-i]}\big)$$

$$\le \frac{1}{n}g\big(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}),\mathbf{f}_{\lambda i}^{[-i]}\big)+\frac{1}{n}\sum_{l=1,l\ne i}^{n} g(\mathbf{y}_l,\mathbf{f}_l)+J_\lambda(\mathbf{f})$$

$$\le \frac{1}{n}g\big(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}),\mathbf{f}_i\big)+\frac{1}{n}\sum_{l=1,l\ne i}^{n} g(\mathbf{y}_l,\mathbf{f}_l)+J_\lambda(\mathbf{f})=I_\lambda\big(\mathbf{f},\mathbf{y}^{[-i]}\big).$$

The first inequality holds by the definition of $\mathbf{f}_\lambda^{[-i]}$. Note that the jth coordinate of $\mathbf{L}(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}))$ is positive only when $\mu_j(\mathbf{f}_{\lambda i}^{[-i]})=-1/(k-1)$, whereas the corresponding jth coordinate of $(\mathbf{f}_{\lambda i}^{[-i]}-\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}))_+$ will be 0 because $f_{\lambda j}^{[-i]}(\mathbf{x}_i)<-1/(k-1)$ for $\mu_j(\mathbf{f}_{\lambda i}^{[-i]})=-1/(k-1)$. As a result, $g(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}),\mathbf{f}_{\lambda i}^{[-i]})=\mathbf{L}(\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}))\cdot(\mathbf{f}_{\lambda i}^{[-i]}-\boldsymbol{\mu}(\mathbf{f}_{\lambda i}^{[-i]}))_+=0$. Thus the second inequality follows by the nonnegativity of the function g. This completes the proof.

APPENDIX B: APPROXIMATION OF $g(\mathbf{y}_i,\mathbf{f}_i^{[-i]})-g(\mathbf{y}_i,\mathbf{f}_i)$

Due to the sum-to-0 constraint, it suffices to consider $k-1$ coordinates of $\mathbf{y}_i$ and $\mathbf{f}_i$ as arguments of g, which correspond to non-0 components of $\mathbf{L}(\mathbf{y}_i)$. Suppose that $\mathbf{y}_i=(-1/(k-1),\ldots,-1/(k-1),1)$; all of the arguments will hold analogously for other class examples. By the first-order Taylor expansion, we have

$$g\big(\mathbf{y}_i,\mathbf{f}_i^{[-i]}\big)-g(\mathbf{y}_i,\mathbf{f}_i)\approx-\sum_{j=1}^{k-1}\frac{\partial}{\partial f_j}g(\mathbf{y}_i,\mathbf{f}_i)\big(f_j(\mathbf{x}_i)-f_j^{[-i]}(\mathbf{x}_i)\big). \tag{B.1}$$

Ignoring nondifferentiable points of g for a moment, we have, for $j=1,\ldots,k-1$,

$$\frac{\partial}{\partial f_j}g(\mathbf{y}_i,\mathbf{f}_i)=\mathbf{L}(\mathbf{y}_i)\cdot\left(0,\ldots,0,\left[f_j(\mathbf{x}_i)+\frac{1}{k-1}\right]_*,0,\ldots,0\right) = L_{ij}\left[f_j(\mathbf{x}_i)+\frac{1}{k-1}\right]_*.$$

Let $(\mu_{i1}(\mathbf{f}),\ldots,\mu_{ik}(\mathbf{f}))=\boldsymbol{\mu}(\mathbf{f}(\mathbf{x}_i))$ and, similarly, $(\mu_{i1}(\mathbf{f}^{[-i]}),\ldots,\mu_{ik}(\mathbf{f}^{[-i]}))=\boldsymbol{\mu}(\mathbf{f}^{[-\mathbf{i}]}(\mathbf{x}_i))$. Using the leave-one-out lemma for

$j = 1, \ldots, k-1$ and the Taylor expansion,

$$f_j(\mathbf{x}_i) - f_j^{[-i]}(\mathbf{x}_i) \approx \left(\frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i1}}, \ldots, \frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i,k-1}}\right) \begin{pmatrix} y_{i1} - \mu_{i1}(\mathbf{f}^{[-i]}) \\ \vdots \\ y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}^{[-i]}) \end{pmatrix}. \quad \text{(B.2)}$$

The solution for the MSVM is given by $f_j(\mathbf{x}_i) = \sum_{i'=1}^{n} c_{i'j} K(\mathbf{x}_i, \mathbf{x}_{i'}) + b_j = -\sum_{i'=1}^{n} (\alpha_{i'j} - \bar{\alpha}_{i'})/(n\lambda) K(\mathbf{x}_i, \mathbf{x}_{i'}) + b_j$. Parallel to the binary case, we rewrite $c_{i'j} = -(k-1) y_{i'j} c_{i'j}$ if the i'th example is not from class j, and $c_{i'j} = (k-1) \sum_{l=1,\, l \neq j}^{k} y_{i'l} c_{i'l}$ otherwise. Hence,

$$\begin{pmatrix} \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i1}} & \cdots & \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i,k-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i1}} & \cdots & \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i,k-1}} \end{pmatrix} = -(k-1) K(\mathbf{x}_i, \mathbf{x}_i) \begin{pmatrix} c_{i1} & 0 & \cdots & 0 \\ 0 & c_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{i,k-1} \end{pmatrix}.$$

From (B.1), (B.2), and $(y_{i1} - \mu_{i1}(\mathbf{f}^{[-i]}), \ldots, y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}^{[-i]})) \approx (y_{i1} - \mu_{i1}(\mathbf{f}), \ldots, y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}))$, we have $g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i) \approx (k-1) K(\mathbf{x}_i, \mathbf{x}_i) \sum_{j=1}^{k-1} L_{ij} [f_j(\mathbf{x}_i) + 1/(k-1)]_* c_{ij} (y_{ij} - \mu_{ij}(\mathbf{f}))$. Noting that $L_{ik} = 0$ in this case, and that the approximations are defined analogously for other class examples, we have $g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i) \approx (k-1) K(\mathbf{x}_i, \mathbf{x}_i) \sum_{j=1}^{k} L_{ij} \times [f_j(\mathbf{x}_i) + 1/(k-1)]_* c_{ij} (y_{ij} - \mu_{ij}(\mathbf{f}))$.

[Received October 2002. Revised September 2003.]

REFERENCES

Ackerman, S. A., Strabala, K. I., Menzel, W. P., Frey, R. A., Moeller, C. C., and Gumley, L. E. (1998), "Discriminating Clear Sky From Clouds With MODIS," *Journal of Geophysical Research*, 103(32),141–157.

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000), "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers," *Journal of Machine Learning Research*, 1, 113–141.

Boser, B., Guyon, I., and Vapnik, V. (1992), "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Vol. 5, pp. 144–152.

Bredensteiner, E. J., and Bennett, K. P. (1999), "Multicategory Classification by Support Vector Machines," *Computational Optimizations and Applications*, 12, 35–46.

Burges, C. (1998), "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 2, 121–167.

Crammer, K., and Singer, Y. (2000), "On the Learnability and Design of Output Codes for Multi-Class Problems," in *Computational Learing Theory*, pp. 35–46.

Cristianini, N., and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines*, Cambridge, U.K.: Cambridge University Press.

Dietterich, T. G., and Bakiri, G. (1995), "Solving Multiclass Learning Problems via Error-Correcting Output Codes," *Journal of Artificial Intelligence Research*, 2, 263–286.

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.

Evgeniou, T., Pontil, M., and Poggio, T. (2000), "A Unified Framework for Regularization Networks and Support Vector Machines," *Advances in Computational Mathematics*, 13, 1–50.

Ferris, M. C., and Munson, T. S. (1999), "Interfaces to PATH 3.0: Design, Implementation and Usage," *Computational Optimization and Applications*, 12, 207–227.

Friedman, J. (1996), "Another Approach to Polychotomous Classification," technical report, Stanford University, Department of Statistics.

Guermeur, Y. (2000), "Combining Discriminant Models With New Multi-Class SVMs," Technical Report NC-TR-2000-086, NeuroCOLT2, LORIA Campus Scientifique.

Hastie, T., and Tibshirani, R. (1998), "Classification by Pairwise Coupling," in *Advances in Neural Information Processing Systems 10*, eds. M. I. Jordan, M. J. Kearns, and S. A. Solla, Cambridge, MA: MIT Press.

Jaakkola, T., and Haussler, D. (1999), "Probabilistic Kernel Regression Models," in *Proceedings of the Seventh International Workshop on AI and Statistics*, eds. D. Heckerman and J. Whittaker, San Francisco, CA: Morgan Kaufmann, pp. 94–102.

Joachims, T. (2000), "Estimating the Generalization Performance of a SVM Efficiently," in *Proceedings of ICML-00, 17th International Conference on Machine Learning*, ed. P. Langley, Stanford, CA: Morgan Kaufmann, pp. 431–438.

Key, J., and Schweiger, A. (1998), "Tools for Atmospheric Radiative Transfer: Streamer and FluxNet," *Computers and Geosciences*, 24, 443–451.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Atonescu, 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 Tchebycheian Spline Functions," *Journal of Mathematics Analysis and Applications*, 33, 82–95.

Lee, Y. (2002), "Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data," unpublished doctoral thesis, University of Wisconsin-Madison, Department of Statistics.

Lee, Y., and Lee, C.-K. (2003), "Classification of Multiple Cancer Types by Multicategory Support Vector Machines Using Gene Expression Data," *Bioinformatics*, 19, 1132–1139.

Lee, Y., Wahba, G., and Ackerman, S. (2004), "Classification of Satellite Radiance Data by Multicategory Support Vector Machines," *Journal of Atmospheric and Oceanic Technology*, 21, 159–169.

Lin, Y. (2001), "A Note on Margin-Based Loss Functions in Classification," Technical Report 1044, University of Wisconsin–Madison, Dept. of Statistics.

——— (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.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J., and Poggio, T. (1999), "Support Vector Machine Classification of Microarray Data," Technical Report AI Memo 1677, MIT.

Passerini, A., Pontil, M., and Frasconi, P. (2002), "From Margins to Probabilities in Multiclass Learning Problems," in *Proceedings of the 15th European Conference on Artificial Intelligence*, ed. F. van Harmelen, Amsterdam, The Netherlands: IOS Press, pp. 400–404.

Price, D., Knerr, S., Personnaz, L., and Dreyfus, G. (1995), "Pairwise Neural Network Classifiers With Probabilistic Outputs," in *Advances in Neural Information Processing Systems 7 (NIPS-94)*, eds. G. Tesauro, D. Touretzky, and T. Leen, Cambridge, MA: MIT Press, pp. 1109–1116.

Schölkopf, B., and Smola, A. (2002), *Learning With Kernels—Support Vector Machines, Regularization, Optimization and Beyond*, Cambridge, MA: MIT Press.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, New York: Springer-Verlag.

——— (1998), *Statistical Learning Theory*, New York: Wiley.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: SIAM.

——— (1998), "Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV," in *Advances in Kernel Methods: Support Vector Learning*, eds. B. Schölkopf, C. J. C. Burges, and A. J. Smola, Cambridge, MA: MIT Press, pp. 69–87.

——— (2002), "Soft and Hard Classification by Reproducing Kernel Hilbert Space Methods," *Proceedings of the National Academy of Sciences*, 99, 16524–16530.

Wahba, G., Lin, Y., Lee, Y., and Zhang, H. (2002), "Optimal Properties and Adaptive Tuning of Standard and Nonstandard Support Vector Machines," in *Nonlinear Estimation and Classification*, eds. D. Denison, M. Hansen, C. Holmes, B. Mallick, and B. Yu, New York: Springer-Verlag, pp. 125–143.

Wahba, G., Lin, Y., and Zhang, H. (2000), "GACV for Support Vector Machines, or, Another Way to Look at Margin-Like Quantities," in *Advances in Large Margin Classifiers*, eds. A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schurmans, Cambridge, MA: MIT Press, pp. 297–309.

Weston, J., and Watkins, C. (1999), "Support Vector Machines for Multiclass Pattern Recognition," in *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, ed. M. Verleysen, Brussels, Belgium: D-Facto Public, pp. 219–224.

Yeo, G., and Poggio, T. (2001), "Multiclass Classification of SRBCTs," Technical Report AI Memo 2001-018, CBCL Memo 206, MIT.

Zadrozny, B., and Elkan, C. (2002), "Transforming Classifier Scores Into Accurate Multiclass Probability Estimates," in *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 694–699.

Zhang, T. (2001), "Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization," Technical Report rc22155, IBM Research, Yorktown Heights, NY.