Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data [1]

by

Yoonkyung Lee, Yi Lin, and Grace Wahba

TECHNICAL REPORT NO. 714

August 18, 2003

Department of Statistics
The Ohio State University
404 Cockins Hall
1958 Neil Avenue
Columbus, OH 43210

[1] This is the second revision of Technical Report 1064, Department of Statistics, University of Wisconsin, September 2002. The first revision, TR 1064r, May 2003, contains some new results concerning the relations between the Bayes rule and other proposed SVM methods for the multicategory case. Some numerical studies have been added to Section 5 for comparison between our proposed method and other methods in this revision.

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

Yoonkyung Lee, Yi Lin, & Grace Wahba[†]

August 18, 2003

Abstract

Two category Support Vector Machines (SVM) have been very popular in the machine learning community for the classification problem. Solving multicategory problems by a series of binary classifiers is quite common in the SVM paradigm. However, this approach may fail under a variety of circumstances. We have proposed 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 leaving-out-one cross validation function, called Generalized Approximate Cross Validation (GACV) 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: nonparametric classification method, reproducing kernel Hilbert space, regularization method, generalized approximate cross validation, quadratic programming

[†]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, Department of Statistics, University of Wisconsin, Madison, WI 53706 (E-mail: yilin@stat.wisc.edu). Grace Wahba is Bascom and I. J. Schoenberg Professor, Department of Statistics, University of Wisconsin, Madison, WI 53706 (E-mail: wahba@stat.wisc.edu). Lee's research was supported in part by NSF Grant DMS 0072292 and NASA Grant NAG5 10273. Lin's research was supported in part by NSF Grant DMS 0134987. Wahba's research was supported in part by NIH Grant EY09946, NSF Grant DMS 0072292 and NASA Grant NAG5 10273.

1 INTRODUCTION

The Support Vector Machine (SVM) has seen the explosion of its popularity in the machine learning literature, and more recently, increasing attention from the statistics community as well. For a comprehensive list of its references, see the web site `http://www.kernel-machines.org`. This paper concerns Support Vector Machines for classification problems especially when there are more than two classes. The SVM paradigm, originally designed for the binary classification problem, has a nice geometrical interpretation of discriminating one class from the other by a hyperplane with the maximum margin. For an overview, see Vapnik (1998). It is commonly known that the SVM paradigm can comfortably sit in the regularization framework where we have a data fit component ensuring the model fidelity to data, and a penalty component enforcing the model simplicity. Wahba (1998) and Evgeniou, Pontil and Poggio (2000) have more details in this regard. 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.

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

Let us turn our attention to the multicategory classification problem. We assume the class label $Y \in \{1, \cdots, k\}$ without loss of generality, where k is the number of classes. Define $p_j(\mathbf{x}) = P(Y = j|X = \mathbf{x})$. In this case, the Bayes decision rule assigns a new $\mathbf{x}$ to the class with the largest $p_j(\mathbf{x})$. There are two strategies in tackling the multicategory problem, in general. One is to solve the multicategory problem by solving a series of binary problems, and the other is to consider all the classes at once. Refer to Dietterich and Bakiri (1995) for a general scheme to utilize 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 popular among the first approaches. The pairwise approach has the disadvantage of potential variance increase since smaller observations are used to learn each classifier. Also, it allows only a simple cost structure when different misclassification costs are concerned. See Friedman (1996) for details. For SVMs, 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 to assign a new $\mathbf{x}$ the class giving the largest $f_j(\mathbf{x})$ for $j = 1, \cdots, k$, 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 SVMs 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 $sign(p_j(\mathbf{x}) - 1/2)$. 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 $\ell = 1, \cdots, k$ since $f_j(\mathbf{x})$ would be near 1, and all the other $f_\ell(\mathbf{x})$ would be close to -1, making a big contrast. However, if there is no dominating class, then all $f_j(\mathbf{x})$'s would be close to -1, leaving the class prediction based on them very obscure. Apparently, it is different from the Bayes decision rule. Thus, there is a demand for a true extension of SVMs to the multicategory case, which would inherit the optimal property of the binary case, and treat the problem in a simultaneous fashion. In fact, there have been alternative multiclass formulations of the SVM considering all the classes at once, such as Vapnik (1998), Weston and Watkins (1999), and 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 multicategory SVM which 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 it is carried out in a unified way, thereby encompassing the version of the binary SVM modification for unequal costs in Lin, Lee and Wahba (2002).

We briefly state the Bayes decision rule in Section 2 for either equal or unequal misclassification costs. The binary Support Vector Machine is reviewed in Section 3. Section 4 is the main part of this paper where we present a formulation of the multicategory SVM. The dual problem for the proposed method is derived, as well as a data adaptive tuning method, analogous to the binary case. A numerical study comprises Section 5 for illustration. Then, cancer diagnosis using gene expression profiles and cloud classification using satellite radiance profiles are presented in Section 6 as its applications. Concluding remarks and future directions are given at the end.

2 CLASSIFICATION PROBLEM AND THE BAYES RULE

We state the theoretically best classification rules derived under a decision theoretic formulation of classification problems in this section. Their derivations are fairly straightforward, and can be found in any general references to classification problems. In the classification problem, we are given a training data set that consists of n observations $(\mathbf{x}_i, y_i)$ for $i = 1, \cdots, n$. $\mathbf{x}_i \in R^d$ represents covariates and $y_i \in \{1, \cdots, k\}$ denotes a class label. The task is to learn a classification rule $\phi(\mathbf{x}) : R^d \to \{1, \cdots, k\}$ that well 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 $P(\mathbf{x}, y)$. Let (X, Y) denote a generic pair of a random realization from $P(\mathbf{x}, y)$, and $p_j(\mathbf{x}) = P(Y = j|X = \mathbf{x})$ for $j = 1, \cdots, k$. If the misclassification costs are all equal, the loss by the classification rule ϕ at $(\mathbf{x}, y)$ is defined as

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

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

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

When the misclassification costs are not equal, which may be common in solving real world problems, we change the loss (1) to reflect the cost structure. First, define $C_{j\ell}$ for $j, \ell = 1, \cdots, k$ as the cost of misclassifying an example from class j to class ℓ. C_{jj} for $j = 1, \cdots, k$ are all zero. The loss function for the unequal costs is then

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

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

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

Besides the concern with different misclassification costs, sampling bias that leads to distortion of the class proportions needs 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 we collect 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 π_j^s be the prespecified proportion of class j examples in a training data set. π_j^s may be different from π_j if sampling bias has occurred. Let (X^s, Y^s) be a random pair obtained by the sampling mechanism used in the data collection stage, and $p_j^s(\mathbf{x}) = P(Y^s = j | X^s = \mathbf{x})$. Then (4) can be rewritten in terms of the quantities for (X^s, Y^s) and π_j's which we assume are known a priori.

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

where $l_{\ell j}$ is defined as $(\pi_\ell/\pi_\ell^s)C_{\ell j}$, which is a modified cost that takes the sampling bias into account together with the original misclassification cost. Following the usage in Lin et al. (2002), we call the case when misclassification costs are not equal or there is a sampling bias, nonstandard, as opposed to the standard case when there are equal misclassification costs without sampling bias.

3 SUPPORT VECTOR MACHINES

We briefly go over the standard Support Vector Machines for the binary case. SVMs have their roots in a geometrical interpretation of the classification problem as a problem of finding a separating hyperplane in a multidimensional input space. For reference, see Boser, Guyon and Vapnik (1992), Vapnik (1998), Burges (1998), Cristianini and Shawe-Taylor (2000), Schölkopf and Smola (2002) and references therein. The class labels y_i are either 1 or -1 in the SVM setting. Generalizing SVM classifiers from hyperplanes to nonlinear ones, 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

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

where $(x)_+ = \max(x, 0)$, and $\|h\|_{H_K}^2$ 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 $\|h\|_{H_K}^2 = \|\mathbf{w}\|^2$, then (6) reduces to the linear SVM. For more information on RKHS, see Wahba (1990). λ is a tuning parameter. The classification rule $\phi(\mathbf{x})$ induced by $f(\mathbf{x})$ is $\phi(\mathbf{x}) = sign(f(\mathbf{x}))$. 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 2 times the misclassification loss.

There are two kinds of theoretical explanations available for the observed good behavior of SVMs. The first and original explanation is represented by theoretical justification of the SVM in Vapnik's structural risk minimization approach, see Vapnik (1998). The arguments there are based on upper bounds of the generalization error in terms of the Vapnik-Chervonenkis dimension. The second kind of explanation was provided by Lin (2002). He 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 $\phi_B(\mathbf{x}) = sign(p_1(\mathbf{x}) - 1/2)$. It was shown that, if the reproducing kernel Hilbert space is rich enough, the decision rule implemented by $sign(f(\mathbf{x}))$ 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 SVMs, the RKHS induced by which is flexible enough to approximate $sign(p_1(\mathbf{x}) - 1/2)$. 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, for example Wahba (2002) where penalized likelihood estimates of probabilities, which could be used to generate a classifier, are discussed in parallel with SVMs. See also Lin (2001) and Zhang (2001) that contain a general treatment of various convex loss functions in relation to the Bayes decision rule. However the efficiency of the SVMs in going straight for the classification rule is valuable in a broad class of practical applications, including the ones to be discussed in this paper. It is worth noting that due to the efficient mechanism that the SVM estimates the most likely class code, not the posterior probability for classification, recovering a real probability from the SVM function would be inevitably limited. A referee said that "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 can be found in Lin (2002) or Wahba (2002).

4 MULTICATEGORY SUPPORT VECTOR MACHINES

In the subsequent sections, we present the extension of the Support Vector Machines 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 will note its relationship to some other multicategory SVMs that have been proposed. Then, straightforward modification follows for the nonstandard case. In the end, we derive its dual formulation via which we obtain the solution, and address how to tune the model controlling parameter(s) involved in the multicategory SVM.

4.1 Standard Case

Assuming that all the misclassification costs are equal and there is no sampling bias in the training data set, 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 $j = 1, \cdots, k$ 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, $\mathbf{y}_i = \mathbf{v}_1 = (1, -1/(k-1), \cdots, -1/(k-1))$. Similarly, if it falls into class k, $\mathbf{y}_i = \mathbf{v}_k = (-1/(k-1), \cdots, -1/(k-1), 1)$. Accordingly, we define a k-tuple of separating functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ with the sum-to-zero constraint, $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ for any $\mathbf{x} \in R^d$. The k functions are constrained by the sum-to-zero condition, $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ in this particular setting, for the same reason as $p_j(\mathbf{x})$'s, the conditional probabilities of k classes are constrained by the sum-to-one 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 $\{1, \cdots, k\}$. The utility of the sum-to-zero constraint will be justified 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 $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})) \in \prod_{j=1}^k(\{1\} + H_{K_j})$, the product space of k reproducing kernel Hilbert spaces H_{K_j} for $j = 1, \cdots, k$. 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 $j = 1, \cdots, k$, we will assume they are the same RKHS denoted by H_K. Define Q as the k by k matrix with 0 on the diagonal, and 1 elsewhere. It represents the cost matrix when all the misclassification costs are equal. Let L be a function which maps a class label $\mathbf{y}_i$ to the jth row of the matrix Q if $\mathbf{y}_i$ indicates class j. So, if $\mathbf{y}_i$ represents class j, then $L(\mathbf{y}_i)$ is a k dimensional vector with 0 in the jth coordinate, and 1 elsewhere. Now, we propose that to find $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})) \in \prod_1^k(\{1\} + H_K)$, with the sum-to-zero constraint, minimizing the following quantity is a natural extension of SVMs methodology:

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

where $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ means $[(f_1(\mathbf{x}_i) - y_{i1})_+, \cdots, (f_k(\mathbf{x}_i) - y_{ik})_+]$ 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 analogous relation to the misclassification loss (1). If $\mathbf{f}(\mathbf{x}_i)$ itself is one of the class codes, $L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ is $k/(k-1)$ times the misclassification loss. When $k = 2$, the generalized hinge loss function reduces to the binary hinge loss. Check that if $\mathbf{y}_i = (1, -1)$ (1 in the binary SVM notation), then $L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ = (0, 1) \cdot [(f_1(\mathbf{x}_i) - 1)_+, (f_2(\mathbf{x}_i) + 1)_+] = (f_2(\mathbf{x}_i) + 1)_+ = (1 - f_1(\mathbf{x}_i))_+$. Likewise, if $\mathbf{y}_i = (-1, 1)$ (-1 in the binary SVM notation), $L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ = (f_1(\mathbf{x}_i) + 1)_+$. Thereby, the data fit functionals in (6) and (7) are identical, f_1 playing the same role as f in (6). Also, note that $(\lambda/2)\sum_{j=1}^2 \|h_j\|_{H_K}^2 = (\lambda/2)(\|h_1\|_{H_K}^2 + \|-h_1\|_{H_K}^2) = \lambda\|h_1\|_{H_K}^2$, by the fact that $h_1(\mathbf{x}) + h_2(\mathbf{x}) = 0$ for any $\mathbf{x}$, to be discussed later. So, the penalties to the model complexity in (6) and (7) are identical. These 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 $E[L(Y) \cdot (\mathbf{f}(X) - Y)_+]$.

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

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

Proof of this lemma and other proofs are 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 were given by Vapnik (1998), Weston and Watkins (1999), and Bredensteiner and Bennett (1999). Guermeur (2000) showed that they are essentially equivalent and amount to using the following loss function with the same regularization terms as in (7):

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

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

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

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 insure 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, let's consider different misclassification costs only, assuming no sampling bias. Instead of the equal cost matrix Q used in the definition of $L(\mathbf{y}_i)$, define a k by k cost matrix C with entry $C_{j\ell}$, the cost of misclassifying an example from class j to class ℓ. Modify $L(\mathbf{y}_i)$ in (7) to the jth row of the cost matrix C if $\mathbf{y}_i$ indicates class j. When all the misclassification costs $C_{j\ell}$ are equal to 1, the cost matrix C becomes Q. So, the modified map $L(\cdot)$ 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 (X, Y) to (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 L whose entry $l_{j\ell}$ is given by $(\pi_j/\pi_j^s)C_{j\ell}$ for $j, \ell = 1, \cdots, k$. Accordingly, define $L(\mathbf{y}_i)$ to be the jth row of the matrix L if $\mathbf{y}_i$ indicates class j. When there is no sampling bias, in other words, $\pi_j = \pi_j^s$ for all j, the generalized cost matrix L reduces to the ordinary cost matrix C. With the finalized version of the cost matrix L and the map $L(\mathbf{y}_i)$, the multicategory SVM formulation (7) still holds as the general scheme. The following lemma identifies the minimizer of the limit of the data fit functional, which is $E[L(Y^s) \cdot (\mathbf{f}(X^s) - Y^s)_+]$.

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

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

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

4.3 The Representer Theorem and Dual Formulation

We explain how to carry out the computation to find the minimizer of (7). The problem of finding constrained functions $(f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ 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) and Wahba (1998). Theorem 1 says that we can still apply the representer theorem to each component $f_j(\mathbf{x})$ with, however some restrictions on the coefficients due to the sum-to-zero constraint.

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

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

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

Switching to a Lagrangian formulation of the problem (7), we introduce a vector of nonnegative slack variables $\xi_i \in R^k$ 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 $L_j \in R^n$ for $j = 1, \cdots, k$ be the jth column of the n by k matrix with the ith row $L(\mathbf{y}_i) \equiv (L_{i1}, \cdots, L_{ik})$. Let $\xi_{\cdot j} \in R^n$ for $j = 1, \cdots, k$ be the jth column of the n by k matrix with the ith row ξ_i. Similarly, $\mathbf{y}_{\cdot j}$ denotes the jth column of the n by k matrix with the ith row $\mathbf{y}_i$. With some abuse of notation, let K be now the n by n matrix with ijth entry $K(\mathbf{x}_i, \mathbf{x}_j)$. Then, the primal problem in vector notation is

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

$$\begin{aligned}
\text{subject to} \qquad & b_j \mathbf{e} + K\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} \le \xi_{\cdot j} && \text{for } j = 1, \cdots, k && (14)\\
& \xi_{\cdot j} \ge 0 && \text{for } j = 1, \cdots, k && (15)\\
& (\textstyle\sum_{j=1}^{k} b_j)\mathbf{e} + K(\textstyle\sum_{j=1}^{k} \mathbf{c}_{\cdot j}) = 0 && && (16)
\end{aligned}$$

It is a quadratic optimization problem with some equality and inequality constraints. We derive its Wolfe dual problem, by introducing nonnegative Lagrange multipliers $\alpha_{\cdot j} = (\alpha_{1j}, \cdots, \alpha_{nj})^t \in R^n$ for (14), nonnegative Lagrange multipliers $\gamma_j \in R^n$ for (15), and unconstrained Lagrange multipliers $\delta_f \in R^n$ for (16), the equality constraints. Then, the dual problem becomes a problem of maximizing

$$\begin{aligned}
L_D \;=\; & \sum_{j=1}^{k} L_j^t \xi_{\cdot j} + \frac{1}{2} n\lambda \sum_{j=1}^{k} \mathbf{c}_{\cdot j}^t K \mathbf{c}_{\cdot j} + \sum_{j=1}^{k} \alpha_{\cdot j}^t (b_j \mathbf{e} + K\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} - \xi_{\cdot j}) \\
& - \sum_{j=1}^{k} \gamma_j^t \xi_{\cdot j} + \delta_f^t \left((\sum_{j=1}^{k} b_j)\mathbf{e} + K(\sum_{j=1}^{k} \mathbf{c}_{\cdot j}) \right)
\end{aligned} \tag{17}$$

$$\begin{aligned}
\text{subject to for } j = 1, \cdots, k \qquad \frac{\partial L_D}{\partial \xi_{\cdot j}} &= L_j - \alpha_{\cdot j} - \gamma_j = 0 && (18)\\
\frac{\partial L_D}{\partial \mathbf{c}_{\cdot j}} &= n\lambda K \mathbf{c}_{\cdot j} + K\alpha_{\cdot j} + K\delta_f = 0 && (19)\\
\frac{\partial L_D}{\partial b_j} &= (\alpha_{\cdot j} + \delta_f)^t \mathbf{e} = 0 && (20)\\
& \alpha_{\cdot j} \ge 0 && (21)\\
& \gamma_j \ge 0 && (22)
\end{aligned}$$

Let $\bar{\alpha}$ be $(\sum_{j=1}^{k} \alpha_{\cdot j})/k$. Since δ_f is unconstrained, one may take $\delta_f = -\bar{\alpha}$ from (20). Accordingly, (20) becomes $(\alpha_{\cdot j} - \bar{\alpha})^t \mathbf{e} = 0$. Eliminating all the primal variables in L_D by the equality constraint

(18) and using relations from (19) and (20), we have the following dual problem.

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

$$\text{subject to} \quad 0 \le \alpha_{\cdot j} \le L_j \quad \text{for } j = 1, \cdots, k \tag{24}$$

$$(\alpha_{\cdot j} - \bar{\alpha})^t \mathbf{e} = 0 \quad \text{for } j = 1, \cdots, k \tag{25}$$

Once the quadratic programming problem is solved, the coefficients can be determined by the relation $\mathbf{c}_{\cdot j} = -(\alpha_{\cdot j} - \bar{\alpha})/(n\lambda)$ from (19). Note that if the matrix 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

$$\alpha_{\cdot j} \perp (b_j \mathbf{e} + K\mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} - \xi_{\cdot j}) \quad \text{for } j = 1, \cdots, k \tag{26}$$

$$\gamma_j = (L_j - \alpha_{\cdot j}) \perp \xi_{\cdot j} \quad \text{for } j = 1, \cdots, k \tag{27}$$

where $\perp$ means that componentwise products are all zero. If $0 < \alpha_{ij} < L_{ij}$ for some i, then ξ_{ij} should be zero 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 $(\alpha_{i1}, \cdots, \alpha_{ik}) = 0$ for the ith example, then $(c_{i1}, \cdots, c_{ik}) = 0$. 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 $\mathbf{c}_i = (c_{i1}, \cdots, c_{ik}) \neq 0$. Hence, depending on the number of support vectors, the multicategory SVM solution may have a sparse representation, which is also one of the main characteristics of the binary SVM. In practice, solving the quadratic programming (QP) problem can be done via available optimization packages for moderate size problems. All the examples presented in this paper 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. There have been various tuning methods proposed for the binary Support Vector Machines, to list a few, Vapnik (1995), Jaakkola and Haussler (1999), Joachims (2000), Wahba, Lin and Zhang (2000), and Wahba, Lin, Lee and Zhang (2002). We derive an approximate leaving-out-one cross validation function, called Generalized Approximate Cross Validation (GACV) for the multicategory SVM. It is based on the leaving-out-one arguments, reminiscent of GACV derivations for penalized likelihood methods.

For concise notations, let $J_\lambda(\mathbf{f}) = (\lambda/2)\sum_{j=1}^k \|h_j\|^2_{H_K}$, and $\mathbf{y} = (\mathbf{y}_1, \cdots, \mathbf{y}_n)$. We denote the objective function of the multicategory SVM (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 $g(\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i)) \equiv L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$. Let $\mathbf{f}_\lambda$ be the minimizer of $I_\lambda(\mathbf{f}, \mathbf{y})$. It would be ideal but only theoretically possible to choose tuning parameters minimizing the generalized comparative Kullback-Leibler (GCKL) distance with respect to the loss function, $g(\mathbf{y}, \mathbf{f}(\mathbf{x}))$ averaged over a data set with the same covariates $\mathbf{x}_i$ and unobserved Y_i, $i = 1, \cdots, n$:

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

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 the usage of GCKL as an ideal tuning measure, and our strategy is to develop a data-dependent computable proxy of GCKL and choose tuning parameters minimizing the proxy of GCKL.

The leaving-out-one cross validation arguments are used 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 $\mathbf{f}_{\lambda i}^{[-i]}$. $f_{\lambda j}(\mathbf{x}_i)$ and $f_{\lambda j}^{[-i]}(\mathbf{x}_i)$ denote the jth component of $\mathbf{f}_\lambda(\mathbf{x}_i)$, and $\mathbf{f}_\lambda^{[-i]}(\mathbf{x}_i)$, respectively. Now, we define the leaving-out-one cross validation function which 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 $D(\lambda) = (1/n)\sum_{i=1}^{n}\left(g(\mathbf{y}_i, \mathbf{f}_{\lambda i}^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_{\lambda i})\right)$. For an approximation of $V_0(\lambda)$ without actually doing the leaving-out-one procedure, which may be prohibitive for large data sets, we will approximate $D(\lambda)$ further using the leaving-out-one lemma. As a necessary ingredient for the lemma, we extend the domain of the function $L(\cdot)$ from a set of k distinct class codes to allow argument $\mathbf{y}$ not necessarily a class code. For any $\mathbf{y} \in \mathbf{R}^k$ satisfying the sum-to-zero constraint, we define $L : R^k \to R^k$ as $L(\mathbf{y}) = (w_1(\mathbf{y})[-y_1 - 1/(k-1)]_*, \cdots, w_k(\mathbf{y})[-y_k - 1/(k-1)]_*)$ where $[\tau]_* = I(\tau \geq 0)$, and $(w_1(\mathbf{y}), \cdots, w_k(\mathbf{y}))$ is the jth row of the extended misclassification cost matrix L with the jl entry $(\pi_j/\pi_j^s)C_{jl}$ if $arg\max_{l=1,\cdots,k} y_l = j$. If there are ties, then $(w_1(\mathbf{y}), \cdots, w_k(\mathbf{y}))$ is defined as the average of the rows of the cost matrix L corresponding to the maximal arguments. We easily check that $L(0, \cdots, 0) = (0, \cdots, 0)$ and the extended $L(\cdot)$ coincides with the original $L(\cdot)$ over the domain of class codes. We define a class prediction $\mu(\mathbf{f})$ 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 $\dfrac{\sum_{j=1}^{k} I(f_j < -\frac{1}{k-1})}{k - \sum_{j=1}^{k} I(f_j < -\frac{1}{k-1})}\left(\dfrac{1}{k-1}\right)$ to satisfy the sum-to-zero constraint. If $\mathbf{f}$ has a maximum component greater than 1, and all the others less than $-1/(k-1)$, then $\mu(\mathbf{f})$ is a k-tuple with 1 on the maximum coordinate and $-1/(k-1)$ elsewhere. So, the function μ maps $\mathbf{f}$ to its most likely class code if there is a class strongly predicted by $\mathbf{f}$. By contrast, if none of the coordinates of $\mathbf{f}$ is less than $-1/(k-1)$, μ maps $\mathbf{f}$ to $(0, \cdots, 0)$. With this definition of μ, the following can be shown.

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

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 above lemma. Details of the approximation are in Appendix B. Let $(\mu_{i1}(\mathbf{f}), \cdots, \mu_{ik}(\mathbf{f})) = \mu(\mathbf{f}(\mathbf{x}_i))$. From the approximation

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

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

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

From a numerical point of view, the proposed GACV may be vulnerable to small perturbations in the solution since 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 gets robust against slight perturbations of the solutions up to a certain precision.

5 NUMERICAL STUDY

In this section, we illustrate the Multicategory Support Vector Machine (MSVM) through numerical examples. Various tuning criteria, some of which are available only in simulation settings, are considered and the performance of GACV is compared with those theoretical criteria. Throughout this section, the Gaussian kernel function, $K(s,t) = \exp\left(-\frac{1}{2\sigma^2}\|s-t\|^2\right)$ was used, and λ and σ were searched over a grid.

We considered a simple three-class example on the unit interval $[0,1]$ with $p_1(x) = 0.97\exp(-3x)$, $p_3(x) = \exp(-2.5(x-1.2)^2)$, and $p_2(x) = 1 - p_1(x) - p_3(x)$. Class 1 is most likely for small x while class 3 is most likely for large x. The in-between interval would be a competing zone for three classes although class 2 is slightly dominant. The three panels in Figure 1 depict the ideal target function $f_1(x)$, $f_2(x)$ and $f_3(x)$ defined in Lemma 1 for this example. $f_j(x)$ assumes the value 1 when $p_j(x)$ is larger than $p_l(x)$, $l \neq j$, and -1/2 otherwise. On the other hand, the ordinary one-vs-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 one-vs-rest scheme do not carry enough information to identify the most probable class at x. In this example, chosen to illustrate how a one-vs-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 hard because the maximum of $p_2(x)$ is barely 0.5 across the interval. To compare the multicategory SVM and the one-versus-rest scheme, we applied both methods to a data set with sample size $n = 200$. The attribute x_i's were generated from the uniform distribution on $[0,1]$, and given x_i, the corresponding class label y_i was randomly assigned according to the conditional probabilities $p_j(x)$. The tuning parameters λ, and σ were jointly tuned to minimize the GCKL distance of the estimate $\mathbf{f}_{\lambda,\sigma}$ from the true distribution.

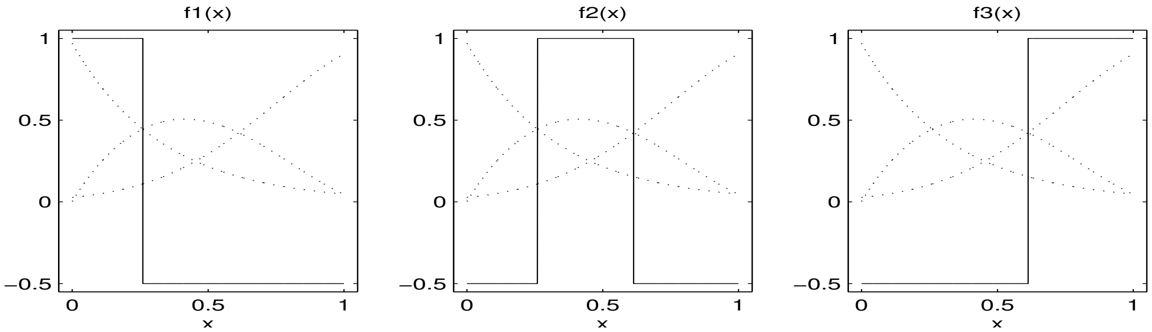

Figure 1: Multicategory SVM target functions for three-class example. The dotted lines are the conditional probabilities of three classes.

Figure 2 shows the estimated functions for both the MSVM and the one-versus-rest methods when both are 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). Whereas, the multicategory SVM 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 multicategory SVM. Table 1 shows the tuning parameters chosen by other tuning criteria alongside GCKL and their inefficiencies for this example. When we treat all the misclassifications equally, the true target GCKL is given by $E_{true}(1/n)\sum_{i=1}^{n} L(\mathbf{Y}_i)\cdot(\mathbf{f}(\mathbf{x}_i)-\mathbf{Y}_i)_+ = (1/n)\sum_{i=1}^{n}\sum_{j=1}^{k}(f_j(\mathbf{x}_i)+1/(k-1))_+(1-p_j(\mathbf{x}_i))$. More directly, the misclassification rate (MISRATE) is available in simulation settings, which is defined as $E_{true}(1/n)\sum_{i=1}^{n} I(\mathbf{Y}_i \neq arg\max_{j=1,\cdots,k} f_{ij}) = (1/n)\sum_{i=1}^{n}\sum_{j=1}^{k} I(f_{ij}=\max_{l=1,\cdots,k} f_{il})(1-p_j(\mathbf{x}_i))$. In addition, to see how good one can 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 bigger λ than that of GCKL. However, we observe that TUNE, the other data adaptive criterion if a tuning set is available, gave a similar outcome. The inefficiency of GACV is 1.048, yielding the misclassification rate 0.4171, slightly bigger than the optimal rate 0.3980. Expectedly, it is a little worse than having an extra tuning set, but almost as good as 10-fold CV which requires about ten times more computations than GACV. 10-fold CV has two minimizers, and they suggest the compromising role between λ and σ for the Gaussian kernel function.

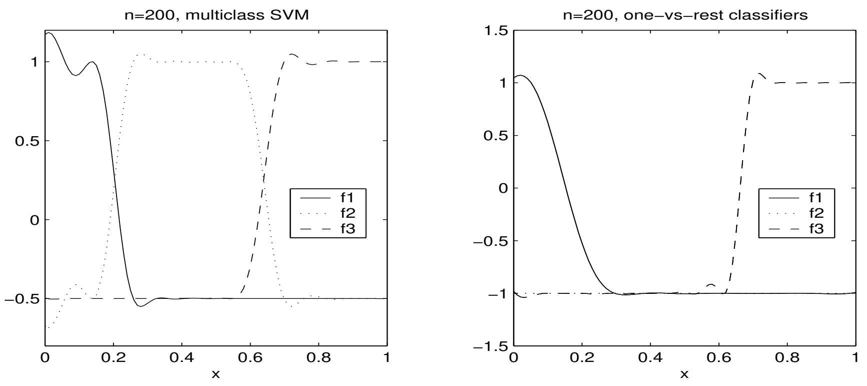

Figure 2: Comparison between the multicategory SVM and one-versus-rest method. The Gaussian kernel function was used, and the tuning parameters λ, and σ were simultaneously chosen via GCKL.

To demonstrate that the estimated functions indeed affect the test error rate, we generated 100 replicate data sets of sample size 200, and applied the multicategory SVM and one-versus-rest SVM classifiers to each data set, combined with GCKL tuning. Based on the estimated classification rules, we evaluated the test error rates for both methods over a test data set of size 10000. For the test data set, the Bayes misclassification rate was 0.3841 while the average test error rate of the multicategory SVM over 100 replicates was 0.3951 with the standard deviation 0.0099 and that of the one-versus-rest classifiers was 0.4307 with its standard deviation 0.0132. The multicategory

SVM gave a smaller test error rate than the one-versus-rest scheme across all the 100 replicates.

Table 1: Tuning criteria and their inefficiencies

Criterion	($\log_2 \lambda$, $\log_2 \sigma$)	Inefficiency
MISRATE	(-11,-4)	*
GCKL	(-9,-4)	0.4001/0.3980=1.0051
TUNE	(-5,-3)	0.4038/0.3980=1.0145
GACV	(-4,-3)	0.4171/0.3980=1.0480
10-fold CV	(-10,-1)	0.4112/0.3980=1.0331
	(-13,0)	0.4129/0.3980=1.0374

Other simulation studies in various settings showed that MSVM outputs approximate coded classes when the tuning parameters are appropriately chosen, and oftentimes GACV and TUNE tend to oversmooth in comparison to 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) $p_1(x) = 0.7 - 0.6x^4$, $p_2(x) = 0.1 + 0.6x^4$, and $p_3(x) = 0.2$: 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) $p_1(x) = 0.45 - 0.4x^4$, $p_2(x) = 0.3 + 0.4x^4$, and $p_3(x) = 0.25$: in this case over a large subset of the domain, there is no dominant class, but there is one class that is clearly more likely than the other two. 3) $p_1(x) = 0.45 - 0.3x^4$, $p_2(x) = 0.35 + 0.3x^4$, and $p_3(x) = 0.2$: again there is no dominant class over a large subset of the domain, and there are two classes that are competitive. Figure 3 depicts the three situations. In each scenario, x_i's of size 200 were generated from the uniform distribution on $[-1, 1]$ and the tuning parameters were chosen by GCKL. Table 2 shows the misclassification rates of the MSVM and the other extension averaged over 10 replicates. Just for reference, the one-vs-rest classification error rates are also given in the table. The error rates were numerically approximated with the true conditional probabilities and the estimated classification rules evaluated on a fine grid. In the scenario 1, the three methods are almost indistinguishable due to the presence of a dominant class mostly over the region. When the lowest two classes compete without a dominant class in the scenario 2, the MSVM and the other extension perform similarly with clearly lower error rates than the one-vs-rest approach. On the other hand, when the highest two classes compete, the MSVM gives smaller error rates than the alternative extension as expected by Lemma 2. 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 0.0137 using the paired 10 replicates in this case.

A small scale empirical study was carried out over four data sets from the UCI data repository. The four data sets are `wine`, `waveform`, `vehicle` and `glass`. As a tuning method, we compared GACV with 10-fold CV, 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 CV. However, with considerable overlaps among classes, we empirically observed that GACV tends to oversmooth and result in a little bigger error rate than 10-fold CV. 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)

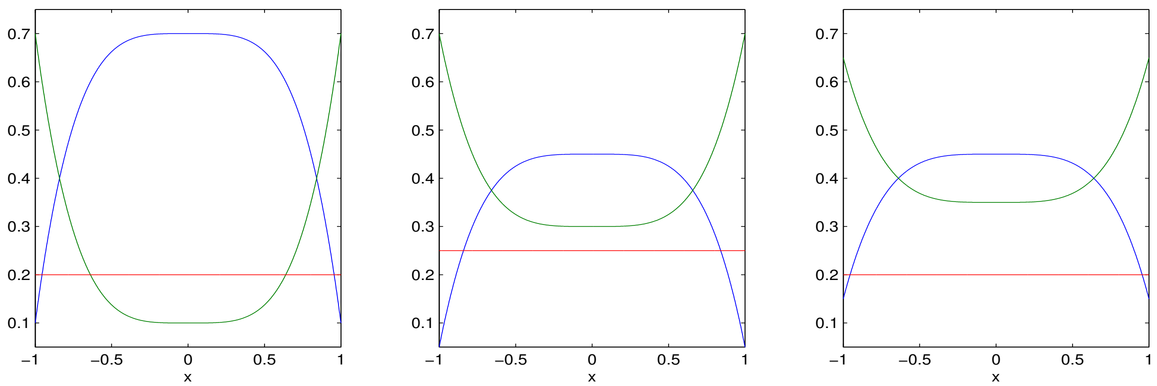

Figure 3: Underlying true conditional probabilities in three situations. Scenario 1: dominant class (left), Scenario 2: the lowest two classes compete (center), and Scenario 3: the highest two classes compete (right). Class 1: blue, Class 2: green and Class 3: red.

Table 2: Approximated classification error rates

Scenario	Bayes rule	MSVM	other extension	one-vs-rest
1	0.3763	0.3817	0.3784	0.3811
		(0.0021)	(0.0005)	(0.0017)
2	0.5408	0.5495	0.5547	0.6133
		(0.0040)	(0.0056)	(0.0072)
3	0.5387	0.5517	0.5708	0.5972
		(0.0045)	(0.0089)	(0.0071)

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

method, the one-vs-rest binary SVM (OVR), and the alternative multiclass extension (AltMSVM). For the one-vs-rest SVM, and the alternative extension, 10-fold CV was used for tuning. Table 3 is a summary of 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`, the error rates were evaluated over the test set of size 4700 and 346, respectively that was held out. MSVM performed the best over the `waveform`, and `vehicle` data sets. Over the `wine` data set, the performance of MSVM is about the same as that of QDA, OVR, and AltMSVM, slightly worse than LDA, and better than NN. Over the `glass` data, MSVM is better than LDA, but is 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 no single classification method is going to dominate all other methods. In practice, simple methods such as the linear discriminant analysis often outperform more sophisticated methods. The multicategory SVM is a general purpose classification method, and we think that it is a useful new addition to the toolbox of the data analyst.

Table 3: Classification error rates

Data set	MSVM	QDA	LDA	NN	OVR	AltMSVM
`wine`	0.0169	0.0169	0.0112	0.0506	0.0169	0.0169
`glass`	0.3645	NA	0.4065	0.2991	0.3458	0.3170
`waveform`	0.1564	0.1917	0.1757	0.2534	0.1753	0.1696
`vehicle`	0.0694	0.1185	0.1908	0.1214	0.0809	0.0925

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

6 APPLICATIONS

Two applications to problems arising in oncology and meteorology are presented. One application is cancer classification using microarray data and the other is cloud detection and classification via satellite radiance profiles. The results are outlined here. Complete details of the cancer classification application appear in Lee and Lee (2003) and details of the cloud detection and classification application appear in Lee, Wahba and Ackerman (2003). 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 to classify tumor types using gene expression data have been studied. Typical microarray training data sets (a set of pairs of a gene expression profile $\mathbf{x}_i$ and the tumor type y_i that it falls into) have a fairly small sample size, usually less than one hundred, while the number of genes involved is in the order of thousands. This poses an unprecedented challenge to some classification methodologies. The Support Vector Machine is one of the methods successfully applied to the cancer diagnosis problems in the previous studies. Since in principle, it can handle input variables much larger than the sample size via its dual formulation, it may be well suited to the microarray data structure.

We revisited the data set in Khan, Wei, Ringner, Saal, Ladanyi, Westermann, Berthold, Schwab, Atonescu, Peterson and Meltzer (2001). Khan et al. (2001) classified the small round blue cell tumors (SRBCTs) of childhood into 4 classes; neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS) using cDNA gene expression profiles. The data set is available from `http://www.nhgri.nih.gov/DIR/Microarray/Supplement/`. 2308 gene profiles out of 6567 genes are given in the data set after filtering for a minimal level of expression. The training set consists of 63 cases (NB: 12, RMS: 20, BL: 8, EWS: 23), and the test set has 20 SRBCT cases (NB: 6, RMS: 5, BL: 3, EWS: 6) and five non SRBCTs. Note that Burkitt lymphoma (BL) is a subset of NHL. Khan et al. (2001) successfully diagnosed the tumor types into four categories using Artificial Neural Networks. Also, Yeo and Poggio (2001) applied k Nearest Neighbor (kNN), weighted voting and linear SVM in one-vs-rest fashion to this four-class problem, and compared the performances of these methods when they are combined with several feature selection methods for each binary classification problem. It was reported that mostly SVM classifiers achieved the smallest test error and leaving-out-one cross validation (LOOCV) error when five to 100 genes (features) were used. For the best results shown in the paper, 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 in 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 select genes with the largest ratios. Table 4 is a summary of the classification results by MSVMs with the Gaussian kernel function. The proposed MSVMs were cross validated for the training set in leaving-out-one fashion, with zero error attained for 20, 60, and 100 genes, as shown in the second column. The last column shows the final test results. Using the top ranked 20, 60, and 100 genes, the MSVMs correctly classify 20 test examples. With all the genes included, one error occurs in LOOCV and the misclassified example is identified as EWS-T13, which was reported to occur frequently as an LOOCV error in Khan et al. (2001) and Yeo and Poggio (2001). The test error using all genes varies from zero to three depending on tuning measures. The MSVM tuned by GACV gives three test errors while LOOCV tuning gives zero to three test errors. The range of test errors is due to the fact that multiple pairs of (λ, σ) gave the same minimum in LOOCV tuning, and all were evaluated in the test phase, with varying results. Perfect classification in cross validation and testing with high dimensional inputs, suggests a possibility of a compact representation of the classifier in a low dimension. See Figure 3 in Lee and Lee (2003) for the principal component analysis of the top 100 genes in the training set. The three principal components contain total 66.5% variation of 100 genes in the training set. They contribute 27.52%, 23.12% and 15.89%, respectively and the fourth component not included in the analysis explains only 3.48% of variation of the training data. With the three principal components (PCs) only, we applied the MSVM. Again, perfect classification was achieved in cross validating and testing.

Table 4: LOOCV error and Test error for SRBCT data set. MSVMs with the Gaussian kernel were applied to the training data set. The last row shows the results by using only three principal components (PCs) from 100 genes.

Number of genes	LOOCV error	Test error
20	0	0
60	0	0
100	0	0
all	1	0 to 3
3 PCs (100)	0	0

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 from the plot that all the 20 test examples from four classes are classified correctly. Note that the test examples are rearranged in the order of EWS, BL, NB, RMS, and non SRBCT. In the test data set, there are five non SRBCT cases.

In medical diagnosis, attaching a confidence statement to each prediction may be useful in identifying such borderline cases. For classification methods with their ultimate output being the

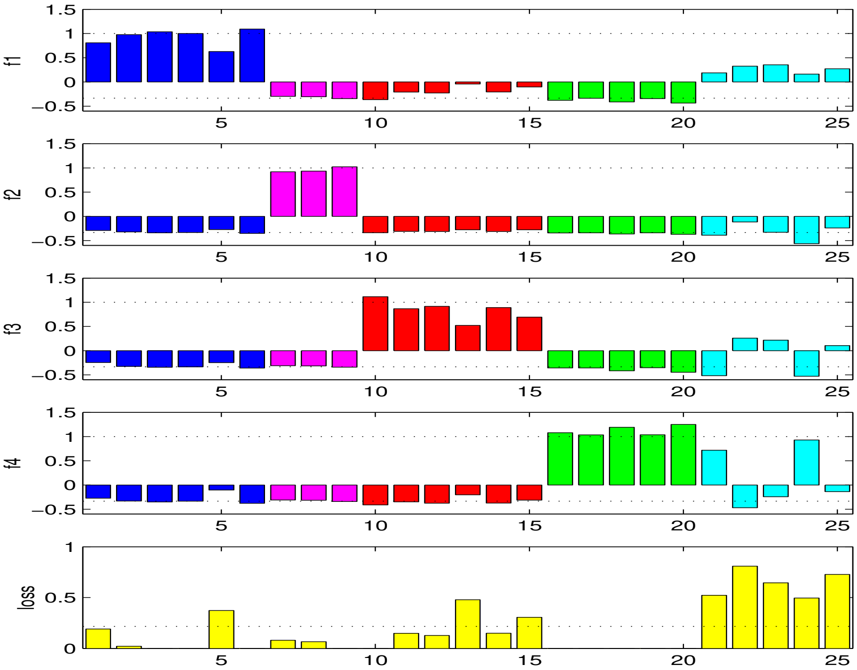

Figure 4: The first four panels show the predicted decision vectors (f_1, f_2, f_3, f_4) at the test examples. The four class labels are coded according as EWS in blue: $(1, -1/3, -1/3, -1/3)$, BL in purple: $(-1/3, 1, -1/3, -1/3)$, NB in red: $(-1/3, -1/3, 1, -1/3)$, and RMS in green: $(-1/3, -1/3, -1/3, 1)$. 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. The last panel depicts the loss for the predicted decision vector at each test example. The last five losses corresponding to the predictions of non SRBCTs all exceed the threshold (the dotted line) below which means a strong prediction. Three test examples falling into the known four classes can not be classified confidently by the same threshold.

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) and Hastie and Tibshirani (1998). However, multiclass problems are treated as a series of binary class problems in each of the references. Although they 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 the classes at once, let alone that the SVM, in particular, is not designed to convey the information of class probabilities. Instead of 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 to be 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 in Mukherjee, Tamayo, Slonim, Verri, Golub, Mesirov and

Poggio (1999). The MSVM decision vector $(f_1, \cdots, f_k)$ 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 $L(\cdot)$, $g(\mathbf{y}, \mathbf{f}(\mathbf{x})) \equiv L(\mathbf{y}) \cdot (\mathbf{f}(\mathbf{x}) - \mathbf{y})_+$ 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 will 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 $g(j, \mathbf{f}(\mathbf{x}))$ stand for $g(\mathbf{v}_j, \mathbf{f}(\mathbf{x}))$. We assume that the probability of a correct prediction given $\mathbf{f}(\mathbf{x})$, $P(Y = arg\max_j f_j(\mathbf{x})|\mathbf{f}(\mathbf{x}))$ 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, $P(Y = arg\max_j f_j(\mathbf{x})|\mathbf{f}(\mathbf{x}))$ 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 it, 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, $P(Y = 1) = \cdots = P(Y = k)$ and $P(\mathbf{f}(\mathbf{x})|Y = y) = P(\pi(\mathbf{f}(\mathbf{x}))|Y = \pi(y))$ for any permutation operator π of $\{1, \cdots, k\}$, it follows that $P(Y = arg\max_j f_j(\mathbf{x})|\mathbf{f}(\mathbf{x})) = P(Y = \pi(arg\max_j f_j(\mathbf{x}))|\pi(\mathbf{f}(\mathbf{x})))$. Consequently, under these symmetry and invariance assumption with respect to k classes, we can pool the pairs of the hinge loss and the indicator for all the classes, and estimate the invariant prediction strength function in terms of the loss, regardless of the predicted class. In almost separable classification problems, we might see the loss values for correct classifications only, impeding the 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 SRBCTs. The last panel in Figure 4 depicts the loss for the predicted MSVM decision vector at each test example including five non SRBCTs. 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. It was set at 0.2171, which is a jackknife estimate of the 95th percentile of the loss distribution from 63 correct predictions in the training data set. The losses corresponding to the predictions of five non SRBCTs all exceed the threshold, while three test examples out of 20 can not be classified confidently by thresholding.

6.2 Cloud Classification with Radiance profiles

The MODIS (MODerate resolution Imaging Spectroradiometer) is a key instrument of the Earth Observing System (EOS). It measures radiances at 36 wavelengths including infrared and visible bands every one to two days with spatial resolution 250 m to 1 km. For more information about the MODIS instrument, see `http://modis.gsfc.nasa.gov/`. Earth Observing System models require knowledge of whether a radiance profile is cloud free, or not. If the profile is not cloud free, it is valuable to have information concerning the type of cloud. For more informations on the MODIS cloud mask algorithm with a simple threshold technique, see Ackerman, Strabala, Menzel, Frey, Moeller and Gumley (1998). We have applied the MSVM to simulated MODIS type channels 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, 12 microns or MODIS channels 1, 2, 3, 4, 5, 6, 7, 27, 28, 29, 31, 32) were simulated using DISORT, driven by STREAMER in Key and Schweiger (1998). Setting atmospheric conditions as simulation parameters, atmospheric temperature and moisture profiles were selected from the 3I TIGR (Thermodynamic Initial Guess Retrieval) data base, and the surface was set to be water. A total of 744 radiance profiles over the ocean (81 clear scenes, 202 liquid clouds and 461 ice clouds) are given in the data set. Each simulated

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	(i) R_2, $\log_{10}(R_5/R_6)$	11.50	14.97	16.58
5	(i)+R_1/R_2, BT_{31}, $BT_{32} - BT_{29}$	10.16	15.24	12.30
12	(ii) original 12 variables	12.03	16.84	20.86
12	log transformed (ii)	9.89	16.84	18.98

radiance profile consists of seven reflectances (R) at .66, .86, .46, .55, 1.2, 1.6, 2.1 microns, and five brightness temperatures (BT) at 6.6, 7.3, 8.6, 11, 12 microns. No single channel seemed to give a clear separation of the three categories. The two variables, $R_{channel_2}$ vs $\log_{10}(R_{channel_5}/R_{channel_6})$ were initially chosen to use for classification based on an understanding of the underlying physics, and an examination of several other scatter plots. To test how predictive the two features, $R_{channel_2}$ and $\log_{10}(R_{channel_5}/R_{channel_6})$ are, we split the data set into a training set and a test set, and applied the MSVM with two features only to the training data. 370 examples, almost half of the original data were selected randomly as the training set. The Gaussian kernel was used and the tuning parameters were tuned by 5-fold CV. The test error rate of the SVM rule over 374 test examples was 11.5% (= 43/374). The left panel of Figure 5 shows the classification boundaries determined by the training data set in this case. Note that a lot of 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 improve the classification accuracy significantly. These variables are given in the second row of Table 5, and again the choice was based on knowledge of the underlying physics and pairwise scatter plots. We could classify correctly just five more examples than the two features only case with the misclassification rate 10.16% (=38/374). Assuming no such domain knowledge regarding which features to look at, we applied the MSVM to the original 12 radiance channels without any transformations or variable selections. It yielded 12.03% test error rate, which is slightly larger than the MSVMs with the tailored two or five features. Interestingly enough, when all the variables are 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, but much more sophisticated than the MODIS cloud mask algorithm. The library `tree` in the R package was used. For each combination of the variables, the size of the fitted tree was determined by the 10-fold cross validation of the training set, and its error rate was estimated over the test set. The results are under the column heading TREE in Table 5. Over all the combinations of the variables considered, the MSVM gives smaller test error rates than the tree method. This suggests the possibility that the proposed MSVM improves the accuracy of the current cloud detection algorithm. To roughly measure how hard the classification problem is due to the intrinsic overlap between class distributions, we applied the nearest neighbor (NN) method with its results in the last column of Table 5. They suggest that the data set is not trivially separable. It would be interesting to investigate further if any sophisticated variable (feature) selection methods may improve the accuracy substantially.

So far, we have treated different types of misclassification equally. However, misclassifying clouds as clear could be more serious than other kinds of misclassifications in practice, since essen-

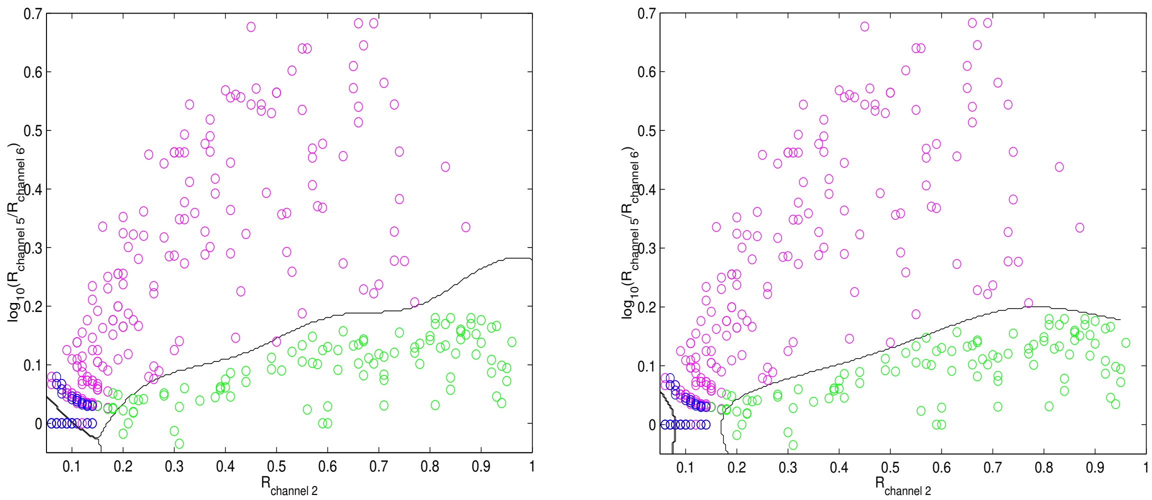

Figure 5: The classification boundaries determined by the MSVM using 370 training examples randomly selected from the data set in the standard case (left) and the nonstandard case (right) where the cost of misclassifying clouds as clear is 1.5 times higher than other types of misclassifications. clear sky: blue, water clouds: green, and ice clouds: purple.

tially this cloud detection algorithm will be used as cloud mask. We considered a cost structure which penalizes misclassifying clouds as clear 1.5 times more than misclassifications of other kinds. Its corresponding classification boundaries are drawn in the right panel of Figure 5. It was observed that if the cost 1.5 is replaced by 2, then there is no region left for the clear sky category at all within the square range of the two features considered here. The approach to estimating the prediction strength 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 Support Vector Machines to the multicategory case so that the resulting classifier approximates the optimal classification rule. The nonstandard Multicategory Support Vector Machine we have proposed allows a unifying formulation when there are possibly nonrepresentative training sets and either equal or unequal misclassification costs. An approximate leaving-out-one cross validation function was derived for tuning the method, and compared 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 k-fold cross validation method, but tends to slightly oversmooth. Then, the usefulness of the multicategory SVM was demonstrated through the 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 achieving the smallest error rate by a classifier, but also having its compact representation for better interpretation. For instance, classification problems in data mining, and bioinformatics often pose a question of which subsets of the variables are most responsible for the class separation. It would be valuable to generalize some variable selection methods for binary SVMs further to the multicategory SVM. Another direction of future work includes establishing the advantage of the multicategory SVM theoretically, such as its convergence rates to the optimal error rate, compared to those indirect ways to classify via estimation of the conditional probability or density functions.

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

APPENDIX A: PROOFS

Proof of Lemma 1. Since $E[L(Y)\cdot(\mathbf{f}(X)-Y)_+] = E(E[L(Y)\cdot(\mathbf{f}(X)-Y)_+|X])$, we can minimize $E[L(Y)\cdot(\mathbf{f}(X)-Y)_+]$ by minimizing $E[L(Y)\cdot(\mathbf{f}(X)-Y)_+|X=\mathbf{x}]$ for every $\mathbf{x}$. If we write out the functional for each $\mathbf{x}$, we have

$$\begin{aligned} E[L(Y)\cdot(\mathbf{f}(X)-Y)_+|X=\mathbf{x}] &= \sum_{j=1}^{k}\Big(\sum_{l\neq j}(f_l(\mathbf{x})+\frac{1}{k-1})_+\Big)p_j(\mathbf{x}) \\ &= \sum_{j=1}^{k}(1-p_j(\mathbf{x}))(f_j(\mathbf{x})+\frac{1}{k-1})_+. \end{aligned} \tag{30}$$

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 $j=1,\cdots,k$, to minimize (30). If any $f_j(\mathbf{x}) < -1/(k-1)$, then we can always find another $\mathbf{f}^*(\mathbf{x})$ which 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 which are greater than $-1/(k-1)$. The existence of such other components is always guaranteed by the sum-to-zero 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. Since 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 $j=1,\cdots,k$. Dropping the truncate functions from (30), and rearranging, we get

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

Without loss of generality, we may assume that $k = arg\max_{j=1,\cdots,k} p_j(\mathbf{x})$ 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 $j=1,\cdots,k-1$ 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-zero 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 $R(\mathbf{f}(\mathbf{x}))$. Since we fix $f_1=-1$, R can be seen as a function of (f_2,f_3):

$$R(f_2,f_3) = (3+f_2)_+p_1+(3+f_3)_+p_1+(1-f_2)_+p_2+(2+f_3-f_2)_+p_2+(1-f_3)_+p_3+(2+f_2-f_3)_+p_3.$$

Now consider (f_2,f_3) in the neighborhood of $(1,1)$: $0<f_2<2$, $0<f_3<2$. 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

$R(1,1) = 4p_1 + 2 + (1-2p_2) + (1-2p_3)$. Since $1/3 < p_2 < 1/2$, if $f_2 > 1$, then $f_2(1-2p_2) + (1-f_2)_+ p_2 = f_2(1-2p_2) > 1-2p_2$; 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 $(f_2, f_3) = (1,1)$. Since 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 above we used the constraint $f_1 = -1$. Other constraints can certainly be used. For example, if we use the constraint $f_1 + f_2 + f_3 = 0$ instead of $f_1 = -1$, 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 the arguments used for the proof of Lemma 1, it can be shown that $E[L(Y^s)\cdot(\mathbf{f}(X^s) - Y^s)_+ | X^s = \mathbf{x}] = 1/(k-1)\sum_{j=1}^k \sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x}) + \sum_{j=1}^k \left(\sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x})\right) f_j(\mathbf{x})$. We can immediately eliminate the first term which does not involve any $f_j(\mathbf{x})$ from our consideration. To make the equation simpler, let $W_j(\mathbf{x})$ be $\sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x})$ for $j = 1, \cdots, k$. Then the whole equation reduces to the following up to a constant.

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

Without loss of generality, we may assume that $k = arg\min_{j=1,\cdots,k} W_j(\mathbf{x})$. To minimize the expected quantity, $f_j(\mathbf{x})$ should be $-1/(k-1)$ for $j = 1, \cdots, k-1$ because of the nonnegativity of $W_j(\mathbf{x}) - W_k(\mathbf{x})$ and $f_j(\mathbf{x}) \geq -1/(k-1)$ for all $j = 1, \cdots, k$. Finally, we have $f_k(\mathbf{x}) = 1$ by the sum-to-zero 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 $j = 1, \cdots, k$ where c_{ij}'s are some constants, and $\rho_j(\cdot)$ is the element in the RKHS orthogonal to the span of $\{K(\mathbf{x}_i, \cdot), i = 1, \cdots, n\}$. By the sum-to-zero 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 $i = 1, \cdots, n$. Then,

$$\begin{aligned} f_j(\mathbf{x}_i) &= b_j + h_j(\mathbf{x}_i) = b_j + (h_j, K(\mathbf{x}_i, \cdot))_{H_K} \\ &= b_j + (\sum_{l=1}^n c_{lj} K(\mathbf{x}_l, \cdot) + \rho_j(\cdot), K(\mathbf{x}_i, \cdot))_{H_K} = b_j + \sum_{l=1}^n c_{lj} K(\mathbf{x}_l, \mathbf{x}_i) \end{aligned}$$

So, the data fit functional in (7) does not depend on $\rho_j(\cdot)$ at all for $j = 1, \cdots, k$. 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 $j = 1, \cdots, k-1$, 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-zero constraint at the data points only is equivalent to minimizing (7) under the constraint for every $\mathbf{x}$. Let K be now the n by n matrix with il entry $K(\mathbf{x}_i, \mathbf{x}_l)$. Let $\mathbf{e}$ be the column vector with n ones, and $\mathbf{c}_{\cdot j} = (c_{1j}, \cdots, c_{nj})^t$. Given the representation (12), consider the problem of minimizing (7) under $(\sum_{j=1}^k b_j)\mathbf{e} + K(\sum_{j=1}^k \mathbf{c}_{\cdot j}) = 0$. For any $f_j(\cdot) = b_j + \sum_{i=1}^n c_{ij} K(\mathbf{x}_i, \cdot)$ satisfying $(\sum_{j=1}^k b_j)\mathbf{e} + K(\sum_{j=1}^k \mathbf{c}_{\cdot j}) = 0$, 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

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

Since the equality holds only when $K\bar{\mathbf{c}} = 0$, that is, $K(\sum_{j=1}^k \mathbf{c}_{\cdot j}) = 0$, we know that at the minimizer, $K(\sum_{j=1}^k \mathbf{c}_{\cdot j}) = 0$, and therefore $\sum_{j=1}^k b_j = 0$. Observe that $K(\sum_{j=1}^k \mathbf{c}_{\cdot j}) = 0$ implies

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

It means $\sum_{j=1}^k \sum_{i=1}^n c_{ij} K(\mathbf{x}_i, \mathbf{x}) = 0$ for every $\mathbf{x}$. Hence, minimizing (7) under the sum-to-zero 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 (Leaving-out-one Lemma) Observe that

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

The first inequality holds by the definition of $\mathbf{f}_\lambda^{[-i]}$. Notice that the jth coordinate of $L(\mu(\mathbf{f}_{\lambda i}^{[-i]}))$ is positive only when $\mu_j(\mathbf{f}_{\lambda i}^{[-i]}) = -1/(k-1)$, while the corresponding jth coordinate of $(\mathbf{f}_{\lambda i}^{[-i]} - \mu(\mathbf{f}_{\lambda i}^{[-i]}))_+$ will be zero since $f_{\lambda j}^{[-i]}(\mathbf{x}_i) < -1/(k-1)$ for $\mu_j(\mathbf{f}_{\lambda i}^{[-i]}) = -1/(k-1)$. As a result, $g(\mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{f}_{\lambda i}^{[-i]}) = L(\mu(\mathbf{f}_{\lambda i}^{[-i]})) \cdot (\mathbf{f}_{\lambda i}^{[-i]} - \mu(\mathbf{f}_{\lambda i}^{[-i]}))_+ = 0$. 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-zero constraint, it suffices to consider $k-1$ coordinates of $\mathbf{y}_i$ and $\mathbf{f}_i$ as arguments of g, which correspond to nonzero components of $L(\mathbf{y}_i)$. Suppose that $\mathbf{y}_i = (-1/(k-1), \cdots, -1/(k-1), 1)$. All the arguments will hold analogously for other class examples. By the first order Taylor expansion, we have

$$g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i) \approx -\sum_{j=1}^{k-1} \frac{\partial}{\partial f_j} g(\mathbf{y}_i, \mathbf{f}_i) \left(f_j(\mathbf{x}_i) - f_j^{[-i]}(\mathbf{x}_i) \right). \tag{31}$$

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

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

Let $(\mu_{i1}(\mathbf{f}),\cdots,\mu_{ik}(\mathbf{f})) = \mu(\mathbf{f}(\mathbf{x}_i))$ and similarly $(\mu_{i1}(\mathbf{f}^{[-i]}),\cdots,\mu_{ik}(\mathbf{f}^{[-i]})) = \mu(\mathbf{f}^{[-\mathbf{i}]}(\mathbf{x}_i))$. Using the leaving-out-one lemma for $j = 1,\cdots,k-1$ and the Taylor expansion,

$$f_j(\mathbf{x}_i) - f_j^{[-i]}(\mathbf{x}_i) \approx \left(\frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i1}},\cdots,\frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i,k-1}}\right)\begin{pmatrix} y_{i1} - \mu_{i1}(\mathbf{f}^{[-i]}) \\ \vdots \\ y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}^{[-i]}) \end{pmatrix}. \tag{32}$$

The solution of k-class SVM 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 (31), (32) and $(y_{i1}-\mu_{i1}(\mathbf{f}^{[-i]}),\cdots,y_{i,k-1}-\mu_{i,k-1}(\mathbf{f}^{[-i]})) \approx (y_{i1}-\mu_{i1}(\mathbf{f}),\cdots,y_{i,k-1}-\mu_{i,k-1}(\mathbf{f}))$, 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 the approximations are defined analogously for other class examples, we have $g(\mathbf{y}_i,\mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i,\mathbf{f}_i) \approx (k-1)K(\mathbf{x}_i,\mathbf{x}_i)\sum_{j=1}^{k} L_{ij}[f_j(\mathbf{x}_i)+1/(k-1)]_* c_{ij}(y_{ij}-\mu_{ij}(\mathbf{f}))$.

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**(D24): 32,141–157.

Allwein, E. L., Schapire, R. E. and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers, *Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp. 9–16.

Boser, B., Guyon, I. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers, *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**(2): 121–167.

Crammer, K. and Singer, Y. (2000). On the learnability and design of output codes for multiclass problems, *Computational Learing Theory*, pp. 35–46.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, 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**(457): 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*, Department of Statistics, Stanford University, Stanford, CA.

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* M. I. Jordan, M. J. Kearns and S. A. Solla (eds), *Advances in Neural Information Processing Systems 10*, MIT Press.

Jaakkola, T. and Haussler, D. (1999). Probabilistic kernel regression models, *Proceedings of the 1999 Conference on AI and Statistics*, Morgan Kaufmann.

Joachims, T. (2000). Estimating the generalization performance of a SVM efficiently, *in* P. Langley (ed.), *Proceedings of ICML-00, 17th International Conference on Machine Learning*, Morgan Kaufmann, Stanford, US, pp. 431–438.

Key, J. and Schweiger, A. (1998). Tools for atmospheric radiative transfer: Streamer and FluxNet, *Computers and Geosciences* **24**(5): 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 Tchebychean Spline functions, *Journal of Mathematics Analysis and Applications* **33**(1): 82–95.

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

Lee, Y. and Lee, C.-K. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data, *Bioinformatics* **19**(9): 1132–1139.

Lee, Y., Wahba, G. and Ackerman, S. (2003). Classification of satellite radiance data by multicategory support vector machines, *Technical Report 1075*, Department of Statistics, University of Wisconsin, Madison WI.

Lin, Y. (2001). A note on margin-based loss functions in classification, *Technical Report 1044*, Department of Statistics, University of Wisconsin, Madison WI.

Lin, Y. (2002). Support vector machines and the Bayes rule in classification, *Data Mining and Knowledge Discovery* **6**: 259–275.

Lin, Y., Lee, Y. and Wahba, G. (2002). Support vector machines for classification in nonstandard situations, *Machine Learning* **46**: 191–202.

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* F. van Harmelen (ed.), *Proceedings of the 15th annual conference on artificial intelligence.*

Price, D., Knerr, S., Personnaz, L. and Dreyfus, G. (1995). Pairwise neural network classifiers with probabilistic outputs, *in* G. Tesauro, D. Touretzky and T. Leen (eds), *Advances in Neural Information Processing Systems 7 (NIPS-94)*, MIT Press, pp. 1109–1116.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press.

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

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York.

Wahba, G. (1990). *Spline Models for Observational Data*, Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia.

Wahba, G. (1998). Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV, *in* B. Schölkopf, C. J. C. Burges and A. J. Smola (eds), *Advances in Kernel Methods: Support Vector Learning*, MIT Press, pp. 69–87.

Wahba, G. (2002). Soft and hard classification by reproducing kernel Hilbert space methods, *Proc.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* D. Denison, M. Hansen, C. Holmes, B. Mallick and B. Yu (eds), *Nonlinear Estimation and Classification*, Springer, 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* A. J. Smola, P. Bartlett, B. Schölkopf and D. Schurmans (eds), *Advances in Large Margin Classifiers*, MIT Press, pp. 297–309.

Weston, J. and Watkins, C. (1999). Support vector machines for multiclass pattern recognition, *Proceedings of the Seventh European Symposium On Artificial Neural Networks.*

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, *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, 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.