

DEPARTMENT OF STATISTICS  
University of Wisconsin  
1210 West Dayton St.  
Madison, WI 53706

TECHNICAL REPORT NO. 1063

September 9, 2002

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

Yoonkyung Lee <sup>1</sup>

---

<sup>1</sup>Presently at Department of Statistics, Ohio State University, Columbus, OH 43210.  
This work was supported in part by NSF Grant DMS 0072292, NIH Grant EY09946 and  
NASA Grant NAG5-10273.

MULTICATEGORY SUPPORT VECTOR  
MACHINES, THEORY, AND  
APPLICATION TO THE CLASSIFICATION  
OF MICROARRAY DATA AND  
SATELLITE RADIANCE DATA

By  
Yoonkyung Lee

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY  
(STATISTICS)

at the  
UNIVERSITY OF WISCONSIN – MADISON  
2002

# Abstract

Two category Support Vector Machines (SVM) have been very popular in the machine learning community for the classification problem. Solving multiclass 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 Multiclass Support Vector Machine (MSVM), which extends the binary SVM to the multiclass 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. Its effectiveness is demonstrated through the applications to cancer classification using microarray data and cloud classification with satellite radiance profiles.

# Acknowledgements

I would like to express my sincerest gratitude to my thesis advisor, Grace Wahba, without whose support and guidance this work could not have been completed. Her enthusiasm and encouragement made the entire course of this study very much enjoyable. Her responsive email correspondences were extremely helpful at various stages of my research. Not only were they full of thoughtful suggestions, but also with a personal touch. Had I saved them all, they could have been longer than a book. Most of all, it was the appreciation and pleasure of scientific endeavors with statistics that I learned from her.

Yi Lin offered me inspirations in tackling with my thesis problems, which I deeply appreciate. His critical readings and sharp comments made this dissertation more rigorous. My dissertation is truly a work of collaboration. I would like to thank Michael Ferris for helping me with computation, and Steve Ackerman for kindly providing the cloud data set which led to an interesting application. The collaboratorship with Cheol-Koo Lee, which came as a serendipity, allowed me better understanding of microarray data and spawned further interests in the field of genetics. I wish to thank Wei-Yin Loh and Kam-Wah Tsui for their careful reading and valuable comments on this dissertation.

I also thank my friends in Madison, too many to name, and the ‘Thursday group’ fellow students for helping me out in many ways and making my life here colorful. My special thanks go to Pil Ho, whose companionship and unselfish support throughout the years ever since we launched our journey together have lighted up my life. Last but not the least, I am grateful for the support from my family and family-in-law.

This research was partly supported by NSF Grant DMS0072292, NIH Grant EY09946, and NASA Grant NAG5 10273.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Outline of the Thesis . . . . .	3
<b>2 Classification problem and the Bayes rule</b>	<b>4</b>
2.1 Equal Cost Case . . . . .	4
2.2 Unequal Cost Case . . . . .	5
<b>3 Support Vector Machines</b>	<b>7</b>
3.1 Linear SVM . . . . .	7
3.2 SVM in Regularization Framework . . . . .	8
3.3 Relation to the Bayes Rule . . . . .	9
3.4 Dual Problem . . . . .	10
<b>4 Multicategory Support Vector Machines</b>	<b>11</b>
4.1 Standard Case . . . . .	11
4.2 Nonstandard Case . . . . .	14
4.3 The Representer Theorem . . . . .	16
4.4 Dual Formulation . . . . .	18
4.5 Implementation and Related Issues . . . . .	21
4.6 Data Adaptive Tuning Criterion . . . . .	22
4.6.1 Generalized Comparative Kullback-Leibler Distance . . . . .	23
4.6.2 Leaving Out One Lemma . . . . .	24
4.6.3 Generalized Approximate Cross Validation . . . . .	25
<b>5 Numerical Study</b>	<b>30</b>
5.1 Simulations . . . . .	30
5.1.1 One dimensional three-class example . . . . .	31
5.1.2 Two dimensional four-class example . . . . .	34
5.1.3 High dimensional six-category example . . . . .	36
5.2 Small Scale Benchmark Study . . . . .	38

<b>6 Applications</b>	<b>41</b>
6.1 Cancer Classification with Microarray Data . . . . .	41
6.1.1 Introduction . . . . .	41
6.1.2 Assessment of Prediction Strength . . . . .	42
6.1.3 Small Round Blue Cell Tumors of Childhood . . . . .	44
6.2 Cloud Classification with Radiance profiles . . . . .	52
6.2.1 Introduction . . . . .	52
6.2.2 Data Description . . . . .	52
6.2.3 Analysis . . . . .	53
<b>7 Conclusion</b>	<b>62</b>
<b>Bibliography</b>	<b>64</b>

# Chapter 1

## Introduction

### 1.1 Overview

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 thesis 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), Burges (1998), and Cristianini and Shawe-Taylor (2000). It is commonly known that the SVM paradigm can comfortably sit in regularization frameworks 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 (1999) 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 a fresh light on the SVM and illuminate its properties in the perspective as other penalized methods have been comprehended. 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 sample from the underlying distribution  $P(\mathbf{x}, y)$ . In the classification problem, the goal is to find a classification rule that generalizes the relation between the covariate and its class label, based on  $n$  realizations of  $(X, Y)$ ,  $(\mathbf{x}_i, y_i)$  for  $i = 1, \dots, n$ , so that for future sample  $\mathbf{x}$ , its class can be predicted with a minimal error rate. The theoretically optimal rule, the so called Bayes rule, minimizes the misclassification error rate and it is given by  $\text{sign}(p_1(\mathbf{x}) - 1/2)$ , 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,  $f(\mathbf{x})$  targets directly  $\text{sign}(p_1(\mathbf{x}) - 1/2)$ , or equivalently  $\text{sign}\left(\log \frac{p_1(\mathbf{x})}{1 - p_1(\mathbf{x})}\right)$  without estimating a conditional probability function  $p_1(\mathbf{x})$ , thus realizing the Bayes rule via the SVM decision rule,  $\text{sign}(f(\mathbf{x}))$ .

Let us turn our attention to the multicategory classification problem. We assume the class label  $Y \in \{1, \dots, k\}$  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 rule assigns a test sample  $\mathbf{x}$  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. 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 samples are used to learn each classifier. Regarding its statistical validity, 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 test sample the class giving the largest  $f_j(\mathbf{x})$  for  $j = 1, \dots, k$ , 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. Learning on the insight that we have from the two category SVM,  $f_j(\mathbf{x})$  will approximate  $\text{sign}(p_j(\mathbf{x}) - 1/2)$ . 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  $\ell = 1, \dots, k$  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 rule. Thus, there is a demand for a rightful 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), Bredensteiner and Bennett (1999) and Crammer and Singer (2000). However, they are rather



algorithmic extensions of the binary SVM and the relation of those formulations to the Bayes rule is unclear. 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. Based on the loss function, we extend the SVM paradigm to the multiclass case and show that this extension ensures the solution to directly target the Bayes 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).

## 1.2 Outline of the Thesis

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

## Chapter 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 chapter. They serve as golden standards for any reasonable classifiers to approximate. The optimal rule for the equal misclassification costs is followed by that for unequal costs. Their derivations are fairly straightforward, and can be found in any general references to classification problems, for instance, Devroye, Györfi and Lugosi (1996).

### 2.1 Equal Cost Case

In the classification problem, we are given a training data set that consists of  $n$  samples  $(\mathbf{x}_i, y_i)$  for  $i = 1, \dots, n$ .  $\mathbf{x}_i \in R^d$  represents covariates and  $y_i \in \{1, \dots, k\}$  denotes the class label of the  $i$ th sample. The task is to learn a classification rule  $\phi(\mathbf{x}) : R^d \rightarrow \{1, \dots, k\}$  that well matches attributes  $\mathbf{x}_i$  to a class label  $y_i$ . We assume that each  $(\mathbf{x}_i, y_i)$  is an independent random sample from a target population with probability distribution  $P(\mathbf{x}, y)$ . Let  $(X, Y)$  denote a generic pair of a random sample from  $P(\mathbf{x}, y)$ , and  $p_j(\mathbf{x}) = P(Y = j | X = \mathbf{x})$  be the conditional probability of class  $j$  given  $X = \mathbf{x}$  for  $j = 1, \dots, k$ . If the misclassification costs are all equal, the loss by the classification rule  $\phi$  at  $(\mathbf{x}, y)$  is defined as

$$l(y, \phi(\mathbf{x})) = I(y \neq \phi(\mathbf{x})) \quad (2.1)$$

where  $I(\cdot)$  is the indicator function, which assumes 1 if its argument is true, and 0 otherwise. The best classification rule with respect to the loss would be the one that minimizes the expected misclassification rate. The best rule, often called the Bayes rule is given by

$$\phi_B(\mathbf{x}) = \arg \min_{j=1, \dots, k} [1 - p_j(\mathbf{x})] = \arg \max_{j=1, \dots, k} p_j(\mathbf{x}). \quad (2.2)$$

If we knew the conditional probabilities  $p_j(\mathbf{x})$ , we can implement  $\phi_B(\mathbf{x})$  easily. However, since we rarely know  $p_j(\mathbf{x})$ 's in reality, we need to approximate the Bayes rule by learning from a training data set. A common way to approximate it is to estimate  $p_j(\mathbf{x})$ 's or equivalently the log odds  $\log[p_j(\mathbf{x})/p_k(\mathbf{x})]$  from data first and to plug them into the rule.

## 2.2 Unequal Cost Case

When the misclassification costs are not equal, which may be more useful in solving real world problems, we change the loss (2.1) to reflect the cost structure. First, define  $C_{j\ell}$  for  $j, \ell = 1, \dots, k$  as the cost of misclassifying an example from class  $j$  to class  $\ell$ .  $C_{jj}$  for  $j = 1, \dots, k$  are all zero since the correct decision should not be penalized. The loss function for the unequal costs is then

$$l(y, \phi(\mathbf{x})) = \sum_{j=1}^k I(y = j) \left( \sum_{\ell=1}^k C_{j\ell} I(\phi(\mathbf{x}) = \ell) \right). \quad (2.3)$$

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

$$\phi_B(\mathbf{x}) = \mathit{arg} \min_{j=1, \dots, k} \sum_{\ell=1}^k C_{\ell j} p_{\ell}(\mathbf{x}). \quad (2.4)$$

Notice that when the misclassification costs are all equal, say,  $C_{j\ell} = 1$ ,  $j \neq \ell$ , then (2.4) nicely reduces to (2.2), the Bayes rule in the equal cost case. Besides the concern with different misclassification costs, sampling bias is an issue that 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 samples. 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. The sampling bias leads to distortion of the class proportions. If we know the prior class proportions, then there is a remedy for the sampling bias by incorporating the discrepancy between the sample proportions and the population proportions into a cost component. Let  $\pi_j$  be the prior proportion of class  $j$  in the general population, and  $\pi_j^s$  be the prespecified proportion of class  $j$  examples in a training data set.  $\pi_j^s$  may be different from  $\pi_j$  if the sampling bias has occurred. Define  $g_j(\mathbf{x})$  the probability density of  $X$  for class  $j$  population,  $j = 1, \dots, k$ , and let  $(X^s, Y^s)$  be a random sample obtained by the sampling mechanism used in the data collection stage.

Then, the difference between  $(X^s, Y^s)$  in the training data and  $(X, Y)$  in the general population becomes clear when we look at the conditional probabilities. While

$$p_j(\mathbf{x}) = P(Y = j|X = \mathbf{x}) = \frac{\pi_j g_j(\mathbf{x})}{\sum_{\ell=1}^k \pi_\ell g_\ell(\mathbf{x})}, \quad (2.5)$$

$$p_j^s(\mathbf{x}) = P(Y^s = j|X^s = \mathbf{x}) = \frac{\pi_j^s g_j(\mathbf{x})}{\sum_{\ell=1}^k \pi_\ell^s g_\ell(\mathbf{x})}. \quad (2.6)$$

Since we learn a classification rule only through the training data, it is better to express the Bayes rule in terms of the quantities for  $(X^s, Y^s)$  and  $\pi_j$  which we assume to know a priori. One can verify that the following is equivalent to (2.4).

$$\phi_B(\mathbf{x}) = \arg \min_{j=1, \dots, k} \sum_{\ell=1}^k \frac{\pi_\ell}{\pi_\ell^s} C_{\ell j} p_\ell^s(\mathbf{x}) = \arg \min_{j=1, \dots, k} \sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x}) \quad (2.7)$$

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. Lin et al. (2002) has more details on the two-category case in treating the Support Vector Machines. Following the distinction in the paper, 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.

## Chapter 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 the multidimensional input space. For reference, see Boser, Guyon and Vapnik (1992), Vapnik (1998), Burges (1998), Cristianini and Shawe-Taylor (2000) and references therein. The class labels  $y_i$  are either 1 or -1 in the SVM setting. The symmetry in the representation of  $y_i$  is very essential in the mathematical formulation of SVMs.

### 3.1 Linear SVM

Let us consider the linearly separable case when the positive examples (with  $y_i = 1$ ) in the training data set can be perfectly separated from the negative examples (with  $y_i = -1$ ) by a hyperplane in  $R^d$ . Then there exists  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ ,  $\mathbf{w} \in R^d$  and  $b \in R$ , satisfying the following conditions for  $i = 1, \dots, n$ :

$$f(\mathbf{x}_i) \geq 1 \quad \text{if } y_i = 1 \quad (3.1)$$

$$f(\mathbf{x}_i) \leq -1 \quad \text{if } y_i = -1 \quad (3.2)$$

Or more succinctly,

$$y_i f(\mathbf{x}_i) \geq 1 \quad \text{for } i = 1, \dots, n. \quad (3.3)$$

Here, the hyperplane  $\mathbf{w} \cdot \mathbf{x} + b = 0$  separates all the positive examples from the negative examples. Among the hyperplanes satisfying (3.3), Support Vector Machines look for the one with the maximum margin. The margin is defined as the sum of the shortest distance from the hyperplane to the closest positive example and the closest negative example. It is given by  $2/\|\mathbf{w}\|$  when the closest positive example lies on the level set of  $f(\mathbf{x}) = 1$  and likewise, the closest negative example lies on  $f(\mathbf{x}) = -1$  level set. Note that finding the hyperplane maximizing  $2/\|\mathbf{w}\|$  is equivalent to finding the one minimizing  $\|\mathbf{w}\|^2$ , subject to (3.3).

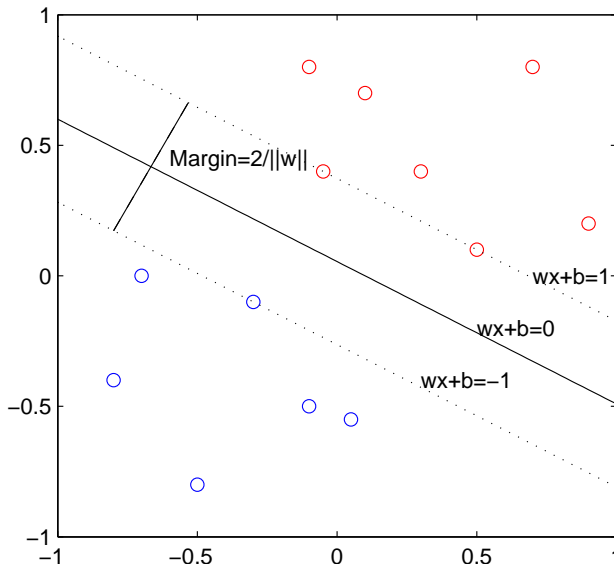


Figure 3.1: A canonical example of the binary Support Vector Machine

Figure 3.1 shows a canonical picture of the SVM in the linearly separable case. The red circles indicate positive examples and the blue circles represent negative examples. The solid line corresponds to the SVM solution which puts positive examples maximally apart from the negative examples. In the nonseparable case, the Support Vector Machine finds  $f(\mathbf{x})$  minimizing  $\|\mathbf{w}\|^2$  together with the loss function  $(1 - y_i f(\mathbf{x}_i))_+$  averaged over all the samples, where  $(x)_+ = \max(x, 0)$ . Essentially, the SVM loss function  $(1 - y_i f(\mathbf{x}_i))_+$ , so-called hinge loss penalizes the violation of the separability condition (3.3).

## 3.2 SVM in Regularization Framework

Further, generalizing SVM classifiers from hyperplanes to nonlinear ones, we get the following SVM formulation with 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

$$\frac{1}{n} \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + \lambda \|h\|_{H_K}^2 \quad (3.4)$$

where  $\|h\|_{H_K}^2$  denotes the square norm of the function  $h$  defined in the RKHS with the reproducing kernel function  $K(\cdot, \cdot)$ . For more information on RKHS, see Aronszajn (1950) and Wahba (1990).  $\lambda$  is a given tuning parameter which balances the data fit measured as the average hinge loss, and the complexity of  $f(\mathbf{x})$ , measured as  $\|h\|_{H_K}^2$ . The classification rule  $\phi(\mathbf{x})$  induced by  $f(\mathbf{x})$  is  $\phi(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$ . The function  $f(\mathbf{x})$  yields the level curve defined by  $f(\mathbf{x}) = 0$  in  $R^d$ , which is the classification boundary of the rule  $\phi(\mathbf{x})$ . 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 former is an upper bound of the latter, 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.

### 3.3 Relation to the Bayes Rule

Theoretical justifications of the SVM in Vapnik's structural risk minimization approach can be found in Vapnik (1995), and Vapnik (1998). These arguments are based on upper bounds of its generalization error in terms of Vapnik-Chervonenkis dimensions, which are often too pessimistic to explain the success of the SVMs in many applications completely. Another explanation as to why the SVM works well has been given in Lin (2002), by identifying the asymptotic target function of the SVM formulation, and associating it with the Bayes rule. Noting that the representation of class label  $Y$  in the binary SVMs is either 1 or -1, one can verify that the Bayes rule in (2.2) is  $\phi_B(\mathbf{x}) = \text{sign}(p_1(\mathbf{x}) - 1/2)$  in this symmetric representation. Lin (2002) showed that, if the reproducing kernel Hilbert space is rich enough, the solution  $f(\mathbf{x})$  approaches the Bayes rule directly, as the sample size  $n$  goes to  $\infty$  for appropriately chosen  $\lambda$ . For example, the Gaussian kernel is one of typically used kernels for SVMs, the RKHS induced by which is flexible enough to approximate  $\text{sign}(p_1(\mathbf{x}) - 1/2)$ . Compared to other popular statistical methods implementing the Bayes rule via density estimations or logistic regressions, the mechanism that Support Vector Machines approximate the optimal rule seems to be particularly efficient for sparse data since  $\text{sign}(p_1(\mathbf{x}) - 1/2)$  would be much simpler to estimate than the probability  $p_1(\mathbf{x})$ . For a discussion of the connection between SVMs and likelihood-based penalized methods, see Wahba (1998).

### 3.4 Dual Problem

For analogous expositions later, we sketch the derivations to get the SVM solution to (3.4). The minimizer  $f(\mathbf{x})$  is known to be of the form  $\sum_{i=1}^n c_i K(\mathbf{x}, \mathbf{x}_i) + b$  by the representer theorem in Kimeldorf and Wahba (1971). Using the reproducing property of  $K$ , (3.4) can be written as a constrained quadratic optimization problem in terms of  $c_1, \dots, c_n$  and  $b$ . Finally, the coefficients  $c_i$  and the constant  $b$  are determined by its dual problem using Lagrange multipliers  $\alpha = (\alpha_1, \dots, \alpha_n)^t$ . The dual problem is given by

$$\min L_D(\alpha) = \frac{1}{2} \alpha^t H \alpha - \mathbf{e}^t \alpha \quad (3.5)$$

$$\text{subject to } 0 \leq \alpha \leq \mathbf{e} \quad (3.6)$$

$$\alpha^t \mathbf{y} = 0 \quad (3.7)$$

where  $H = \left( \frac{1}{2n\lambda} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right)$ ,  $\mathbf{y} = (y_1, \dots, y_n)^t$ , and  $\mathbf{e} = (1, \dots, 1)^t$ . Once  $\alpha_i$ 's are obtained from the quadratic programming problem above, we have  $c_i = \frac{\alpha_i y_i}{2n\lambda}$  by the primal-dual relation, and  $b$  is determined from the examples with  $0 < \alpha_i < 1$  by the Karush-Kuhn-Tucker optimality conditions. For more details on mathematical programming, see Mangasarian (1994). Usually, some fraction of  $\alpha_i$ 's are zero. Thus, the SVM solution permits a sparse expansion depending only on the samples with nonzero  $\alpha_i$ , which are called support vectors. The support vectors are typically either near the classification boundaries or misclassified samples. The modification of the standard SVM setting for the nonstandard case is treated in detail in Lin et al. (2002). Similarly, it has been shown that the modified SVM implements the optimal classification rule in the same way as the standard SVM.



## Chapter 4

# Multicategory Support Vector Machines

Now, we propose to extend the whole machinery of the SVM for the multiclass case, from its optimization problem formulation to its theoretical properties. 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 for the multicategory case, and show that the generalized formulation encompasses that of the two-category SVM, retaining desirable properties of the binary SVM. 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

Throughout this section, we assume 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  $j = 1, \dots, k$  as a  $k$ -dimensional vector with 1 in the  $j$ th coordinate and  $-\frac{1}{k-1}$  elsewhere. Then,  $\mathbf{y}_i$  is coded as  $\mathbf{v}_j$  if example  $i$  belongs to class  $j$ . For instance, if example  $i$  belongs to class 1,  $\mathbf{y}_i = \mathbf{v}_1 = (1, -\frac{1}{k-1}, \dots, -\frac{1}{k-1})$ . Similarly, if it belongs to class  $k$ ,  $\mathbf{y}_i = \mathbf{v}_k = (-\frac{1}{k-1}, \dots, -\frac{1}{k-1}, 1)$ . Accordingly, we define a  $k$ -tuple of separating functions  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  with the sum-to-zero constraint,  $\sum_{j=1}^k f_j(\mathbf{x}) = 0$  for any  $\mathbf{x} \in R^d$ . Note that the constraint holds implicitly for coded class labels  $\mathbf{y}_i$ . Analogous to the two-category case, we consider  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \in \prod_{j=1}^k (\{1\} + H_{K_j})$ , the product space of  $k$  reproducing kernel Hilbert spaces  $H_{K_j}$  for  $j = 1, \dots, k$ . 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  $j = 1, \dots, k$ , 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  $j$ th 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  $j$ th coordinate, and 1 elsewhere. Now, we propose that to find  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \in \prod_1^k(\{1\} + H_K)$ , with the sum-to-zero constraint, minimizing the following quantity is a natural extension of SVMs methodology.

$$\frac{1}{n} \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ + \frac{1}{2} \lambda \sum_{j=1}^k \|h_j\|_{H_K}^2 \quad (4.1)$$

where  $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$  means  $[(f_1(\mathbf{x}_i) - y_{i1})_+, \dots, (f_k(\mathbf{x}_i) - y_{ik})_+]$  by taking the truncate function  $(\cdot)_+$  componentwise, and  $\cdot$  operation in the data fit functional indicates the Euclidean inner product. The classification rule induced by  $\mathbf{f}(\mathbf{x})$  is naturally

$$\phi(\mathbf{x}) = \arg \max_j f_j(\mathbf{x}). \quad (4.2)$$

As with the hinge loss function in the binary case, the proposed loss function has analogous relation to the misclassification loss (2.1) in the multicategory case. 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  $\frac{k}{k-1}$  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 (3.4) and (4.1) are identical,  $f_1$  playing the same role as  $f$  in (3.4). Also, note that  $\frac{1}{2} \lambda \sum_{j=1}^2 \|h_j\|_{H_K}^2 = \frac{1}{2} \lambda (\|h_1\|_{H_K}^2 + \|-h_1\|_{H_K}^2) = \lambda \|h_1\|_{H_K}^2$ , 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 (3.4) and (4.1) are identical. These verify that the binary SVM formulation (3.4) is a special case of (4.1) when  $k = 2$ .

An immediate justification for this new formulation generalizing the binary SVM paradigm is that it carries over the efficiency of implementing the Bayes rule in the same fashion. In the binary case, Lin (2002) adopted the approach of Cox and O'Sullivan (1990) to establish that the SVM directly targets the optimal classification rule, bypassing the estimation of a possibly more complex probability function. Cox and O'Sullivan (1990) have provided a theoretical

framework for analyzing the asymptotics of penalized methods. It is the very first step to identify the asymptotic target function of a penalized method, which is a minimizer of its limit data fit functional. Having said that the SVM paradigms in general are penalized methods, we first identify the asymptotic target function of (4.1) in this direction. The limit of the data fit functional in (4.1) is  $E[L(Y) \cdot (\mathbf{f}(X) - Y)_+]$ .

**Lemma 4.1.** *The minimizer of  $E[L(Y) \cdot (\mathbf{f}(X) - Y)_+]$  under the sum-to-zero constraint is  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  with*

$$f_j(\mathbf{x}) = \begin{cases} 1 & \text{if } j = \arg \max_{\ell=1, \dots, k} p_\ell(\mathbf{x}) \\ -\frac{1}{k-1} & \text{otherwise} \end{cases} \quad (4.3)$$

Proof: 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

$$\begin{aligned} & E[L(Y) \cdot (\mathbf{f}(X) - Y)_+ | X = \mathbf{x}] \\ &= \sum_{j=1}^k \left( \sum_{\ell \neq j} (f_\ell(\mathbf{x}) + \frac{1}{k-1})_+ \right) p_j(\mathbf{x}) \end{aligned} \quad (4.4)$$

$$= \sum_{j=1}^k \left( \sum_{\ell \neq j} p_\ell(\mathbf{x}) \right) (f_j(\mathbf{x}) + \frac{1}{k-1})_+ \quad (4.5)$$

$$= \sum_{j=1}^k (1 - p_j(\mathbf{x})) (f_j(\mathbf{x}) + \frac{1}{k-1})_+. \quad (4.6)$$

Here, we claim that it is sufficient to search over  $\mathbf{f}(\mathbf{x})$  with  $f_j(\mathbf{x}) \geq -\frac{1}{k-1}$  for all  $j = 1, \dots, k$ , to minimize (4.6). If any  $f_j(\mathbf{x}) < -\frac{1}{k-1}$ , 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  $-\frac{1}{k-1}$  and subtract the surplus  $-\frac{1}{k-1} - f_j(\mathbf{x})$  from other component  $f_\ell(\mathbf{x})$ 's which are greater than  $-\frac{1}{k-1}$ . 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  $(f_j^*(\mathbf{x}) + \frac{1}{k-1})_+ \leq (f_j(\mathbf{x}) + \frac{1}{k-1})_+$  for each  $j$ . Since the expected loss is a nonnegatively weighted sum of  $(f_j(\mathbf{x}) + \frac{1}{k-1})_+$ , it is sufficient to consider  $\mathbf{f}(\mathbf{x})$  with  $f_j(\mathbf{x}) \geq -\frac{1}{k-1}$  for all  $j = 1, \dots, k$ . Dropping the truncate functions

from (4.6), and rearranging, we get

$$\begin{aligned} & E[L(Y) \cdot (\mathbf{f}(X) - Y)_+ | X = \mathbf{x}] \\ &= \sum_{j=1}^k (1 - p_j(\mathbf{x})) \left( f_j(\mathbf{x}) + \frac{1}{k-1} \right) \end{aligned} \quad (4.7)$$

$$= 1 + \sum_{j=1}^{k-1} (1 - p_j(\mathbf{x})) f_j(\mathbf{x}) + (1 - p_k(\mathbf{x})) \left( - \sum_{j=1}^{k-1} f_j(\mathbf{x}) \right) \quad (4.8)$$

$$= 1 + \sum_{j=1}^{k-1} (p_k(\mathbf{x}) - p_j(\mathbf{x})) f_j(\mathbf{x}). \quad (4.9)$$

Without loss of generality, we may assume that  $k = \arg \max_{j=1, \dots, k} p_j(\mathbf{x})$  by the symmetry in the class labels. This implies that to minimize the expected loss,  $f_j(\mathbf{x})$  should be  $-\frac{1}{k-1}$  for  $j = 1, \dots, k-1$  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.  $\square$

Indeed, Lemma 4.1 is a multicategory extension of Lemma 3.1 in Lin (2002) which showed that  $f(\mathbf{x})$  in ordinary SVMs approximates  $\text{sign}(p_1(\mathbf{x}) - 1/2)$  asymptotically. If the reproducing kernel Hilbert space is flexible enough to approximate the minimizer in Lemma 4.1, and  $\lambda$  is chosen appropriately, the solution  $\mathbf{f}(\mathbf{x})$  to (4.1) approaches it as the sample size  $n$  goes to  $\infty$ . Notice that the minimizer is exactly the code of the most probable class. Then, the classification rule induced by  $\mathbf{f}(\mathbf{x})$  in Lemma 4.1 is  $\phi(\mathbf{x}) = \arg \max_j f_j(\mathbf{x}) = \arg \max_j p_j(\mathbf{x}) = \phi_B(\mathbf{x})$ , the Bayes rule (2.2) for the standard multicategory case.

## 4.2 Nonstandard Case

In this section, we allow different misclassification costs and the possibility of sampling bias mentioned earlier. Necessary modification of the multicategory SVM (4.1) to accommodate such differences is straightforward. 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  $C_{j\ell}$  for  $j, \ell = 1, \dots, k$  meaning the cost of misclassifying an example from class  $j$  to class  $\ell$ . All the diagonal entries  $C_{jj}$  for  $j = 1, \dots, k$  would be zero. Modify  $L(\mathbf{y}_i)$  in (4.1) to the  $j$ th 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 Chapter 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  $j, \ell = 1, \dots, k$ . Accordingly, define  $L(\mathbf{y}_i)$  to be the  $j$ th 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 (4.1) 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 4.2.** *The minimizer of  $E[L(Y^s) \cdot (\mathbf{f}(X^s) - Y^s)_+]$  under the sum-to-zero constraint is  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  with*

$$f_j(\mathbf{x}) = \begin{cases} 1 & \text{if } j = \arg \min_{\ell=1, \dots, k} \sum_{m=1}^k l_{m\ell} p_m^s(\mathbf{x}) \\ -\frac{1}{k-1} & \text{otherwise} \end{cases} \quad (4.10)$$

Proof: Parallel to all the arguments used for the proof of Lemma 4.1, it can be shown that

$$\begin{aligned} & E[L(Y^s) \cdot (\mathbf{f}(X^s) - Y^s)_+ | X^s = \mathbf{x}] \\ &= \frac{1}{k-1} \sum_{j=1}^k \sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x}) + \sum_{j=1}^k \left( \sum_{\ell=1}^k l_{\ell j} p_\ell^s(\mathbf{x}) \right) f_j(\mathbf{x}) \end{aligned} \quad (4.11)$$

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  $j = 1, \dots, k$ . Then the whole equation reduces to the following up to a constant.

$$\sum_{j=1}^k W_j(\mathbf{x}) f_j(\mathbf{x}) = \sum_{j=1}^{k-1} W_j(\mathbf{x}) f_j(\mathbf{x}) + W_k(\mathbf{x}) \left( -\sum_{j=1}^{k-1} f_j(\mathbf{x}) \right) \quad (4.12)$$

$$= \sum_{j=1}^{k-1} (W_j(\mathbf{x}) - W_k(\mathbf{x})) f_j(\mathbf{x}) \quad (4.13)$$

Without loss of generality, we may assume that  $k = \arg \min_{j=1, \dots, k} W_j(\mathbf{x})$ . To minimize the expected quantity,  $f_j(\mathbf{x})$  should be  $-\frac{1}{k-1}$  for  $j = 1, \dots, k-1$

because of the nonnegativity of  $W_j(\mathbf{x}) - W_k(\mathbf{x})$  and  $f_j(\mathbf{x}) \geq -\frac{1}{k-1}$  for all  $j = 1, \dots, k$ . Finally, we have  $f_k(\mathbf{x}) = 1$  by the sum-to-zero constraint.  $\square$

It is not hard to see that Lemma 4.1 is a special case of the above lemma. Like the standard case, Lemma 4.2 has its existing counterpart when  $k = 2$ . See Lemma 3.1 in Lin et al. (2002) with the caution that  $\mathbf{y}_i$ , and  $L(\mathbf{y}_i)$  are defined differently than here. Again, the lemma implies that if the reproducing kernel Hilbert space is rich enough to approximate the minimizer in Lemma 4.2, for appropriately chosen  $\lambda$ , we would observe the solution to (4.1) to be very close to the minimizer for a large sample. Analogously, the classification rule derived from the minimizer in Lemma 4.2 is  $\phi(\mathbf{x}) = \arg \max_j f_j(\mathbf{x}) = \arg \min_{j=1, \dots, k} \sum_{\ell=1}^k l_{\ell j} p_{\ell}^s(\mathbf{x}) = \phi_B(\mathbf{x})$ , the Bayes rule (2.7) for the nonstandard multicategory case.

### 4.3 The Representer Theorem

We explain how to carry out the computation to find the minimizer of (4.1). First, the problem of finding constrained functions  $(f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  minimizing (4.1) is transformed into that of finding finite dimensional coefficients instead, 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). The following theorem 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 4.1.** *To find  $(f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \in \prod_1^k (\{1\} + H_K)$ , with the sum-to-zero constraint, minimizing (4.1) is equivalent to find  $(f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$  of the form*

$$f_j(\mathbf{x}) = b_j + \sum_{i=1}^n c_{ij} K(\mathbf{x}_i, \mathbf{x}) \quad \text{for } j = 1, \dots, k \quad (4.14)$$

*with the sum-to-zero constraint only at  $\mathbf{x}_i$  for  $i = 1, \dots, n$ , minimizing (4.1).*

Proof. Consider  $f_j(\mathbf{x}) = b_j + h_j(\mathbf{x})$  with  $h_j \in H_K$ . Decompose

$$h_j(\cdot) = \sum_{\ell=1}^n c_{\ell j} K(\mathbf{x}_{\ell}, \cdot) + \rho_j(\cdot) \quad (4.15)$$

for  $j = 1, \dots, k$  where  $c_{ij}$ 's are some constants, and  $\rho_j(\cdot)$  is the element in the RKHS orthogonal to the span of  $\{K(\mathbf{x}_i, \cdot), i = 1, \dots, n\}$ . 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  $i = 1, \dots, n$ . Then,

$$f_j(\mathbf{x}_i) = b_j + h_j(\mathbf{x}_i) = b_j + (h_j, K(\mathbf{x}_i, \cdot))_{H_K} \quad (4.16)$$

$$= b_j + \left( \sum_{\ell=1}^n c_{\ell j} K(\mathbf{x}_\ell, \cdot) + \rho_j(\cdot), K(\mathbf{x}_i, \cdot) \right)_{H_K} \quad (4.17)$$

$$= b_j + \sum_{\ell=1}^n c_{\ell j} K(\mathbf{x}_\ell, \mathbf{x}_i) \quad (4.18)$$

So, the data fit functional in (4.1) does not depend on  $\rho_j(\cdot)$  at all for  $j = 1, \dots, k$ . On the other hand, we have  $\|h_j\|_{H_K}^2 = \sum_{i,\ell} c_{ij} c_{\ell j} K(\mathbf{x}_\ell, \mathbf{x}_i) + \|\rho_j\|_{H_K}^2$  for  $j = 1, \dots, k-1$ , and  $\|h_k\|_{H_K}^2 = \left\| \sum_{j=1}^{k-1} \sum_{i=1}^n c_{ij} K(\mathbf{x}_i, \cdot) \right\|_{H_K}^2 + \|\sum_{j=1}^{k-1} \rho_j\|_{H_K}^2$ . To minimize (4.1), obviously  $\rho_j(\cdot)$  should vanish. It remains to show that minimizing (4.1) under the sum-to-zero constraint at the data points only is equivalent to minimizing (4.1) under the constraint for every  $\mathbf{x}$ . With some abuse of notation, let  $K$  be now the  $n$  by  $n$  matrix with  $i\ell$  th entry  $K(\mathbf{x}_i, \mathbf{x}_\ell)$ . Let  $\mathbf{e}$  be the column vector with  $n$  ones, and  $\mathbf{c}_{\cdot j} = (c_{1j}, \dots, c_{nj})^t$ . Given the representation (4.14), consider the problem of minimizing (4.1) 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

$$f_j^*(\cdot) = b_j^* + \sum_{i=1}^n c_{ij}^* K(\mathbf{x}_i, \cdot) = (b_j - \bar{b}) + \sum_{i=1}^n (c_{ij} - \bar{c}_i) K(\mathbf{x}_i, \cdot) \quad (4.19)$$

where  $\bar{b} = \frac{1}{k} \sum_{j=1}^k b_j$  and  $\bar{c}_i = \frac{1}{k} \sum_{j=1}^k c_{ij}$ . Then  $f_j(\mathbf{x}_i) = f_j^*(\mathbf{x}_i)$ , and

$$\sum_{j=1}^k \|h_j^*\|_{H_K}^2 = \sum_{j=1}^k \mathbf{c}_{\cdot j}^t K \mathbf{c}_{\cdot j} - k \bar{\mathbf{c}}^t K \bar{\mathbf{c}} \leq \sum_{j=1}^k \mathbf{c}_{\cdot j}^t K \mathbf{c}_{\cdot j} = \sum_{j=1}^k \|h_j\|_{H_K}^2. \quad (4.20)$$

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

$$\left( \sum_{j=1}^k \mathbf{c}_{\cdot j} \right)^t K \left( \sum_{j=1}^k \mathbf{c}_{\cdot j} \right) = \left\| \sum_{i=1}^n \left( \sum_{j=1}^k c_{ij} \right) K(\mathbf{x}_i, \cdot) \right\|_{H_K}^2 = \left\| \sum_{j=1}^k \sum_{i=1}^n c_{ij} K(\mathbf{x}_i, \cdot) \right\|_{H_K}^2 = 0. \quad (4.21)$$

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 (4.1) under the sum-to-zero constraint at the data points is equivalent to minimizing (4.1) 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  $\mathbf{x}$ .  $\square$

**Remark 4.1.** *If the reproducing kernel  $K$  is strictly positive definite, then the sum-to-zero constraint at the data points can be replaced by the equality constraints  $\sum_{j=1}^k b_j = 0$  and  $\sum_{j=1}^k \mathbf{c}_{.j} = 0$ .*

## 4.4 Dual Formulation

Switching to a Lagrangian formulation of the problem (4.1), 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 4.1, we can write the primal problem in terms of  $b_j$  and  $c_{ij}$  only. Since the problem involves  $k$  class components symmetrically, we may rewrite it more succinctly in vector notation. Let  $L_j \in R^n$  for  $j = 1, \dots, k$  be the  $j$ th column of the  $n$  by  $k$  matrix with the  $i$ th row  $L(\mathbf{y}_i)$ . Let  $\xi_{.j} \in R^n$  for  $j = 1, \dots, k$  be the  $j$ th column of the  $n$  by  $k$  matrix with the  $i$ th row  $\xi_i$ . Similarly,  $\mathbf{y}_{.j}$  denotes the  $j$ th column of the  $n$  by  $k$  matrix with the  $i$ th row  $\mathbf{y}_i$ . Then, the primal problem in vector notation is

$$\min L_P(\xi, \mathbf{c}, \mathbf{b}) = \sum_{j=1}^k L_j^t \xi_{.j} + \frac{1}{2} n \lambda \sum_{j=1}^k \mathbf{c}_{.j}^t K \mathbf{c}_{.j} \quad (4.22)$$

$$\text{subject to} \quad b_j \mathbf{e} + K \mathbf{c}_{.j} - \mathbf{y}_{.j} \leq \xi_{.j} \quad \text{for } j = 1, \dots, k \quad (4.23)$$

$$\xi_{.j} \geq 0 \quad \text{for } j = 1, \dots, k \quad (4.24)$$

$$(\sum_{j=1}^k b_j) \mathbf{e} + K(\sum_{j=1}^k \mathbf{c}_{.j}) = 0 \quad (4.25)$$

It is a quadratic optimization problem with some equality and inequality constraints. The duality theory in nonlinear programming allows us to solve its dual problem, which is easier than, but equivalent to the primal problem. See Mangasarian (1994) for an overview of the duality results of nonlinear programming. To derive its Wolfe dual problem, we introduce nonnegative Lagrange multipliers  $\alpha_{.j} = (\alpha_{1j}, \dots, \alpha_{nj})^t \in R^n$  for (4.23), nonnegative Lagrange multipliers  $\gamma_j \in R^n$  for (4.24), and unconstrained Lagrange multipliers  $\delta_f \in R^n$  for (4.25), the equality constraints. Then, the dual problem becomes a problem of



maximizing

$$\begin{aligned}
L_D = & \sum_{j=1}^k L_j^t \xi_{.j} + \frac{1}{2} n \lambda \sum_{j=1}^k \mathbf{c}_{.j}^t K \mathbf{c}_{.j} + \sum_{j=1}^k \alpha_{.j}^t (b_j \mathbf{e} + K \mathbf{c}_{.j} - \mathbf{y}_{.j} - \xi_{.j}) \\
& - \sum_{j=1}^k \gamma_j^t \xi_{.j} + \delta_f^t \left( \left( \sum_{j=1}^k b_j \right) \mathbf{e} + K \left( \sum_{j=1}^k \mathbf{c}_{.j} \right) \right)
\end{aligned} \tag{4.26}$$

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

$$\frac{\partial L_D}{\partial \xi_{.j}} = L_j - \alpha_{.j} - \gamma_j = 0 \tag{4.27}$$

$$\frac{\partial L_D}{\partial \mathbf{c}_{.j}} = n \lambda K \mathbf{c}_{.j} + K \alpha_{.j} + K \delta_f = 0 \tag{4.28}$$

$$\frac{\partial L_D}{\partial b_j} = (\alpha_{.j} + \delta_f)^t \mathbf{e} = 0 \tag{4.29}$$

$$\alpha_{.j} \geq 0 \tag{4.30}$$

$$\gamma_j \geq 0 \tag{4.31}$$

Let  $\bar{\alpha}$  be  $\frac{1}{k} \sum_{j=1}^k \alpha_{.j}$ . Since  $\delta_f$  is unconstrained, one may take  $\delta_f = -\bar{\alpha}$  from (4.29). Accordingly, (4.29) becomes  $(\alpha_{.j} - \bar{\alpha})^t \mathbf{e} = 0$ . Eliminating all the primal variables in  $L_D$  by the equality constraint (4.27) and using relations from (4.28) and (4.29), we have the following dual problem.

$$\min L_D(\alpha) = \frac{1}{2} \sum_{j=1}^k (\alpha_{.j} - \bar{\alpha})^t K (\alpha_{.j} - \bar{\alpha}) + n \lambda \sum_{j=1}^k \alpha_{.j}^t \mathbf{y}_{.j} \tag{4.32}$$

$$\text{subject to } 0 \leq \alpha_{.j} \leq L_j \quad \text{for } j = 1, \dots, k \tag{4.33}$$

$$(\alpha_{.j} - \bar{\alpha})^t \mathbf{e} = 0 \quad \text{for } j = 1, \dots, k \tag{4.34}$$

Matching the dual variable  $\alpha_i$  in the binary case with the corresponding dual vector  $(\alpha_{i1}, \alpha_{i2})$  in the multiclass case,

$$\alpha_i = \begin{cases} \alpha_{i2} & \text{with } \alpha_{i1} = 0 \quad \text{if } y_i = 1 \text{ or } (1, -1) \\ \alpha_{i1} & \text{with } \alpha_{i2} = 0 \quad \text{if } y_i = -1 \text{ or } (-1, 1) \end{cases} \tag{4.35}$$

and consequently

$$\alpha_i y_i = \alpha_{i2} - \alpha_{i1} = -2(\alpha_{i1} - \bar{\alpha}_i) = 2(\alpha_{i2} - \bar{\alpha}_i). \tag{4.36}$$

From these relations, it can be verified that the above dual formulation, although disguised in its form, reduces to the binary SVM dual problem (3.5), (3.6), and (3.7), when  $k = 2$  and the costs are all equal. Once the quadratic programming problem is solved, the coefficients can be determined by the relation  $\mathbf{c}_{\cdot j} = -\frac{1}{n\lambda}(\alpha_{\cdot j} - \bar{\alpha})$  for  $j = 1, \dots, k$  from (4.28). 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 should satisfy

$$\alpha_{\cdot j} \perp (b_j \mathbf{e} + K \mathbf{c}_{\cdot j} - \mathbf{y}_{\cdot j} - \xi_{\cdot j}) \quad \text{for } j = 1, \dots, k \quad (4.37)$$

$$\gamma_j = (L_j - \alpha_{\cdot j}) \perp \xi_{\cdot j} \quad \text{for } j = 1, \dots, k \quad (4.38)$$

where  $\perp$  means that componentwise products are all zero. If  $0 < \alpha_{ij} < l_{ij}$  for some  $i$ , then  $\xi_{ij}$  should be zero from (4.38), and this implies that  $b_j + \sum_{\ell=1}^n c_{\ell j} K(\mathbf{x}_{\ell}, \mathbf{x}_i) - y_{ij} = 0$  from (4.37). If there is no example satisfying  $0 < \alpha_{ij} < l_{ij}$  for some class  $j$ ,  $\mathbf{b} = (b_1, \dots, b_k)$  is determined as the solution to the following problem:

$$\min_{\mathbf{b}} \frac{1}{n} \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{h}_i + \mathbf{b} - \mathbf{y}_i)_+ \quad (4.39)$$

$$\text{subject to } \sum_{j=1}^k b_j = 0 \quad (4.40)$$

where  $\mathbf{h}_i = (h_{i1}, \dots, h_{ik}) = (\sum_{\ell=1}^n c_{\ell 1} K(\mathbf{x}_{\ell}, \mathbf{x}_i), \dots, \sum_{\ell=1}^n c_{\ell k} K(\mathbf{x}_{\ell}, \mathbf{x}_i))$ .

It is worth noting that if  $(\alpha_{i1}, \dots, \alpha_{ik}) = 0$  for the  $i$ th example, then  $(c_{i1}, \dots, c_{ik}) = 0$ . Removing such example  $(\mathbf{x}_i, \mathbf{y}_i)$  would have no effect on the solution at all. Carrying over the notion of support vectors to the multiclass case, we define support vectors as examples with  $\mathbf{c}_i = (c_{i1}, \dots, c_{ik}) \neq 0$  for  $i = 1, \dots, n$ . Hence, depending on the number of support vectors, the multiclass SVM solution may have a sparse representation, which is also one of the main characteristics of the binary SVM. This seemingly novel concept of the support vectors has more plain interpretation of the active constraints in nonlinear programming problems. The only difference is that the notion is attached to each sample in the training data set, rather than each constraint. Notice that a single sample is related to multiple constraints in the multiclass SVM.

## 4.5 Implementation and Related Issues

In practice, solving the quadratic programming (QP) problem can be done via available optimization packages for moderate size problems. For instance, <http://www.numerical.rl.ac.uk/qp/qp.html> returned by Google search gives a list of QP solving packages and codes. All the examples presented in this thesis were done via MATLAB 6.1 with an interface to PATH 3.0, an optimization package implemented by Ferris and Munson (1999). It is helpful to put (4.32), (4.33), and (4.34) in a standard QP format for use of some existing QP solvers.

$$\min L_D(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^t \left[ (I_k - \frac{1}{k} J_k) \otimes K \right] \boldsymbol{\alpha} + n\lambda \mathbf{Y}^t \boldsymbol{\alpha} \quad (4.41)$$

$$\text{subject to} \quad 0 \leq \boldsymbol{\alpha} \leq \mathbf{L} \quad (4.42)$$

$$\left[ (I_k - \frac{1}{k} J_k) \otimes \mathbf{e} \right]^t \boldsymbol{\alpha} = 0 \quad (4.43)$$

where

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{.1} \\ \vdots \\ \alpha_{.k} \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_{.1} \\ \vdots \\ \mathbf{y}_{.k} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} L_1 \\ \vdots \\ L_k \end{pmatrix},$$

$I_k$  is the  $k$  by  $k$  identity matrix, and  $J_k$  is the  $k$  by  $k$  matrix with ones.  $\otimes$  means the Kronecker product. Note that due to the upper bound  $\mathbf{L}$  having  $n$  zeros in (4.42), the number of nontrivial dual variables is  $(k-1)n$ . Compared to solving  $k$  QP problems with  $n$  dual variables in the one-versus-rest approach, the multiclass formulation amounts to solving a bigger problem once. Hsu and Lin (2002) compared several methods to solve multiclass problems using SVMs in terms of their performance and computing time. Although our method presented here was not included in the study, the comparison results are still relevant because some multiclass extensions in the study share the same computational complexity. It was reported that considering all the classes at once tends to be slower than solving a series of binary problems, however, the former needed fewer support vectors, which was also mentioned in Weston and Watkins (1999). In terms of accuracy, the two different approaches were pretty comparable with nonlinear kernels. In contrast to this, using the linear kernel resulted in the worst accuracy to the one-vs-rest approach among those methods investigated.

To make the computation amenable to large data sets, one may borrow implementation ideas successfully exercised in binary SVMs. Studies have shown

that slight modification of the problem gives a fairly good approximation to the solution in binary case, and its computational benefit is immense for massive data. For example, SOR (Successive OverRelaxation) in Mangasarian and Musicant (1999), and SSVM (Smooth SVM) in Lee and Mangasarian (2001) are strategies in this vein. Decomposition algorithms are the other very popular approach for the binary SVM, the main idea of which is to solve a smaller piece of the problem each time and update the solution iteratively until it satisfies the optimality conditions. SMO (Sequential Minimal Optimization) in Platt (1999), the chunking method in Boser et al. (1992), and SVM<sup>light</sup> in Joachims (1999) are the examples of this kind. Another possibility to make the proposed method computationally feasible for massive datasets is to exploit the specific structure of the QP problem. Noting that the whole issue is approximating some step functions by basis functions determined by kernel functions evaluated at data points, we may consider reducing the number of basis functions as well. For a large dataset, subsetting basis functions would not lead to any significant loss in accuracy, while we get a computational gain by doing so. How to ease computational burden of the proposed multiclass approach is an ongoing research problem.

## 4.6 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 (2001).

We derive an approximate leaving-out-one cross validation function, called Generalized Approximate Cross Validation (GACV) for the multiclass Support Vector Machines. It is based on the leaving-out-one arguments, reminiscent of GACV derivations for penalized likelihood methods in Xiang and Wahba (1996). It is quite parallel to the binary GACV in Wahba et al. (2000) except that the sum-to-zero constraints upon coefficients should be taken care of, due to the characterization of the multiclass SVM solution. Throughout the derivation, it is desirable to formulate GACV symmetrically with respect to each class, since exchanging class labels nominally would not change the problem at all.

### 4.6.1 Generalized Comparative Kullback-Leibler Distance

It would be ideal but only theoretically possible to choose tuning parameters minimizing the generalized comparative Kullback-Leibler (GCKL) distance with respect to the multiclass SVM loss function,  $g(\mathbf{y}_i, \mathbf{f}_i) \equiv L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$  averaged over a data set with the same covariates  $\mathbf{x}_i$  and unobserved  $Y_i$ ,  $i = 1, \dots, n$ :

$$GCKL(\lambda) = E_{true} \frac{1}{n} \sum_{i=1}^n g(\mathbf{Y}_i, \mathbf{f}_i) = E_{true} \frac{1}{n} \sum_{i=1}^n L(\mathbf{Y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{Y}_i)_+. \quad (4.44)$$

By Lemma 4.1 and Lemma 4.2, the convex multiclass SVM loss function asymptotically surrogates (2.1) and (2.3), the commonly used loss functions in the standard and the nonstandard classification problems. 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. For concise notations, let  $J_\lambda(\mathbf{f}) = \frac{\lambda}{2} \sum_{j=1}^k \|h_j\|_{H_K}^2$ , and  $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ . We denote the objective function of the multicategory SVM (4.1) by  $I_\lambda(\mathbf{f}, \mathbf{y})$ . That is,

$$I_\lambda(\mathbf{f}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n g(\mathbf{y}_i, \mathbf{f}_i) + J_\lambda(\mathbf{f}). \quad (4.45)$$

Let  $\mathbf{f}_\lambda$  be the minimizer of  $I_\lambda(\mathbf{f}, \mathbf{y})$  and  $\mathbf{f}_\lambda^{[-i]}$  be the solution to the variational problem when the  $i$ th sample is left out, minimizing

$$\frac{1}{n} \sum_{\substack{\ell=1 \\ \ell \neq i}}^n g(\mathbf{y}_\ell, \mathbf{f}_\ell) + J_\lambda(\mathbf{f}). \quad (4.46)$$

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  $j$ th 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  $GCKL(\lambda)$ :

$$V_0(\lambda) = \frac{1}{n} \sum_{i=1}^n g(\mathbf{y}_i, \mathbf{f}_{\lambda i}^{[-i]}). \quad (4.47)$$

$V_0(\lambda)$  can be reexpressed as the sum of  $OBS(\lambda)$ , the observed fit to the data measured as the average loss and  $D(\lambda)$ , where

$$OBS(\lambda) = \frac{1}{n} \sum_{i=1}^n g(\mathbf{y}_i, \mathbf{f}_{\lambda i}) \quad (4.48)$$

$$D(\lambda) = \frac{1}{n} \sum_{i=1}^n \left( g(\mathbf{y}_i, \mathbf{f}_{\lambda i}^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_{\lambda i}) \right) \quad (4.49)$$

To obtain a computable 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 following lemma.

### 4.6.2 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  $\mathbf{y} \in \mathbf{R}^k$  with sum-to-zero constraint, we define  $L : \mathbf{R}^k \rightarrow \mathbf{R}^k$  as  $L(\mathbf{y}) = (w_1(\mathbf{y})[-y_1 - \frac{1}{k-1}]_*, \dots, w_k(\mathbf{y})[-y_k - \frac{1}{k-1}]_*)$  where  $[\tau]_* = I(\tau \geq 0)$ , and  $(w_1(\mathbf{y}), \dots, w_k(\mathbf{y}))$  is the  $j$ th row of the extended misclassification cost matrix  $L$  with the  $jl$  entry  $\frac{\pi_j}{\pi_j^s} C_{jl}$  if  $\arg \max_{l=1, \dots, k} y_l = j$ . If there are ties, then  $(w_1(\mathbf{y}), \dots, w_k(\mathbf{y}))$  is defined as the average of the rows of the cost matrix  $L$  corresponding to the maximal arguments. We easily check that  $L(0, \dots, 0) = (0, \dots, 0)$  and the extended  $L(\cdot)$  coincides with the original  $L(\cdot)$  over the domain of class representations. We define a class prediction  $\mu(\mathbf{f})$  given the SVM output  $\mathbf{f}$  as a function truncating any component  $f_j < -\frac{1}{k-1}$  to  $-\frac{1}{k-1}$  and replacing the rest by  $\frac{\sum_{j=1}^k I(f_j < -\frac{1}{k-1})}{k - \sum_{j=1}^k I(f_j < -\frac{1}{k-1})} \left( \frac{1}{k-1} \right)$  to satisfy the sum-to-zero constraint. If  $\mathbf{f}$  has a maximum component greater than 1, and all the others less than  $-\frac{1}{k-1}$ , then  $\mu(\mathbf{f})$  is a  $k$ -tuple with 1 on the maximum coordinate and  $-\frac{1}{k-1}$  elsewhere. So, the function  $\mu$  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  $-\frac{1}{k-1}$ ,  $\mu$  maps  $\mathbf{f}$  to  $(0, \dots, 0)$ . For the binary case, the class prediction function is explicitly

$$\mu((f_1, f_2)) = \begin{cases} (1, -1) & \text{if } f_1 > 1 \\ (-1, 1) & \text{if } f_1 < -1 \\ (0, 0) & \text{if } f_1 \in [-1, 1] \end{cases} \quad (4.50)$$

**Lemma 4.3 (Leaving-out-one Lemma).** *The minimizer of  $I_\lambda(\mathbf{f}, \mathbf{y}^{[-i]})$  is  $\mathbf{f}_\lambda^{[-i]}$ , where  $\mathbf{y}^{[-i]} = (\mathbf{y}_1, \dots, \mathbf{y}_{i-1}, \mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{y}_{i+1}, \dots, \mathbf{y}_n)$ .*

Proof. Observe that

$$\begin{aligned} & I_\lambda(\mathbf{f}_\lambda^{[-i]}, \mathbf{y}^{[-i]}) \\ &= \frac{1}{n} g(\mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{f}_{\lambda i}^{[-i]}) + \frac{1}{n} \sum_{\substack{\ell=1 \\ \ell \neq i}}^n g(\mathbf{y}_\ell, \mathbf{f}_{\lambda \ell}^{[-i]}) + J_\lambda(\mathbf{f}_\lambda^{[-i]}) \end{aligned} \quad (4.51)$$

$$\leq \frac{1}{n} g(\mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{f}_{\lambda i}^{[-i]}) + \frac{1}{n} \sum_{\substack{\ell=1 \\ \ell \neq i}}^n g(\mathbf{y}_\ell, \mathbf{f}_\ell) + J_\lambda(\mathbf{f}) \quad (4.52)$$

$$\leq \frac{1}{n} g(\mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{f}_i) + \frac{1}{n} \sum_{\substack{\ell=1 \\ \ell \neq i}}^n g(\mathbf{y}_\ell, \mathbf{f}_\ell) + J_\lambda(\mathbf{f}) \quad (4.53)$$

$$= I_\lambda(\mathbf{f}, \mathbf{y}^{[-i]}) \quad (4.54)$$

The first inequality holds by the definition of  $\mathbf{f}_\lambda^{[-i]}$ . Notice that the  $j$ th coordinate of  $L(\mu(\mathbf{f}_{\lambda i}^{[-i]}))$  is positive only when  $\mu_j(\mathbf{f}_{\lambda i}^{[-i]}) = -\frac{1}{k-1}$ , while the corresponding  $j$ th coordinate of  $(\mathbf{f}_{\lambda i}^{[-i]} - \mu(\mathbf{f}_{\lambda i}^{[-i]}))_+$  will be zero since  $f_{\lambda j}^{[-i]}(\mathbf{x}_i) < -\frac{1}{k-1}$  for  $\mu_j(\mathbf{f}_{\lambda i}^{[-i]}) = -\frac{1}{k-1}$ . As a result,

$$g(\mu(\mathbf{f}_{\lambda i}^{[-i]}), \mathbf{f}_{\lambda i}^{[-i]}) = L(\mu(\mathbf{f}_{\lambda i}^{[-i]})) \cdot (\mathbf{f}_{\lambda i}^{[-i]} - \mu(\mathbf{f}_{\lambda i}^{[-i]}))_+ = 0. \quad (4.55)$$

Thus, the second inequality follows by the nonnegativity of the function  $g$ . This completes the proof.  $\square$

For notational simplicity, we suppress the subscript  $\lambda$  from  $\mathbf{f}$  and  $\mathbf{f}^{[-i]}$ . Due to the sum-to-zero constraint,  $g$  depends only on  $k-1$  components of each  $\mathbf{f}_i$  and  $\mathbf{y}_i$ . Thus, 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  $L(\mathbf{y}_i)$ . We approximate  $g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i)$ , the contribution of the  $i$ th example to  $D(\lambda)$  depending on the class  $\mathbf{y}_i$ .

### 4.6.3 Generalized Approximate Cross Validation

We illustrate the case when the  $i$ th example is from class  $k$ . All the arguments will hold analogously for other class examples. Suppose that  $\mathbf{y}_i =$

$(-\frac{1}{k-1}, \dots, -\frac{1}{k-1}, 1)$ . By the first order Taylor expansion, we have

$$\begin{aligned} & g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i) \\ & \approx - \left( \frac{\partial}{\partial f_1} g(\mathbf{y}_i, \mathbf{f}_i), \dots, \frac{\partial}{\partial f_{k-1}} g(\mathbf{y}_i, \mathbf{f}_i) \right) \begin{pmatrix} f_1(\mathbf{x}_i) - f_1^{[-i]}(\mathbf{x}_i) \\ \vdots \\ f_{k-1}(\mathbf{x}_i) - f_{k-1}^{[-i]}(\mathbf{x}_i) \end{pmatrix} \end{aligned} \quad (4.56)$$

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

$$\frac{\partial}{\partial f_j} g(\mathbf{y}_i, \mathbf{f}_i) = L(\mathbf{y}_i) \cdot \left( 0, \dots, 0, [f_j(\mathbf{x}_i) + \frac{1}{k-1}]_*, 0, \dots, 0 \right) \quad (4.57)$$

$$= L_{ij} [f_j(\mathbf{x}_i) + \frac{1}{k-1}]_*. \quad (4.58)$$

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

$$\begin{aligned} & f_j(\mathbf{x}_i) - f_j^{[-i]}(\mathbf{x}_i) \\ & \approx \left( \frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i1}}, \dots, \frac{\partial f_j(\mathbf{x}_i)}{\partial y_{i,k-1}} \right) \begin{pmatrix} y_{i1} - \mu_{i1}(\mathbf{f}^{[-i]}) \\ \vdots \\ y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}^{[-i]}) \end{pmatrix}. \end{aligned} \quad (4.59)$$

Thus we have the following approximation in matrix notation.

$$\begin{aligned} & \begin{pmatrix} f_1(\mathbf{x}_i) - f_1^{[-i]}(\mathbf{x}_i) \\ \vdots \\ f_{k-1}(\mathbf{x}_i) - f_{k-1}^{[-i]}(\mathbf{x}_i) \end{pmatrix} \\ & \approx \begin{pmatrix} \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i1}} & \dots & \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i,k-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i1}} & \dots & \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i,k-1}} \end{pmatrix} \begin{pmatrix} y_{i1} - \mu_{i1}(\mathbf{f}^{[-i]}) \\ \vdots \\ y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}^{[-i]}) \end{pmatrix}. \end{aligned} \quad (4.60)$$

Recall that in the binary SVM, we could get  $\frac{\partial f(\mathbf{x}_i)}{\partial y_i} = \frac{\alpha_i}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i)$  from

$$f(\mathbf{x}_i) = \sum_{i'=1}^n c_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}) + b = \frac{1}{2n\lambda} \sum_{i'=1}^n \alpha_{i'} y_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}) + b. \quad (4.61)$$



$f(\mathbf{x})$  in the binary SVM corresponds to  $f_1(\mathbf{x})$  of the multiclass SVM, and so does the class code  $\mathbf{y} = (y_1, \dots, y_n)$  to  $\mathbf{y}_{\cdot 1} = (y_{11}, \dots, y_{n1})$ . The correspondence between two equivalent expressions is exploited to extend the two-class GACV to  $k$ -class GACV. The solution of  $k$ -class SVM is given by

$$f_j(\mathbf{x}_i) = \sum_{i'=1}^n c_{i'j} K(\mathbf{x}_i, \mathbf{x}_{i'}) + b_j = - \sum_{i'=1}^n \frac{(\alpha_{i'j} - \bar{\alpha}_{i'})}{n\lambda} K(\mathbf{x}_i, \mathbf{x}_{i'}) + b_j. \quad (4.62)$$

Here is the explicit relation between the coefficients  $c_i$  of the ordinary SVM solution and  $c_{ij}$  of the multiclass SVM through the dual variables  $\alpha_i$  and  $\alpha_{ij}$ , when  $k = 2$ . All the coefficients  $c_i$  and the dual solutions  $\alpha_i$  from the binary SVM are single-indexed, as opposed to those double-indexed in the multiclass SVM. From (4.35), and (4.36), we have

$$c_{i1} = -\frac{(\alpha_{i1} - \bar{\alpha}_i)}{n\lambda} = \frac{\alpha_i y_i}{2n\lambda} = c_i. \quad (4.63)$$

Hinted by the expression of the solution in the binary case, we rewrite  $c_{i'j} = -y_{i'j}(k-1)c_{i'j}$  if the  $i'$ th example is not from class  $j$ , and  $c_{i'j} = (k-1) \sum_{\substack{l=1 \\ l \neq j}}^k y_{i'l} c_{i'l}$  otherwise. When the  $i$ th example is from class  $k$ , we get for  $j$  and  $l = 1, \dots, k-1$

$$\frac{\partial f_j(\mathbf{x}_i)}{\partial y_{il}} = \begin{cases} -(k-1)c_{ij}K(\mathbf{x}_i, \mathbf{x}_i) & \text{if } l = j \\ 0 & \text{if } l \neq j. \end{cases} \quad (4.64)$$

For instance, if the  $i$ th example is from class 2 in the two-class case,

$$\frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i1}} = -c_{i1}K(\mathbf{x}_i, \mathbf{x}_i) = -\frac{\alpha_i y_{i1}}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i) = \frac{\alpha_i}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i). \quad (4.65)$$

By the sum-to-zero constraint,

$$\begin{aligned} \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i1}} &= \frac{\partial f_2(\mathbf{x}_i)}{\partial y_{i2}} = -c_{i2}K(\mathbf{x}_i, \mathbf{x}_i) = c_{i1}K(\mathbf{x}_i, \mathbf{x}_i) \\ &= \frac{\alpha_i y_{i1}}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i) = \frac{\alpha_i}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i) \end{aligned} \quad (4.66)$$

if the  $i$ th example is from class 1. Thus, (4.64) returns to the ordinary form  $\frac{\partial f(\mathbf{x}_i)}{\partial y_i} = \frac{\alpha_i}{2n\lambda} K(\mathbf{x}_i, \mathbf{x}_i)$  when  $k = 2$ . For a class  $k$  example in general, (4.64)

gives

$$\begin{pmatrix} \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i1}} & \cdots & \frac{\partial f_1(\mathbf{x}_i)}{\partial y_{i,k-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i1}} & \cdots & \frac{\partial f_{k-1}(\mathbf{x}_i)}{\partial y_{i,k-1}} \end{pmatrix} = -(k-1)K(\mathbf{x}_i, \mathbf{x}_i) \begin{pmatrix} c_{i1} & 0 & \cdots & 0 \\ 0 & c_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{i,k-1} \end{pmatrix}. \quad (4.67)$$

Define

$$\mathbf{D}g_i = \left( \frac{\partial}{\partial f_1}g(\mathbf{y}_i, \mathbf{f}_i), \cdots, \frac{\partial}{\partial f_{k-1}}g(\mathbf{y}_i, \mathbf{f}_i) \right), \quad (4.68)$$

and

$$\Delta \mathbf{Y}_i = (y_{i1} - \mu_{i1}(\mathbf{f}), \cdots, y_{i,k-1} - \mu_{i,k-1}(\mathbf{f}))^t. \quad (4.69)$$

From (4.56), (4.60) 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 reach an approximation for a class  $k$  example that

$$g(\mathbf{y}_i, \mathbf{f}_i^{[-i]}) - g(\mathbf{y}_i, \mathbf{f}_i) \approx \mathbf{D}g_i H_{ii} \Delta \mathbf{Y}_i \quad (4.70)$$

where  $H_{ii} = (k-1)K(\mathbf{x}_i, \mathbf{x}_i) \text{diag}(c_{i1}, \cdots, c_{i,k-1})$ . In general,  $\mathbf{D}g_i$ ,  $\Delta \mathbf{Y}_i$ , and  $H_{ii}$  are analogously defined for other class examples, and they are readily available once a minimizer  $\mathbf{f}_\lambda$  of  $I_\lambda(\mathbf{f}, \mathbf{y})$  is found. Averaging over the examples in the data set, we have

$$\begin{aligned} D(\lambda) &\approx \frac{1}{n} \sum_{i=1}^n \mathbf{D}g_i H_{ii} \Delta \mathbf{Y}_i \\ &= \frac{1}{n} \sum_{i=1}^n (k-1)K(\mathbf{x}_i, \mathbf{x}_i) \sum_{j=1}^k L_{ij} [f_j(\mathbf{x}_i) + \frac{1}{k-1}]_* c_{ij} (y_{ij} - \mu_{ij}(\mathbf{f})). \end{aligned}$$

Finally, the Generalized Approximate Cross Validation (GACV) for the multi-category SVM is given by

$$\begin{aligned} &GACV(\lambda) \\ &= \frac{1}{n} \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ \\ &+ \frac{1}{n} \sum_{i=1}^n (k-1)K(\mathbf{x}_i, \mathbf{x}_i) \sum_{j=1}^k L_{ij} [f_j(\mathbf{x}_i) + \frac{1}{k-1}]_* c_{ij} (y_{ij} - \mu_{ij}(\mathbf{f})). \end{aligned} \quad (4.71)$$

In a numerical point of view, the proposed GACV may be vulnerable to a small perturbation of solutions since it involves sensitive computations such as checking the condition  $f_j(\mathbf{x}_i) < -\frac{1}{k-1}$  or evaluating the step function  $[f_j(\mathbf{x}_i) + \frac{1}{k-1}]_*$ . To enhance the stability of GACV computation, we introduce a tolerance term  $\epsilon$ . The nominal condition  $f_j(\mathbf{x}_i) < -\frac{1}{k-1}$  is implemented as  $f_j(\mathbf{x}_i) < -\frac{1+\epsilon}{k-1}$ , and the step function  $[f_j(\mathbf{x}_i) + \frac{1}{k-1}]_*$  is replaced by  $[f_j(\mathbf{x}_i) + \frac{1+\epsilon}{k-1}]_*$  likewise. 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.

# Chapter 5

## Numerical Study

In this chapter, we illustrate the Multicategory Support Vector Machine (MSVM) through numerical examples. For empirical validation of its theoretical properties, we present simulated 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. In addition, a small scale benchmark study is conducted to test the utility of the multicategory SVM for real data sets. Throughout this chapter, we use the Gaussian kernel function,  $K(s, t) = \exp\left(-\frac{1}{2\sigma^2}\|s - t\|^2\right)$ , and  $\lambda$  and  $\sigma$  are searched over a grid. Since  $\sigma$  can be interpreted as a smoothing bound that controls the influence of neighboring samples to each sample, we set up a rule of thumb for an initial search range of  $\sigma$ . Beginning with the interval between the 10th percentile and the 90th percentile of within-class pairwise distances of the samples for  $2\sigma$ , we expand the initial interval, depending on evaluated tuning criteria. Often, searching beyond the rule of thumb upper bound was necessary.

### 5.1 Simulations

We demonstrate the effectiveness of the multicategory SVM through simulated examples. Additionally, we check the performance of the proposed tuning measure, GACV compared to the true target GCKL or more directly to the misclassification rate (MISRATE), treating all the misclassifications equally.

In the standard case,

$$\begin{aligned} & GCKL(\lambda, \sigma) \\ &= E_{true} \frac{1}{n} \sum_{i=1}^n L(\mathbf{Y}_i) \cdot (\mathbf{f}_{\lambda, \sigma}(\mathbf{x}_i) - \mathbf{Y}_i)_+ \end{aligned} \quad (5.1)$$

$$= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k \left( f_j(\mathbf{x}_i) + \frac{1}{k-1} \right)_+ (1 - p_j(\mathbf{x}_i)), \quad (5.2)$$

and

$$\begin{aligned} & \text{MISRATE}(\lambda, \sigma) \\ &= E_{\text{true}} \frac{1}{n} \sum_{i=1}^n L(\mathbf{Y}_i) \cdot \left( I(f_{i1} = \max_{1 \leq j \leq k} f_{ij}), \dots, I(f_{ik} = \max_{1 \leq j \leq k} f_{ij}) \right) \end{aligned} \quad (5.3)$$

$$= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k I(f_{ij} = \max_{1 \leq \ell \leq k} f_{i\ell}) (1 - p_j(\mathbf{x}_i)). \quad (5.4)$$

Note that they are available only in simulation settings. To see how good one can expect from data adaptive tuning procedures, we will generate a tuning set of the same size as the training set for each example below, and use the misclassification rate over the tuning set (TUNE), as a yardstick. More explicitly, let  $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$  and  $\{(\mathbf{x}_i^*, y_i^*), i = 1, \dots, n\}$  denote the training set and the tuning set, respectively. For comparison, we define

$$\begin{aligned} & \text{TUNE}(\lambda, \sigma) \\ &= \frac{1}{n} \sum_{i=1}^n L(\mathbf{y}_i^*) \cdot \left( I(f_1(\mathbf{x}_i^*) = \max_j f_j(\mathbf{x}_i^*)), \dots, I(f_k(\mathbf{x}_i^*) = \max_j f_j(\mathbf{x}_i^*)) \right) \end{aligned} \quad (5.5)$$

where  $(f_1, \dots, f_k)$  is the estimated MSVM from the training set. We investigate the inefficiency of each tuning criterion defined as the ratio of MISRATE at its minimizer to the minimum MISRATE. Thus, the inefficiency of a tuning criterion suggests how much misclassification would be incurred, relative to the smallest possible error rate by the MSVM if we know the underlying probabilities.

### 5.1.1 One dimensional three-class example

We considered a simple three-class example in which  $x$  lies in the unit interval  $[0, 1]$ . Let the conditional probabilities of each class given  $x$  be  $p_1(x) = 0.97 \exp(-3x)$ ,  $p_3(x) = \exp(-2.5(x - 1.2)^2)$ , and  $p_2(x) = 1 - p_1(x) - p_3(x)$ . They are shown in the top left panel of Figure 5.1. 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 subsequent three panels depict the true target function  $f_j(x)$ ,  $j = 1, 2, 3$  defined in Lemma 4.1 for this example. It assumes 1 when  $p_j(x)$  is maximum, and  $-1/2$  otherwise, whereas the target functions under one-versus-rest schemes are

$f_j(x) = \text{sign}(p_j(x) - 1/2)$ . 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.

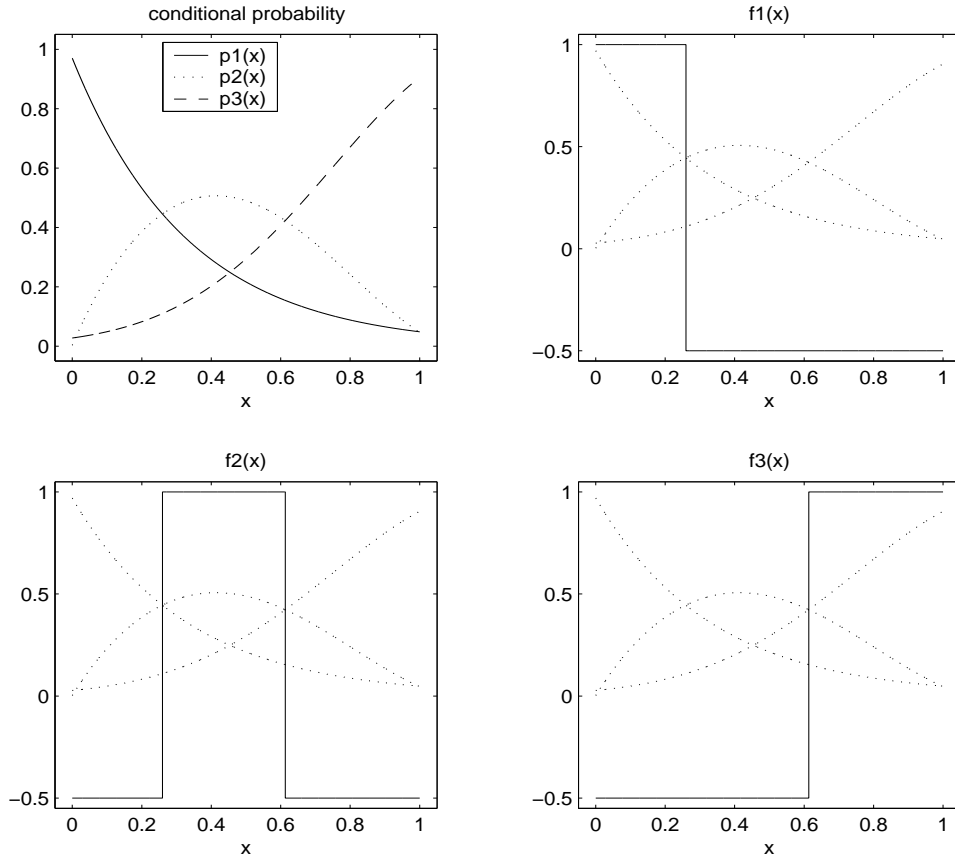


Figure 5.1: Conditional probabilities and multcategory SVM target functions for three-class example.

To compare the multcategory 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  $p_j(x)$ ,  $j = 1, 2, 3$ . The tuning parameters  $\lambda$ , and  $\sigma$  were jointly tuned to minimize GCKL distance of the estimate  $\mathbf{f}_{\lambda, \sigma}$  from the true distribution.

Figure 5.2 shows the estimated functions for both methods when 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 indeterminate region would

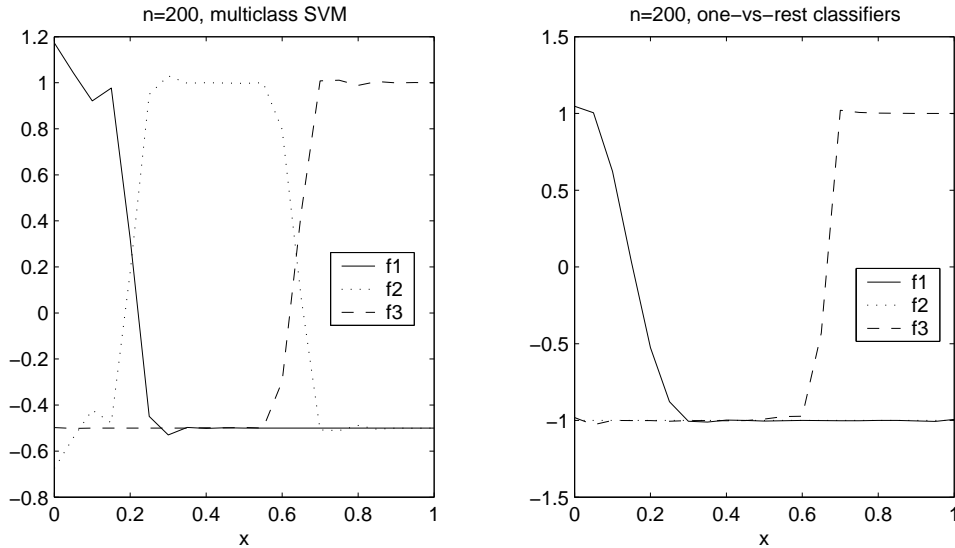


Figure 5.2: Comparison between the multicategory SVM and one-versus-rest method. The Gaussian kernel function is used, and the tuning parameters  $\lambda$ , and  $\sigma$  were simultaneously chosen via GCKL.

amplify the effectiveness of the proposed multicategory SVM. Table 5.1 shows the tuning parameters chosen by other tuning criteria alongside GCKL and their inefficiencies. As it is often observed in the binary case, GACV tends to pick bigger  $\lambda$  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, and it is a little worse than TUNE.

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 5.1: Tuning criteria and their inefficiencies (Example 1)

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

### 5.1.2 Two dimensional four-class example

The second example is a four-class problem in 2 dimensional input space. We generated uniform random vectors  $\mathbf{x}_i = (x_{i1}, x_{i2})$  on the unit square  $[0, 1]^2$ , and assigned class labels to each  $\mathbf{x}_i$  according to the following conditional probabilities:

$$\begin{aligned}
 p_1(x) &= C(x) \exp(-8[x_1^2 + (x_2 - 0.5)^2]) \\
 p_2(x) &= C(x) \exp(-8[(x_1 - 0.5)^2 + (x_2 - 1)^2]) \\
 p_3(x) &= C(x) \exp(-8[(x_1 - 1)^2 + (x_2 - 0.5)^2]) \\
 p_4(x) &= C(x) \exp(-8[(x_1 - 0.5)^2 + x_2^2])
 \end{aligned}$$

where  $C(x)$  is a normalizing function at  $x$  so that  $\sum_{j=1}^4 p_j(x) = 1$ . Note that four peaks of the conditional probabilities are at the middle points of the four sides of the unit square, and by symmetry the ideal classification boundaries are formed by two diagonal lines joining the opposite vertices of the unit square. Thus the target functions  $f_j$ ,  $j = 1, 2, 3, 4$  would be indicator-like functions that take 1 on the corresponding  $j$ th triangle and  $-1/3$  elsewhere. Figure 5.3 shows the estimated functions via multicategory SVM for a data set of size  $n = 300$ . The estimated classification boundaries derived from  $\hat{f}_j(x)$  are illustrated in Figure 5.4 together with the ideal classification boundary induced by the Bayes rule. Again the multicategory SVM was tuned via GCKL for this example.



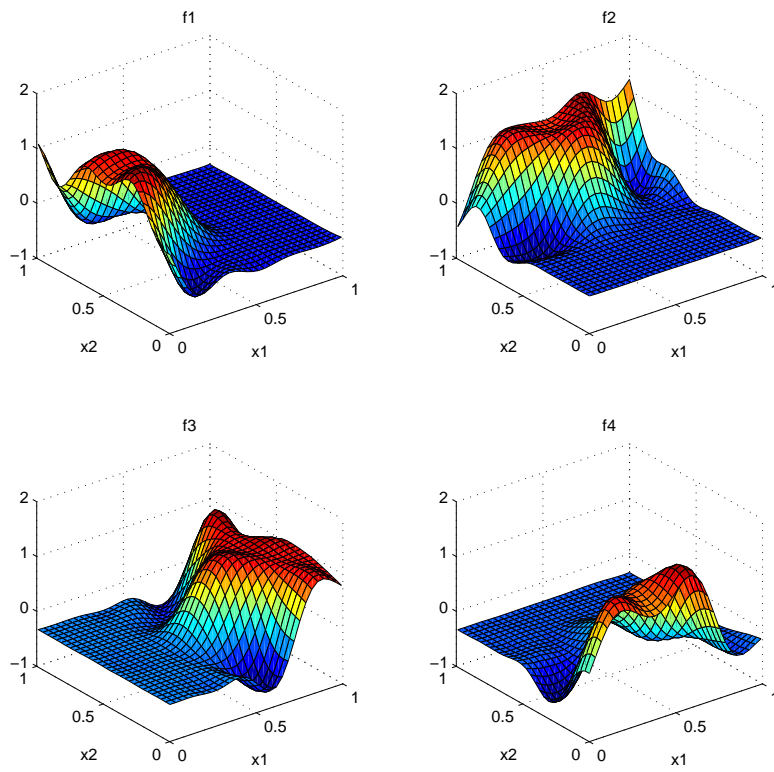


Figure 5.3: Estimated functions via the multiclass SVM for 2 dimensional example with sample size 300. The Gaussian kernel function was used, and the tuning parameters were fixed at the minimizer of GCKL.

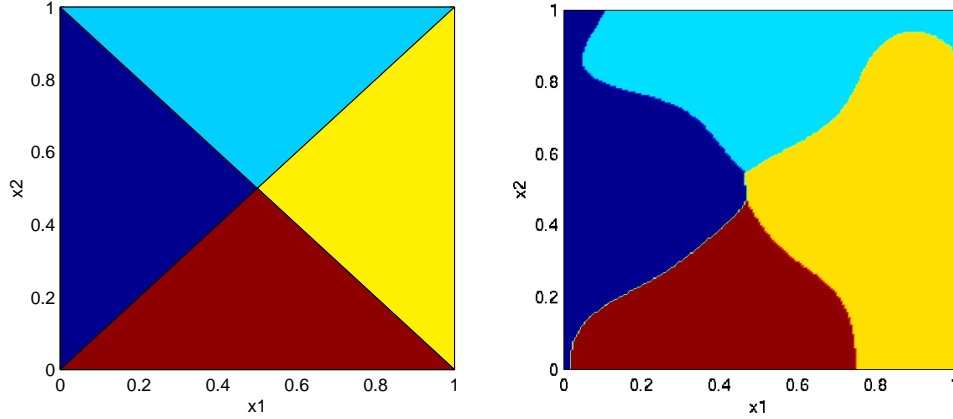


Figure 5.4: The classification boundaries determined by the Bayes rule (left) and the estimated classification boundaries by the multicategory SVM (right).

### 5.1.3 High dimensional six-category example

In theory, the Support Vector Machine can be applied to very high dimensional data without altering its formulation. Such capacity is well suited to data mining tasks and small  $n$  but large  $d$  structures like microarray data. To illustrate that the multicategory SVM formulation fits itself into a high dimensional scenario, we set up an example as follows. The dimension of the input data space,  $d$  is 80, and the number of categories,  $k$  is set to be 6. The data are basically produced by adding independent noise to an 80-dimensional mean vector for each class. Let each attribute  $X$  be  $(X_1, \dots, X_{80})$ . Then, the mean of  $X_i$ ,  $i = 1, \dots, 80$  is specified as  $E(X_i) = 0.5 \sin\left(\frac{2\pi}{r}(i + j - 1)\right)$  for class  $j = 1, \dots, 6$ .  $r$  governs the period of the sine curves, and we may vary  $r$  to control the degree of difficulty of the classification task. It was set at 35 in this example. Figure 5.5 depicts the mean sine curves for 6 classes. To generate the variables  $X_i$   $i = 1, \dots, 40$ , we added independent normal noise  $0.2Z$  to the mean for all the classes where  $Z$  stands for a standard normal variable. For the next quarter of variables  $X_i$ ,  $i = 41, \dots, 60$ , we added  $0.2T(5)$  as a noise where  $T(5)$  is a random variable for t-distribution with d.f. 5, and similarly for the last 20 variables, we used  $0.2T(10)$  as a noise added to the mean. This simulation setting hypothesized a situation that no single variable easily yields a clear class separation, but most of the variables contribute equally in making distinctions. We generated 60

samples from each class with the total sample size 360.

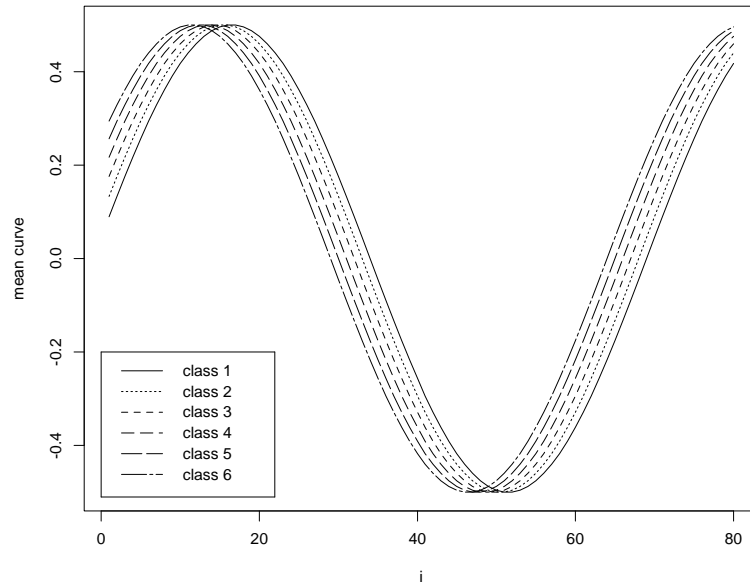


Figure 5.5: The mean sine curves for six classes

Table 5.2 shows the tuning results. Using the rule of thumb for a priori bound of  $\sigma$ , we searched from 0.2 to 0.65 by increment 0.05 on log base 2 scale. The range for  $\lambda$  was from -11 to -3 with increment 0.5. GACV picked larger  $\lambda$ , but smaller  $\sigma$  than those chosen via GCKL. In terms of the inefficiency, GACV is slightly worse than having an extra tuning set, but they are quite comparable. The Bayes error rate estimated by the Monte Carlo method from 3000 samples was 0.4367 with its standard deviation 0.0091. The MSVM error rate 0.5166 even when we tuned it using the true probabilities is not quite close to the Monte Carlo estimate of the Bayes rate, possibly due to the curse of dimensionality. A limited study in Hastie, Tibshirani and Friedman (2001, p. 384-385) showed that the presence of irrelevant variables degrades the accuracy of the Support Vector Machine, although our example is not this case. The effect of dimension reduction methods on the performance of the SVM was not investigated over this data set. However, it would be quite interesting to explore how the high dimensionality affects the SVM methodology in respect to its computational stability, the efficiency of tuning, and the consequent impact on its accuracy.

Table 5.2: Tuning criteria for the high dimensional data

Criterion	$(\log_2 \lambda, \log_2 \sigma)$	Inefficiency
MISRATE	(-7.5,0.3)	*
GCKL	(-9.5,0.3)	0.5232/0.5166=1.0128
TUNE	(-9.0,0.2)	0.5232/0.5166=1.0128
GACV	(-7.5,0.2)	0.5242/0.5166=1.0148

## 5.2 Small Scale Benchmark Study

We applied the multicategory SVM to some datasets from UCI Repository, <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Table 5.3 gives a short description of the data sets that we used. Two goals of this benchmark study were to investigate the performance of the MSVM in comparison with other methods and to see the effectiveness of GACV and  $k$ -fold cross validation, another commonly used tuning method in the absence of the true probabilities.

Table 5.3: Data sets from UCI Data Repository

Data set	sample size ( $n$ )	attributes ( $d$ )	classes ( $k$ )
wine	178	13	3
glass	214	9	6
waveform	300	21	3
vehicle	846	18	4

For brief comparison, we included three basic classification methods, LDA (linear discriminant analysis), QDA (quadratic discriminant analysis), and 1-NN (the nearest neighbor) together with the one-vs-rest SVMs. The Gaussian kernel was used for the MSVM and the one-vs-rest SVMs, so there are two parameters,  $\lambda$  and  $\sigma$  to be tuned. The estimated error rates of these classifiers over the four data sets are graphically summarized in Figure 5.6. The vertical distance of each tick mark from the bottom represents the misclassification rate of the method labeled beside for a given data set. Thus, the classifiers placed lower in the plot are the better ones.

**wine** Using the quantities of 13 constituents found in the three types of wines, the task is to predict the type of wines. To estimate the test error rates, we

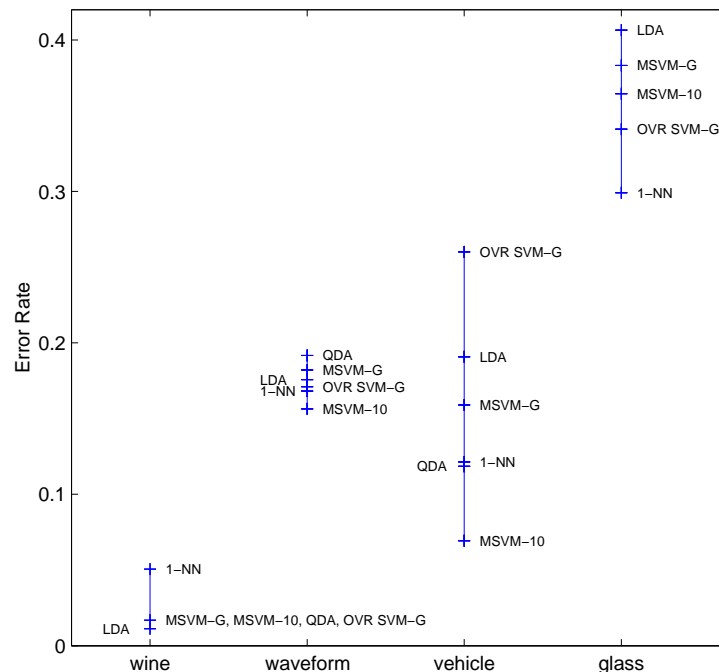


Figure 5.6: The error rates of various classifiers over the four data sets. (LDA: linear discriminant analysis, QDA: quadratic discriminant analysis, 1-NN: the nearest neighbor, MSVM-G: tuned via GACV, MSVM-10: tuned via 10-fold CV, and OVR SVM-G: the one-vs-rest SVMs tuned via GACV)

performed 10-fold cross validation. For 10 disjoint splits of the data set, each time the 90% data set was used as a training set for the classifiers, and then their error rates were estimated by the misclassification rates over the remaining 10% of the data set. The error rates in Figure 5.6 represent the average of the misclassification rates over 10-splits for **wine** data set. As we notice from the test error rates, this classification problem is not challenging. Many classifiers including the MSVMs, performed similarly over this data set. We observed that when the MSVM was tuned with the 10-fold cross validation method for each split, its tuning surface was pretty flat, again confirming that the problem is almost separable.

**glass** Based on 9 physical or chemical measurements of glasses, the problem

is to predict the type of glasses such as building window, vehicle window, tableware, and so on. The class frequencies in the data set are (70,76,17,13,9,29). To get the error rates, 10-fold cross validation was conducted likewise. The nearest neighbor method gave the smallest error rate while the MSVMs tuned via either GACV or 10-fold CV performed slightly worse than the one-vs-rest binary SVMs. QDA could not be used due to the rank deficiency of the sample covariance matrix.

**waveform** As a simulated data, each class was generated from a combination of 2 of 3 “base” waves. For details, see Breiman, Friedman, Olshen and Stone (1984, p. 49-55). The optimal Bayes classification rule has 86% accuracy. Using the test set of size 4700, the error rates of the classifiers were simply estimated by their misclassification rates over the test set. The MSVM tuned by 10-fold CV was the best with its accuracy 84.36 % close to that of the Bayes rule. The one-vs-rest binary SVMs were slightly better than the GACV tuned MSVM.

**vehicle** Given a set of features extracted from a silhouette, we classify each silhouette as one of four types of vehicle. Randomly selected 500 samples were used as a training set and the error rates were evaluated over the remaining 346 samples as a test set. The MSVM tuned via 10-fold CV gave the smallest error rate followed by QDA and 1-NN. Using binary SVMs in the one-vs-rest fashion resulted in the biggest error rate.

In our experiment, we neither screened attributes nor considered any transformations of them. We poured all the attributes to learning procedures after standardization. So, there may still exist a room for improvement by more sophisticated analyses. Overall performance of the MSVM was pretty comparable to other methods, and when it was tuned via 10-fold CV, it could outperform other methods over two data sets. As a tuning method, GACV seems effective when the problem is almost separable, so the misclassification count based methods give a flat tuning surface. However, with considerable overlaps between classes, we empirically observe that GACV tends to oversmooth and result in a little bigger error rate than  $k$ -fold CV. Advantages of the MSVM over the one-vs-rest approach appear to vary depending on data sets.

# Chapter 6

## Applications

Two applications to problems arising in the oncology and the meteorology are presented. One application is cancer classification using microarray data and the other is cloud detection and classification via satellite radiance profiles. Specific issues pertaining to each application will be addressed in the corresponding subsection.

### 6.1 Cancer Classification with Microarray Data

#### 6.1.1 Introduction

The advent of microarray gene expression technology has opened the possibility of investigating the activity of thousands of genes simultaneously. Gene expression profiles are the measurements of relative abundance of mRNA corresponding to the genes. Since transcriptional changes sensibly reflect the status of disease including cancers, gene expression profiles can be used to classify the different types of cancers accurately. See DeRisi, Penland, Brown, Bittner, Meltzer, Ray, Chen, Su and Trent (1996), Zhang, Zhou, Velculescu, Kern, Hruban, Hamilton, Vogelstein and Kinzler (1997), Perou, Jeffrey, van de Rijn, Rees, Eisen, Ross, Pergamenschikov, Williams, Zhu, Lee, Lashkari, Shalon, Brown and Botstein (1999), Schummer, Ng, Bumgarner, Nelson, Schummer, Bednarski, Hassell, Baldwin, Karlan and Hood (1999), and Jiang, Harlocker, Molesh, Dillon, Stolk, Houghton, Repasky, Badaro, Reed and Xu (2002) for reference. Currently, cancer diagnosis highly depends on a variety of histological observations, which have limitations due to morphological similarity. Accurate diagnosis promotes the efficacy of a proper treatment of cancers. 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 in Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing, Caligiuri, Bloomfield and Lander (1999), Mukherjee, Tamayo, Slonim, Verri, Golub, Mesirov and Poggio (1999), Dudoit, Fridlyand and Speed (2000), Furey, Cristianini, Duffy, Bednarski, Schummer and Haussler (2000), Khan,

Wei, Ringner, Saal, Ladanyi, Westermann, Berthold, Schwab, Atonescu, Peterson and Meltzer (2001), Yeo and Poggio (2001), and references therein. Typical microarray training data sets (a set of pairs of a gene expression profile and the tumor type 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 (SVM) 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. Before demonstrating the usage of the MSVM in a particular cancer classification problem using gene expression profiles, we discuss how to assess the strength of prediction made by the MSVM and some heuristics to reject weak predictions first. Attaching a confidence statement to each prediction may be important for medical diagnosis since making a wrong prediction could be more serious than reserving a call. For weakly diagnosed examples, getting further information from a specialized investigation or expert opinion would be an appropriate procedure for a more informative call.

### 6.1.2 Assessment of Prediction Strength

By measuring the strength or the confidence of a class prediction made by Support Vector Machines, we wish to reject any prediction weaker than a specified threshold. For classification methods with their ultimate output being the estimated conditional probability of each class at  $\mathbf{x}$ , we can simply set a threshold such that the classification is made only when the estimated probability of the predicted class exceeds the threshold. Whereas, SVMs target the representation of the most probable class itself without any probability estimate when flexible kernel functions are used. Linear SVMs do not provide probability estimates, either. The mechanism of the Support Vector Machine to extract the necessary information for the minimum error rate seems very simple and efficient, but inevitably limited in restoring the probability from the estimated class code.

Nevertheless, there have been a couple of empirical approaches to address this issue for SVMs in the binary case, and solving a series of binary SVMs in the multiclass case. The idea is based on that the SVM decision functions diminish in magnitude near classification boundaries by their geometrical formulation in the input space. Naturally, one may consider a confidence measure based on the SVM decision function,  $f(\mathbf{x})$  at  $\mathbf{x}$ ; the bigger  $f(\mathbf{x})$  in the absolute value, the stronger the prediction. Mukherjee et al. (1999) proposed a confidence



measure for an SVM output  $f(\mathbf{x})$  and its induced class prediction with this notion. Assuming that  $P(Y|X = \mathbf{x}) \approx P(Y|f(X) = f(\mathbf{x}))$ ,  $P(Y = 1) = P(Y = -1)$  and  $P(f|Y = 1) = P(-f|Y = -1)$ , they asserted that a confidence of the SVM prediction  $f$  can be quantified using the relation that  $P(Y|f) \propto P(f|Y)P(Y)$ . For the estimation of  $P(f|Y)$ , they used leave-one-out estimates of  $f$  values from the training data set, along with the class label  $y$  of each example left out. However, for almost separable classification problems, the proposed computations can not be done properly due to the complete or quasi-complete separation. Hence, they heuristically defined the confidence level of an SVM prediction  $f$  as  $1 - \hat{F}(|f|)$  at the end of their applications, using the symmetry assumption that  $P(f|Y = 1) = P(-f|Y = -1)$ . Here,  $\hat{F}$  is the estimated cumulative distribution function of SVM outputs  $|f|$ . This heuristic measure implicitly assumes that the probability of a correct prediction given  $f$  depends only on the margin  $\text{sign}(f) \cdot f = |f|$  and realizes the initial notion that the bigger the margin  $|f|$ , the stronger the prediction. So, the confidence for an SVM output  $f$  can be interpreted as the proportion of SVM predictions stronger than  $f$ . The proportion can be inferred from jackknife (LOO) estimates or any variants of cross validation of training samples. The confidence level seems to be a misnomer in that the smaller the confidence level, the stronger the prediction. In their application, it was limited to at most 95% to allow a class prediction.

Here is a simple variant of the method for MSVMs. The MSVM output is a vector of the decision functions  $(f_1, \dots, f_k)$  evaluated at  $\mathbf{x}$ . A decision vector close to a class code in the multiclass case may mean a strong prediction away from the classification boundary. The multiclass hinge loss with the standard cost function  $L(\cdot)$ ,  $g(\mathbf{y}, \mathbf{f}) \equiv L(\mathbf{y}) \cdot (\mathbf{f} - \mathbf{y})_+$  sensibly measures the proximity between an MSVM decision vector and a coded class, reflecting how strong their association is in the classification context. It considers the sign and the magnitude of each coordinate of a decision vector simultaneously. 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 without causing much confusion. Recall that given an MSVM decision vector  $(f_1, \dots, f_k)$ , the predicted class is  $\arg \max_j f_j$ . Analogous to the binary case, we assume that the probability of a correct prediction given  $\mathbf{f}(\mathbf{x}) = (f_1, \dots, f_k)$  at  $\mathbf{x}$ ,  $P(Y = \arg \max_j f_j | \mathbf{f})$  depends on  $\mathbf{f}$  only through the multiclass hinge loss,  $g(\arg \max_j f_j, \mathbf{f})$  for the predicted class. Now, the smaller the hinge loss, the stronger the prediction. Then the strength of the MSVM prediction,  $P(Y = \arg \max_j f_j | \mathbf{f})$  can be inferred from the training data similarly by cross validation. For example, leaving out the  $i$ th example  $(\mathbf{x}_i, y_i)$ , we get the MSVM decision vector  $\mathbf{f}(\mathbf{x}_i) = (f_1, \dots, f_k)$  at  $\mathbf{x}_i$  based on the remaining

samples. 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  $I(y_i = \arg \max_j f_j(\mathbf{x}_i))$ , and repeat this calculation marching through the samples in the training data set.  $P(Y = \arg \max_j f_j | \mathbf{f})$ , as a function of  $g(\arg \max_j f_j, \mathbf{f})$  can be estimated then from the collection of pairs of the hinge loss and the indicator. If we further assume the complete symmetry of  $k$  classes, that is,  $P(Y = 1) = \dots = P(Y = k)$  and  $P(\mathbf{f} | Y = y) = P(\pi(\mathbf{f}) | Y = \pi(y))$  for any permutation operator  $\pi$  of  $\{1, \dots, k\}$ , it follows that  $P(Y = \arg \max_j f_j | \mathbf{f}) = P(Y = \pi(\arg \max_j f_j) | \pi(\mathbf{f}))$ . 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. Again, we can apply similar 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 will be exercised in the following application.

The second approach to reject a prediction by SVMs naturally arises in solving multiclass problems by binary classifiers in the one-vs-rest fashion. Breaking a multiclass problem into a series of unrelated binary problems is apt to yield unresolved calls such as non-membership prediction (it does not belong to any of the known classes) and conflicting prediction (it falls into more than one class). Though the possibility of having unresolved calls may not be desirable in general, such indecisive prediction does mean a weak call subject to rejection. Yeo and Poggio (2001) successfully demonstrated the idea of rejecting the two kinds of predictions in a tumor classification problem. It is worth mentioning that the population version of this approach is equivalent to making a prediction only when the predicted class has more than a 50% chance of being correct. If a more stringent classification rule is necessary in making prediction for unseen examples, then one can use the nonstandard binary SVMs in Lin et al. (2002) which adjust the costs for two different types of misclassification to achieve a required accuracy.

### 6.1.3 Small Round Blue Cell Tumors of Childhood

We revisit the small round blue cell tumors (SRBCTs) of childhood data set in Khan et al. (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

Table 6.1: Class distribution of SRBCTs data set

Data set	NB	RMS	BL	EWS	total
Training set	12	20	8	23	63
Test set	6	5	3	6	20
Total	18	25	11	29	83

(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 samples falling into 4 categories each, while the test set contains 20 SRBCT samples and 5 non SRBCTs (2 normal muscle tissues and 3 cell lines including an undifferentiated sarcoma, osteosarcoma, and a prostate carcinoma). Table 6.1 shows the distribution of the four distinct tumor categories in the training set and the test set. 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 ( $k$ NN), 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 5 to 100 genes (features) were used. For the best results shown in the paper, perfect classification was possible in testing the blind 20 samples as well as in cross validating 63 training samples. Since the one-vs-rest scheme needs four binary classifiers in this problem, the maximum number of distinct features used in learning a complete classification rule is four times the number of features for each binary classifier.

For comparison, we applied the MSVM to the problem after taking the logarithm base 10 of the expression levels and standardizing arrays. Finding the best subset of genes out of 2308 would be combinatorically formidable as a variable selection problem. Instead, the marginal relevance of each gene in class separation was evaluated, following a simple criterion used in Dudoit et al. (2000). For gene  $\ell$ , we define the ratio of between classes sum of squares to

within class sum of squares as its relevance measure;

$$\frac{BSS(\ell)}{WSS(\ell)} = \frac{\sum_{i=1}^n \sum_{j=1}^k I(y_i = j) (\bar{x}_{\cdot\ell}^{(j)} - \bar{x}_{\cdot\ell})^2}{\sum_{i=1}^n \sum_{j=1}^k I(y_i = j) (x_{i\ell} - \bar{x}_{\cdot\ell}^{(j)})^2} \quad (6.1)$$

where  $n$  is the training sample size,  $\bar{x}_{\cdot\ell}^{(j)}$  indicates the average expression level of gene  $\ell$  for class  $j$  samples, and  $\bar{x}_{\cdot\ell}$  is the overall mean expression levels of gene  $\ell$  in the training set. We select genes with the largest ratios. Table 6.2 shows the list of top 20 genes and their functions for the SRBCTs data.

Table 6.3 is a summary of the classification results by MSVMs with the Gaussian kernel function. Though the previous studies showed that linear classifiers are good enough to achieve almost perfect classification, we find that flexible basis functions such as the Gaussian kernel are particularly effective for multiclass problems. The classification results with the linear kernel function are not shown in the table, but we observed that linear MSVMs achieve similar performances as Gaussian MSVMs although their evaluated decision vectors are less specific to the class representation than those of the Gaussian kernel. The second column indicates the optimal tuning parameters pair  $\lambda$  and  $\sigma$  on log 2 scale chosen by the GACV tuning measure (4.71). In fact, the LOOCV tuning error as a function of the tuning parameters was zero at multiple minima. The phenomenon that LOOCV tuning error has multiple minima while the multiple minima include the optimal tuning parameters given by GACV was observed in this experiment as well. The zero LOOCV tuning errors imply that the classification task is not challenging at all. The number of Support Vectors (SVs) in the third columns indicates how many samples out of 63 have nonzero coefficients in the expression of the solution (4.14). Removing non Support Vectors does not change the solution, and the number of SVs is related to the fraction of the training data near the classification boundary induced by the SVM. We observe from the table that a large number of features involved tend to produce a large number of SVs. It seems due to the sparsity of the data in a high dimensional space. 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 fourth 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 0 to 3 depending on tuning measures. The MSVM tuned by GACV gives 3 test errors while LOOCV tuning gives 0

Table 6.2: 20 genes with the largest ratios in SRBCT data set

Id	Gene	Description
770394	<i>Fc fragment of IgG, receptor, transporter, alpha</i>	Regulate serum IgG level.
796258	<i>Sarcoglycan, alpha (50 kDa dystrophin-associated glycoprotein)</i>	Structure.
784224	<i>Fibroblast growth factor receptor 4</i>	Bind both acidic and basic FGF.
814260	<i>Follicular lymphoma variant translocation 1 (FVT-1)</i>	
295985	Unknown	Unknown.
377461	<i>Caveolin 1, caveolae protein, 22 kDa</i>	Structural component of caveolae.
859359	<i>Quinone oxidoreductase homolog</i>	Metabolism.
769716	<i>Neurofibromin 2 (bilateral acoustic neuroma)</i>	Possible tumor suppressor <sup>1</sup> .
365826	<i>Growth arrest-specific 1</i>	Growth regulation.
1435862	<i>MIC2 (CD99)</i>	Transmembrane glycoprotein and tumor marker <sup>2</sup> .
866702	<i>Protein tyrosine phosphatase, non-receptor type 13 (APO-1/CD95 (Fas)-associated phosphatase)</i>	May involve in Fas-mediated apoptosis.
296448	<i>Insulin-like growth factor 2 (somatomedin A)</i>	Growth regulation.
740604	<i>Interferon stimulated gene (20 kDa) (ISG20)</i>	
241412	<i>E74-like factor 1 (ets domain transcription factor) (Elf-1)</i>	Ets family transcription factor.
810057	<i>Cold shock domain protein A</i>	Probable transcriptional factor for Y-box.
244618	Unknown	Unknown.
52076	<i>Olfactomedin related ER localized protein</i>	
21652	<i>Catenin (cadherin-associated protein), alpha 1 (102 kDa)</i>	Structure.
43733	<i>Glycogenin 2 (GYG2)</i>	lycogen synthesis.
236282	<i>Wiskott-Aldrich syndrome protein (WASP)</i>	Related to X-linked immunodeficiency.

<sup>1</sup> Zhu and Parada (2001), <sup>2</sup> Weidner and Tjoe (1994), Ramani, Rampling and Link (1993), and Fellingner, Garin-Chesa, Glasser, Huvos and Rettig (1992).

Table 6.3: LOOCV error and Test error for SRBCT data set. MSVMs with the Gaussian kernel are applied to the training data set. The second column indicates the optimal tuning parameters pair,  $\lambda$  and  $\sigma$  on log 2 scale chosen by the GACV. The third column presents the number of Support Vectors in the final solution of the MSVM with the number of genes specified as in the first column. The last row shows the results by using only three principal components (PCs) from 100 genes.

Number of genes	$(\log_2 \lambda, \log_2 \sigma)$	no. of SVs	LOOCV error	Test error
20	(-22,1.4)	45	0	0
60	(-23,2.4)	63	0	0
100	(-23,2.6)	58	0	0
all	(-25,4.8)	63	1	0 to 3
3 PCs (100)	(-19,1.6)	22	0	0

to 3 test errors.

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. The main obstacle of analyzing high dimensional data like gene expression data is that we are not capable of visualizing the raw data in their original space, and consequently it is hard to make judicious calls in fitting and assessing models. However, such high dimensional data oftentimes reside in a low dimensional subspace. Using dimension reduction techniques such as the principal component analysis, we can visualize the data approximately in a much lower dimension than that of the original space. Figure 6.1 displays the three principal components of the top 100 genes in the training set as circles. Squares represent the corresponding three principal coordinates of the test set when we apply the linear combinations obtained from the training set to the test samples. Different colors identify four different tumor types; EWS in blue, BL in purple, NB in red, RMS in green, and non SRBCT in cyan. Notice that the principal coordinates of 5 non SRBCTs in the test set land on ‘no man’s land’, encircled by the samples from the four known classes. It clearly shows that three linear combinations of the 100 gene expression profiles are informative enough to differentiate 4 tumor types. 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.

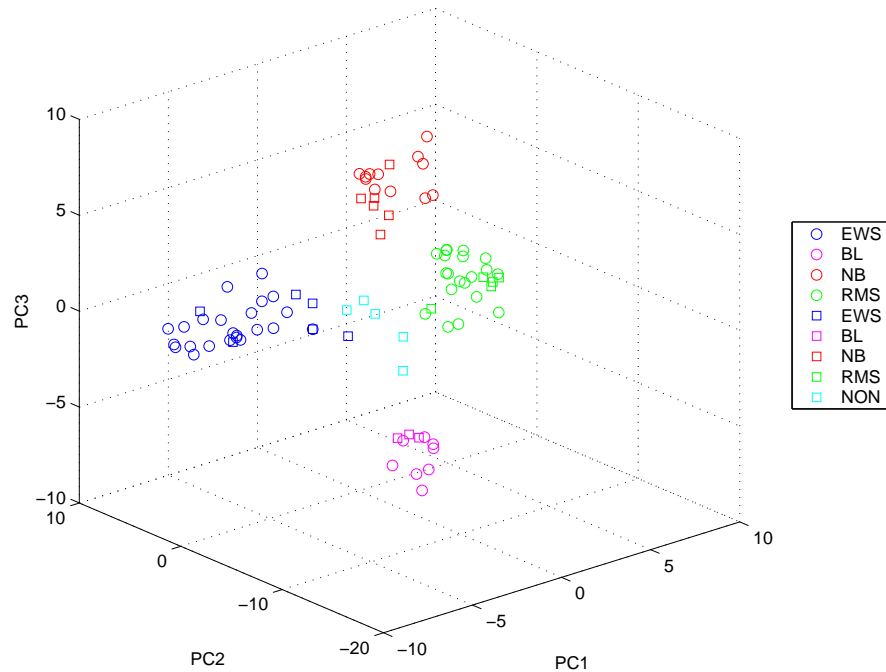


Figure 6.1: Three principal components of 100 gene expression levels in the training set are plotted as circles. The squares represent the corresponding principal coordinates of the test samples including non SRBCT samples. The tumor types are distinguished by colors (EWS: blue, BL: purple, NB: red, RMS: green, and non SRBCT: cyan). We can see a nice separation of the four tumor types through three principal components. Non SRBCT samples lie amid four-class samples.

With the three principal components (PCs) only, we applied the MSVM, and the corresponding classification result is in the last row of Table 6.3. Again, perfect classification was achieved in cross validating and testing. Indeed, the zero test error is no surprise from the picture, and we have checked that QDA (quadratic discriminant analysis), a simple traditional method, which could not be applied when the dimension of input space exceeds the sample size, gives the same zero test error once the data are represented by three PCs. Other benefit of the dimension reduction for MSVMs is that the number of SVs is noticeably reduced to about a third of the training sample size. Figure 6.2 shows the predicted decision vectors  $(f_1, f_2, f_3, f_4)$  at the test samples. The four class labels

are coded according as EWS:  $(1, -1/3, -1/3, -1/3)$ , BL:  $(-1/3, 1, -1/3, -1/3)$ , NB:  $(-1/3, -1/3, 1, -1/3)$ , and RMS:  $(-1/3, -1/3, -1/3, 1)$ . We follow the color scheme in Figure 6.1, to indicate the true class identities of the test samples. For example, the blue bars correspond to EWS samples, and the ideal decision vector  $(f_1, f_2, f_3, f_4)$  for them is  $(1, -1/3, -1/3, -1/3)$ . The estimated decision vectors are pretty close to the ideal representation and their maximum components are the first one, meaning correct classification. We can see from the plot that all the 20 test examples from 4 classes are classified correctly. Note that the test examples are rearranged in the order of EWS, BL, NB, RMS, and non SRBCT, so the horizontal coordinates do not match with the test id's given in the original data set. In the test data set, there are 5 non SRBCT samples (2 normal muscle tissues and 3 cell lines). The fitted MSVM decision vectors for the 5 samples are plotted in cyan color in Figure 6.2. In clinical settings, it is important to be able to reject classification whenever samples not falling into the known classes are given. Now, we demonstrate that the MSVM predictions are specific enough to identify the peculiarity of the 5 non SRBCTs. The multi-class hinge loss with the standard cost function  $L$  is used to measure the MSVM prediction strength at unseen examples, as described before. The last panel in Figure 6.2 depicts the loss for the predicted MSVM decision vector at each test sample including 5 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 5 non SRBCTs all exceed the threshold, while 3 test samples out of 20 can not be classified confidently by thresholding.

Overall, comparable to alternative methods, the MSVM method appears to achieve perfect or near perfect classification for cancer diagnosis problems using microarray data. We believe it has a great potential for such medical diagnosis problems. For another MSVM application to the leukemia data set, see Lee and Lee (2002). The tumor diagnosis problems using gene expression profiles available so far are observed to be very separable and not a challenging task once the dimension of the input space is reduced. This implies that gene expression profiles are informative enough to differentiate several tumor types. If this is a prevalent characteristic of the cancer diagnosis problem with gene expressions, then the accuracy of any reasonable classifier may not be significantly different. Differences, if any, will get evident as we accumulate more information on this kind of data. Still, there are certain advantages of flexible classifiers. The Support Vector Machine, for one, is often advocated not only for its accuracy



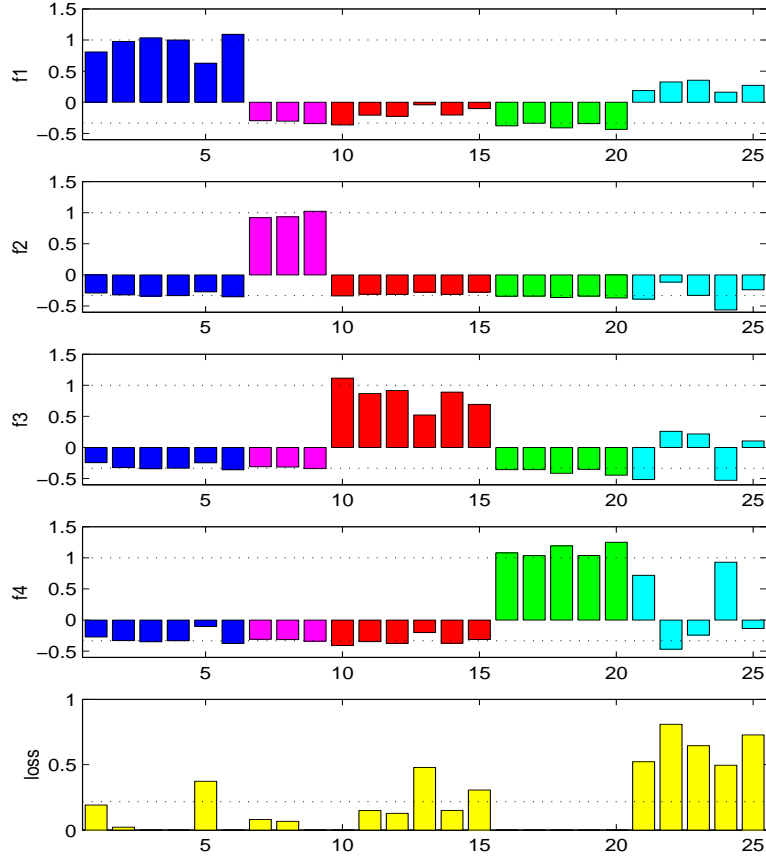


Figure 6.2: The first four panels show the predicted decision vectors  $(f_1, f_2, f_3, f_4)$  at the test samples. The four class labels are coded according as EWS in blue:  $(1, -1/3, -1/3, -1/3)$ , BL in purple:  $(-1/3, 1, -1/3, -1/3)$ , NB in red:  $(-1/3, -1/3, 1, -1/3)$ , and RMS in green:  $(-1/3, -1/3, -1/3, 1)$ . The colors indicate the true class identities of the test samples. We can see from the plot that all the 20 test examples from 4 classes are classified correctly and the estimated decision vectors are pretty close to their ideal class representation. The fitted MSVM decision vectors for the 5 non SRBCT samples are plotted in cyan. The last panel depicts the loss for the predicted decision vector at each test sample. The last 5 losses corresponding to the predictions of non SRBCTs all exceed the threshold (the dotted line) below which means a strong prediction. Three test samples falling into the known four classes can not be classified confidently by the same threshold.

but also its versatile formulation to handle high dimensional data. However, a caveat is that it is not completely free from the curse of dimensionality either. Not only for the sake of the parsimony, dimension reduction methods including gene selection, therefore will be indispensable to improve the accuracy.

## 6.2 Cloud Classification with Radiance profiles

### 6.2.1 Introduction

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 1 to 2 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 MODIS cloud mask algorithm with a simple threshold technique, see Strabala, Ackerman and Menzel (1994) and Ackerman, Strabala, Menzel, Frey, Moeller and Gumley (1998). We illustrate the potential of the multicategory SVM as an efficient cloud detection algorithm. We have applied the MSVM to simulated MODIS type channels data to classify the radiance profiles as clear, liquid clouds, or ice clouds.

### 6.2.2 Data Description

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. Total 744 radiance profiles over the ocean (81 clear scenes, 202 liquid clouds and 461 ice clouds) are given in the data set. Each simulated radiance profile consists of 7 reflectances at .66, .86, .46, .55, 1.2, 1.6, 2.1 microns, and 5 brightness temperatures at 6.6, 7.3, 8.6, 11, 12 microns.

Figure 6.3 shows boxplots of the reflectances and the brightness temperatures along 12 radiance channels for each type. Generally, clouds are characterized by higher reflectance and lower temperature than the underlying Earth surface.

The boxplots confirm this general characteristic of clouds compared to clear sky. Here, we use the abbreviations of  $R$  and  $BT$  for the reflectance and the brightness temperature. The top panels show the profiles of clear scenes, the middle panels for water clouds and the bottom panels for ice clouds. No single channel seems to give a clear separation of the three categories. To see the individual profile patterns, 10 profiles were randomly selected from each category and illustrated in Figure 6.4. Clear sky profiles are in blue, water clouds are in green, and ice clouds are in purple. Again, we observe a fair amount of overlaps in the profiles among the three types. Figure 6.5 displays scatterplots of some features (either variable or transformation of variables) of interest, which have been used conventionally to distinguish the scenes. They are deduced from domain knowledge of the physics underlying weather phenomena. The scatterplot of  $BT_{channel_{31}}$  versus  $BT_{channel_{32}} - BT_{channel_{29}}$  is in the top left, while the pairs of  $R_{channel_1}/R_{channel_2}$  and  $R_{channel_2}$  are in the top right. Although the features in the top two panels are partially effective in distinguishing the three types of scenes,  $R_{channel_2}$  and  $\log_{10}(R_{channel_5}/R_{channel_6})$  in the bottom left panel appear to be most informative.

### 6.2.3 Analysis

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 samples, almost half of the original data were selected randomly from the bottom left panel in Figure 6.5 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 samples was 11.5% (= 43/374). Figure 6.6 shows the classification boundaries. Most of the misclassifications occurred due to the considerable overlap between ice clouds and clear sky samples at the lower left corner of the plot. Table 6.4 shows the cross tabulation of the predicted category based on the classifier over the test set. QDA (quadratic discriminant analysis) gave the misclassification rate 14.44% (=54/374), while the test error rate of LDA (linear discriminant analysis), which assumes the equal covariance across the classes, was 36.1%. Such low accuracy of LDA is expected since the covariance structures for the three types seem to be heterogeneous unarguably from the plot. It turned out that adding three more features in Figure 6.5 to the MSVM did not improve the classification accuracy significantly. We could classify correctly just 5 more examples than the two features only case with the misclassification rate 10.16% (=38/374).

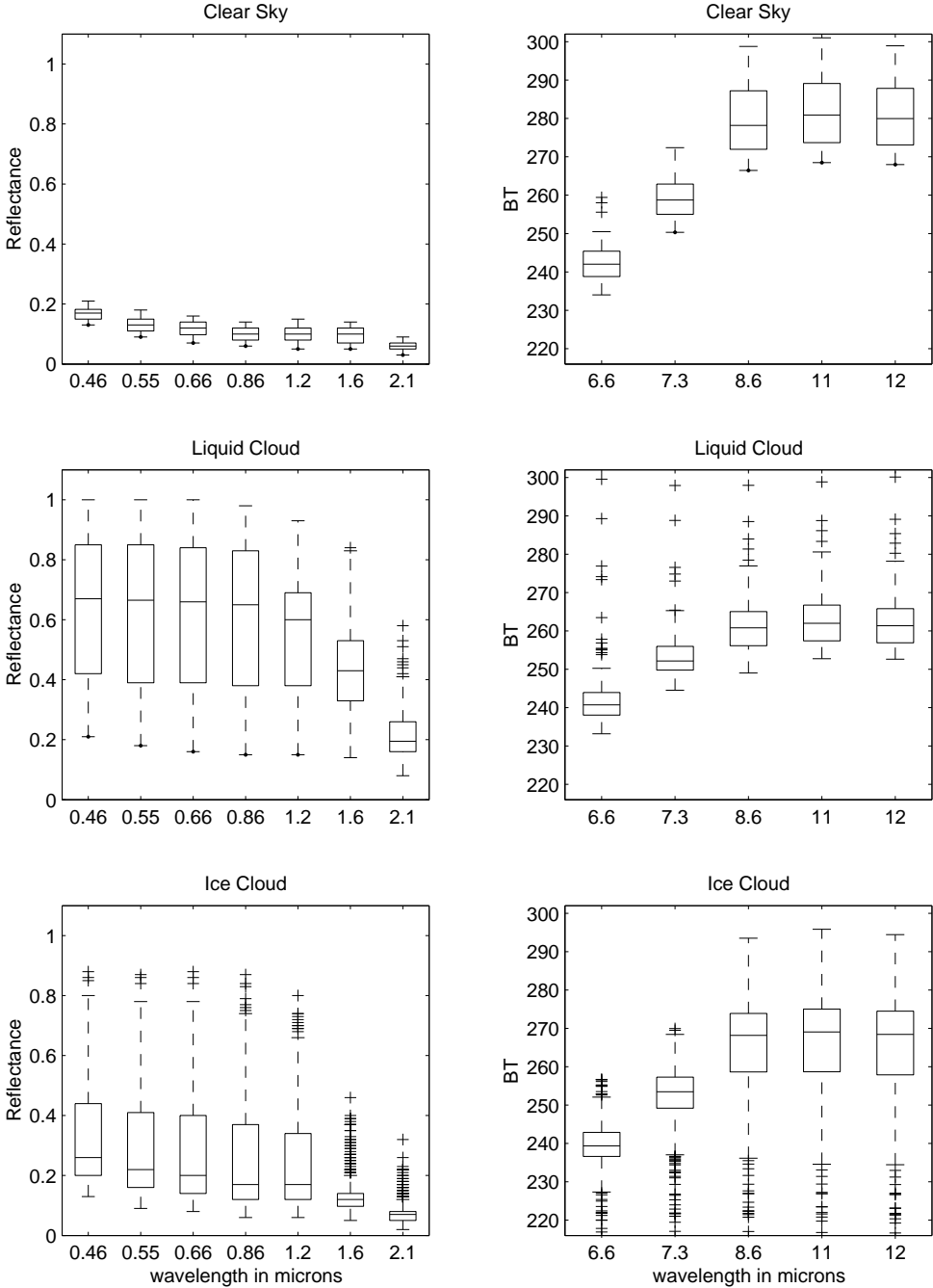


Figure 6.3: The boxplots of 7 reflectances and 5 brightness temperatures for clear sky, liquid clouds, and ice clouds over the ocean.

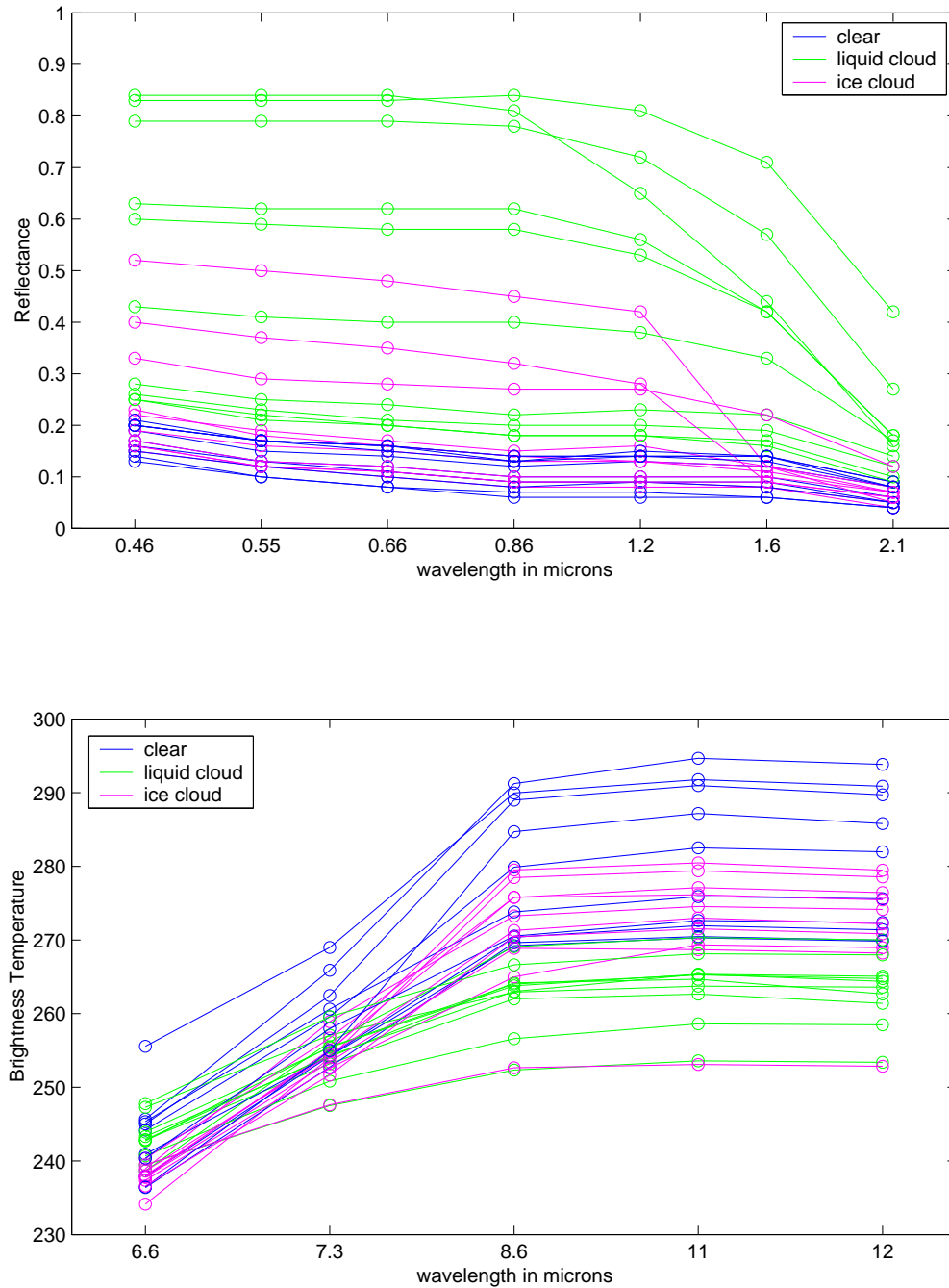


Figure 6.4: We selected 10 radiance profiles at random from the three scenes. The top panel displays the reflectance profiles of the 10 random samples for each category, and the bottom panel shows the brightness temperature profiles (clear sky: blue, liquid clouds: green, ice clouds: purple).

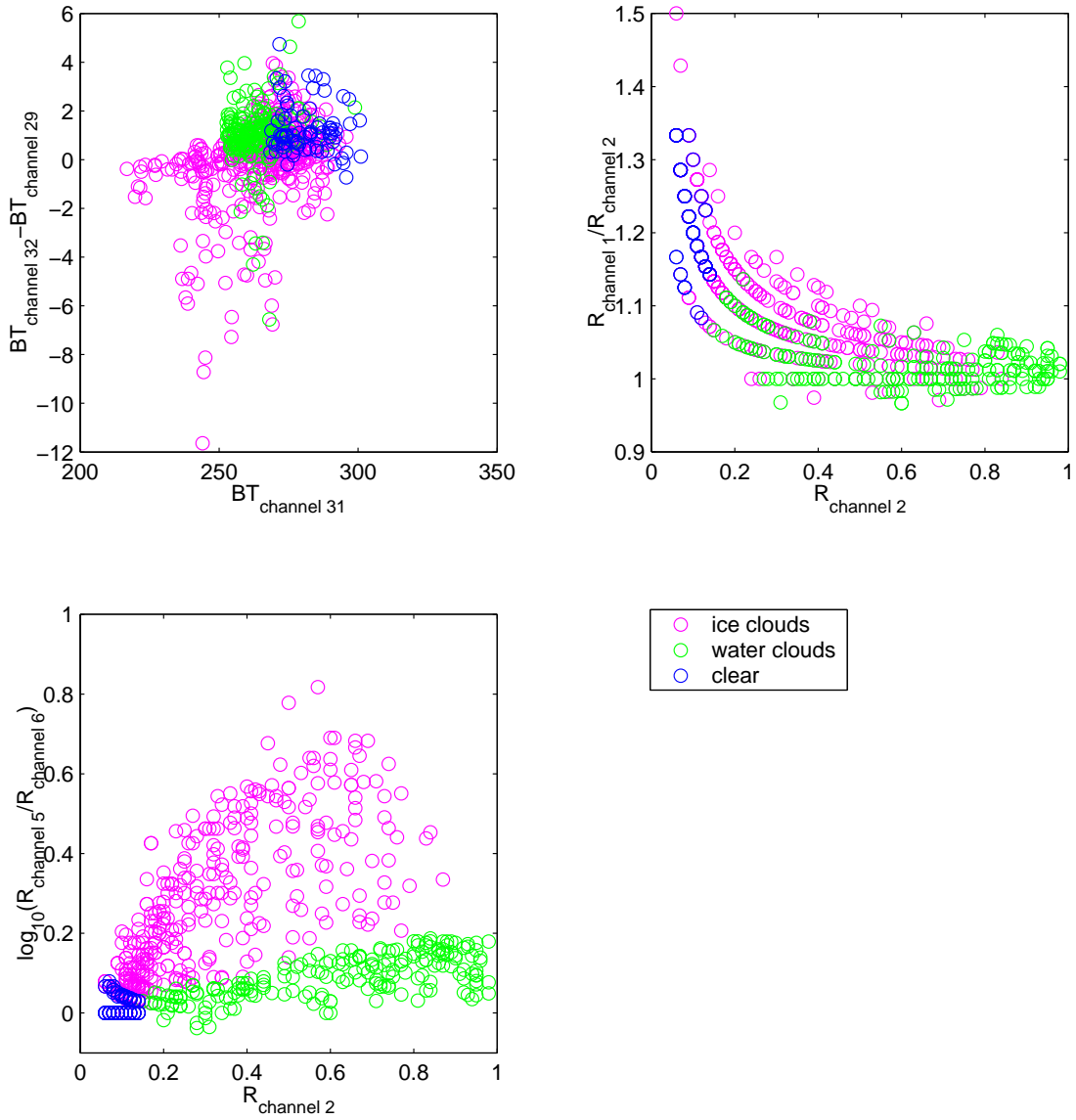


Figure 6.5: Scatterplots of  $BT_{channel\ 31}$  vs  $BT_{channel\ 32} - BT_{channel\ 29}$  (top left),  $R_{channel\ 1}/R_{channel\ 2}$  vs  $R_{channel\ 2}$  (top right), and  $R_{channel\ 2}$  vs  $\log_{10}(R_{channel\ 5}/R_{channel\ 6})$  (bottom left).

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 2 or 5 features. However, interestingly QDA with the original 12 channels gave a reduced test error rate of 9.36%. Not conclusive but very likely account for this is that the data gen-

Table 6.4: Distribution of the predicted class based on the MSVM with two features

True category	Predicted category			total
	clear sky	water clouds	ice clouds	
clear sky	18	0	23	41
water clouds	0	100	2	102
ice clouds	14	4	213	231

eration procedure involved extensive use of the normal distributions, for which QDA would be the theoretically optimal and efficient classifier. When all the variables are transformed by the logarithm function, the MSVM achieved its minimum error rate while the QDA returned one additional misclassification. All the results are summarized in Table 6.5. We have observed that clear sky samples are more clumped than the other two types of samples for all the combinations of features considered in Table 6.5. This explains why QDA performed much better than LDA for all the cases presented in the table, by the order of almost 2. It was checked that such difference between LDA and QDA is reproducible over other random splits of the data set into a training set and a test set. To roughly measure how hard the classification problem is due to the intrinsic overlap between class distributions, we applied the nearest neighbor (NN) method. The inequality in Cover and Hart (1967) relates the misclassification rate of the nearest neighbor method to the Bayes risk, the smallest error rate theoretically achievable, in the asymptotic sense. The inequality says that the probability of error for the NN is no more than twice the Bayes error rate as the size of a training set goes to infinity. The last column in Table 6.5 shows the test error rates of the nearest neighbor method. They suggest that the data set is not trivially separable. The relations between the half of the NN test error rates and the actual error rates incurred by the MSVM and the QDA, are reasonably close, if not very tight. It would be interesting to investigate further if any sophisticated variable (feature) selection methods may improve

Table 6.5: Test error rates for the combinations of variables and classifiers.

Number of variables	Variable descriptions	Test error rates (%)			
		MSVM	QDA	LDA	1-NN
2	(i) $R_2, \log_{10}(R_5/R_6)$	11.50	14.44	36.10	16.58
5	(i) $+R_1/R_2, BT_{31}, BT_{32} - BT_{29}$	10.16	12.83	27.81	12.30
12	(ii) original 12 variables	12.03	9.36	17.11	20.86
12	log transformed (ii)	9.89	9.63	24.06	18.98

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 essentially this cloud detection algorithm will be used as cloud mask for the Earth Observing System (EOS). The following cost matrix was considered, which penalizes misclassifying clouds as clear 1.5 times more than misclassifications of other kinds:

$$C = \begin{pmatrix} 0 & 1 & 1 \\ 1.5 & 0 & 1 \\ 1.5 & 1 & 0 \end{pmatrix} \quad (6.2)$$

where we coded clear as class 1, water clouds as class 2, and ice clouds as class 3. Its corresponding classification boundaries are drawn in Figure 6.7. 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.

Now, we illustrate how to exercise the idea of measuring prediction strength in the cloud classification problem. In the standard case, we want to associate the classification accuracy rate with each class prediction by MSVMs. Whereas, in nonstandard situations, we associate the expected misclassification cost with each prediction analogously. Once these relations are established, we will make a decisive prediction based on an MSVM output only when its accuracy exceeds a threshold or its cost is no higher than a threshold. In the equal cost case, we have proposed to use the pairs of the multiclass hinge loss for SVM prediction at each sample in a reference data set and its actual class to gauge classification accuracy as a function of the loss. The reference data set needs to be independent of the training samples. Cross validation is an alternative



to reserving a reference set for this purpose. Just for simple demonstration, we used the test set as a reference set, and evaluated the MSVM decision vectors at 374 test samples for the trained MSVM with two features in Figure 6.6. Then the reference samples were sorted out according to their predicted class. The reason why we did not pool three predicted categories altogether is that the classes are so lopsided that the symmetry assumption does not appear to hold for this problem. For each predicted category, we estimated the probability of

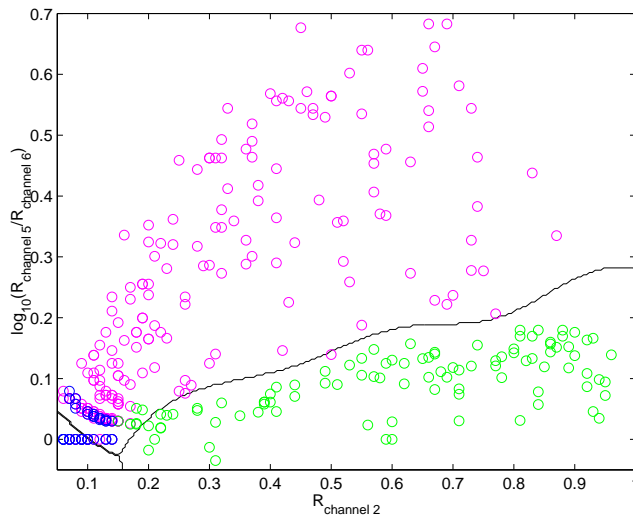


Figure 6.6: The classification boundaries determined by the MSVM using 370 training samples randomly selected from the bottom left plot in Figure 6.5.

a correct prediction, as a function of the multiclass hinge loss for MSVM prediction via a simple logistic regression. The three plots in Figure 6.8 depict these estimated accuracies for each class. Red tick marks represent the actual data pairs derived from the test set, and used for the logistic regression. The solid lines are the estimated probabilities and the dotted lines are non-simultaneous pointwise 95% confidence intervals. Undoubtedly, for clear sky, the estimated coefficient for the loss came out to be insignificant while those for liquid clouds and ice clouds are significant. For a small loss, liquid clouds predictions could be more confident than ice clouds from the plots. Some multiplicity that the data exhibit, especially evident in the clear samples, seems a little bothering, but the details of the original data generation procedure was not known.

Similarly when different misclassification costs are considered, we can infer

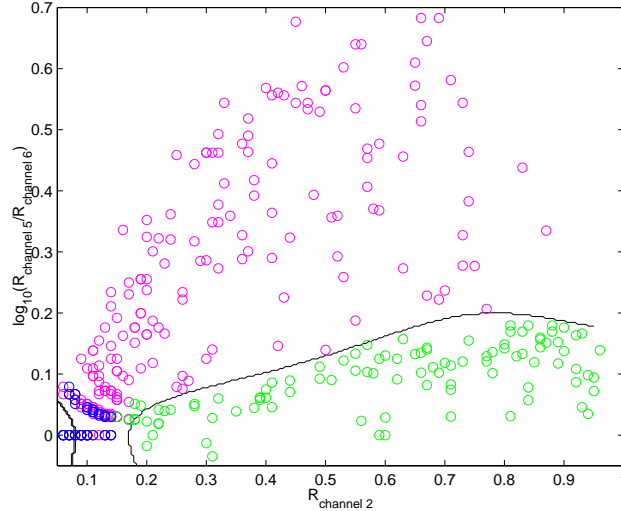


Figure 6.7: The classification boundaries determined by the nonstandard MSVM when the cost of misclassifying clouds as clear is 1.5 times higher than other types of misclassifications.

the expected cost given the MSVM prediction as a function of the loss, using a reference data set. Instead of associating the loss with the indicator of whether the predicted class is the same as the actual class, we associate the standard multiclass hinge loss with the actual misclassification cost incurred by the MSVM prediction. For the cost structure in (6.2), we applied this idea using the test set as a reference set, and obtained the three plots in Figure 6.9. Again, the red tick marks indicate the pairs of the loss and the misclassification cost for the test set. Likewise, the coefficient of the loss was not significant for the clear sky prediction. Given a loss, liquid clouds predictions seem to bring smaller costs than ice clouds.

Although this study is preliminary in its scope, the results are promising. It is believed that the MSVM will be very useful for other classification problems in atmospheric sciences as well.

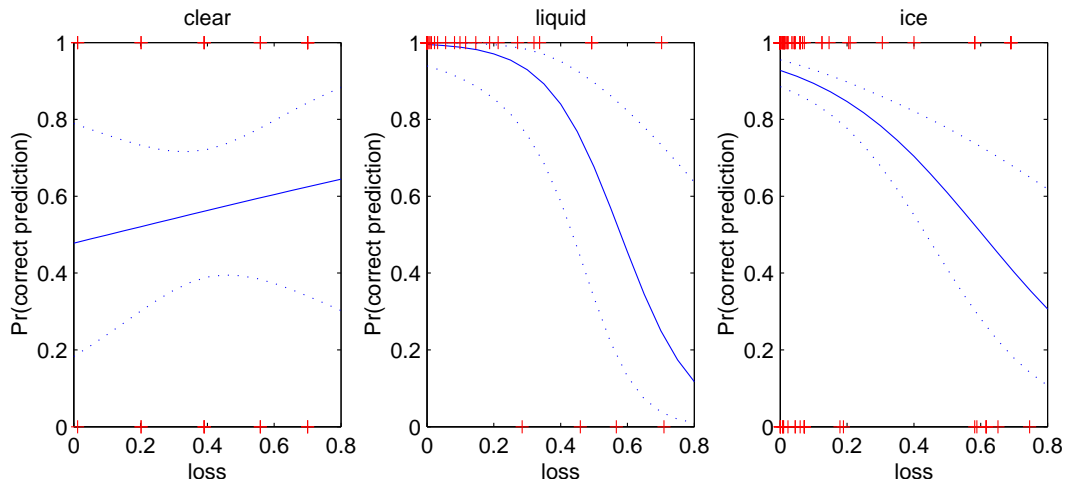


Figure 6.8: The estimated MSVM prediction accuracy as a function of the loss via a simple logistic regression, for each predicted class.

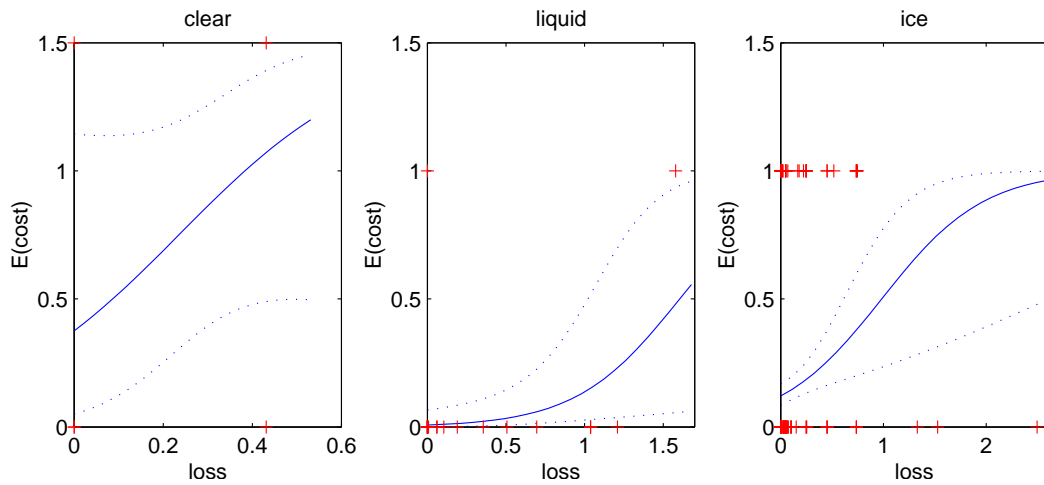


Figure 6.9: The estimated expected misclassification costs as a function of the loss for three predicted categories.

# Chapter 7

## Conclusion

We have proposed a loss function deliberately tailored to target the coded class with the maximum conditional probability for multiclass classification problems. Using the loss function, we have extended the classification paradigm of Support Vector Machines to the multiclass case so that the resulting classifier approximates the optimal classification rule. The extended Support Vector Machines allow a unifying formulation when there are 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. It would be interesting to investigate further whether this seeming bias is systematic. Then, the usefulness of the multiclass 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 samplewise data reduction through support vectors. Some works to integrate variable selection with binary SVMs have been done by Bradley and Mangasarian (1998), Weston, Mukherjee, Chapelle, Pontil, Poggio and Vapnik (2000), and Guyon, Weston, Barnhill and Vapnik (2002). Note that some of the methods were limited to linear SVMs only. 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. In the microarray data analysis presented, we screened predictive genes by a criterion, which measures the association between the gene and class distinction marginally, and trained classifiers based on the prescreened genes. It is interesting to know how different results would be obtained if some subsets of genes

were considered jointly and the training was done simultaneously with gene selection steps. For answering such questions, it would be valuable to generalize the variable selection methods for binary SVMs further to the multiclass SVM.

Another direction of future work includes establishing the advantage of the multiclass 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. Lin (2000) and Steinwart (2001) have made some theoretical endeavors for the binary SVM in some special cases. It would be intriguing to compare the Support Vector Machine paradigm with traditional methods, based on a lucid theoretical criterion.

The MSVM methodology is a generic approach to multiclass problems treating all the classes simultaneously. Solving a series of binary problems instead, in one-vs-rest fashion has potential drawbacks when classes overlap considerably. The pairwise approach often exhibits large variability since each binary classifier is estimated from a small subset of the training data. If we restrict classifiers to simple ones, say, those yielding linear boundaries only, and pooling some classes into a hyperclass gives much simpler classification boundaries, then this simultaneous approach may not be very efficient, let alone its increased computational complexity. Nevertheless, the difficulty is how to form such hyperclasses inductively from the data. If classifiers are flexible enough to provide arbitrary boundaries, then its advantages of aggregating classes become murky. Considering all these, we believe that the multiclass Support Vector Machine with flexible kernel functions will be a useful and theoretically sound alternative for classification problems in general.

# Bibliography

- Ackerman, S. A., Strabala, K. I., Menzel, W., Frey, R. A., Moeller, C. and Gumley, L. (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.
- Aronszajn, N. (1950). Theory of reproducing kernel, *Transactions of the American Mathematical Society* **68**: 3337–404.
- 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.
- Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines, in J. Shavlik (ed.), *Machine Learning Proceedings of the Fifteenth International Conference*, Morgan Kaufmann, San Francisco, California, pp. 82–90.
- Bredensteiner, E. J. and Bennett, K. P. (1999). Multicategory classification by support vector machines, *Computational Optimizations and Applications* **12**: 35–46.
- Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth.
- Burges, C. (1998). A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* **2**(2): 121–167.
- Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13**(1): 21–7.
- Cox, D. and O’Sullivan, F. (1990). Asymptotic analysis of penalized likelihood and related estimators, *The Annals of Statistics* **18**: 1676–1695.

- Crammer, K. and Singer, Y. (2000). On the learnability and design of output codes for multiclass problems, *Computational Learning Theory*, pp. 35–46.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press.
- DeRisi, J., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P. S., Ray, M., Chen, Y., Su, Y. A. and Trent, J. M. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer., *Nat Genet* **14**: 457–60.
- Devroye, L., Györfi, L. and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag.
- 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. (2000). Comparison of discrimination methods for the classification of tumors using gene expression data, *Technical Report 576*, Department of Statistics, University of California, Berkeley. *J. Am. Stat. Assoc.*, 97(457):77–87, 2002.
- Evgeniou, T., Pontil, M. and Poggio, T. (1999). A unified framework for regularization networks and support vector machines, *Technical Report AI Memo 1654*, MIT.
- Fellinger, E. J., Garin-Chesa, P., Glasser, D. B., Huvos, A. G. and Rettig, W. J. (1992). Comparison of cell surface antigen HBA71 (p30/32MIC2), neuron-specific enolase, and vimentin in the immunohistochemical analysis of Ewing’s sarcoma of bone., *Am J Surg Pathol* **16**: 746–755.
- 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.
- Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M. and Hausler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics* **16**(10): 906–914.

- Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* **286**: 531–537.
- Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1-3): 389–422.
- Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*, Springer Verlag, New York.
- Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks* **13**: 415–425.
- Jaakkola, T. and Haussler, D. (1999). Probabilistic kernel regression models, *Proceedings of the 1999 Conference on AI and Statistics*, Morgan Kaufmann.
- Jiang, Y., Harlocker, S. L., Molesh, D. A., Dillon, D. C., Stolk, J. A., Houghton, R. L., Repasky, E. A., Badaro, R., Reed, S. G. and Xu, J. (2002). Discovery of differentially expressed genes in human breast cancer using subtracted cDNA libraries and cDNA microarrays., *Oncogene* **21**: 2270–82.
- Joachims, T. (1999). Making large-scale svm learning practical, in B. Schölkopf, C. Burges and A. Smola (eds), *Advances in Kernel Methods - Support Vector Learning*, MIT Press.
- 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. and Lee, C.-K. (2002). Classification of multiple cancer types by multicategory support vector machines using gene expression data, *Technical Report 1051*, Department of Statistics, University of Wisconsin-Madison.
- Lee, Y.-J. and Mangasarian, O. (2001). SSVM: A smooth support vector machine, *Computational Optimization and Applications* **20**: 5–22.
- Lin, Y. (2000). Some asymptotic properties of the support vector machine, *Technical Report 1029*, Department of Statistics, University of Wisconsin-Madison.
- 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.
- Mangasarian, O. (1994). *Nonlinear Programming*, Classics in Applied Mathematics, Vol. 10, SIAM, Philadelphia.
- Mangasarian, O. and Musicant, D. (1999). Successive overrelaxation for support vector machines.
- 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.
- Perou, C. M., Jeffrey, S. S., van de Rijn, M., Rees, C. A., Eisen, M. B., Ross, D. T., Pergamenschikov, A., Williams, C. F., Zhu, S. X., Lee, J. C., Lashkari, D., Shalon, D., Brown, P. O. and Botstein, D. (1999). Distinctive gene expression patterns in human mammary epithelial cells and breast cancers., *Proc Natl Acad Sci U S A* **96**: 9212–7.
- Platt, J. (1999). Sequential minimal optimization: A fast algorithm for training support vector machines.
- Ramani, P., Rampling, D. and Link, M. (1993). Immunocytochemical study of 12E7 in small round-cell tumours of childhood: an assessment of its sensitivity and specificity., *Histopathology* **23**: 557–561.

- Schummer, M., Ng, W. V., Bumgarner, R. E., Nelson, P. S., Schummer, B., Bednarski, D. W., Hassell, L., Baldwin, R. L., Karlan, B. Y. and Hood, L. (1999). Comparative hybridization of an array of 21,500 ovarian cDNAs for the discovery of genes overexpressed in ovarian carcinomas., *Gene* **238**: 375–85.
- Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines, *Journal of Machine Learning Research* **2**: 67–93.
- Strabala, K. I., Ackerman, S. A. and Menzel, W. (1994). Cloud properties inferred from 8-12- $\mu$ m data, *Journal of Applied Meteorology* **33**: 212–229.
- 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., 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.
- Wahba, G., Lin, Y., Lee, Y. and Zhang, H. (2001). Optimal properties and adaptive tuning of standard and nonstandard support vector machines, *Proceedings of the MSRI Berkeley Workshop on Nonlinear Estimation and Classification*. To appear.
- Weidner, N. and Tjoe, J. (1994). Immunohistochemical profile of monoclonal antibody O13: antibody that recognizes glycoprotein p30/32MIC2 and is useful in diagnosing Ewing's sarcoma and peripheral neuroepithelioma., *Am J Surg Pathol* **18**: 486–494.
- Weston, J. and Watkins, C. (1999). Support vector machines for multiclass pattern recognition, *Proceedings of the Seventh European Symposium On Artificial Neural Networks*.

- Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2000). Feature selection for SVMs, *Neural Information Processing Systems*, pp. 668–674.
- Xiang, D. and Wahba, G. (1996). A generalized approximate cross validation for smoothing splines with non-gaussian data, *Statistica Sinica* **6**: 675–692.
- Yeo, G. and Poggio, T. (2001). Multiclass classification of SRBCTs, *Technical Report AI Memo 2001-018 CBCL Memo 206*, MIT.
- Zhang, L., Zhou, W., Velculescu, V. E., Kern, S. E., Hruban, R. H., Hamilton, S. R., Vogelstein, B. and Kinzler, K. W. (1997). Gene expression profiles in normal and cancer cells., *Science* **276**: 1268–72.
- Zhu, Y. and Parada, L. F. (2001). Neurofibromin, a tumor suppressor in the nervous system., *Exp Cell Res* **264**: 19–28.