Cloud Classification of Satellite Radiance Data by Multicategory Support Vector Machines [1]

Yoonkyung Lee [2]

Department of Statistics, The Ohio State University, Columbus OH

Grace Wahba [3]

Department of Statistics, University of Wisconsin, Madison WI

Steven A. Ackerman[4]

Department of Atmospheric and Oceanic Sciences, University of Wisconsin, Madison WI

July 29, 2003

[1]Corresponding author address: Prof. Grace Wahba, Department of Statistics, University of Wisconsin, 1210 W. Dayton St., Madison WI 53706.

[2]Research supported in part by NASA Grant NAG5-1073 and NSF Grant DMS-0072292

[3]Research supported in part by NASA Grant NAG5-1073 and NSF Grant DMS-0072292

[4]Research supported by NASA Grant NAS5-31367

Abstract

Two category Support Vector Machines (SVMs) have become very popular in the machine learning community for classification problems, and have recently been shown to have good optimality properties for classification purposes. Treating multicategory problems as a series of binary problems is common in the SVM paradigm. However, this approach may fail under a variety of circumstances. The Multicategory Support Vector Machine (MSVM), which extends the binary SVM to the multicategory case in a symmetric way, and has good theoretical properties, has recently been proposed. The proposed MSVM in addition provides a unifying framework when there are either equal or unequal misclassification costs, and when there is a possibly nonrepresentative training set.

In this paper, we illustrate the potential of the MSVM as an efficient cloud detection and classification algorithm for use in Earth Observing System models, which 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 example ice or water. We have applied the MSVM to simulated MODIS channel data to classify the radiance profiles as coming from clear, water clouds, or ice clouds, and the results are promising. It can be seen in simple examples, and application to MODIS observations, that the method is an improvement over channel by channel partitioning. It is believed that the MSVM will be a very useful tool for classification problems in atmospheric sciences.

1 Introduction

The MODIS (MODderate resolution Imaging Spectroradiometer) is a key instrument developed for the NASA Earth Observing System (EOS) Terra and Aqua satellites. It measures radiances at 36 wavelengths including infrared and visible bands with spatial resolution 250 m to 1 km. 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. Cloud mask algorithms for MODIS which use a series of sequential tests on the radiances or their associated brightness temperatures, may be found in Strabala, Ackerman & Menzel (1994), Ackerman, Strabala, Menzel, Frey, Moeller & Gumley (1998), Platnick, King, Ackerman, Menzel, Baum, Riedi & Frey (2003) where their description as part of the MODIS Cloud Products Suites is described. See also Heidinger, Anne & Dean (2002). As readers of this journal are no doubt aware, the supervised machine learning literature contains many possibilities for classification (e. g. neural nets). A relatively new classification procedure, the Support Vector Machine (SVM) (Vapnik (1998), Scholkopf, Burges & Smola (1999), Wahba (1999), Cristianini & Shawe-Taylor (2000), Scholkopf & Smola (2002), Lin, Lee & Wahba (2002)) has become popular, for various reasons, some of which we will detail below. The original SVM method classified into one of two categories, and most of the literature used various combinations of the two category method to handle the multicategory case. The original SVM has been recently generalized to a truly multicategory classification scheme, which, moreover handles unequal misclassification costs and nonrepresentative examples in a principled way, see Lee & Lee (2003), Lee, Lin & Wahba (2001), Lin et al. (2002), Lee (2002). It appears that this Multicategory Support Vector Machine (MSVM) is well suited for classifying radiance profiles simultaneously according as they are cloudy or not, and, if cloudy, categorizing them as to type of cloud. The purpose of this paper is to introduce this MSVM to the meteorological literature and to describe how it may be applied to MODIS profiles.

In Section 2 we review the theory of optimal classification, and the relation of the (standard, two-category) SVM to it. In Section 3 we describe the MSVM, and in Section 4, we apply it to simulated MODIS observations. Section 5 applies the MSVM method to actual MODIS observations that have been classified (labeled) by an expert, and compares the results with the MODIS algorithm on the same labeled data set. Section 6 gives a summary and conclusions.

2 Optimal Classification, Bayes Rule, Support Vector Machines, and Other Margin Based Classifiers

Let $\boldsymbol{x} \in \mathcal{X}$ be an attribute vector that is going to be used in the future to classify. Here $\mathcal{X}$ is Euclidean m-space and $\boldsymbol{x}$ is an $m-$vector of observations from m MODIS channels. For expository purposes we first describe the two class problem, later the results for the general k-class problem will be given. Suppose we knew the probability densities $g_{\mathcal{A}}(\boldsymbol{x}), g_{\mathcal{B}}(\boldsymbol{x})$ for class $\mathcal{A}$ and class $\mathcal{B}$, and let $\pi_{\mathcal{A}}$ = probability the next observation (Y) is an $\mathcal{A}$, and let $\pi_{\mathcal{B}} = 1 - \pi_{\mathcal{A}}$ = probability that the next observation is a $\mathcal{B}$. Then

$$p_{\mathcal{A}}(\boldsymbol{x}) \equiv prob\{Y = \mathcal{A}|\boldsymbol{x}\} = \frac{\pi_{\mathcal{A}} g_{\mathcal{A}}(\boldsymbol{x})}{\pi_{\mathcal{A}} g_{\mathcal{A}}(\boldsymbol{x}) + \pi_{\mathcal{B}} g_{\mathcal{B}}(\boldsymbol{x})}.$$

Let $C_{\mathcal{A}}$ = cost to falsely call a $\mathcal{B}$ an $\mathcal{A}$ and $C_{\mathcal{B}}$ = cost to falsely call an $\mathcal{A}$ a $\mathcal{B}$. A (two-category) classifier ϕ is a rule which assigns $\boldsymbol{x}$ to one of $\{\mathcal{A}, \mathcal{B}\}$. The optimal (Bayes) classifier, ϕ_{OPT} which minimizes the expected cost is

$$\phi_{\mathrm{OPT}}(\boldsymbol{x}) = \begin{cases} \mathcal{A} & \text{if } \frac{p_{\mathcal{A}}(\boldsymbol{x})}{1-p_{\mathcal{A}}(\boldsymbol{x})} > \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}}, \\ \mathcal{B} & \text{if } \frac{p_{\mathcal{A}}(\boldsymbol{x})}{1-p_{\mathcal{A}}(\boldsymbol{x})} < \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}}. \end{cases} \tag{1}$$

If $C_{\mathcal{A}}/C_{\mathcal{B}} = 1$, and f is the log odds ratio $f(\boldsymbol{x}) = log\frac{p_{\mathcal{A}}(\boldsymbol{x})}{1-p_{\mathcal{A}}(\boldsymbol{x})}$, the optimal classifier is

$$f(\boldsymbol{x}) > 0 \text{ (equivalently, } p_{\mathcal{A}}(\boldsymbol{x}) - \tfrac{1}{2} > 0) \rightarrow \mathcal{A}$$
$$f(\boldsymbol{x}) < 0 \text{ (equivalently, } p_{\mathcal{A}}(\boldsymbol{x}) - \tfrac{1}{2} < 0) \rightarrow \mathcal{B}$$

Given a training set $\{y_i, \boldsymbol{x}_i\}_{i=1}^n, y_i \in \{\mathcal{A}, \mathcal{B}\}, \boldsymbol{x}_i \in \mathcal{X}$, where y_i is the class label for the ith member of the set, then f and hence p can, in principle be estimated by the method of penalized likelihood, see O'Sullivan, Yandell & Raynor (1986), Wahba (1990), Wahba, Wang, Gu, Klein & Klein (1994), Wahba, Wang, Gu, Klein & Klein (1995), Wahba (2002), and the *sign* of f is used as the classifier. In theory, with a large enough representative training set, it is known under very general conditions that f estimated this way does converge to the 'true' f if the penalty or smoothing parameter (λ in Equation (2) below) is chosen well. In practice however, there is not always a large enough training set to estimate f well, and, furthermore in regions of the domain $\mathcal{X}$ where the classification can be carried out with 100% accuracy, $f = \pm\infty$. Ideally, for purely classification purposes it would be good to have a practical estimate targeted *directly* at *sign* f. The SVM is known to do this when the SVM uses a sufficiently flexible kernel and is tuned well (see Lin (2002), the result has been known since Lin (1999)), which explains one of the reasons for the popularity of the SVM.

A regularized, margin based classifier is a classifier f_λ which is obtained as the solution to the optimization problem: Find f of the form $f(\boldsymbol{x}) = b + h(\boldsymbol{x})$, where $h \in \mathcal{H}_K$, to minimize

$$\mathcal{I}\{\boldsymbol{y}, f\} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{C}(y_i f(\boldsymbol{x}_i)) + \lambda \|h\|_K^2 \tag{2}$$

where $\boldsymbol{y} = (y_1, \cdots, y_n)$ and y_i is coded as $+1$ if the ith example is in $\mathcal{A}$ and -1 if it is in $\mathcal{B}$. The kernel $K = K(\boldsymbol{s}, \boldsymbol{t}), \boldsymbol{s}, \boldsymbol{t} \in \mathcal{X}$ is some positive definite function, that is, it is a *covariance* and $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS) associated with K, however, the only facts about RKHS that are relevant here will be given below. K can be any positive definite function which must be chosen with the particular problem in mind, although there are several general purpose ones that work well in a variety of circumstances. $\mathcal{C}(\tau)$ can be one of a variety of functions which satisfy some mild conditions. $y_i f(\boldsymbol{x}_i)$ is called the margin for the ith example, if it is positive then y_i will be classified correctly by $f(\boldsymbol{x}_i)$ and if it is negative, it will be classified incorrectly. Under general conditions (Kimeldorf & Wahba (1971)), the minimizer of $\mathcal{I}\{\boldsymbol{y}, f\}$ with h in $\mathcal{H}_K$ has a representation of the form:

$$f(\boldsymbol{x}) = b + \sum_{i=1}^{n} c_i K(\boldsymbol{x}_i, \boldsymbol{x}). \tag{3}$$

and

$$\| \sum_{i=1}^{n} c_i K(\boldsymbol{x}_i, \cdot) \|_K^2 = \boldsymbol{c}' \mathsf{K_n} \boldsymbol{c} \tag{4}$$

where $\mathsf{K_n}$ is the $n \times n$ matrix with i, jth entry $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. b and the coefficient vector $\boldsymbol{c} = (c_1, \cdots, c_n)'$ are found by substituting (3) into the first term in (2), and (4) into the second and minimizing.

It can be shown, with the $\pm$ coding for y_i, and letting $\tau = y_i f(\boldsymbol{x}_i)$ that setting $\mathcal{C}(\tau) = log(1 + e^{-\tau})$ in (2) gives the penalized log likelihood estimate. The SVM corresponds to $\mathcal{C}(\tau) = (1 - \tau)_+$ where $(\tau)_+ = \tau, \tau > 0$, and 0 otherwise. The ideal cost function for a margin based classifier might be $\mathcal{C}(\tau) = (-\tau)_*$ where $(\tau)_* = 1, \tau \geq 0$ and 0 otherwise, since $\frac{1}{n} \sum_{i=1}^{n} (-y_i f(\boldsymbol{x}_i))_*$ is the fraction of misclassified examples in the training set when f is the classifier (with the convention $f(\boldsymbol{x}_i) = 0$ is a misclassification of y_i). However, this $\mathcal{C}(\tau)$ leads to a nonconvex optimization problem. The SVM $\mathcal{C}(\tau)$ can be seen to be the closest convex function to $(-\tau)_*$ with derivative -1 at 0, see Figure 1. A good source of references and further information regarding SVMs may be found at the website `http://www.kernel-machines.org`. Further discussion on the comparison between penalized likelihood, SVM and some other regularized, margin based classifiers may be found in Wahba (2002)

3 Multiple Categories, Unequal Costs, Nonrepresentative Examples

In this section we describe the general nonstandard MSVM, as given in Lee (2002), Lee, Lin & Wahba (2002), Lee et al. (2001).

We now consider the case of k categories, with the costs of misclassification possibly different for different mistakes. Let C_{jr} be the cost of classifying an object in category j as an r, with $C_{jj} = 0$. Then the Bayes rule (which minimizes expected cost) is to choose the j for which $\sum_{\ell=1}^{k} C_{\ell j} p_\ell(\boldsymbol{x})$ is minimized, where $p_\ell(\boldsymbol{x})$ is the probability that an object in the population as a whole, with attribute vector $\boldsymbol{x}$ is in category ℓ.

We next allow the case that the training set is not representative of the population as a whole. Let $\pi_j, j = 1, \cdots, k$ be the proportions of the different categories in the population as a whole. and let π_j^s be the proportions of the different categories in the training set. Let $p_j^s(\boldsymbol{x})$ be the probability that an example in the training set with attribute vector $\boldsymbol{x}$ is in category j. Let

$$L_{jr} = (\pi_j/\pi_j^s) C_{jr}. \tag{5}$$

It can be shown that the optimum (Bayes) classifier chooses the j for which $\sum_{\ell=1}^{k} L_{\ell j} p_\ell^s(\boldsymbol{x})$ is minimized.

In the MSVM of the papers noted at the start of this section, the class label for the ith example is coded as $\boldsymbol{y}_i$, a k dimensional vector with 1 in the jth position if example i is in category j and $-\frac{1}{k-1}$ otherwise. Thus $\boldsymbol{y}_i \equiv (y_{i1}, \cdots y_{ik}) = (1, -\frac{1}{k-1}, \cdots, -\frac{1}{k-1})$ indicates that the ith example is in category 1. We define a k-tuple of functions $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \cdots f^k(\boldsymbol{x}))$, with each $f^j = b^j + h^j$ with $h^j \in \mathcal{H}_K$, and which are required to satisfy the sum-to-zero constraint $\sum_{j=1}^{k} f^j(\boldsymbol{x}) = 0$, for all $\boldsymbol{x}$ in $\mathcal{X}$. Define $cat(i) \equiv j$ if the ith example is in category j. Then, $L_{cat(i)r} = L_{jr}$. The MSVM is defined as the vector of functions $\boldsymbol{f}_\lambda = (f_\lambda^1, \cdots, f_\lambda^k)$, with each h^j in $\mathcal{H}_K$, satisfying the sum-to-zero constraint, which minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{r=1}^{k} L_{cat(i)r} (f^r(\boldsymbol{x}_i) - y_{ir})_+ + \frac{\lambda}{2} \sum_{j=1}^{k} \|h^j\|_{\mathcal{H}_K}^2. \tag{6}$$

It is not hard to show that the $k = 2$ case reduces to the 2-category margin based SVM which has just been discussed under the assumption that $L_{12} = L_{21} = 1, L_{11} = L_{22} = 0$.

It is shown in Lee et al. (2001) and Lee (2002) that the target for this general MSVM is $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \cdots, f^k(\boldsymbol{x}))$ with $f^j(\boldsymbol{x}) = 1$ for the j which minimizes $\sum_{\ell=1}^k L_{\ell j} p_\ell^s(\boldsymbol{x})$ and $-\frac{1}{k-1}$ otherwise. Thus the MSVM is directly estimating the class label which implements the Bayes rule. A simple demonstration will be given later.

The problem of finding constrained functions $(f^1(\boldsymbol{x}), \cdots, f^k(\boldsymbol{x}))$ minimizing (6) can be shown, as before, equivalent to the problem of finding a set of finite dimensional coefficients. It was shown in Lee et al. (2001) that to find $\boldsymbol{f}_\lambda$ with the sum-to-zero constraint, minimizing (6) is equivalent to finding each $(f^1(\boldsymbol{x}), \cdots, f^k(\boldsymbol{x}))$ of the form

$$f^r(\boldsymbol{x}) = b^r + \sum_{\ell=1}^n c_{\ell r} K(\boldsymbol{x}_\ell, \boldsymbol{x}) \quad \text{for } r = 1, \cdots, k \tag{7}$$

with the sum-to-zero constraint only at $\boldsymbol{x}_i$ for $i = 1, \cdots, n$, minimizing (6).

To find $\boldsymbol{f}_\lambda$, (7) is first substituted into (6). Then, by introducing nonnegative Lagrange multipliers, $\boldsymbol{\alpha}^j \in R^n, j = 1, \cdots, k$, the following dual problem can be obtained:

$$\min_{\boldsymbol{\alpha}^j} L_D = \frac{1}{2n} \sum_{j=1}^k (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})' \mathsf{K}_n (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}}) + \lambda \sum_{j=1}^k \boldsymbol{\alpha}^{j\prime} \boldsymbol{y}^j \tag{8}$$

$$\text{subject to} \quad 0 \le \boldsymbol{\alpha}^j \le \boldsymbol{L}^j \quad \text{for } j = 1, \cdots, k \tag{9}$$

$$(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})' \mathbf{e} = 0 \quad \text{for } j = 1, \cdots, k \tag{10}$$

where $\boldsymbol{L}^j \in R^n$ is the jth column of the n by k matrix with the ith row $L(\boldsymbol{y}_i) \equiv (L_{cat(i)1}, \cdots, L_{cat(i)k})$, $\boldsymbol{y}^j$ denotes the jth column of the n by k matrix with the ith row $\boldsymbol{y}_i$, and $\mathbf{e}$ is the n dimensional column vector of all ones. Once the quadratic programming problem is solved, the coefficients can be determined from the relation $\boldsymbol{c}^j = (c_{1j}, \cdots, c_{nj})' = -\frac{1}{n\lambda}(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}}), j = 1, \cdots, k$. Note that if K_n is not strictly positive definite, then $\boldsymbol{c}^j$ is not uniquely defined. According to the Karush-Kuhn-Tucker complementarity conditions the b^j can be found from any one of the components of $\boldsymbol{\alpha}^j$ (call it α_{ij}) which satisfies $0 < \alpha_{ij} < L_{cat(i)j}$ as

$$b^j = y_{ij} - \sum_{\ell=1}^n c_{\ell j} K(\boldsymbol{x}_\ell, \boldsymbol{x}_i). \tag{11}$$

If there is no such unbound α_{ij}, then $\boldsymbol{b} \equiv (b^1, \cdots, b^k)'$ is found as the solution to

$$\min_{\boldsymbol{b}} \quad \frac{1}{n}\sum_{i=1}^{n} L(\boldsymbol{y}_i)'(\boldsymbol{h}^i + \boldsymbol{b} - \boldsymbol{y}_i)_+ \tag{12}$$

$$\text{subject to } \sum_{j=1}^{k} b^j = 0 \tag{13}$$

where $\boldsymbol{h}_i = (h_{i1}, \cdots, h_{ik}) = (\sum_{\ell=1}^{n} c_{\ell 1} K(\boldsymbol{x}_\ell, \boldsymbol{x}_i), \cdots, \sum_{\ell=1}^{n} c_{\ell k} K(\boldsymbol{x}_\ell, \boldsymbol{x}_i))$. Details of the derivation may be found in Lee (2002), Lee et al. (2001), see also Mangasarian (1994).

Solving the quadratic programming (QP) problem of (8)-(10) can be done with available optimization packages for moderate sized problems. The calculations in this paper were done via MATLAB 6.1 with an interface to PATH 3.0, an optimization package implemented by Ferris & Munson (1999).

It is worth noting that if $(\alpha_{i1}, \cdots, \alpha_{ik}) = (0, \cdots, 0)$ for the ith example, then $(c_{i1}, \cdots, c_{ik}) = (0, \cdots, 0)$, so removing such an example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ would not affect the solution. In the two-category SVM, those data points with nonzero coefficient are called support vectors. To carry over the notion of support vectors to the multicategory case, we define support vectors as examples with $\mathbf{c}_i = (c_{i1}, \cdots, c_{ik}) \neq (0, \cdots, 0)$ for $i = 1, \cdots, n$. Thus, the multicategory SVM retains the sparsity of the solution in the same way as the binary SVM. For proofs, and further details about the MSVM and its implementation, refer to Lee et al. (2001) and Lee et al. (2002).

As with other regularization methods, the efficiency of the method depends on the ability to choose the tuning parameters well. An approximate leaving-out-one cross validation function, called Generalized Approximate Cross Validation (GACV) has been derived for the MSVM in Lee et al. (2002), analogous to the GACV proposed by Wahba, Lin & Zhang (2000) in the binary case. Alternatively 5-fold (or 10-fold) cross validation may be used. The GACV and 5-fold cross validation behave similarly and have relative advantages and disadvantages, depending on the problem.

Figure 2 describes a simulated example to suggest the result from Lee et al. (2002) that

the target of the MSVM is the class label (vector) implementing the Bayes rule. In this example a representative training set and equal misclassification costs are assumed.

In this example $\boldsymbol{x} = x \in [0, 1]$. The leftmost panel of Figure 2 gives $p_j(x), j = 1, 2, 3$ which will be used to generate data for this example. The other three panels give the three optimum f^j, superimposed on the p_j. The f^j take on only the values 1 and $-\frac{1}{2} \equiv -\frac{1}{k-1}$. For the experiment $n = 200$ values of x_i were chosen according to a uniform distribution on the unit interval, and the class label $j = 1, 2$ or 3 is assigned to an observation at x_i with probability $p_j(x_i)$. The Gaussian kernel $K(x, x') = e^{-\frac{1}{2\sigma^2}(x-x')^2}$ was used to calculate the f^j. The left panel of Figure 3 gives the estimated f^1, f^2 and f^3. For this example, λ and σ were chosen with the knowledge of the 'right' answer. It is strongly suggestive that the target functions are close to the step functions as claimed. In the second from left panel both λ and σ were chosen by 5-fold cross validation in the MSVM and in the third panel they were chosen by GACV. These two tuning methods gave somewhat different estimates of λ and σ and also different from the first (ideal) panel, but the resulting classification rules are similar. In the rightmost panel in Figure 3 the classification is carried out by a one-vs-rest method. This is the kind of example where the MSVM will beat a one-vs-rest two-category SVM: category 2 would be missed since the probability of category 2 is less that the probability of not 2 over a region, even though it is the most likely category there.

The GACV and 5-fold cross validation are used and compared in Lee & Lee (2003). Only 5-fold cross validation results will be given for the simulated MODIS data and MODIS observations analyzed below.

4 MSVM Cloud Classification With Radiance profiles

a. Introduction

As noted in the introduction, MODIS is a key instrument of the Earth Observing System (EOS). A description of the MODIS instrument may be found at `http://modis.gsfc.nasa.gov/`. MODIS cloud mask algorithms using sequential thresholding tests on channel observations one at a time are in Strabala et al. (1994), Ackerman et al. (1998), Platnick et al. (2003). In this section, 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, water clouds, or ice clouds.

b. 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 & 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 water clouds and 461 ice clouds) are given in the data set. Clouds were randomly placed within a given TIGR profile atmospheric layer. Cloud layers colder than 253K were assigned as ice and warmer than 273K were assigned water. Clouds with layer temperatures between these limits were randomly selected as either a water or cloud. Water contents within a cloud layer were randomly selected and ranges between 0.05 and .5 g m^{-3} for water clouds and 0.0007 and 0.11 g m^{-3} for ice clouds. The effective radius for water and ice clouds range between 2.5 and 20 microns and 10 and 80 microns respectively, and was randomly selected in the simulation. 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. Note that differing surface conditions that affect the observations in ways that are important for cloud classification should have their own training sets.

Figure 4 shows boxplots of the reflectances and the brightness temperatures along 12 spectra 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 R and BT for reflectance and 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. We observe a fair amount of overlap in the profiles among the three types. Figure 5 displays scatterplots of some features (either variable or transformation of variables) of interest, which have been used conventionally to distinguish between categories. 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 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.

c. 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 these two features only to the training data. To have a fair evaluation of this or any other flexible classification algorithm, it is appropriate to evaluate the algorithm on a test set that was not used in building it, since the training set error will in general be an underestimate of the accuracy on future observations. 370 examples, almost half of the original data were

selected randomly from the bottom left panel in Figure 5 as the training set. The Gaussian kernel was used and the tuning parameters λ and σ were tuned by 5-fold CV. The test error rate of the SVM rule over the 374 test examples was 11.5% (= 43/374). Figure 6 shows the classification boundaries. Most of the misclassifications occurred due to the considerable overlap between ice clouds and clear sky examples at the lower left corner of the plot. Table 1 shows the cross tabulation of the predicted category based on the classifier over the test set. It turned out that adding three more features $(R_1/R_2, BT_{31}, BT_{32} - BT_{29})$, to the MSVM to make 5 dimensional attribute vectors instead of a two dimensional ones 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).

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 selection. It yielded a 12.03% test error rate, which is slightly larger than the MSVMs with the tailored 2 or 5 features. Interestingly enough, when all the variables were transformed by the logarithm function, the MSVM achieved a test error rate of 9.89 %. The results are summarized in Table 2. We have observed that clear sky examples are more clumped than the other two types of examples for all the combinations of features considered in Table 2. To roughly measure how hard the classification problem is due to the intrinsic overlap between class distributions, we applied the nearest neighbor (NN) method. An inequality in Cover & Hart (1967) relates the misclassification rate of the NN method to the Bayes risk, the smallest error rate theoretically achievable, as the training sample size becomes infinitely large. The inequality says that the probability of error for the NN method is no more than twice the Bayes error rate as the size of a training set goes to infinity. The last column in Table 2 shows the test error rates of the NN method. They suggest that the data set is not separable in any simple way. The relations between one-half of the NN test error rates and the actual error rates incurred by the MSVM are reasonably close, if not very tight. It would

be interesting to investigate further if any sophisticated variable (feature) selection methods may improve the accuracy substantially.

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

$$C = \begin{pmatrix} 0 & 1 & 1 \\ 1.5 & 0 & 1 \\ 1.5 & 1 & 0 \end{pmatrix} \tag{14}$$

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 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. Just to suggest how the boundaries can be manipulated by changing the costs, Figure 8 plots the boundaries for the nonstandard MSVM when the cost matrix is

$$C = \begin{pmatrix} 0 & 4 & 4 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{15}$$

We note that in an operational system it would be easy with this method to examine the effects of different cost matrices on the overall data analysis system.

d. Assessing Prediction Strength

As noted previously, the MSVM is not estimating probabilities, but the hinge loss at $\boldsymbol{x}_*$, which measures how close the MSVM is to the class label of the class it has identified, may be used as a yardstick of the strength of the classification at $\boldsymbol{x}_*$. Letting $\mathcal{L}_{hinge}(\boldsymbol{x}_*)$ be

the hinge loss of the classification of the attribute vector $\boldsymbol{x}_*$ with respect to $\boldsymbol{f}_\lambda$, the fitted MSVM, then the hinge loss is

$$\mathcal{L}_{hinge}(\boldsymbol{x}_*) = \sum_{r=1}^{k} L_{cat(*)r}(f_\lambda^r(\boldsymbol{x}_*) - y_{*r})_+ \tag{16}$$

where y_{*r} is the rth component of the class label assigned by the MSVM $\boldsymbol{f}_\lambda(\boldsymbol{x}_*)$ and $cat(*)$ is the category assigned. Thus for example if the largest component of $\boldsymbol{f}_\lambda(\boldsymbol{x}_*)$ occurs for $r = 1$ then (for the standard case) $L_{cat(*)r} = 0$ for $r = 1$ and 1 otherwise, and $\mathcal{L}_{hinge}(\boldsymbol{x}_*) = \sum_{r=2}^{k}(f_\lambda^r(\boldsymbol{x}_*) + \frac{1}{k-1})_+$ and will be increasingly positive as the $f_\lambda^r(\boldsymbol{x}_*)$ increase above $-\frac{1}{k-1}$ for $r \neq 1$.

The hinge loss could be calibrated in various ways. The calibration set should be independent of the training examples. Here we use the 374 test examples. The test examples were sorted according to their predicted class. Within each class the hinge loss based on the MSVM that was used in constructing Figure 6 was computed for each test example and saved along with an indicator as to whether the classification was correct or not. For each (prediction) class, the probability of a correct prediction as a function of the hinge loss was then (roughly) estimated using linear logistic regression on the pairs of hinge losses and indicators. The two plots in Figure 9 depict these estimated probabilities of an accurate classification for liquid and ice clouds. Red tick marks represent the actual data pairs derived from the test set, and used for the logistic regression. The corresponding plot for the clear sky category is not shown, as the estimated probability of an accurate classification was essentially independent of the observed hinge loss. This is easily explained by inspection of Figure 6 where the clear attribute vectors are very closely bunched compared to the other attribute vectors, and overlaid by ice cloud attribute vectors.

5 Comparison with the MODIS algorithm

a. Labeled MODIS scenes and MODIS analysis.

The MODIS instrument provides an opportunity for applying the MSVM algorithm to satellite observations. A comprehensive remote sensing algorithm for cloud masking has been developed by members of the MODIS atmosphere science team. In this section we compare the MSVM and the MODIS algorithm on MODIS observations that have been identified by an expert.

Assessing any cloud algorithm is difficult. One validation approach is to use an expert analyst to label pixels as clear or cloudy through visual inspection of the spectral, spatial, and temporal features in a set of composite satellite images. The analyst uses knowledge of and experience with cloud and surface spectral properties to identify clear, water cloud and ice clouds. In this study, 1536 MODIS scenes over the Gulf of Mexico in July 2002 were classified as clear, ice cloud, or water cloud by a satellite expert. There were 256 clear, 952 ice cloud and 328 water clouds identified. Each of these three groups were divided in half by a random mechanism, and the first halves were set aside as a training set for the MSVM, leaving 128, 476 and 164 clear, ice cloud and water cloud profiles for a test set of 768 profiles. Training and testing was done using the same channels as in the simulation.

As a reference, the expert analysis is compared with the operational MODIS cloud mask detection algorithm on the test set. The MODIS cloud mask classifies each pixel as either confident clear, probably clear, uncertain, or cloudy. The cloud mask algorithm (see Ackerman et al. (1998)) uses a series of threshold tests to detect the presence of clouds in the instrument field-of-view. Designed to operate globally during the day and night, the specific tests executed are a function of surface type, including land, water, snow/ice, desert, and coast, and solar illumination.

For many regions of the globe, the uncertain classification can be considered probably cloudy. For comparison with the expert analysis, confident clear and probably clear are considered clear pixels and the uncertain and cloudy confidences are labeled as cloudy. 115 of the clear pixels in the test set were misclassified as cloudy, and 26 of the cloudy pixels

were misclassified as being clear. Thus the test error rate of the MODIS cloud detection algorithm for these scenes is approximately $18\% = (115+26)/(768)$. This is consistent with the clear-sky bias of the cloud mask algorithm, in the sense that if one of the tests indicates that the pixel is cloud contaminated, the pixel is flagged as cloudy or uncertain. In these particular scenes, the cloud mask misclassified clear pixels with a low reflectance in channel 2, but a high reflectance ratio between channels 2 and 1. The cloud mask flags these pixels as "uncertain", which are interpreted as cloud.

b. MSVM analysis of the MODIS labeled pixels

We now turn to the results of first training the MSVM on the set-aside MODIS scenes, and then testing it on the same set of 768 scenes used to test the MODIS algorithm against the expert's labels. Before presenting the results it is interesting to compare the labeled MODIS data set with the simulated data.

Figure 10 plots the 1536 labeled MODIS scenes, and may be compared with Figure 5. The set of labeled MODIS scenes are easier to classify than the simulated scenes of Section 4, however, we note how qualitatively similar they are. This illustrates an interesting result - in developing an operational MSVM algorithm for MODIS under observing conditions other than those shown here, it is likely that the simulated data, which is cheap to generate, can reasonably be used for preliminary experimental training and testing of the MSVM algorithm. This would then be followed by the collecting of the much more expensive expert-labeled observational data sets which would be used to build an actual operational MSVM algorithm.

The MSVM misclassification rates on the labeled MODIS test set were under 1.0% for all three of the cases using 5 or 12 variables, the details are in Table 3.

It is of course hard to visualize what the MSVM or any other classification method is doing on 5 or more variables. To visualize just how powerful an MSVM trained algorithm can be, Figure 11 plots the first two variables in the training set, along with the classification

boundaries given by the MSVM trained on these two variables. The error rate on the test set was 4.69%, from Table 3.

6 Summary and Conclusions

We have described the usual two category Support Vector Machine, and the recent generalization, the Multicategory Support Vector Machine. The MSVM is estimating the Bayes rule under appropriate conditions and so can be expected to have favorable properties as a classification algorithm for classifying attribute vectors into one of several categories. We have demonstrated the potential of this method for classifying MODIS observations into clear, water cloud or ice cloud, from simulated MODIS data, and from MODIS observational data that has been classified by an expert. The MSVM can be adjusted, if desired, to take into account nonrepresentative training sets and unequal costs of misclassification, and a rudimentary procedure for assessing the strength of the prediction is proposed. The method clearly has benefits over the existing MODIS algorithms, which use thresholds on individual components of the attribute vectors. (Those classification boundaries would look like segments of horizontal or vertical lines if applied to the attributes in Figure 11). Both the simulated data and the observational MODIS data represent ocean conditions. In practice training sets for the different conditions which materially affect the MODIS observations would have to be collected and labels established. It is believed that this method has important potential for improving the ability of the MODIS data analysis to efficiently classify clear and different kinds of cloudy observations.

Acknolegdments. This research supported by NASA Grants NAG5-1073 and NAS5-31367 and NSF Grant DMS-0072292.

References

Ackerman, S., Strabala, K., Menzel, W., Frey, R., Moeller, C. & Gumley, L. (1998), 'Discriminating clear sky from clouds with MODIS', *J. Geophysical Research* **103**, 32,141–32,157.

Cover, T. & Hart, P. (1967), 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory* **13**(1), 21–7.

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

Ferris, M. C. & Munson, T. S. (1999), 'Interfaces to PATH 3.0: Design, implementation and usage', *Computational Optimization and Applications* **12**, 207–227.

Heidinger, A., Anne, V. & Dean, C. (2002), 'Using MODIS to estimate cloud contamination of the AVHRR record', *J. Atmos. Ocean Tech.* **19**, 586–597.

Key, J. & Schweiger, A. (1998), 'Tools for atmospheric radiative transfer: Streamer and FluxNet', *Computers and Geosciences* **24**(5), 443–451.

Kimeldorf, G. & Wahba, G. (1971), 'Some results on Tchebycheffian spline functions', *J. Math. Anal. Applic.* **33**, 82–95.

Lee, Y. (2002), Multicategory Support Vector Machines, theory, and Application to the Classification of Microarray Data and Satellite Radiance Data, PhD thesis, Technical Report 1062, Department of Statistics, University of Wisconsin, Madison WI.

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

Lee, Y., Lin, Y. & Wahba, G. (2001), Multicategory support vector machines, Technical Report 1043, Department of Statistics, University of Wisconsin, Madison WI. To appear, *Computing Science and Statistics*, 33.

Lee, Y., Lin, Y. & Wahba, G. (2002), Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, Technical Report 1064, Department of Statistics, University of Wisconsin, Madison WI.

Lin, Y. (1999), Support vector machines and the Bayes rule in classification, Technical Report 1014, Department of Statistics, University of Wisconsin, Madison WI, to appear, *Data Mining and Knowledge Discovery*.

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

Lin, Y., Lee, Y. & 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.

O'Sullivan, F., Yandell, B. & Raynor, W. (1986), 'Automatic smoothing of regression functions in generalized linear models', *J. Amer. Statist. Assoc.* **81**, 96–103.

Platnick, S., King, M., Ackerman, S., Menzel, W., Baum, B., Riedi, J. & Frey, R. (2003), 'The MODIS cloud products: Algorithms and examples from Terra', *To appear, IEEE Trans. Geoscience and Remote Sensing*.

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

Scholkopf, B., Burges, C. & Smola, A. (1999), *Advances in Kernel Methods-Support Vector Learning*, MIT Press.

Strabala, K., Ackerman, S. & Menzel, W. (1994), 'Cloud properties inferred from 8-12 μm data', *J. Applied Meteorology* **33**, 212–229.

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

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM. CBMS-NSF Regional Conference Series in Applied Mathematics, v. 59.

Wahba, G. (1999), Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV, *in* B. Scholkopf, C. Burges & A. Smola, eds, 'Advances in Kernel Methods-Support Vector Learning', MIT Press, pp. 69–88.

Wahba, G. (2002), 'Soft and hard classification by reproducing kernel Hilbert space methods', *Proc.National Academy of Sciences* **99**, 16524–16530.

Wahba, G., Lin, Y. & 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 & D. Schurmans, eds, 'Advances in Large Margin Classifiers', MIT Press, pp. 297–309.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1994), Structured machine learning for 'soft' classification with smoothing spline ANOVA and stacked tuning, testing and evaluation, *in* J. Cowan, G. Tesauro & J. Alspector, eds, 'Advances in Neural Information Processing Systems 6', Morgan Kauffman, pp. 415–422.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1995), 'Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy', *Ann. Statist.* **23**, 1865–1895. Neyman Lecture.

List of Tables

List of Figures

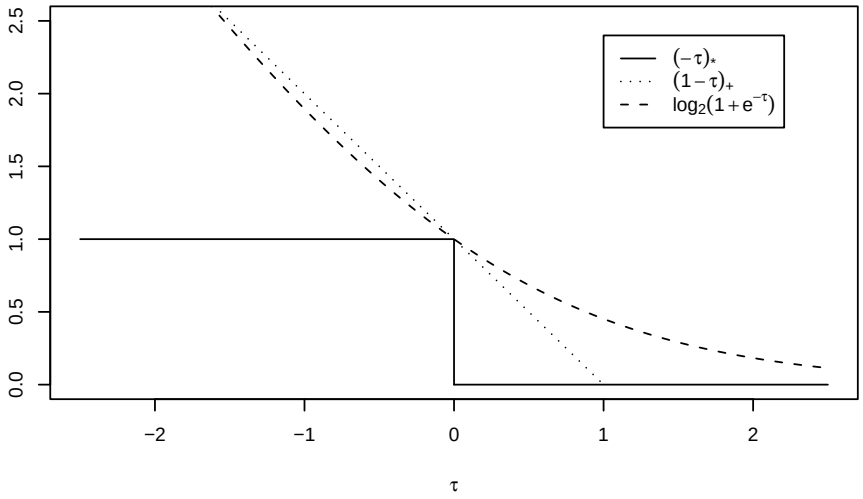

Figure 1: Comparison of $(-\tau)_*, (1-\tau)_+$ and $log_2(1+e^{-\tau})$.

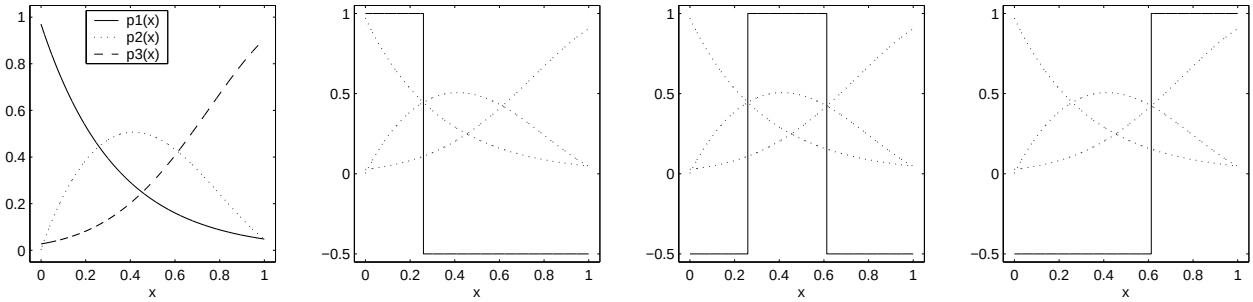

Figure 2: Probabilities and optimum f^j's for three category SVM demonstration.

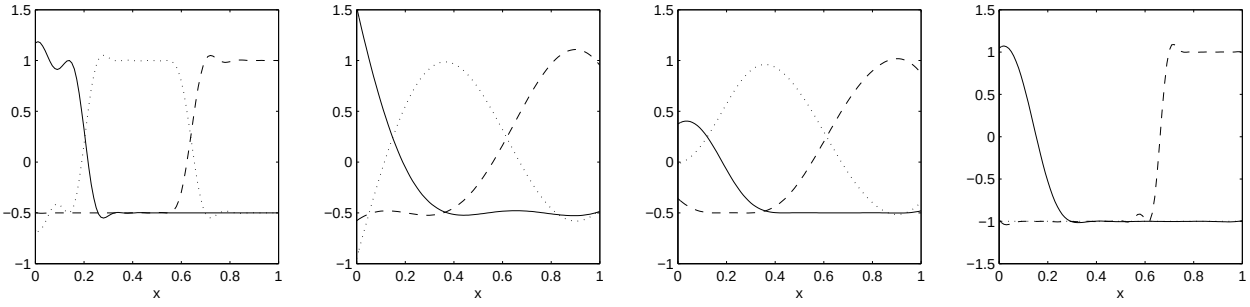

Figure 3: Left to Right: Optimum MSVM tuning, 5-fold cross validation tuning of the MSVM, GACV MSVM tuning, one-vs-many SVM.

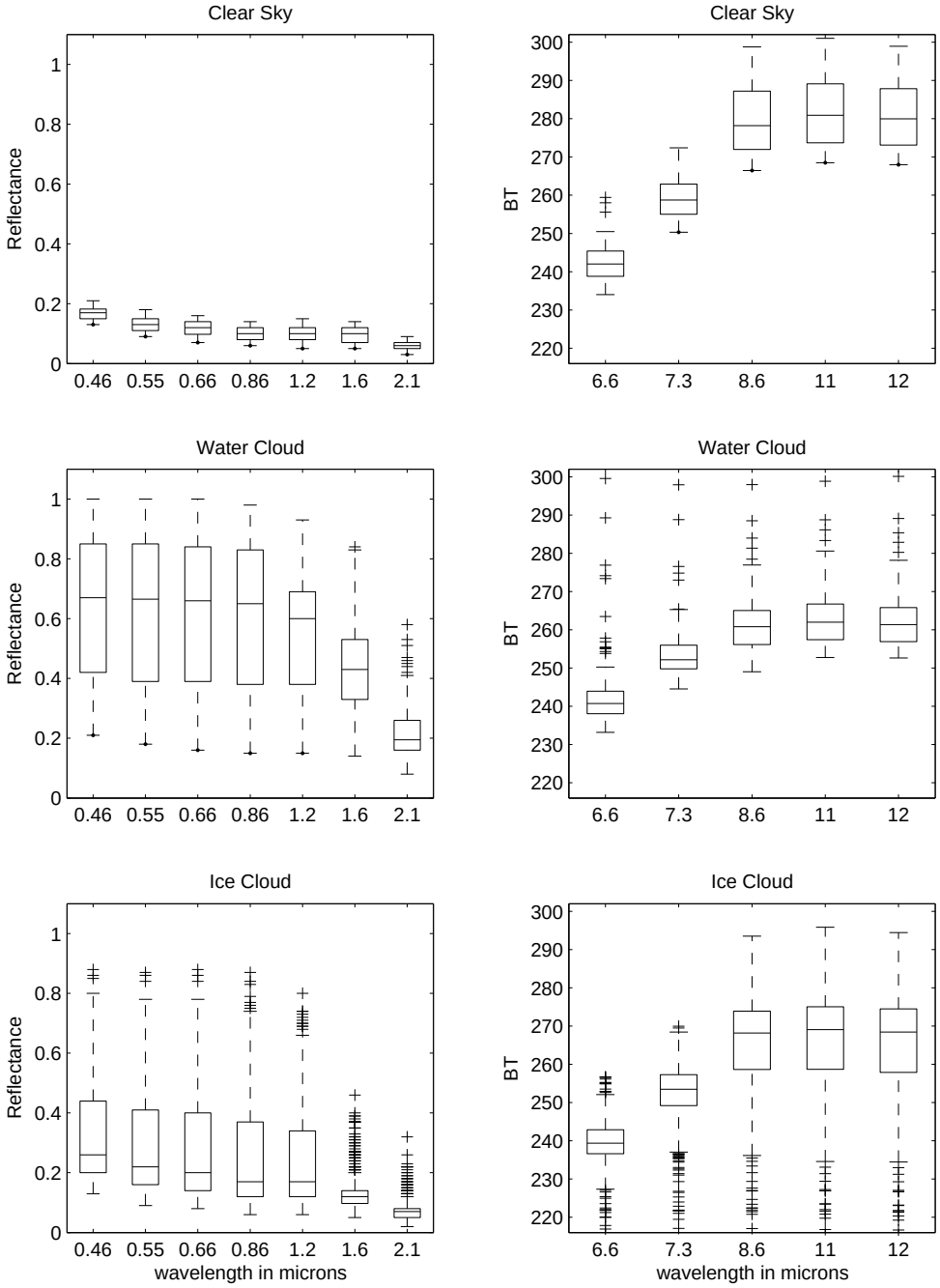

Figure 4: The boxplots of 7 reflectances and 5 brightness temperatures for clear sky, water clouds, and ice clouds over the ocean.

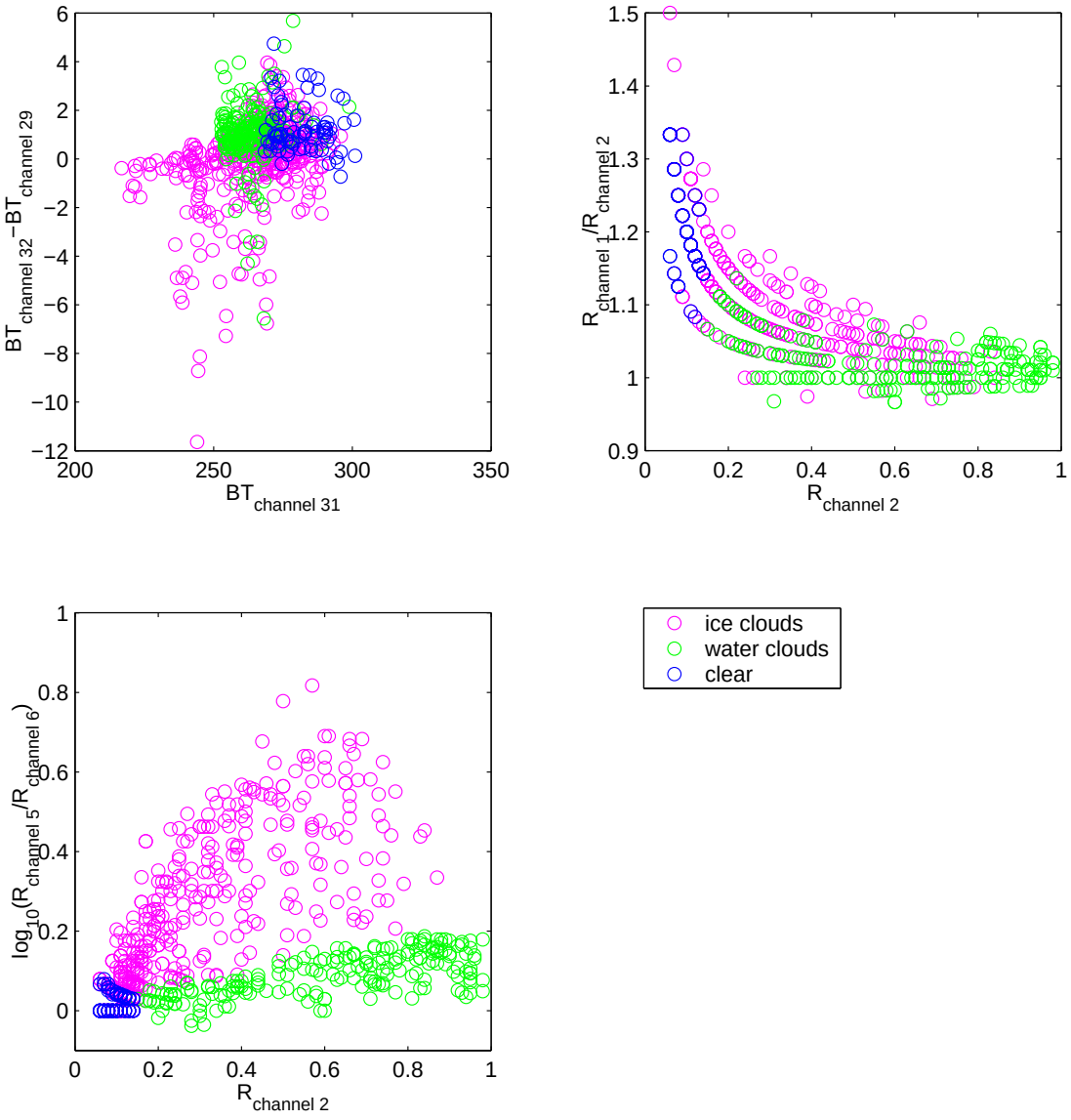

Figure 5: Scatterplots of $BT_{channel_{31}}$ vs $BT_{channel_{32}} - BT_{channel_{29}}$ (top left), $R_{channel_1}/R_{channel_2}$ vs $R_{channel_2}$ (top right), and $R_{channel_2}$ vs $\log_{10}(R_{channel_5}/R_{channel_6})$ (bottom left).

Table 1: Distribution of the predicted class based on the MSVM with two features

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

Table 2: Test error rates for the combinations of variables and classifiers.

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

Table 3: Test error rates for the real data.

Number of variables	Variable descriptions	Test error rates (%) MSVM	
2	(i) R_2, $\log_{10}(R_5/R_6)$	36/768	= 4.69
5	(i)+R_1/R_2, BT_{31}, $BT_{32} - BT_{29}$	2/768	= 0.26
12	(ii) original 12 variables	6/768	= 0.78
12	log transformed (ii)	5/768	= 0.65

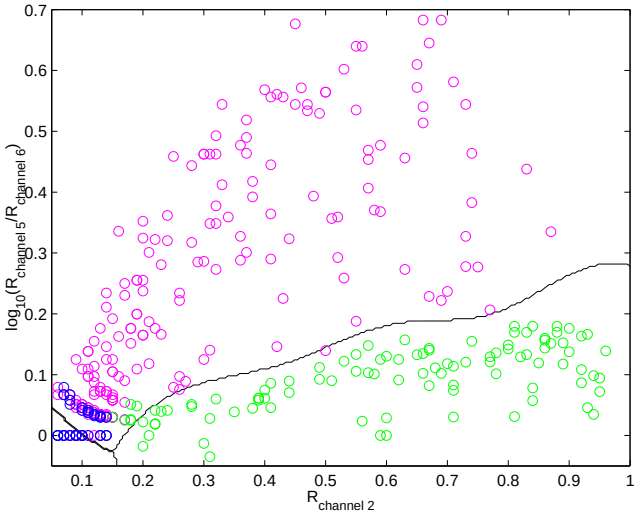

Figure 6: The classification boundaries determined by the MSVM using 370 training examples randomly selected from the bottom left plot in Figure 5.

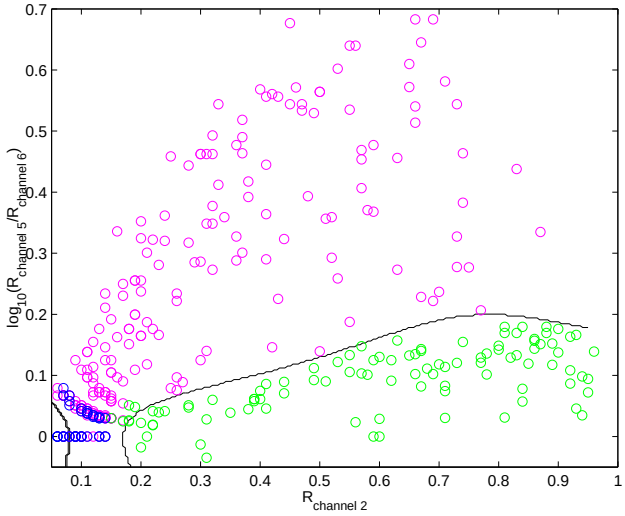

Figure 7: The classification boundaries determined by the nonstandard MSVM when the cost of misclassifying clouds as clear is 1.5 times higher than other types of misclassifications.

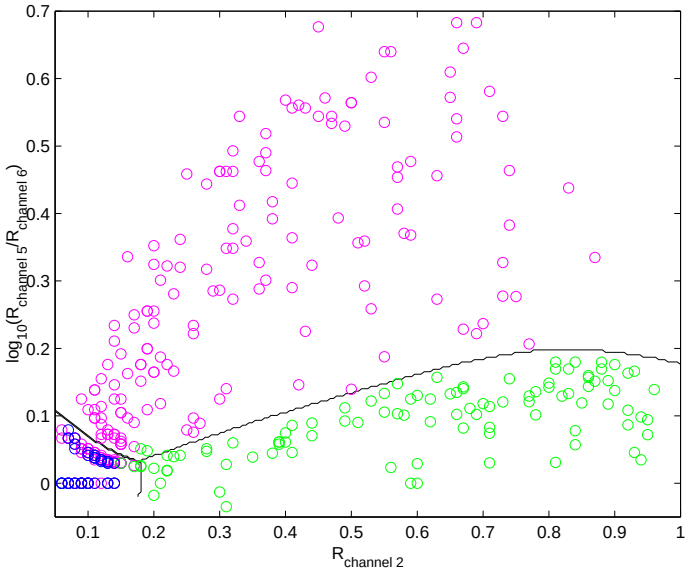

Figure 8: The classification boundaries determined by the nonstandard MSVM when the cost of misclassifying clear sky examples is four times as high as other types of misclassifications.

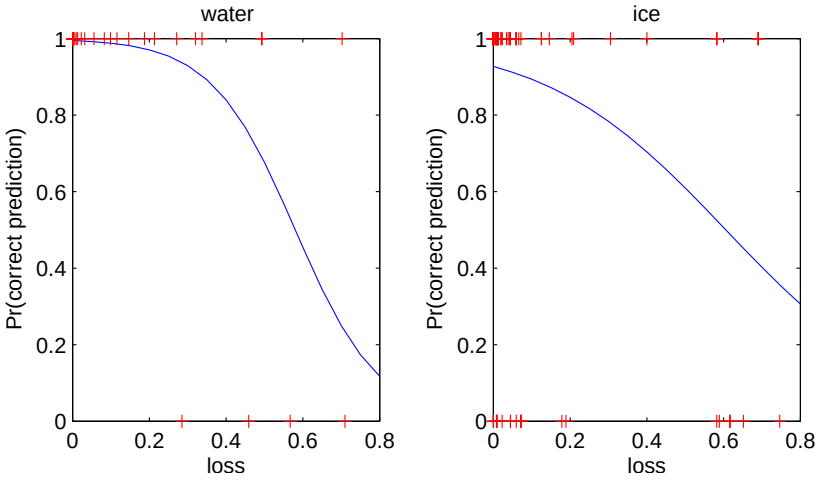

Figure 9: The estimated MSVM prediction accuracy as a function of the loss estimated via linear logistic regression, for the water and ice cloud predicted classes. Red ticks are the actual pairs of the hinge loss and the indicator of correct prediction (1:correct, 0:incorrect) for each test example.

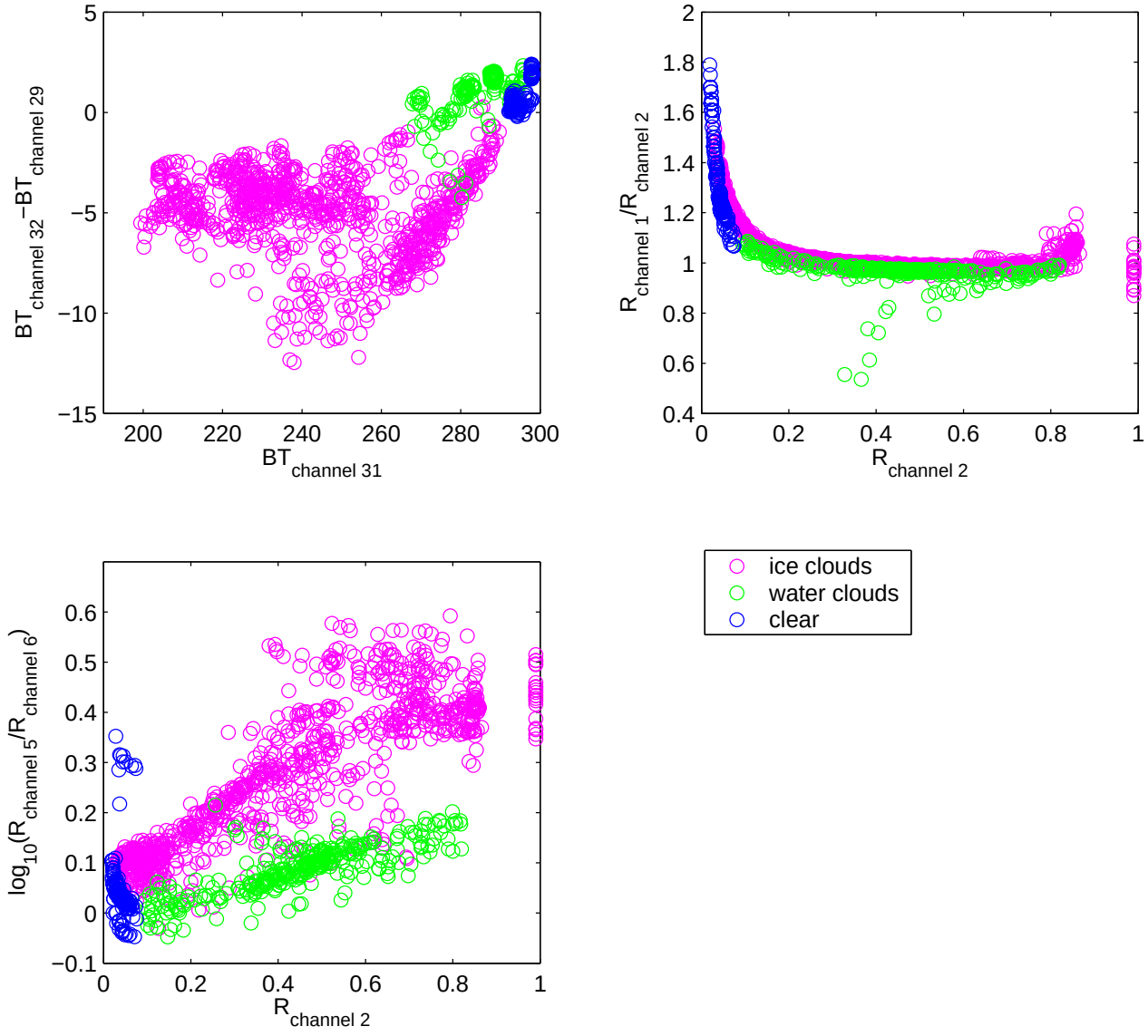

Figure 10: Scatterplots of $BT_{channel_{31}}$ vs $BT_{channel_{32}} - BT_{channel_{29}}$ (top left), $R_{channel_1}/R_{channel_2}$ vs $R_{channel_2}$ (top right), and $R_{channel_2}$ vs $\log_{10}(R_{channel_5}/R_{channel_6})$ (bottom left), labeled MODIS observations

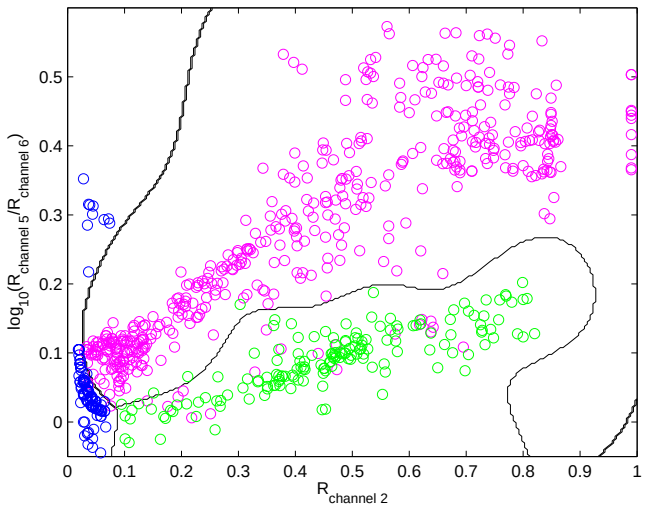

Figure 11: Classification boundaries on the training set based on the MSVM trained on two variables only.