Support Vector Machines for Hyperspectral Remote Sensing Classification

J. Anthony Gualtieri[a] and R. F. Cromp[b]

[a]Applied Information Sciences and Global Science and Technology
Code 935, NASA/GSFC, Greenbelt, MD 20771, U.S.

[b]Earth and Space Data Computing Division
Code 930, NASA/GSFC Greenbelt, MD 20771, U.S.

ABSTRACT

The Support Vector Machine provides a new way to design classification algorithms which learn from examples (supervised learning) and generalize when applied to new data. We demonstrate its success on a difficult classification problem from hyperspectral remote sensing, where we obtain performances of 96%, and 87% correct for a 4 class problem, and a 16 class problem respectively. These results are somewhat better than other recent results on the same data. A key feature of this classifier is its ability to use high-dimensional data *without* the usual recourse to a feature selection step to reduce the dimensionality of the data. For this application, this is important, as hyperspectral data consists of several hundred contiguous spectral channels for each exemplar. We provide an introduction to this new approach, and demonstrate its application to classification of an agriculture scene.

Keywords: Support Vector Machine, Classifier, Hyperspectral, Supervised Learning, AVIRIS

1. INTRODUCTION

The Support Vector Machine (SVM) is a relatively recent approach introduced by Boser, Guyon, and Vapnik,[1] [2] for solving *supervised* classification and regression problems, or more colloquially learning from examples. In the following we will discuss only classification.

This work is in part motivated by the recent profusion of high-dimensional data in remote sensing, where hyperspectral imaging sensors for research,[3] or commercial use,[4] ,[5] measure radiance at hundreds of contiguous channels for each ground pixel. For this data, part of the challenge is for classifiers that perform well in such high-dimensional spaces.

Traditionally, classifiers model the underlying density of the various classes and then find a separating surface. However density estimation in high-dimensional spaces suffers from the Hughes effect,[6] [7]: For a fixed amount of training data the classification accuracy as a function of number of bands reaches a maximum and then declines, because there is limited amount of training data to estimate the large number of parameters needed. Thus usually, a feature selection step is first performed on the high-dimensional data to reduce its dimensionality.

As we will demonstrate, the SVM approach does not suffer this limitation and uses the full dimensionality of the hyperspectral data. Support Vector Machines directly seek a separating surface through an optimization procedure that finds the exemplars that form the *boundaries* of the classes. These exemplars are called the *support vectors*. This is significant because it is usually the case that there are a small subset of all the training data that are involved in defining the separating surface, i.e., those examples that are closest to the separating surface.

In addition, the Support Vector Machine approach uses the kernel method, discussed below, to map the data with a non-linear transformation to a *higher dimensional* space and in that space trys to find a *linear* separating surface between the two classes. The transformation to a higher dimensional space tends to spread the data out in a way that facilitates the finding of a linear separating surface. In this way the separating surfaces that would be non-linear (not a hyperplane) in the original data space can become linear (a hyperplane) in the higher dimensional space. Instead of being penalized by the *curse of dimensionality* and its attendant Hughes effect, the Support Vector

Further author information: (Send correspondence to J. A. Gualtieri)
J.A. Gualtieri: E-mail: gualt@peep.gsfc.nasa.gov
R.F. Cromp: E-mail: cromp@sauquoit.gsfc.nasa.gov

Machine can use the full dimensionality of the hyperspectral data without the feature selection preprocessing step. Why the curse of dimensionality is not a problem for the kernel method is discussed below.

A number of useful introductions are available in publications and on the world wide web,[8] ,[9] ,[10] .[11] In what follows we will first focus on *binary* classification – in the class or not in the class. Subsequently we will handle multiple classes by building separate classifier for each pair of classes and follow this with a voting strategy to choose the class label.

The plan for the paper is to give an overview of the mathematical formulation for binary classification. Then we introduce the optimal margin hyperplane, the transformation of its resulting optimization problem by means of Lagrange multipliers, and its solution. This is done for both the separable and non-separable cases. We then discuss the kernel method and the generalization to multiple classes. Following this, a section is devoted to describing the hyperspectral data we have used for demonstrating the classifier. Then we discuss implementation details and present the results. The conclusion summarizes the results and suggests further development of the method.

2. MATHEMATICAL FORMULATION

2.1. Classification

For classification, a set of examples consisting of pairs of class labels and feature vectors is known, and you desire to find a classifier function that gives correct answers on these examples and has low generalization error, meaning it gives good results for the class labels when applied to feature vector inputs it has not seen before.

We are given l training pairs, $(y_i, \mathbf{x}_i)$ $i = 1, \dots l$, consisting of class labels, $y_i \in \{1, -1\}$, and n-dimensional feature vectors, $\mathbf{x}_i \in \mathbf{R}^n$. We wish to find a function $f(\;\; ;\alpha) : \mathbf{x} \mapsto y$ that represents the classifier $y = f(\mathbf{x}; \alpha)$, where α are all the parameters of the classifier.

2.2. Optimal Margin Method for Separable Data

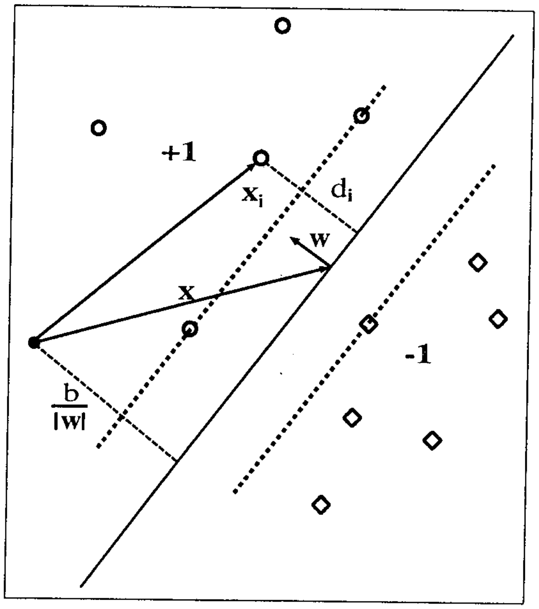

Figure 1. Schematic of separable data in $\mathbf{R}^2$. The circles are feature vectors in class +1 and the diamonds are feature vectors in class −1. The placement of the hyperplane shown is optimal.

Vapnik and Chervonenkis,[12] and Vapnik[13] originated the optimal margin method for separable data. With reference to Fig. 1 the problem is how to place a hyperplane such that:

1. All data belonging to class +1 lies on one side of the hyperplane and all data belonging to class -1 lies on the other side.

2. The hyperplane is placed so that the distance of the closest vectors in both classes are the furthest they can be from the hyperplane.

The hyperplane is defined by the equation

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \tag{1}$$

where $\mathbf{x}$ is a point on the hyperplane, $\mathbf{w}$ is the n-dimensional vector perpendicular to the hyperplane, and b is the distance of the closest point on the hyperplane to the origin. The classifier is then

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w} \cdot \mathbf{x} + b). \tag{2}$$

The orientation of the hyperplane is chosen so that $\mathbf{w}$ points towards the class labeled with $+1$. Let d_i, be the perpendicular distance of vector $\mathbf{x}_i$ from any point $\mathbf{x}$ on the hyperplane,

$$d_i = y_i \frac{\mathbf{w}}{|\mathbf{w}|} \cdot (\mathbf{x}_i - \mathbf{x}). \tag{3}$$

By pre-multiplying by y_i we guarantee that all the d_i are positive. Using the hyperplane equation, Eq. (1), to eliminate $\mathbf{x}$ in Eq. (3) we obtain

$$d_i = y_i \frac{(\mathbf{w} \cdot \mathbf{x}_i + b)}{|\mathbf{w}|}. \tag{4}$$

We may then pose the problem as minimize, over all the training vectors, the distance of the hyperplane from all the training vectors, and then maximize those distances over all placements of the hyperplane:

$$\max_{\mathbf{w}, b} \min_{i=1,\ldots,l} \left[y_i \frac{(\mathbf{w} \cdot \mathbf{x}_i + b)}{|\mathbf{w}|} \right]. \tag{5}$$

The particular vectors that are found to be nearest the hyperplane are called *support vectors* and are the central result of the approach. We note that the parameters describing the hyperplane, $\mathbf{w}$ and b, can be scaled by a constant without changing the hyperplane. To remove this ambiguity we choose a canonical form of the hyperplane by scaling $\mathbf{w}$ and b such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \begin{cases} = & 0 \quad \text{if } i \text{ is a support vector} \\ > & 0 \quad \text{if } i \text{ is not a support vector,} \end{cases}$$

or

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \ldots, l, \tag{6}$$

With this normalization the distance of the hyperplane to the nearest feature vector is $|\mathbf{w}|^{-1}$. When used in Eq. (5) we obtain

$$\max_{\mathbf{w}, b} |\mathbf{w}|^{-1}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \ldots, l.$$

It is convenient to replace maximization of $|\mathbf{w}|^{-1}$ with the equivalent minimization of $\frac{1}{2}|\mathbf{w}|^2$, where the factor of $\frac{1}{2}$ is cosmetic.

In summary we have: To find the optimal hyperplane for separable data, solve the *Quadratic Optimization* problem given by

$$\begin{array}{l} \min\limits_{\mathbf{w}, b} \ \frac{1}{2}|\mathbf{w}|^2 \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \ldots, l. \end{array} \tag{7}$$

2.3. Lagrange Undetermined Multipliers

The quadratic optimization problem in Eq. (7) can be simplified so as to replace the inequalities with a simpler form by transforming the problem to a *dual* space representation using Lagrangian multipliers. The motivation for doing this comes from a method invented by Lagrange in mechanics. We give a short digression to motivate the subsequent transformations.

Suppose we have a a potential energy function of dynamical variables, where the dynamical variables are restricted to certain ranges. For example a mass, m, which can move only vertically in a gravitational field g is attached to a string of length l. The mass has potential energy mgz, but there are also the constraints $z \leq l$, and $z \geq -l$. We desire to find the equilibrium position (minimum energy) while taking into effect the constraints. Lagrange suggested adding additional terms to the original energy function that represent the forces that come into existence only when a dynamical variable cannot change freely because the constraint extreme is reached (the inequality becomes an equality). In our example the mass m feels the force of gravity *and* a constraint force when $z = -l$ or $z = l$. We can either explicitly restrict the range of values to $-l \leq z \leq l$, or allow z to vary over $[-\infty, \infty]$ and put into the potential terms which apply constraint forces which are zero unless $z \leq -l$ or $z \geq l$. In this way the constraints are built into the energy function and the dynamical variables are no longer subjected to explicit constraints, though now we must solve for the additional forces as part of the problem.

2.4. The Dual Optimization Problem for Separable Data

With this as motivation we absorb the constraints into the minimization problem by defining Lagrange undetermined multipliers (the "constraint forces" discussed above),

$$\lambda_i \geq 0 \quad i = 1, \ldots l. \tag{8}$$

Defining our extended potential to be

$$\mathcal{L}(\mathbf{w}, b, \lambda_1, \ldots, \lambda_l) = \frac{|\mathbf{w}|^2}{2} - \sum_{i=1}^{l} \lambda_i \; [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1], \tag{9}$$

the optimization problem becomes

$$\begin{aligned} &\max_{\lambda_1 \ldots \lambda_l} \; \min_{\mathbf{w}, b} \; \mathcal{L}(\mathbf{w}, b, \lambda_1, \ldots, \lambda_l) \\ &\lambda_i \geq 0 \quad i = 1, \ldots, l. \qquad (10) \\ &y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \ldots, l. \end{aligned}$$

By choosing $\lambda_i \geq 0$ in Eq. (8) and putting the constraints in with a minus sign in Eq. (10) we must maximize over λ_i. More can be said concerning the solution of this extended problem. The Lagrange undetermined multipliers are zero only when the constraint is an equality. This is because in putting in the Lagrange multipliers into the minimization problem, they only makes a contribution when a constraint equality is reached. Thus

$$\lambda_i \; [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0 \quad i = 1, \ldots l, \tag{11}$$

which is called a complementarity condition. Assuming that $\mathcal{L}$ is a differentiable function of $\mathbf{w}, b$ we then have the further condition for a minimum in $\mathbf{w}, b$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = 0. \tag{13}$$

A more formal presentation is given by Fletcher[14] and a very readable recent account is given by Boyd and L. Vandenberghe.[15] Eqs. (6), (8), (11), (12), (13) are called the Karush-Kuhn-Tucker (KKT) optimality conditions and a formal derivation and interpretation can be found in these references.

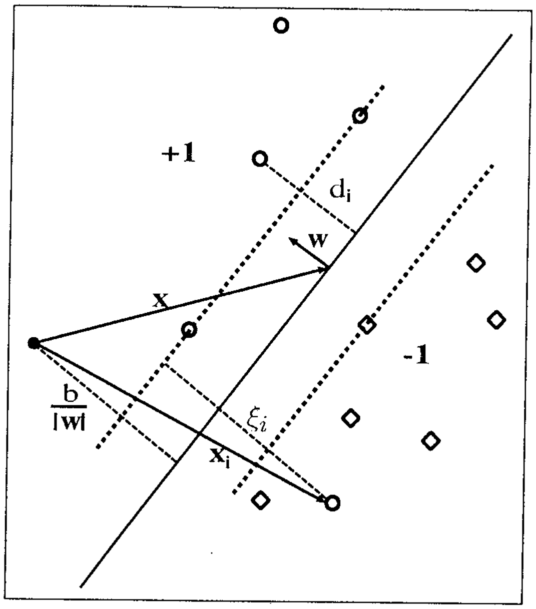

Figure 2. Schematic of non-separable data in $\mathbf{R}^2$. The circles are feature vectors in class $+1$ and the diamonds are feature vectors in class -1. There is one feature vector that is not separable.

Performing the derivatives in Eqs. (12), (13) we have

$$\mathbf{w} = \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i \tag{14}$$

$$\sum_{i=1}^{l} \lambda_i y_i = 0, \tag{15}$$

which when substituted in Eq. (9) allows us to eliminate $\mathbf{w}$ and b in Eq. (10) to obtain an equivalent quadratic optimization problem. This is called the *dual* problem optimization and it has simplified constraints:

$$\begin{aligned} &\max_{\lambda_1 \ldots \lambda_l} \left[-\tfrac{1}{2} \textstyle\sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i y_i (\mathbf{x}_i \cdot \mathbf{x}_j) y_j \lambda_j + \sum_{i=1}^{l} \lambda_i \right] \\ &\lambda_i \geq 0 \quad i = 1, \ldots, l \\ &\textstyle\sum_{i=1}^{l} \lambda_i y_i = 0. \end{aligned} \tag{16}$$

In the process of obtaining a solution, we expect that some of the λ_i will be 0, and the remaining ones will be associated with the support vectors. From the solution for the λ_i we obtain $\mathbf{w}$ from Eq. (14), and b from Eq. (11) for any $\lambda_i > 0$. Methods of solving the optimization problem are taken up in the section on implementation.

2.5. Non-separable Data

Cortes and Vapnik generalized the optimal margin methods to non-separable data,[16 17] which we discuss next. With reference to Fig. 2 the problem is now that there is no way to place a hyperplane such that we can separate the training data into two classes.

Cortes and Vapnik,[16 17] gave the following solution and named it the soft margin classifier. Relax the restriction that every training vector of a given class lie on the same side of the optimal hyperplane by introducing new variables,

ξ_i, $i = 1, \ldots, l$, that take the values $\xi_i \geq 0$, This generalizes Eq. (6) to be

$$y_i(\mathbf{w} \cdot \mathbf{x}_j + b) - 1 + \xi_i \geq 0 \quad i = 1, \ldots, l. \tag{17}$$

Then add a new term, $C \sum_{i=1}^{l} \xi_i$ (C is a positive constant $\infty > C > 0$), to Eq. (7) that balances the contribution of minimizing $\frac{1}{2}|\mathbf{w}|^2$ with penalizing solutions for which ξ_i get large. The non-separable optimization problem is then

$$\begin{array}{ll} \underset{\mathbf{w}, b, \xi_1, \ldots \xi_l}{\mathbf{min}} & \left[\frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^{l} \xi_i\right] \\ & \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0 & i = 1, \ldots, l \\ & \\ \xi_i \geq 0 & i = 1, \ldots, l. \end{array} \tag{18}$$

Note that as $C \to \infty$ the effect of any ξ_i deviating from 0 is increasingly more costly to the minimization. Thus in the limit $C \to \infty$ the optimization reduces to the formulation for the separable case. Note that $C \to 0$ is *not* the separable case, because then there is no effect on the cost if $\xi_i > 0$, and the optimal hyperplane would be that one that placed itself at the midpoint of those two feature vectors, one from each class, with the largest separation.

As in the separable case, a dual form can be obtained using two sets of Lagrange multipliers, $\lambda_i, \mu_i, i = 1, \ldots, l$, to handle the two sets of constraints in Eq. (18). We obtain

$$\begin{aligned} &\mathcal{L}(\mathbf{w}, b, \lambda_1, \ldots, \lambda_l, \mu_1, \ldots, \mu_l) = \\ &\left[\frac{|\mathbf{w}|^2}{2} + C \sum_{i=1}^{l} \xi_i\right] - \sum_{i=1}^{l} \lambda_i \; [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{l} \mu_i \xi_i. \end{aligned} \tag{19}$$

Assembling the constraint inequalities, the properties of the undetermined multipliers, and assuming $\mathcal{L}$ is a differentiable function which we seek to minimize over $\mathbf{w}$, b, and λ_i for $i = 1, \ldots, l$, we have the KKT conditions for the non-separable problem:

$$\begin{aligned} [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] &\geq 0 \quad i = 1, \ldots, l && (20) \\ \xi_i &\geq 0 \quad i = 1, \ldots, l && (21) \\ \lambda_i &\geq 0 \quad i = 1, \ldots, l && (22) \\ \mu_i &\geq 0 \quad i = 1, \ldots, l && (23) \\ \lambda_i \, [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] &= 0 \quad i = 1, \ldots, l && (24) \\ \mu_i \xi_i &= 0 \quad i = 1, \ldots, l && (25) \\ \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i &= 0 && (26) \\ \frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{l} \lambda_i y_i &= 0 && (27) \\ \frac{\partial \mathcal{L}}{\partial \xi_i} = C - \lambda_i - \mu_i &= 0 \quad i = 1, \ldots, l && (28) \end{aligned}$$

Eqs. (20), (21) are the constraint inequalities, Eqs. (22), (23) are part of the definition of the Lagrange undetermined multipliers, Eqs. (24), (25) are the complementarity conditions of the Lagrange multipliers, and Eqs. (26), (27), (28) are minimization conditions for a differentiable function. When Eqs. (25), (26), and (27) are substituted in Eq. (19) we obtain the dual problem

$$\begin{array}{l} \underset{\lambda_1 \ldots \lambda_l}{\mathbf{max}} \; \left[-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i y_i (\mathbf{x}_i \cdot \mathbf{x}_j) y_j \lambda_j + \sum_{i=1}^{l} \lambda_i\right] \\ \\ C \geq \lambda_i \geq 0 \quad i = 1, \ldots, l \\ \\ \sum_{i=1}^{l} \lambda_i y_i = 0 \quad i = 1, \ldots, l \end{array} \tag{29}$$

Note the only difference from the dual of the separable case, Eq. (16), is that the "constraint forces", λ_i are bounded above by C, reflecting the fact that the original inequality constraint, Eq. (6), holds only while $\xi_i = 0$ and then becomes *soft* when $\xi_i > 0$, which implies the constraint force saturates. Thus the term *soft margin* for this approach to the non-separable case has intuitive meaning. Note that Eq. (28) combined with Eq. (25) gives $\xi_i = 0$ if $\lambda_i < C$ – the constraint force is not saturated. The terms with $\lambda_i = C$ label the non-separable points.

Knowing the solutions λ_i, we can find $\mathbf{w}$ for the hyperplane using Eq. (26), and b from any one or more solutions for which $C > \lambda_i > 0$, using Eq. (24) with $\xi_i = 0$.

2.6. Kernel Method

Up to this point we have only dealt with classification as a linear function of the training data – the decision surface is a hyperplane defined by linear equations on the training data. However, it can be the case that no hyperplane exists to separate the data. The non-separable method provides one way to deal with this. As an alternative we would like a way to build a *non-linear* decision surface. An extremely useful generalization which can give non-linear decision surfaces and improved separation of the training data is possible using the following idea, first introduced by Aizerman, Braverman and Rozoner,[18] and incorporated into machine learning as part of the Support Vector Machine by Boser, Guyon, and Vapnik.[1]

Note the way that the training data enters the optimization problems, Eqs. (16), (29), is as dot products. Suppose that we map the feature vectors, $\mathbf{x} \in \mathbf{R}^n$ into a higher dimensional Euclidean space, $\mathcal{H}$, by means of a non-linear vector function $\mathbf{\Phi} : \mathbf{R}^n \mapsto \mathcal{H}$. Then we may again pose the optimal margin problem in the space $\mathcal{H}$ by replacing $\mathbf{x_i} \cdot \mathbf{x_j}$, by $\mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j})$. Then, as before, solve the optimization problem for the λ_i. This finds the support vectors among the transformed vectors, $\mathbf{\Phi}(\mathbf{x_i})$, by association with the $\lambda_i > 0$. We then use these to build the classifier function:

$$f(\mathbf{x}, \lambda_1, \ldots, \lambda_l) = sgn\left(\sum_{i=1}^{l} \lambda_i y_i \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x}) + b\right). \tag{30}$$

Now suppose there exists a *kernel function* K such that

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j}), \tag{31}$$

then everywhere $\mathbf{x_i} \cdot \mathbf{x_j}$ occurred, we could replace it with $K(\mathbf{x_i}, \mathbf{x_j})$. We need not explicitly compute $\mathbf{\Phi}(\mathbf{x})$, which could be computationally expensive, but only need compute the kernel functions. In fact we need not have an explicit representation of $\mathbf{\Phi}$ at all, but only K. The restrictions on what functions can qualify as kernel functions is discussed in Burges.[8]

What is gained is that we have moved the data into a larger space where the training data may be spread further apart and a larger margin may be found for the optimal hyperplane. In the cases where we can explicitly find $\mathbf{\Phi}$, then we can use the inverse of $\mathbf{\Phi}$ to construct the non-linear separator in the original space. Clearly there is a lot of freedom in choosing the Kernel function and recent work has gone into the study of this idea both for SVM and for other problems.[19]

With respect to the curse of dimensionality, we never explicitly work in the higher dimensional space, so we are never confronted with computing the large number of vector components in that space.

For the results presented below, we have used the inhomogeneous polynomial kernel function

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d, \tag{32}$$

with $d = 7$, though we found little difference in our results for $d = 2, \ldots, 6$. The choice of the inhomogeneous polynomial kernel is based on other workers success using this Kernel function in solving the handwritten digit problem.[1]

In fact there are principled ways to choose among kernel functions and to choose the parameters of the kernel function. Vapnik[2] has pioneered a body of results from probability theory that provide a principled way to approach these questions in the context of the Support Vector Machine.

Class Name	Number of Ground Truth Vectors	Number of Training Vectors	Number of Test Vectors
Corn-notil	1008	201	807
Soybean-notill	727	145	582
Soybean-mintill	1926	385	1541
Grass-Trees	732	146	586

Table 1. Data description of the Indian Pines subset scene.

2.7. Multi-Class Classifiers

Two simple ways to generalize a binary classifier to a classifier for K classes are:

1. Train K binary classifiers, each one using training data from one of the K classes and training data from all the remaining $K-1$ classes. Apply all K classifier to each vector of the test data, and select the label of the classifier with the largest margin, the value of the argument of the *sgn* function in Eq. (30).

2. Train $\binom{K}{2} = K(K-1)/2$ binary classifiers on all pairs of training data. Apply all $K(K-1)/2$ binary classifiers to each vector of the test data and for each outcome give one vote to the winning class. Select the label of the class with the most votes. For a tie, apply a tie breaking strategy.

We chose the second approach, and though it requires building more classifiers, it keeps the size of the training data smaller and is faster for training.

3. HYPERSPECTRAL DATA

In this work, hyperspectral data was obtained from the AVIRIS imaging spectrometer which has (on the ER-2 aircraft) a ground pixel size of 17 m × 17 m and a spectral resolution of 224 channels, covering the range from 400 nm to 2500 nm centered at 10 nm intervals. We focus on a part of data taken in June 12, 1992 in the northern part of Indiana, U.S. This data[20] has been studied by D. Landgrebe and students, and his website has a companion paper[21] describing its analysis by a free software package.[22] The data consists of 145 × 145 pixels by 220 bands that has been approximately converted from the radiance measured at the sensor to the reflectance, which is an intrinsic property of the surface.[23] Data from bands where there is a large amounts of water absorption in the atmosphere have been replaced by a constant value.

The scene consists of about two-thirds agriculture, and one-third forest or other natural perennial vegetation. There are two major dual lane highways, a rail line, as well as some low density housing, other built structures, and smaller roads. Since the scene is taken in June some of the crops present, corn, soybeans, are in early stages of growth with less than 5% coverage. The ground truth available is designated into sixteen classes and is not all mutually exclusive.

In order to compare to the recent results of S. Tadjudin and D. Landgrebe,[24] [25] we have studied two scenes also used in their work.

1. A part of the 145 × 145 scene, called the subset scene, consisting of pixels $[27-94] \times [31-116]$ for a size of 68×86. [Upper left in the original scene is at $(1,1)$]. There is ground truth for over 75% of this scene and it is comprised of the three row crops, Corn-notill, Soybean-notill, Soybean-mintill, and Grass-Trees. Table 1 gives further details.

2. The full 145 × 145 scene for which there is ground truth covering 49% of the scene and it is divided amoung 16 classes ranging in size from 20 pixels to 2468 pixels.

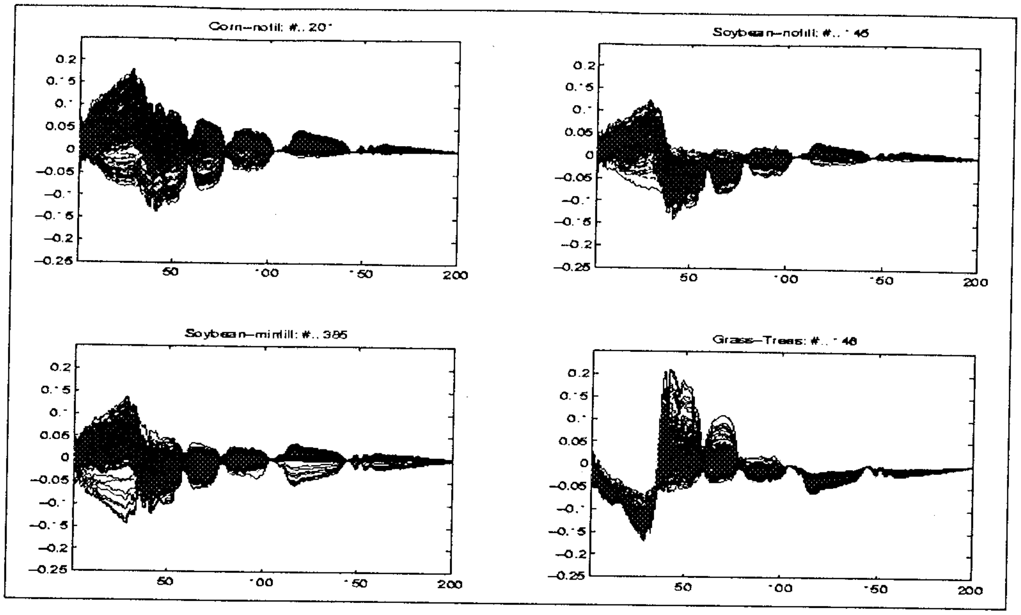

Figure 3. Training data for classification in the subset scene. The data has been centered.

Following Tadjudin and Landgrebe's work, we have also reduced the number of bands to 200 by removing bands covering the region of water absorption: $[104-108], [150-163], 220$. For each band at each pixel in the subset scene, the data was rescaled from the input two byte short integer by dividing by 10000 to make a floating point number in the range $[0, 1]$.

Then the data was *centered*, which means that for the whole scene, for each band, the mean was found and this was subtracted from all the data in that band. This distributes the data around 0 and considerably speeds up the optimization routines. Fig. 3 shows this centered data for the four classes. Note that there is substantial overlap between Corn-notill, Soybean-notill, and Soybean-mintill, while Grass-Trees is well separated from the other three classes.

4. APPLICATION OF SVM TO HYPERSPECTRAL DATA AND RESULTS

4.1. SVM Implementation

We have implemented the Support Vector Machine first by building on free Matlab software available from S. Gunn[9] and de Ridder.[11] For the separable case, Gunn has shown that the problem may be recast as non-linear least squares and he has provided a routine to perform this. However for the non-separable case, quadratic optimization is required. We first used the quadratic optimizer package available from Matlab, which can be slow for larger data sets. Subsequently we have adapted the software package from T. Joachims,[26] which works with A. Smola's optimization code,[27] and which takes advantage of the KKT conditions in the non-separable case to provide good performance for large training sets.

4.2. Classifier Results for the Subset Scene

From the subset scene, a random sample of 20% of the pixels was chosen from the known ground truth of the four classes: Corn-notill, Soybean-notill, Soybean-mintill, Grass-Trees. This was used to train six binary classifiers, one for each pair of classes. The trained classifiers were then applied to the remaining 80% of the known ground pixels in the scene, with the voting strategy above. Ties were broken by a random choice.

This procedure was repeated in five trials using a different random seed for the selection of the 20% of the training data. Table 2 shows the contingency table for a typical trial. For a trial, the overall performance is the sum of the number of samples correctly labeled for each class in the test set divided by the total number of samples in the test

Class Name	Percent Correct	Number of Samples	Corn-notill	Soybean-notill	Soybean-mintill	Grass-Trees
Corn-notill	94.3	807	**761**	4	38	4
Soybean-notill	95.7	582	1	**557**	23	1
Soybean-mintill	96.1	1541	39	21	**1481**	0
Grass-Trees	100	586	0	0	0	**586**
TOTAL	96.3	3516	801	582	1542	591

Table 2. A typical result for the Indian Pines subset scene. The entries in rows 2 - 5 and colums 4 - 7 are the contingency table for results for the subset scene. For each horizontal line labeled on the left by class name A, the entries under the four classnames B_1, B_2, B_3, B_4 give the distribution of the all the testing ground truth pixels of class A into the four classes. Bold face numbers are correctly classified samples. A perfect result would have all zeros except on the diagonal. The overall performance is computed by the ratio of the sum of the diagonal elements to the sum of all entries of the contingency table.

Trial	Overall	Class correct(%)			
	correct (%)	Corn-notill	Grass-Trees	Soybean-notill	Soybean-mintill
1	96.3	94.3	100.0	96.1	95.7
2	95.8	92.8	99.8	95.7	96.0
3	96.1	95.2	99.8	95.7	94.7
4	95.5	94.7	100.0	95.1	93.5
5	95.6	95.7	99.8	94.8	93.3
Average	95.9	94.5	99.9	95.5	94.6

Table 3. Summary of trials on SVM classifier for the Indian Pines subset scene.

set. Table 3 summarizes the five trials. Note that Grass-Trees was classified almost completely correctly as might be expected from the lack of overlap in the training data.

The results across the five trials were consistent within one percent and the average performance was 96%, which is somewhat better than 93% from recent results of Tadjudin and Landgrebe[24] for their best classifier, bLOOC+DAFE+ECHO, on the same data. Table 4 summarizes the results for the subset scene comparing the SVM classifier, bLOOC+DAFE+ECHO, and a simple Euclidean classifier.[24] The Euclidean classifier uses only the first order statistics of the training data. Its poor performance is expected for this data due to the overlap of the classes. The details of the bLOOC+DAFE+ECHO classifier is covered in Tadjudin and Landgrebe.[24]

METHOD	PERFORMANCE	
	Subset Scene	Full scene
Support Vector Machine	95.9%	87.3%
bLOOC+DAFE+ECHO	93.5%	82.9%
Euclidean	66.7%	48.2%

Table 4. A comparison of results for the Indian Pines subset scene (68 x 86 pixels) and the full scene (145 x 145 pixels). The results labeled bLOOC+DAFE+ECHO and Euclidean are taken from the recent work of Tadjudin and Landgrebe[2425] and represent the best classifier results reported for this scene in that work. Also note that their results for the full scene are for 17 classes compared to our 16. The difference is explained in the text. All training is based on 20% of the ground truth and testing on the remaining 80%.

4.3. Classifier Results for the Full Scene

Results for the full scene were produced using only one trial. Here we used the sixteen ground truth classes given in Landgrebe's data.[20] We made a random selection of 20% of the ground truth data and tested on the remaining 80%. A difference with the data and results reported by Tadjudin and Landgrebe,[24 25] is that they studied the scene using 17 classes whereas we only used 16. The difference being that they further resolved the class Soybeans-notill into two subclasses of Soybeans-notill based on fields that were in different locations in the full scene. The results are reported in Table 4 show the Support Vector Machine to be somewhat better, although the difference in the number of classes may have some effect.

5. CONCLUSIONS

We have described a new approach to building a supervised learning machine called the Support Vector Machine, and applied it to classify hyperspectral remote sensing data. The inherent high dimensionality of this data is challenging for traditional classifiers, due to the Hughes effect, and usually a feature selection preprocessing step is performed to first reduce the dimensionality of the data. The Support Vector Machine does not suffer from this handicap, and is thus suitable for use with hyperspectral data. The results we have obtained show it to be competitive with other recently developed classifiers for hyperspectral data when applied to the same data sets.

In this work the choice of kernel function is ad-hoc, as are the choices of the kernel function parameter d, and the separability parameter, C. However, the Support Vector Machine can be placed into the Structural Risk Minimization approach of Vapnik,[2] and using rigorous bounds from recent results from probability theory, a more rigorous approach can be taken to choosing these parameters.

Also we note that all the results we have shown are completely in the spectral domain and no aspect of the spectral coherence of the data has been used. The results would be identical if all the classifier bands were permuted consistently throughout the data. And, we have not utilized the *spatial coherence* of the data. We note recent studies on the classification of hand written digits[28] show that performance gains can be made by incorporating prior knowledge into the construction of the Support Vector Machine and we believe similar gains can be made for classifying hyperspectral data using the coherence in the data.

REFERENCES

1. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, ACM, June 1992.
2. V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.
3. R. O. Green, ed., *Summaries of the Sizth Annual JPL Airborne Earth Science Workshop. Volume I. AVIRIS Workshop*, 1996.
4. J. Okkomen, T. Hyvärinen, and E. Herrala, "Aisa airborne imaging spectrometer-on its way from hyerspectral research to operative use," in *Proc. of the Third International Airborne Remote Sensing Conference and Exhibition.I*, pp. 189–196, 1997.
5. R. Holasek, F. Portigal, G. Mooradian, M. Voelker, D. Even, M. Fené, P. Owensby, and D. Breitwieser, "HSI mapping of marine and cosatal environments using the advanced airborne hyperspectral imaging system (aahis)," in *Algorithms for Multispectral and Hyperspectral Imagery III*, pp. 169–180, SPIE, 1997.
6. G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory* **14**(1), 1968.
7. D. Landgrebe, "Information extraction principles and methods for multispectral and hyperspectral image data," in *Information Processing for Remote Sensing*, H. Chen, ed., ch. 1, World Scientific Publishing Co., 1999. Also available at http://dynamo.ecn.purdue.edu/l̃andgreb/publications.html.
8. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **to appear**, 1998. Also available at http://svm.research.bell-labs.com/SVMdoc.html.
9. S. Gunn, "Support vector machines for classification and regression," tech. rep., Image Speech and Intelligent Systems Group, University of Southampton, November 1997. This site contains the tutorial and matlab code which impliments SVM's.
10. M. S. Schmidt, "Identifying talkers with support vector networks," in *Sydney International Statistical Congress*, (Sydney, Australia), July 1996. Available at http://www.stat.uga.edu/ lynne/symposium/paper1i3.ps.gz.

11. D. Tax, D. de Ridder, and R. P. Duin, "Support vector classifiers: a first look," in *Proceedings of the Third Annual Conference of the Advanced School for Computing and Imaging*, June 1997. Available at http://www.ph.tn.tudelft.nl/ davidt/papers.html. Matlab source code is also available at http://valhalla.ph.tn.tudelft.nl/feature_extraction/source/svc/.
12. V. N. Vapnik and A. J. Chervonenkis, *Theory of pattern Recognition*, Nauka, 1974. In Rusian.
13. V. N. Vapnik, *Estimation of dependencies based on empirical data*, Springer, 1982.
14. R. Fletcher, *Practical methods of optimization (2nd edition)*, J. Wiley, 1987.
15. S. Boyd and L. Vandenberghe, "Convex optimization," 1997. Course notes from Stanford University for EE364, Introduction to Convex Optimization with Engineering Applications. Available at http://www.stanford.edu/class/ee364/.
16. C. Cortes, *Prediction of Generalization Ability in Learning Machines*. PhD thesis, University of Rochester, 1995.
17. C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning* **20**, pp. 1–25, 1995.
18. M. Aizerman, E. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition," *Automation and Remote Control* **25**, pp. 821–837, 1964.
19. S. A. J, B. Schölkopf, and R. J. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks* **to appear**, 1998.
20. D. Landgrebe, "Indian pines aviris hyperpsectral reflectance data: 92av3c," 1992. A 145 × 145 pixel and 220 band subset in BIL format of reflectance data that has been scaled to lie in [0,10000] as 2bit signed integers. It is available at ftp://shay.ecn.purdue.edu/pub/biehl/MultiSpec/92AV3C. Ground truth is also available at ftp://shay.ecn.purdue.edu/pub/biehl/PC_MultiSpec/ThyFiles.zip The full data set is called Indian Pines 1 920612B as listed in the AVIRIS JPL repository (http://makalu.jpl.nasa.gov/locator/index.html). The flight line is [Lat_Start: 40:36:39 Long_Start: -87:02:21 Lat_End: 40:10:17 Long_End: -87:02:21 Start_Time:19:42:43 End_Time:19:46:2].
21. D. Landgrebe, "Multispectral data analysis: A signal theory perspective," 1998. Available at http://dynamo.ecn.purdue.edu/ biehl/MultiSpec/Signal_Theory.pdf.
22. D. Landgrebe and L. Bieh, "Multispec," 1998. A free hyperspectral analysis package is available for the Mac and PC at http://dynamo.ecn.purdue.edu/ biehl/MultiSpec/.
23. Center for Study of Earth from Space (CSES), University of Colorado, Boulder, CO, *ATmospheric REMoval Program (ATREM)*, version 3.0 ed., July 1997. This software is avaialable for anonymous ftp at cses.colorado.edu:/pub/atrem.
24. S. Tadjudin and D. Landgrebe, *Classification of High Dimensional Data with Limited Training Samples*. PhD thesis, School of Electrical Engineering and Computer Science, Purdue University, May 1998. available as TR-ECE-98-9 from http://dynamo.ecn.purdue.edu/ landgreb/Saldju_TR.pdf.
25. S. Tadjudin and D. Landgrebe, "Covaraince estimation for limited training samples," in *Int. Geoscience and Remote Sensing Symposium*, IEEE, (Seattle, WA), July 1998. Available at http://dynamo.ecn.purdue.edu/ landgreb/SaldjuCovarEst.pdf.
26. T. Joachims. The SVM^{light} package is written in gcc and distributed for free for scientific use from ftp://ftp-ai.cs.uni-dortmund.ed/FORSCHUNG/ VERFAHREN/SVM_LIGHT/svm_light.eng.html It can use one of several quadratic optimizers. For our application we used the A. Smola's PR_LOQO package.[27]
27. A. Smola. The PR_LOQO quadratic optimization package is distributed for research purposes by A. Smola at http://svm.first.gmd.edu.de/software/loqosurvey.html.
28. B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," in *Advances in Neural Information Processing Systems,10*, M. I. Jordan, M. Kearns, and S. A. Solla, eds., pp. 640–646, MIT Press, 1998.