# Linear Spectral Mixture Models and Support Vector Machines for Remote Sensing

Martin Brown, Hugh G. Lewis, and Steve R. Gunn

***Abstract*—Mixture modeling is becoming an increasingly important tool in the remote sensing community as researchers attempt to resolve subpixel, area information. This paper compares a well-established technique, linear spectral mixture models (LSMM), with a much newer idea based on data selection, support vector machines (SVM). It is shown that the constrained least squares LSMM is equivalent to the linear SVM, which relies on proving that the LSMM algorithm possesses the "maximum margin" property. This in turn shows that the LSMM algorithm can be derived from the same optimality conditions as the linear SVM, which provides important insights about the role of the bias term and rank deficiency in the pure pixel matrix within the LSMM algorithm. It also highlights one of the main advantages for using the linear SVM algorithm in that it performs automatic "pure pixel" selection from a much larger database. In addition, extensions to the basic SVM algorithm allow the technique to be applied to data sets that exhibit spectral confusion (overlapping sets of pure pixels) and to data sets that have nonlinear mixture regions. Several illustrative examples, based on an area-labeled Landsat TM dataset, are used to demonstrate the potential of this approach.**

***Index Terms*—Area estimation, linear spectral mixture models, support vector machines.**

## I. Introduction: Mixed-Pixel Classification

THE MAPPING of land cover and land use is a key application of remotely sensed data [18]. Information in the form of area estimates and crop yields (during harvest periods, for example) has particular interest for government agencies responsible for economic, social or environmental policy. Typically, these agencies require complete and accurate estimates of area. When remotely sensed data has been considered for monitoring the landscape, it has been suggested that an acceptable accuracy limit for land cover maps is 85% [2]. In many areas, vegetation and crops occur within small land parcels and the remotely sensed data that is used requires a high spatial resolution to produce land cover maps of an acceptable accuracy. Government data collection methods have utilized aerial photography for this reason, but the frequency of acquisition, financial, and practical constraints involved in the use of aerial photography has since led to the investigation of satellite imagery to supplement these existing data collection methods [17], [35].

Traditionally, the production of maps from satellite imagery assumes that each pixel of the image can be assigned to a single land cover class. A crisp classification process then aims to attach one of a number of mutually exclusive labels within a closed world environment. In this context, the algorithms are formulated for two reasons: discrimination, which is the practice of dividing up the feature space into a number of nonoverlapping regions, and statistical pattern recognition, which is the practice of modeling the posterior (or prior) distributions for a predefined number of classes.

In such problems, it is typically assumed that the process of generating observable data may be decomposed into a number of independent classes $C_j$, each of which is a subprocess generating data $\boldsymbol{x}$ according to a particular class-conditional density for that class. When the class conditional densities do not overlap, a discrimination approach is appropriate, but when they do overlap, a statistical pattern recognition approach that models the posterior probabilities $p(C_j|\boldsymbol{x})$ is more suitable. The statistical approach is generally necessary, as the chosen feature vector does not contain enough information to separate all the classes completely.

Implicit in the traditional classification process is the concept that each feature vector should be mapped into one of the classes of interest. However, the spatial characteristics of satellite sensors are such that a significant amount of confusion can arise from variable land cover in the instantaneous field of view (IFOV). In addition, energy transfer into and from neighboring IFOVs leads to further pixel ambiguity. In order to meet the accuracy requirements imposed upon the production of land cover maps, it is necessary to resolve these ambiguities. The process of mixed-pixel classification is therefore to model the class mixing proportions [15] (percentage ground cover area) rather than estimate the probability that the signature corresponds to a particular class label. In practice, it is often found that the posterior probabilities are correlated with the area predictions but that they represent orthogonal information. Errors due to statistical uncertainty and measurements based on area mixing, respectively. The land parcel associated with each pixel cannot be labeled exactly using any of the $c$ classes, even when the input information is perfect [23].

This paper compares the conventional constrained least squares linear spectral mixture modeling (CLS LSMM) technique [1], [11], [15], [30], which has been developed in the remote sensing community with the recently developed support vector machines (SVM) [24], [25], [33], which are based on the principle of optimal class separation. It is shown that the

Manuscript received: September 3, 1998; revised: July 28, 1999. This work was supported by the EU Framework IV FLIERS Project (ENV4-CT96-0305) and the EPSRC Osiris project (GR/K55110).

M. Brown is with Unilever Research, Port Sunlight, Bebington, U.K. (e-mail Martin.Q.Brown@unilever.com).

H. G. Lewis and S. R. Gunn are with the Image, Speech, and Intelligent Systems Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton, U.K. (email: hgl@ecs.soton.ac.uk; srg@ecs.soton.ac.uk).

Publisher Item Identifier S 0196-2892(00)06378-6.

CLS LSMM algorithm is equivalent to the linear SVM's when both models are designed using the same data set and the same thresholding and normalizing operations are applied. This is an important result for a number of reasons as it shows the following.

1) The linear SVM algorithm implements the same model as the CLS LSMM but also performs pure pixel selection automatically, from a potentially large data set.
2) The CLS LSMM algorithm satisfies the "margin maximization" property, which effectively constrains the mixing region's orientation when the number of classes (pure pixels) is less than or equal to the number of spectral bands.
3) The CLS LSMM algorithms can be derived from a different set of optimality constraints, similar to the SVM algorithm, where the margin maximization property is explicit.
4) The pure pixel matrix in the CLS LSMM can be rank deficient in certain circumstances and highlights the role of the bias term in the original LSMM.

In addition, the more general SVM framework provides techniques for dealing with cases where the potential sets of pure pixels for different classes overlap (due to spectral confusion), and also where the mixture regions could be nonlinear. Overlapping sets of pure pixels are dealt with by requiring the model to make as few "misclassifications" on the pure pixels as possible, yet also requiring the mixing region to be as large as possible. Classification errors are regarded as arising from the spectral confusion, whereas the mixing region represents the deterministic spectral signature mixing. These two competing constraints are weighted against each other according to a "regularization" parameter that limits the model's flexibility. The parameter's value can be fixed by the designer or optimized using a cross-validation technique. Nonlinear mixture regions can be formed using kernels such as radial basis functions, polynomials, piecewise polynomial splines and in certain cases sigmoidal nodes [8]. In each case, the input space is mapped to a higher dimensional "kernel-space" and the same processes are performed in this transformed space as normally occurs for the linear SVM algorithms. The range of different kernel nodes include several traditional nonlinear modeling algorithms as well as some neuronally inspired techniques. In all these cases, one of the key points about the SVM algorithms is that they perform automatic pure pixel selection from potential sets of pure pixels. In many general classification problems, it has been found that between 3 and 5% of the potential pure pixels are selected and that the overall performance of the system is at least as good as that obtained using conventional empirical modeling and classification algorithms [8].

## II. Linear Spectral Mixture Models

Spectral unmixing has been used as a technique for analyzing the mixture of components in remotely sensed images for almost 30 years, [15]. During that time, it has been used to study the rock and soil components collected on the Viking Mars missions [1] and to estimate ground and vegetation cover, [11], [30], [32], for instance. The technique is based on the assumptions that several primitive classes of interest can be selected, that each of these primitive classes has a unique spectral signature (a so called endmember or pure pixel), which can be identified and that the mixing between these classes can be adequately modeled as a linear combination of the spectral signatures. Each of these three assumptions may be satisfied to varying degrees, but it is worthwhile commenting on each.

Theoretically, the classes should be mutually exclusive (able to be completely discriminated from the other classes), and the chosen classes should be complete (closed world assumption). This is to ensure that the chosen classes cover every possible situation within the application domain, hence these classes are termed primitives, [23]. In practice, these assumptions cannot be met, otherwise 100% accurate classification results for conventional remote sensing pattern recognition would have been achieved. The closed world assumption is often approached by introducing general classes such as shade, cloud and others.

The identification of the pure pixel value is often difficult. The two methods that have been proposed in the literature are to measure the spectral properties in a laboratory and then modify the results to account for atmospheric and satellite processes and to select a "pure pixel" from a labeled image and use measured spectral values as the spectral signature for that class. For the former technique, the accurate modeling of atmospheric and satellite effects has been difficult to achieve, hence most algorithms use the empirical approach where the collected data sets implicitly contains information about the physical processes (forwards model) in satellite measurement system. The data set used to design the area-based classifier then consists of the identified pure pixels.

Assuming that a primitive class can be represented by a single vector of spectral values is often unrealistic. Atmospheric, temporal and spatial variations occur in most of the classes of interest and whilst the satellite imagery should be pre-processed as much as possible to remove these effects, variations in theoretical pure spectra are inevitable. Many of the classes of interest can be termed composite classes, made up from simpler primitive classes. This has been noted in the literature: "With the generalized classes used in linear mixing there is clearly the possibility that the resultant spectra may be generally unrepresentative of the majority of pixels in the study area," [31]. Predicting the mixture components of such composite classes is difficult [23] unless they satisfy certain conditions that state the composite classes are mutually exclusive and formed from the additive union of simpler primitive classes for which the linear mixing assumptions hold. In practice, this will never be completely true, so effort must be directed at measuring the effect of this on the predictions.

Finally, the linear mixing assumption limits the number of primitive classes, $c$, which can be identified by $c \leq n+1$, where $n$ is the number of measured spectral bands. When the composite classes of interest are formed from combinations of these primitive classes, the bound should still hold for the underlying primitive classes, if the predicted outputs are to be interpreted as mixing proportions. Hence, the bound on the number of composite classes of interest may in fact be much tighter as the spectral mixing occurs between the primitive classes, and this is a fundamental constraint on any set of derived composite classes.

This is illustrated in Fig. 4, where there exists three primitive classes and two composite classes.

### A. Algorithmic Implementation

Assuming there exist $c$ primitive[1] classes of interest and $n$ spectral bands/features, which are used to model the class mixture, the user must specify a $n \times c$ matrix $\boldsymbol{R}$, which contains the spectral features of the "pure pixels," one from each class. The assumed linear model is therefore of the form

$$\boldsymbol{x} = \boldsymbol{R}\boldsymbol{y} \tag{1}$$

where $\boldsymbol{x}$ is the vector of spectral inputs and $\boldsymbol{y}$ is the vector of mixing proportions. It is assumed that $\boldsymbol{R}$ is full-rank, i.e., all the pure pixels are linearly independent. This may or not be true and is further discussed in Section II-B.2. This specification has been termed a "classical estimator," as it describes how the satellite's measurements are related to the land cover mixing proportions [20], [26]. In this context, an "inverse model" specifies how the land cover components are related to the satellite's spectral measurements $\boldsymbol{y} = f(\boldsymbol{x})$.

Let $\boldsymbol{V}$ denote the covariance error matrix of the observations $\boldsymbol{x}$, so

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \epsilon$$
$$\boldsymbol{V} = E(\epsilon\epsilon^T)$$

Then the goal of predicting the class mixtures is formulated as minimizing the weighted sum squared error

$$J = (\hat{\boldsymbol{x}} - \boldsymbol{x})\boldsymbol{V}^{-1}(\hat{\boldsymbol{x}} - \boldsymbol{x}) \tag{2}$$
$$= (\hat{\boldsymbol{x}} - \boldsymbol{R}\boldsymbol{y})^T\boldsymbol{V}^{-1}(\hat{\boldsymbol{x}} - \boldsymbol{R}\boldsymbol{y}). \tag{3}$$

In addition, to ensure that estimated mixtures sum to unity, an additional linear constraint is introduced into the optimization goal, namely

$$\sum_{j=1}^{c} y_j = 1 \tag{4}$$

(an alternative set of constraints is described in Section III-B). This linear constraint can be combined with the quadratic error loss criteria to produce a closed-form, constrained least squares (CLS) estimate [9] of the mixing proportions

$$\boldsymbol{y} = \boldsymbol{C}^{-1}\boldsymbol{e} - \frac{\boldsymbol{C}^{-1}\boldsymbol{1}}{\boldsymbol{1}^T\boldsymbol{C}^{-1}\boldsymbol{1}}(\boldsymbol{1}^T\boldsymbol{C}^{-1}\boldsymbol{e} - 1) \tag{5}$$

where $\boldsymbol{C} = \boldsymbol{R}^T\boldsymbol{V}^{-1}\boldsymbol{R}$ is the weighted autocorrelation matrix, $\boldsymbol{e} = \boldsymbol{R}^T\boldsymbol{V}^{-1}\boldsymbol{x}$ is the weighted cross correlation vector, and $\boldsymbol{1}$ is a unity column vector. This is derived using a standard Lagrange analysis of a quadratic programming problem with linear constraints [9] and implicitly assumes that the inverse of the matrix $\boldsymbol{C}$ exists, a point further discussed in Section II-B.2. Note that the first term on the right hand side of this equation is simply a weighted least squares estimator and the second ensures that the estimates sum to unity by introducing a bias term and "correcting" the gains associated with each spectral band. In addition, the output mixing proportions can be re-expressed as

$$\boldsymbol{y} = \boldsymbol{C}^{-1}\left(\boldsymbol{I} - \frac{\boldsymbol{1}\boldsymbol{1}^T\boldsymbol{C}^{-1}}{\boldsymbol{1}^T\boldsymbol{C}^{-1}\boldsymbol{1}}\right)\boldsymbol{R}^T\boldsymbol{V}^{-1}\boldsymbol{x} + \frac{\boldsymbol{C}^{-1}\boldsymbol{1}}{\boldsymbol{1}^T\boldsymbol{C}^{-1}\boldsymbol{1}} \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix. This is obviously an affine inverse model

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \tag{7}$$

where the linear gain matrix $\boldsymbol{W}$ and the vector of biases $\boldsymbol{b}$ are given by equating terms with (6).

The partition of unity constraint in (4) does not ensure that the estimates lie in the unit interval however, and in order to impose this on the solution explicitly, a common practice is to set $y_j$ to 0 when $y_j < 0$ and then normalize the remaining estimates. There exists many variations on the basic CLS LSMM algorithm to re-estimate the mixing proportions when any are estimated to be negative or greater than one, but these only modify the mixtures in a limited domain and the relative merits of the different approaches lie outside the scope of this paper. Note however, that it may be useful to threshold the outputs when they are greater than 1 as well, prior to normalizing the remaining values, and this will be further discussed in Section III.

### B. Interpretation

The CLS LSMM equations describe how the class mixtures $\boldsymbol{y}$ are related to the measured spectral measurements $\boldsymbol{x}$. In this section, the structure of such mixtures is analyzed and some remarks are made about the way CLS LSMM algorithms are represented and implemented. First, some terminology needs to be defined. A class **core** is defined as the area in feature space where the pixels are full member of that class, i.e., the $j$th class' core is defined as

$$c_j = \{\boldsymbol{x}\colon \mu_j(\boldsymbol{x}) = 1\} \tag{8}$$

where $\mu()$ denotes the (area-based) degree of membership of that class.

The two contours formed by each unthresholded, non-normalized LSMM output, $y_j$, evaluated at 0 and 1, are parallel hyperplanes which pass through the pure pixels for the other classes and for the exemplar class, respectively. These contours are termed margin boundaries, where a "boundary" is defined as

$$\boldsymbol{b}_j^{\theta} = \{\boldsymbol{x}\colon y_j(\boldsymbol{x}) = \theta\}. \tag{9}$$

Each model is a linear function in its margin $m_j$, where the margin is defined as the volume in feature space which lies between the two contour extrema

$$m_j = \{\boldsymbol{x}\colon 0 < y_j(\boldsymbol{x}) < 1\}. \tag{10}$$

The area of linear mixing for the overall multi-output model is given by the intersection of the single output models' margins

$$m = \cap_{j=1}^{c} m_j. \tag{11}$$

When $c = n+1$, the intersection forms a closed simplex in $n$-dimensional feature space, as illustrated in Fig. 1. When the input

[1]This analysis also follows for composite classes formed from the addition of primitive classes that satisfy the appropriate constraints.

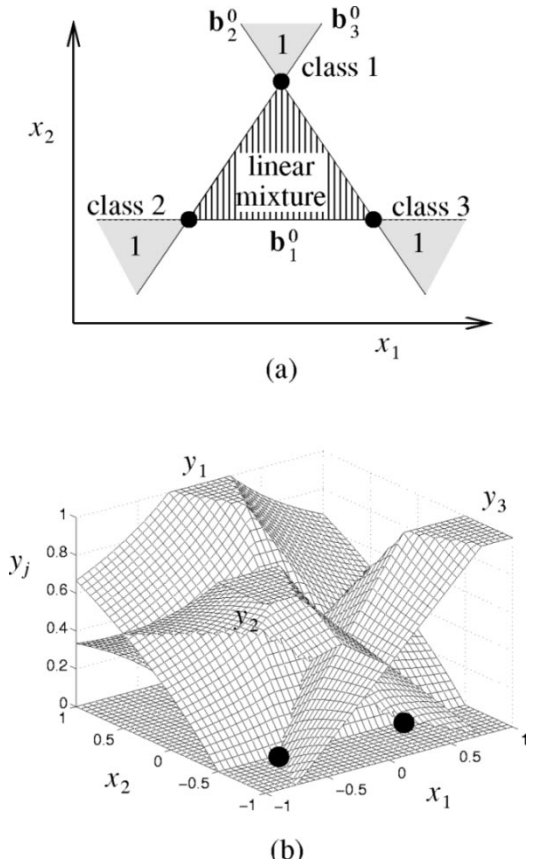


Fig. 1. Linear spectral mixture model for two inputs and three classes, where the pure pixels occur at (0, 0.5), (−0.5, −0.5), (0.5, −0.5), where the origin has been chosen to lie at the center of the simplex. The division of the feature space is shown in (a), while the corresponding mixture surface is shown in (b).

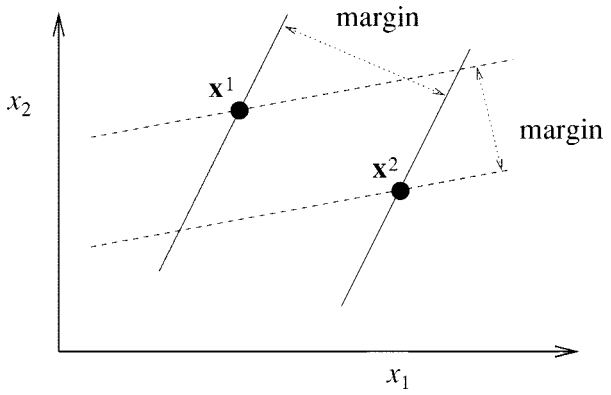


Fig. 2. Two potential mixing margins for a two-input, two-class problem with pure pixels $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$. The solid lines denote the zero and one boundaries for the solution, which maximizes the margin's width, yet the dashed lines are also a potential solution.

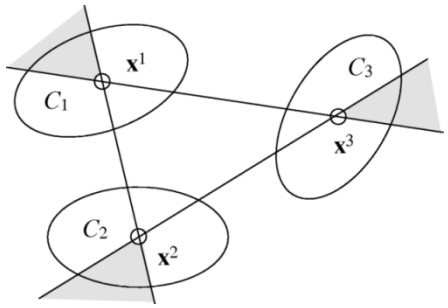


Fig. 3. Representing the pure class variation by the means when it is assumed that the variation is due to spectral confusion.

lies outside this simplex, the unthresholded output for at least one of the models is negative, and performing the thresholding and renormalizing operation means that the overall model is nonlinear in these regions. However, in many cases, it is nearly linear.

When $c < n+1$, the intersection of each of the individual margins produces an infinite linear mixture region, as illustrated in Fig. 2. It can be argued that the specification of the least squares constraint in (2) is under constrained, as the only information (data) that exists, for which errors can be calculated are the values of the pure pixels. It has been assumed that these are linearly independent, hence $J$ is identically zero at these points, independent of the value of $\boldsymbol{V}$. The sum to unity constraint ensures that the zero boundary of the first model is identical to the one boundary of the second (and vice versa), yet these boundaries could occur at almost any orientation as long as they are not equal. The solution produced by the Lagrange derivation (subject to the rank deficiency problem discussed in Section II-B.2, is shown as the solid lines, yet the dashed boundaries are also valid solutions. This is because the least squares constraint in (2) is redundant as it has been assumed that a solution exists ($\boldsymbol{R}$ is full rank), so the minimum value is always zero and a pseudo-inverse solution does not need to be performed (see Section II-B.3). However, it is argued in this paper that the margin produced by the Lagrange implementation is desirable as it maximizes the margin's size (width), a property which is fundamental to deriving the relationship with the linear SVM's in Section III.

*1) Pure Pixel Definition for Composite Classes:* In deciding how to deal with potential sets of pure pixels, it must be determined whether the variation is due to spectral confusion or inherent, measurable natural variation. Spectral confusion results from random perturbations around some true pure pixel value, and this should be treated as a stochastic modeling problem [15]. The stochastic perturbation of the pure pixels could be due to errors in the spectral measurements or due to natural variation within the scene which introduces a stochastic variation within the sensed spectral bands. This is the normal assumption made when the LSMM algorithms are derived [15], [30], and the optimal estimates for the pure pixels are the class means. The class variance can then be used to provide error estimates for the mixing proportion predictions.

When composite (meta) land cover class labels are used, there exist volumes of the feature space that can be considered as pure exemplars of that class, and no mixing should be modeled within these regions. Assuming that these regions can be modeled as linear mixtures between exemplar pure pixels is often inappropriate. Replacing the class distribution by just its mean introduces errors in the models' estimates. However, selecting pure pixels which lie on the boundary closest to the other classes is also inappropriate, as there is no mixing between the class cores and estimated class cores are substantially different from the original shapes. The concept of linear mixing in these cases

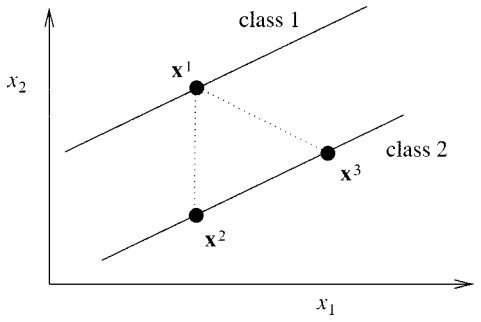


Fig. 4. Mixing margins for two composite classes and their corresponding primitives. Class 2 is composed of the union of $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$, and the solid lines denote the margin of the composite classes and the dotted lines denote the primitive classes' mixing margin.

is only valid if it is assumed that the composite classes are composed of mutually exclusive sets of so-called primitive classes [23], for which the pure pixel/linear mixing assumptions are appropriate ($c \leq n+1$ etc.), as illustrated in Fig. 4.

*2) Rank Deficiency and the Bias Parameter:* Implicit in the Lagrange derivation of the CLS LSMM algorithm, is the condition that $\boldsymbol{C}$ must be full rank. However, this is not true in certain cases. The most obvious is when $c = n+1$ as $\boldsymbol{C}$ is an $(n+1) \times (n+1)$ matrix, which is formed from the inner product of an $n \times (n+1)$ matrix. Hence, the rank of $\boldsymbol{C}$ is at most $n$. As a simple example, consider a single spectral input ($n = 1$), which has two primitive classes at $x^1 = -0.5$ and $x^2 = 0.5$ and $\boldsymbol{V} = \boldsymbol{I}$,

$$\boldsymbol{C} = \begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix} \tag{12}$$

which is obviously rank 1. In addition, this rank deficiency can occur even with what may appear to be well specified problems. Consider the case when $c = n = 2$, and one pure pixel is located at $\boldsymbol{x}^1 = [-0.5, -0.5]^T$, and another at $\boldsymbol{x}^2 = [0.5, 0.5]^T$

$$\boldsymbol{C} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix} \tag{13}$$

which again is obviously rank 1.

The reason why this occurs is due to the role of the bias term in the original set of linear equations. With the CLS LSMM, the bias term is implicitly incorporated by the sum to unity constraint, yet the bias term is often neglected when standard least squares solutions are proposed, [14], [30]. However, the authors would argue that it does change the solution as a bias term no longer exists in the original set of linear equations, and this has the effect that the spectral input $\boldsymbol{x} = \boldsymbol{0}$ always lies on the zero boundary of every mixing domain. This is always true if the LSMM equations are implemented without an implicit or explicit bias term.

One explicit method for incorporating a bias term in the sets of linear equations is to simply introduce an extra input $x_0$ (or equivalently $x_{n+1}$), which has a constant value of 1. This augments $\boldsymbol{R}$ by an extra row of ones, and it can be shown that [29] this does not alter the solution when the sum to unity constraint is implemented, yet it resolves the rank deficiency problem. It also allows the sensible comparison of the constrained and unconstrained least squares solutions (as both models are equivalent). The difference is that the unconstrained least squares solution will give a minimum norm solution when $c < n+1$, which measures the size of the weight vector and the bias term (an affine function), whereas the effect of the sum to unity constraint is to produce a minimum norm solution for the weights only (a linear function) which maximizes the resulting linear mixture region, as shown in Section III-B. It is also worthwhile noting that when the minimum norm solution of an affine function is calculated, the solution is no longer symmetrical for the 2 class case, i.e., $y_1 \neq 1 - y_2$. However, this relationship does hold with the linear minimum norm solution.

Previously, Kent and Mardia [19] have stated that the matrix of mean differences $[\boldsymbol{x}^2 - \boldsymbol{x}^1, \cdots, \boldsymbol{x}^c - \boldsymbol{x}^1]$ should have full rank, which is equivalent to adding a bias term as described above. In addition, Horwitz *et al.* [15], state that the input vectors $[1, \boldsymbol{x}^i]$ should be linearly independent, which simply appends a bias term, as described above. These are equivalent to the comments made above. Finally, the rank deficiency problem mentioned above only occurs because of the Lagrange technique used to derive the closed form solution. It is possible to use the direct elimination method to solve the constrained quadratic program, [9], [14], which does not suffer from the rank deficiency problem.

*3) Infinity of Solutions and Margin Width Maximization:* When $c < n+1$, there are an infinity of solutions which satisfy the original LSMM equations. This is because the original constraints do not uniquely constrain the solution, which is due to the form of the quadratic error in the spectral values, equation (2). These equations can be solved exactly and there is no error, hence the quadratic form has the same constraining power as a set of $c$ linear equations (assuming $\boldsymbol{R}$ is rank $c$). The fact that it is treated as a least squares problem and solved using the normal pseudo-inverse is an implementation constraint which is not part of the original set. Instead of solving the set of linear equations

$$\boldsymbol{x} = \boldsymbol{R}\boldsymbol{y} \tag{14}$$

the quadratic constraint with the pseudo inverse solution solves the equivalent set of linear equations

$$\boldsymbol{R}^T\boldsymbol{x} = \boldsymbol{R}^T\boldsymbol{R}\boldsymbol{y}. \tag{15}$$

However, because the equations can be solved exactly, it is possible to solve the following set of linear equations:

$$\boldsymbol{S}^T\boldsymbol{x} = \boldsymbol{S}^T\boldsymbol{R}\boldsymbol{y} \tag{16}$$

where $\boldsymbol{S} = \boldsymbol{V}^{-1}\boldsymbol{R}$ and $\boldsymbol{V}$ is an arbitrary, symmetric, positive definite matrix. Obviously, $\boldsymbol{V}$ is related to the input noise covariance matrix but has been used here to illustrate how it is possible to rotate the margin through almost 180°, and still produce a solution which satisfies the quadratic form sampled at the pure pixels.

Therefore, while the CLS LSMM solution found using Lagrange multipliers is unique, the orientation of the margin is determined by the noise covariance matrix introduced via the least squares criteria in equation (2). In Section III-B, it is shown that the CLS LSMM solution is arguably the most appropriate as it maximizes the size of the margin, and a set of different

optimality criteria are proposed which explicitly represent the unique solution.

## III. Support Vector Machines

Support Vector Machines (SVMs) are classification and regression methods which have been derived from the basic principles described in Vapnik's Statistical Learning Theory [33] The classification techniques are based on the principle of "optimal separation", where if the classes are separable, the solution is chosen which maximally separates the classes. This uniquely defines the optimal discriminant, and in doing so, selects the data points which lie on the class boundary closest to the neighboring classes [13]. Hence the process performs automatic "pure pixel" selection, where the pure pixels (termed support vectors) are those lying on the class boundary. In handwriting [24] and face recognition [25] problems it has been found that between 3 and 5% of the data points are detected as support vectors, and the remainder can be discarded from the calculation.[2]

An empirical model approach assumes that a set of exemplar data, drawn at random from a distribution which represents how the model will be deployed, is used to design a model of the form

$$y = M_D(\boldsymbol{x}, \boldsymbol{w}) \tag{17}$$

where

- $M$ structure of the selected model;
- $D$ data set of input/output exemplar pairs;
- $\boldsymbol{w}$ parameter vector and it has been assumed that the model is single output.

Little attention is often given to selecting a suitable data set $D = \{\boldsymbol{x}^i, t^i\}_{i=1}^{l}$ of $l$ input/output exemplar pairs, yet the effect of training a model $M$ on an unsuitable data set $D$ can be extremely undesirable. This can be expressed by the bias/variance dilemma [6], where the model's ability to underfit the true relationship (bias) and overfit the training data (variance) are related. One of the potential advantages of the SVM approach is that it is less sensitive to biases which can be introduced by poor data collection, yet it should be noted that if the information is not contained in the data, **no** empirical model can automatically extract it.

Implicit in the SVM approach is that the property that a model only has a single output, so a range of $c$ models are necessary to predict the mixtures, one for each class. Hence, the exemplar data set is labeled using a 1-of-$c$ encoding, which represents the normal closed world, mutually exclusive assumptions. Other encoding schemes are possible [34], and these may be useful for dealing with the problem of selecting different sets of support vectors pixels for a particular output. The basic linear SVM classification algorithms are derived from a discrimination perspective, yet this is in keeping with the linear mixture model approach. A discrimination boundary of the form

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0.5 \tag{18}$$

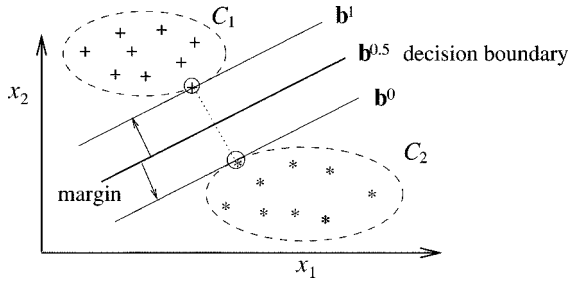


Fig. 5. Linear SVM for two inputs and two classes. The stars and crosses represent the labeled ("pure") training data given to the algorithm and the circles denote the selected support vectors which determine the linear model's boundaries.

for a linear model with a weight vector $\boldsymbol{w}$ and bias term $b$ defines a set of points that constitutes the boundary between the two classes. The threshold value of 0.5 is arbitrary but is appropriate in this case because it lies between the values of 0 and 1 which denote none and full class membership, respectively[3]. The set of points for which this equation equals 0 is the zero boundary, and the one boundary is similarly defined. When the discrimination boundary is linear, these three linear boundaries are all parallel but have a different bias term, as illustrated in Fig. 5. It can be easily shown that the size of the margin is inversely related to the size of the weight vector $\boldsymbol{w}$ in (18) (note that the bias term, $b$, is explicitly represented and separate from the weight vector), hence in order to maximize the margin, it is necessary to minimize the size of the weight vector $||\boldsymbol{w}||_2^2$.

Margin maximization proceeds by identifying the closest points to the boundary (the support vectors) in each set of class exemplars. These data points contain all the information necessary to calculate the weights and bias terms which maximize the margin's width. Therefore, the process of calculating the weights and bias terms can be regarded as an automatic technique for pure pixel selection when it is assumed that the pure pixels lie on the edge of the class margins, as illustrated in Fig. 5.

Linear SVMs use linear discrimination (models) to separate classes which are linearly separable. This approach is considered first in this paper, and it is shown that the algorithm is identical to the CLS LSMM when the same set of pure pixels is used as the data set for both techniques. This is an important result as it shows that the simplest SVM is equivalent to a widely used algorithm in the remote sensing literature, and that the CLS LSMM technique implicitly maximizes the mixing margin. It establishes that the set of SVM optimality constraints can be used to derive the CLS LSMM algorithm which highlights how the technique can be easily extended to include pure pixel selection (support vector identification). In addition, it is not essential to use a simple, linear discriminant, and other higher-order, nonlinear kernel-based classifiers can be used both for nonlinear discrimination and for modeling. Also, overlapping data clusters can be handled by introducing a measure of misclassification, which can be controlled either by the designer or using computer intensive, cross validation algorithms. The SVM frame-

[2]As will be seen later, when a single pure pixel for each class is not available, the number of support vectors in the mixture domain may be much greater.

[3]Normally, a threshold value of 0 is used in the SVM algorithms as class membership is a member of the set $\{-1, 1\}$, but the bipolar class membership representation will only be used in this paper when the algorithms are derived.

work thus provides algorithms for automatic pure pixel selection, for both linear and nonlinear modeling and discrimination problems, and generalizes the basic CLS LSMM algorithm.

### A. Linear Support Vector Machines

In applying a linear SVM for pure discrimination, it is assumed that the problem has two classes (as each model represents a single term in the 1-of-$c$ encoding) and the data can be linearly separated. The task is then to find the linear discriminator which maximally separates the classes. In the previous section, it was assumed that the output of an SVM lay in the unit interval, [0, 1]. However, for the purposes of deriving the algorithms, it is more convenient to assume the output lies in the bi-polar interval $[-1, 1]$. This can be done without loss of generality.

The training set, $\{\boldsymbol{x}^i, t^i\}_{i=1}^{l}$, is composed of exemplars of this class $t^i = 1$, or exemplars of the other class $t^i = -1$. It is assumed that no spectral confusion is present in the data, so the aim is to maximize the size of the mixing margin such that all the data are correctly labeled as either 1 or $-1$. This can be specified as an optimal linear separation problem, which can be formulated as the following quadratic programming (QP) problem:

$$\begin{aligned} &\text{minimise} && \|\boldsymbol{w}\|_2^2 \\ &\text{subject to} && (\boldsymbol{w}^T\boldsymbol{x}^i + b)t^i \geq 1 \qquad i = 1, 2, \cdots, l \end{aligned} \tag{19}$$

where $\boldsymbol{w}$ is the weight vector for this model, which describes the orientation of the margin and boundaries and $b$ is the bias term. The second constraint can be interpreted as requiring the data set to be linearly separable, and the first constraint minimizes the size of the weight vector (maximizes the margin's size) in order to separate the data "optimally."

The solution is given by finding the saddle point of the Lagrange functional

$$L(\boldsymbol{w}, b, \alpha) = \tfrac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_{i=1}^{l} \alpha^i(((\boldsymbol{x}^i.\boldsymbol{w}) + b)t^i - 1) \tag{20}$$

where the Lagrange multipliers $\alpha^i$ denote the importance of each data point. A zero value of $\alpha^i$ means that the exemplar data pair doesn't influence the calculation of the weight vector. The Lagrangian has to be minimized with respect to $\boldsymbol{w}$ and $b$ and maximized with respect to $\alpha^i \geq 0$. The Lagrange multipliers can be found using the dual problem

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha^i\alpha^j t^i t^j(\boldsymbol{x}^i.\boldsymbol{x}^j) + \sum_{i=1}^{l}\alpha^i \tag{21}$$

subject to the constraints

$$\alpha^i \geq 0 \qquad i = 1, 2, \cdots, l \tag{22}$$

$$\sum_{i=1}^{l}\alpha^i t^i = 0 \tag{23}$$

where the second constraint is given from $\partial L/\partial b = 0$. The weight vector and bias term can then be calculated as

$$\boldsymbol{w} = \sum_{i=1}^{l}\alpha^i\boldsymbol{x}^i t^i \tag{24}$$

$$b = -\tfrac{1}{2}\boldsymbol{w}^T(\boldsymbol{x}^1 + \boldsymbol{x}^2) \tag{25}$$

where $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ are any two vectors from $C_1$ and $C_2$, respectively, which have non-zero Lagrange multipliers. The expression for the weight vector is obtained from the equation $\partial L/\partial \boldsymbol{w} = 0$.

The Lagrange multipliers $\alpha^i$ effectively weight each data point according to its importance in determining a solution, as they represent the sensitivity of the optimization criteria with respect to a particular constraint [9]. For the linearly separable two class problem just described, only those data points which lie on the margin's boundary have a non-zero Lagrange multiplier. Hence, solving the QP problem is equivalent to performing data selection, and once the Lagrange multipliers $\alpha^i$ are determined, the weight vector and bias can be calculated directly.

### B. Linear SVM and CLS Linear Spectral Mixture Models

*Theorem 1:* The models formed by a linear SVM and the CLS LSMM are equivalent when the same set of pure pixels (one pure pixel for each class) is available for both techniques and the same nonlinear mixture re-estimation algorithm is used when $y_j < 0$ or $y_j > 1$.

*Proof:* To prove this equivalence, first consider the case with the input noise covariance matrix $\boldsymbol{V} = \boldsymbol{I}$. This will be relaxed later. In addition, the proof is based on showing that the estimation of the linear models for both techniques is the same, and it has been assumed that the same subsequent nonlinear bounding and normalizing operations are applied. All that needs to be shown therefore is that the linear models are the same within the model's margin (intersection of all the individual margins).

When the data point lies within the margin of all the classes, the proof relies on showing that the computed weight vectors and bias terms for each technique for all the models are equivalent. Without loss of generality, this will be done for the $j$th model only. First, note that the linear models formed by both techniques when $c = n+1$ are equivalent, as there are $n+1$ linearly independent degrees of freedom in the data set (basic assumption of the CLS LSMM algorithm) which uniquely constrains the $n+1$ linear parameters in each linear model. This has assumed that the CLS LSMM pure pixel matrix has an augmented row of biases to guarantee a solution, see Section II-B.2.

When $c < n+1$, there are an infinite number of solutions which pass through the training data, (see Fig. 2). The linear SVM is unique in that it maximizes the size of the margin by minimizing the size of the weight vector. To show the CLS LSMM algorithm is equivalent, it is necessary to show that the two calculated weight vectors are identical. It is assumed that the CLS LSMM weight vector is calculated as shown in (5) and (6), and that it produces the unique CLS LSMM solution (although

there are in fact an infinite number that satisfy the original optimization goals, as discussed in Section II-B.3. From (6) and (7)

$$\boldsymbol{W} = \boldsymbol{C}^{-1}\left(\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^T\boldsymbol{C}^{-1}}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}}\right)\boldsymbol{R}^T \tag{26}$$

but $\boldsymbol{b} = \boldsymbol{C}^{-1}\mathbf{1}/\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}$, so

$$\boldsymbol{W} = \boldsymbol{C}^{-1}(\boldsymbol{I} - \mathbf{1}\boldsymbol{b}^T)\boldsymbol{R}^T. \tag{27}$$

The weight vector for the $j$th mixture model is given by transposing this equation and extracting the $j$th column, which gives

$$(\boldsymbol{W}^T)_j = (\boldsymbol{R}\boldsymbol{C}^{-1})_j - \boldsymbol{R}\boldsymbol{C}^{-1}\mathbf{1}b_j \tag{28}$$

where the notation $()_j$ has been introduced to represent extracting the $j$th column.

The aim is to now show that the weight vector for the $j$th linear SVM is equal to this expression. In performing this derivation, it is assumed that every spectral vector in the pure pixel matrix $\boldsymbol{R}$ will be selected as a support vector, hence the inequality constraint in equation (19) will be replaced by an equality. This is valid because each pure pixel is the only example of each class so the output must be either 1 or 0 at these points. Hence, they will all lie on the margin's boundary and so the constraints on the Lagrange multipliers are no longer appropriate. The $j$th linear SVM is designed by solving the following QP problem:

$$\begin{aligned} &\text{minimise} && \boldsymbol{w}^T\boldsymbol{w} \\ &\text{subject to} && \boldsymbol{R}^T\boldsymbol{w} + \mathbf{1}b = \boldsymbol{t} \end{aligned} \tag{29}$$

where the subscript to denote the $j$th model has been dropped. In addition, the target vector $\boldsymbol{t}$ has a single value of 1, which corresponds to the pure pixel for class $j$ and the remaining elements are zero. The Lagrange functional for the $j$th linear SVM is therefore given by

$$L(\boldsymbol{w}, b, \alpha) = \tfrac{1}{2}\,||\boldsymbol{w}||_2^2 - \sum_{i=1}^{l}\alpha^i((\boldsymbol{x}^i.\boldsymbol{w}) + b - t^i) \tag{30}$$

where the $\alpha^i \geq 0$ constraint no longer applies, as it is a QP problem with equality constraints. Now using the direct elimination method [9] to solve this QP problem gives

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{R}\boldsymbol{C}^{-1}(\boldsymbol{t} - \mathbf{1}b) \\ &= (\boldsymbol{R}\boldsymbol{C}^{-1})_j - \boldsymbol{R}\boldsymbol{C}^{-1}\mathbf{1}b \end{aligned}$$

which is the same as (28). Therefore, to show that the linear SVM and the CLS LSMM weight vectors are the same, it needs to be established that the bias terms are also equivalent. To obtain a closed form solution for the linear SVM bias term, the solution of the Lagrange equations [9] gives:

$$\alpha = -\boldsymbol{C}^{-1}(\boldsymbol{t} - b\mathbf{1}) \tag{31}$$

and differentiating the Lagrange functional in (30) with respect to $b$ gives

$$\alpha^T\mathbf{1} = 0. \tag{32}$$

Combining these two equations, and rearranging the terms gives

$$b = \frac{(\boldsymbol{C}^{-1}\mathbf{1})_j}{\mathbf{1}^T\boldsymbol{C}^{-1}\mathbf{1}}. \tag{33}$$

The bias terms are the same and hence, the weight vectors are the same for each row of $\boldsymbol{W}$.

As long as the same thresholding and normalizing procedures are applied outside the mixture regions, when the models calculate a negative output or one greater than one, then the two techniques are equivalent in every other domain as well, as the basic linear calculations are equivalent.

At the start of the proof, it was assumed that the input noise covariance matrix $\boldsymbol{V} = \boldsymbol{I}$. This restriction can be removed by considering the following QP problem for the linear SVM algorithm:

$$\begin{aligned} &\text{minimise} && \boldsymbol{w}^T\boldsymbol{V}\boldsymbol{w} \\ &\text{subject to} && \boldsymbol{R}^T\boldsymbol{w} + b\mathbf{1} = \boldsymbol{t} \end{aligned} \tag{34}$$

and, using a derivation similar to the one described above, it can be shown that the two algorithms are identical. The only difference is that the size of the weight vector is now measured with respect to the matrix $\boldsymbol{V}$ and can be written as $||\boldsymbol{w}||_V^2$, where it has been assumed that the standard Euclidean norm is used. Note that the covariance matrix could equivalently be absorbed into the input inner product [16]. □

*1) On the Relationship Between CLS Linear Spectral Mixture Models and Linear SVM:* This result has a number of important implications for both techniques. Probably the most important is that the CLS LSMM model satisfies the margin maximization property when $c < n+1$. As discussed in Section II-B.3, it can be argued that in this redundant situation, the most sensible solution is one which is perpendicular to the projection of the pure pixel onto the hyperplane containing the other pure pixels. This is exactly the same as maximizing the margin. When the variance-covariance matrix $\boldsymbol{V}$ is not equal to the identity matrix, the minimum norm solution is produced for the space $\boldsymbol{x}' = \boldsymbol{V}^{-0.5}\boldsymbol{x}$. Therefore the effect of introducing the variance–covariance matrix is to "rotate and stretch" the input axes and define a minimum norm solution in this new space.

The optimal formulation of the linear SVM guarantees a unique solution to the problem and directly represents the margin maximization principle and so it could be argued that it is more fundamental than setting up the LSMM modeling problem as a least squares solution, in the knowledge that it can be solved exactly.

The relationship also shows that the margin maximization property implicitly implies that the resulting linear SVM mixture estimates will sum to unity when the same set of pure pixels are used as support vectors for each model. This may not seem at first obvious, but is true because each model forms a linear mixture across the same $c-1$-dimensional hyperplane which is formed from the space spanned by the pure pixels. Within this $c-1$-dimensional hyperplane, there exist $c$ pure pixels. Hence, the linear mixture problem is uniquely defined and the estimates will sum to unity within the model's margin. Every other point in the $n$-dimensional space is projected onto this subspace, and the mixture estimates are then calculated. Thus, the sum to unity

property is implicit in the margin maximization approach used by SVM's.

The SVM formulation thresholds the mixture estimates at both 0 and 1 [see (19)], whereas they are generally only thresholded at 0 in the normal CLS LSMM algorithm. The thresholding at 1 is important in the SVM algorithms as a data point that causes the unthresholded output to be less than zero or greater than one does not lie on the margin boundary and is not considered as a support vector. Only those data points for which the unthresholded output equals zero or one are considered to be support vectors. In practice, the different thresholding strategies make little difference in the solutions calculated by the two algorithms, as can be seen by comparing Fig. 6 with Fig. 1. There is no unique way to enforce the non-negative, sum-to-unity constraints in the LSMM literature. Hence, the previous proof simply assumed that the thresholding SVM conditions were adopted and the resulting values were normalized to sum to unity.

### C. Linearly Nonseparable Mixture Modeling

When the data is nonseparable or the designer may decide that the support vectors should not lie at the edge of the distribution, the above technique can be modified to allow a number of "misclassifications." This typically occurs in remote sensing when the classes are not necessarily spectrally distinct or some of the variation in the potential set of pure pixels is due to spectral confusion. Rather than modeling these pixels as pure class members, they may be located within the mixing margin, and the process of optimal discrimination should be modified. Therefore, the QP problem is modified to minimize the number and size of such misclassifications, which are presumed due to the input measurements being corrupted by zero-mean random noise. This is achieved by introducing an extra set of non-negative variables $\xi^i \geq 0$, $i = 1, \cdots, l$ which specify how far from the bipolar values the current output is

$$t^i(\boldsymbol{w}^T\boldsymbol{x}^i + b) \geq 1 - \xi^i. \tag{35}$$

Therefore, the $\xi^i$'s form extra penalty terms that should be minimized. This can be solved by finding the saddle point of the Lagrangian

$$L(\boldsymbol{w}, b, \xi, \alpha, \beta) = \tfrac{1}{2}\,\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{l}\xi^i - \sum_{i=1}^{l}\alpha^i \cdot \left((\boldsymbol{x}^i.\boldsymbol{w}+b)t^i - 1 + \xi^i\right) - \sum_{i=1}^{l}\beta^i\xi^i \tag{36}$$

where $C$ is a given smoothness constraint, and the $\alpha^i$ and $\beta^i$ terms are Lagrange multipliers. This expression should be minimized with respect to $\boldsymbol{w}$, $b$ and $\xi^i$ and maximized with respect to $\alpha^i$, $\beta^i \geq 0$, and results in maximizing the dual problem

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha^i\alpha^j t^i t^j(\boldsymbol{x}^i.\boldsymbol{x}^j) + \sum_{i=1}^{l}\alpha^i \tag{37}$$

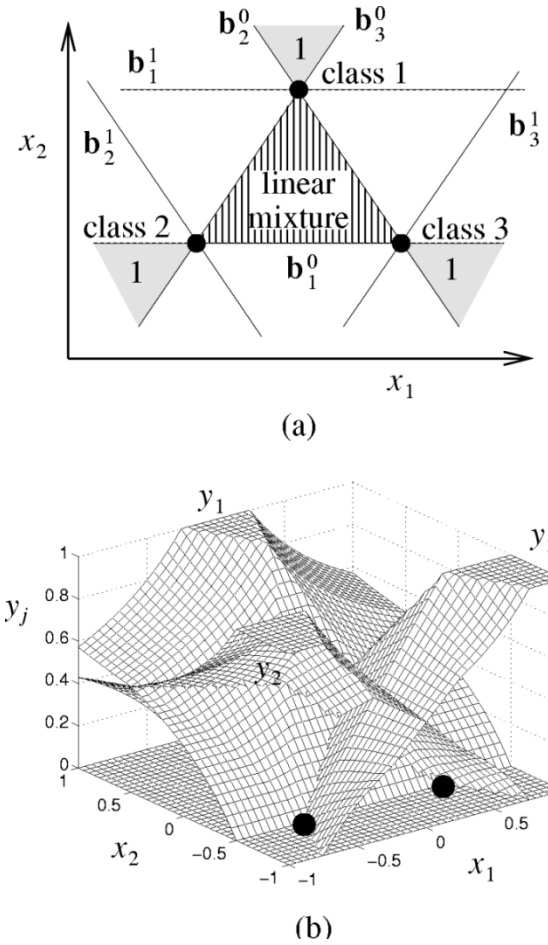


Fig. 6. Three linear SVM's and their margins, which are denoted by the margins' boundaries $\boldsymbol{b}_j^0$ and $\boldsymbol{b}_j^1$, for a two input problem. The division of the feature space is shown in (a), while the corresponding mixture surface is shown in (b).

subject to the constraints

$$C \geq \alpha^i \geq 0 \qquad i = 1, 2, \cdots, l \tag{38}$$

$$\sum_{i=1}^{l}\alpha^i t^i = 0. \tag{39}$$

Therefore, the loss function is composed of a term that tries to maximize the size of the margin $\|\boldsymbol{w}\|_2^2$, and a term that minimizes the size of the errors $\sum_{i=1}^{l}\xi^i$. The parameter $C$ weights these two competing goals. When $C \to 0$, less emphasis is placed on the classification performance of the system applied to the training set, and the margin becomes wider. A lower bound is effectively reached when all the Lagrange multipliers are constrained by $C$. Similarly, when $C \to \infty$, more emphasis is placed on the classification performance and the margin will become narrower. A point is reached when the linearly separable QP problem becomes either solvable or unfeasible, and increasing $C$ further does not affect the form of this solution. The bound $C$ can also be viewed as limiting the VC dimension of the model [33]. It is obviously a design parameter which must be specified prior to running the QP algorithms, but it may be optimized using computer intensive, cross validation algorithms. In

some respects, it is similar to the regularization parameters used in artificial neural networks (ANN), and statistical re-estimation algorithms may therefore be applied to optimize its value [22]. However, despite a lot of research that has been performed in the ANN and Bayesian statistical modeling communities, it is still often necessary to manually tune the calculated value.

### D. Nonlinear Mixture Modeling

So far, the classification algorithm has been based on a linear discriminant for the separable and the nonseparable cases. However, the approach is not limited to simply linear decision boundaries as kernel-based, nonlinear mappings [8], [33] can be used to construct more flexible decision boundaries. In pursuing this approach, all of the previous analysis holds for forming the decision boundary, and the only change that needs to be made is to substitute a kernel function for the inner product between two training vectors in the Lagrange functionals (21) or (37). Instead of performing the data selection in the original feature space, a higher dimensional, nonlinear transformation is used instead. This is the purpose of the kernel functions: to map the data into an alternative space in which the problem may be linearly separable, and common choices for the kernel functions include the following.

*Polynomial kernel*: $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = (\boldsymbol{x}^i.\boldsymbol{x}^j + 1)^d$ where $d = 1, 2, \cdots$.
*Gaussian RBF*: $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp(-\boldsymbol{x}^i - \boldsymbol{x}^j\|_2^2/2\sigma^2)$.
*Ridge functions*: $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \tanh(b(\boldsymbol{x}^i.\boldsymbol{x}^j) - c)$ for certain values of $b$ and $c$.

Also included are various spline functions [13]. The nonlinear kernels can therefore be used to produce a wide range of decision boundaries, and the data selection procedure is equivalent to the selection of the kernel functions in the network, i.e., only those kernels with a non-zero Lagrange multiplier $\alpha^i$ will contribute to the network's decision. In addition, when the transformed feature space is large (for instance a polynomial of degree 5 with six inputs has $C_6^{5+6}$ terms), only the dot product in the original input space needs to be formed, it is not necessary to calculate the expansion in the transformed space [27].

Various nonlinear mixture models have been proposed in the literature, including polynomial transformations [7], which aims to overcome the $c \leq n+1$ constraint. By extending the basic model space to include nonlinear kernels, all of the previous results that apply to implementing separable and nonseparable mixtures still apply. In particular, the techniques used to automatically determine the support vectors for the separable and nonseparable distributions are still appropriate. However, any nonlinear empirical modeling algorithm is subject to the "curse of dimensionality" [4], which refers to the exponential increase in data and model resources required to evenly populate the feature space as the number of inputs increases. Nonlinear margins are obviously a lot more flexible. However, they should only be used if there is strong evidence for a particular margin shape (quadratic, cubic, etc.) or there is a lot of data to support more flexible kernels such as Gaussian or spline basis function. For SVM's, complexity is controlled by adapting the size of the margin, i.e., finding an appropriate value of $C$ in (36). This in turn determines how many "pure pixels" (support vectors) lie within the mixing margin.

### E. Regression SVM Algorithms

So far, the mixture modeling problem has been posed as finding an algorithm which "models" regions of pure pixels, i.e., the only data used are labeled with a 1 (to denote an exemplar pure pixel) or a 0 (to denote a pure pixel which is not a member of that class). Errors in the predictions were introduced to handle spectral confusion and nonlinear kernels have been proposed as one method for producing nonlinear mixture models. However, remotely sensed training data is dominated by examples of mixed data, where the target values lie in the unit interval, [0, 1] and conform to the closed world assumption (sum to unity). Often it is difficult to identify single pure pixels in the data sets, so techniques need to be developed which can make use of these large data sets containing mixed pixels.

Within the remote sensing community, ANN's have often been used to model this type of data [3], [5], [10], [32] which represent a very different approach from the conventional CLS LSMM approach, which uses only pure pixel training data. Hence, the position and orientation of the mixture region is determined by the pure pixels (support vectors) at the edge of the class cores. However, using exemplar data based on actual mixtures allows a nonlinear regression scheme to adapt to the actual local mixtures and as such is a more flexible approach, although the previous comments made about the curse of dimensionality apply. In adopting a statistical regression approach to mixture modeling, it is assumed that the empirical, sampled data set is representative of how the model will be applied. This is characterized by the bias/variance dilemma [6] and Vapnik's statistical learning theory [33], which state that the true performance of a model is only partially related to its empirical performance on the sampled training data, as the training data may be biased, and the model may overfit this information.

Instead of using ANN's, it is possible to extend the SVM algorithms to a regression scenario. The models that can be used are identical to ones used for classification, so they can be linear or use any of the kernel functions described in Section III-D. These nonlinear transformations are similar to the ones often employed in ANN's, and the main difference between the two techniques is the method for choosing and training the nonlinear nodes. ANN's typically have a fixed number of nodes in the hidden layer, whose parameters are optimized using a nonlinear training algorithm to best fit the training data, usually using a summed squared errors criteria. However, SVM's consider a kernel node to be associated with every kernel node and the ones chosen with a non-zero Lagrange multiplier constitute the model's "hidden layer." The output of the $i$th kernel node for a query pattern $\boldsymbol{x}$ is given by $K(\boldsymbol{x}^i, \boldsymbol{x})$, so the node's weights are the corresponding training input pattern. In addition, the Lagrange multipliers are the linear output layer's weights, so the SVM models can be represented as a type of three-layer ANN.

A range of different loss function can be used within the SVM framework, from the normal summed square errors to summed

absolute errors and a robust version of these measures which introduces an $\epsilon$-insensitive dead-zone about 0, such that only absolute errors larger than $\epsilon$ influence the loss function [28]. Strictly speaking, only the loss functions that employ a $\epsilon$-insensitive dead-zone perform data and kernel selection, as otherwise, all the Lagrange multipliers will be non-zero and the SVM model will contain a kernel associated with every data point in the training set. However, the CLS-LSMM algorithm are derived on the principle of minimizing a least squares cost function. Like the complexity control parameter $C$, the dead-zone's size influences the complexity of the final model. A large dead-zone will produce simple models, although the predicted trends between the selected support vectors are not influenced by the intermediate training data points. A small dead-zone means that most data points will be selected as support vectors and the model will consist of a large number of kernels. One of the problems with formulating the mixture problem as a regression scenario is to constrain the outputs to sum to unity and to lie in the unit interval. Often this is done by an explicit transformation of the outputs in the same way as classification is performed. At present, the authors have not applied SVM's to true mixture data and as such have not quantified how much this approach will improve the predictions of the mixtures.

## IV. Applications

The examples described in this section illustrate the potential of applying the SVM algorithms to remotely sensed data. Three situations are considered.

1) Where the pure pixel data sets are linearly separable and a linear mixing region is presumed to exist between the class cores. This illustrates automatic pure pixel selection on the linear margin boundaries.
2) When the class cores are subject to noise and spectral confusion may occur. This illustrates how pure pixel selection occurs within the mixture region and the relationship between the size of the margin and the model complexity parameter $C$.
3) When nonlinear mixture regions are required.

While it has been shown that for the same set of pure pixels, the linear SVM algorithm is identical to the standard CLS LSMM algorithm, a comparison of the CLS LSMM algorithm with the linear, nonseparable SVM algorithm has been performed. In this case, the construction of the CLS LSMM algorithm assumed that the data followed a normal distribution and the means for each class were used as the pure pixel spectra, whereas the linear SVM automatically selected pure pixels from the training set to form the mixing model. The real-word dataset was produced as part of the European Union FLIERS project [21]. Here, the data is split into two base classes:

- developed and other (including slate, tarmac, concrete, etc);
- undeveloped and vegetation (including sand, water, soil, grass, shrubs, woodland, etc).

The data is taken from a Landsat TM image of the Leicester suburbs, as shown in Fig. 7.

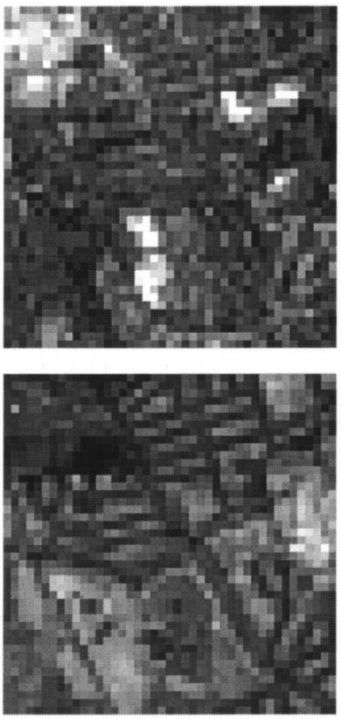

Fig. 7. Bands 1 (top) and 4 (bottom) of the Landsat TM image used to evaluate the linear SVM mixture model.

In the following examples, only two out of the seven available bands (1 and 4) of the Landsat TM imagery are used as spectral features. This allows the mixing margins and the selected support vectors to be plotted. These $33 \times 33$ (1200 potential data points) grey-scale spectral images are shown in Fig. 7, and the data sets used in the following examples are subsets of these images.

Data with an area membership of greater than 0.95 of these two classes were considered as potential "pure pixels," and this produced a reduced data set of 313 pure pixel training pairs (76 from the Developed and Other class and 237 from the Undeveloped and Vegetation class). This formed the basis for the three examples. The first data set was reduced slightly (by 20 data points), as some potential pixels were removed to produce two linearly separable sets of pure pixels and the other two examples used all 313 training pairs. The remaining 794 mixed pixels in the image for which the area membership in the Developed and Other class was $> 0.0$ and $< 0.95$ were also used to evaluate the performance of the CLS LSMM algorithm and the linear, nonseparable SVM algorithm in the second example.

### A. Linear, Separable SVM Algorithm

To show how the linear SVM (linearly separable) algorithm can be applied, the data set described above was filtered to remove points which, due to the spectral confusion, were lying in the region where linear mixing was presumed to occur. This may be unrealistic, but the aim of this example is to illustrate how the support vector (pure pixel) selection process works, rather than show that the technique is "optimal" for this data set. The same assumption is made when the LSMM's are designed.

Running the linearly separable algorithm on the (reduced) data set gave the linear margin and support vectors shown in

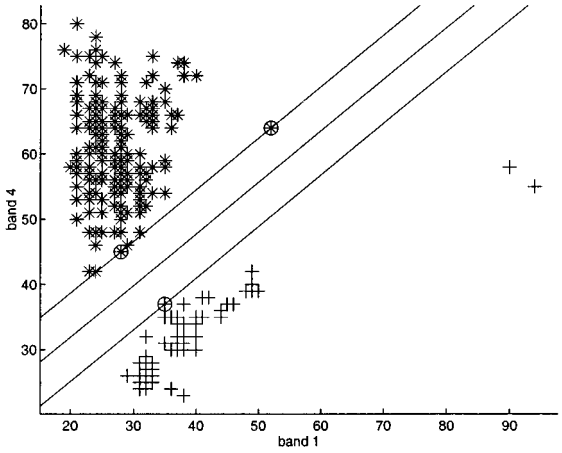


Fig. 8. Linear SVM mixture predictions and selected support vectors for Landsat bands 1 and 4. Crosses are used to represent the developed and other class data, and stars represent the undeveloped and vegetation class data. The three data points selected as support vectors are circled, and the three solid lines represent the margin's boundaries and center.

Fig. 8. Because the problem has been approached from a linearly separable classification viewpoint, it has been implicitly assumed that the aim of mixture modeling is to use pure pixels which lie on the edge of the classes' cores. If this is not the case, this approach will obviously be inappropriate. An appropriate mixture margin has been calculated by the SVM, and only three of the potential 293 data points have been selected as support vectors. The algorithms for performing model construction and the quadratic programming data selection procedure took approximately 9 s to execute on a Pentium 166MHz PC [12].

In this example, it could be argued that the support vectors lying on the edge of the classes' cores could be selected manually. However, incorrectly selecting pixels that are not "opposite" each other will change the orientation of the margin, and while this is relatively easy to visualize and optimize by eye when there are only two spectral features, it is extremely difficult using all six Landsat bands.

### *B. Linear, Nonseparable SVM Algorithm*

In order to show how spectral confusion can be handled within the SVM framework, the data set consisting of 313 data points was used to design a mixture model. In this case these points lie within the mixing margin or within the other class core. This is presumed due to spectral confusion, and the degree with which this occurs is controlled by the model complexity parameter $C$.

Note that the *a priori* selection of a set of pure pixels for the classes cores would be extremely problematic, and for the nonseparable linear SVM algorithm, this simply reduces to estimating a single bound parameter $C$ that determines the width of the margin. In practice, this can be difficult. The designer must select a value for which pure pixels that are incorrectly labeled due to the stochastic spectral confusion lie in the margin, and for which pure pixels that have variable spectral signatures due to the inherent deterministic natural variation in the class lie on either side of the margin. Often cross-validation procedures are used to estimate a suitable value for the parameter.

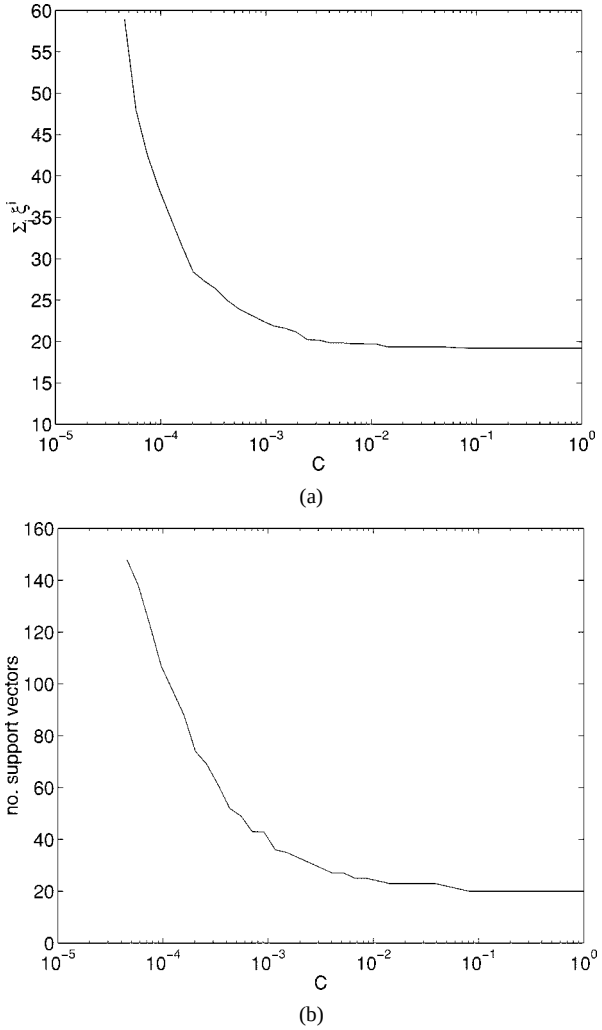


Fig. 9. Diagrams showing (a) the sensitivity of the errors and (b) the number of selected support vectors with respect to the model complexity parameter $C$.

Diagrams showing the variations in the summed errors $\sum_i \xi^i$, and the number of selected support vectors with respect to this model complexity parameter are shown in Fig. 9. In general, it can be seen that as $C$ decreases, more pure pixels are considered as being support vectors (lying in the margin) and that the error rate computed for the pure pixels increases.

The error rate

$$\text{SSE} = \sum_{i=1}^{k} (y(\boldsymbol{x}^i) - t^i)^2 \tag{40}$$

was also computed for the 794 mixed pixels with respect to the model complexity parameter, $C$. The variation in the summed errors for the mixed pixels and the number of selected support vectors are shown in Fig. 10. A value of $C = 0.0001$ was found to minimize the SSE and produced a linear, nonseparable SVM algorithm using 122 pure pixels as support vectors (39% of the pure pixel training data). The performance of this model calculated over the $k = 794$ data points in the mixed pixel data set was SSE $= 58.803$ (root mean square error [RMSE] $= 0.272$ pixels). This took approximately 14 s to execute on a Pentium 166 MHz PC. Fig. 11 shows the result of applying this SVM model to the data set.

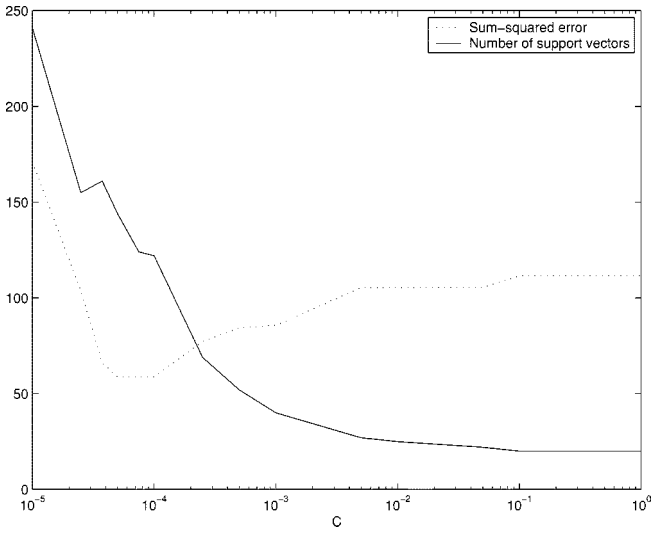


Fig. 10. Sensitivity of the errors in the estimation of components of mixed pixels and the number of selected support vectors with respect to the model complexity parameter $C$.

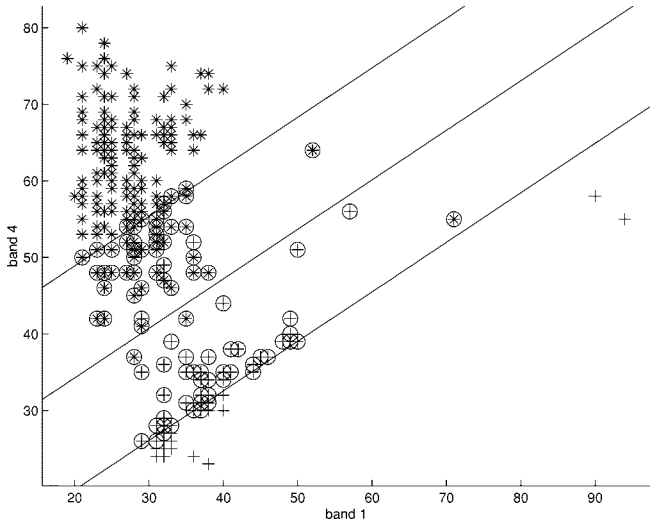


Fig. 11. Linear, nonseparable SVM mixture predictions and selected support vectors for $C = 0.0001$ and Landsat bands 1 and 4. Crosses are used to represent the developed and other class data, and stars represent the undeveloped and vegetation class data. The 122 data points selected as support vectors are circled.

### C. Linear Spectral Mixture Model

When the data are nonseparable, there is no reliable way in the LSMM algorithm to select *a priori* a set of pure pixels to represent the classes' cores. In this case, it is typically assumed that the data for each class follows a normal distribution and consequently the means of the class distributions are generally chosen as the "pure pixel" spectral signatures. For the Landsat data set, the LSMM solution constructed using the mean spectral signatures of the two classes predicted the components of the 794 mixed pixels with SSE $= 87.251$ (RMSE $= 0.332$ pixels). The LSMM mixture boundaries for this solution are shown in Fig. 12.

Since the choice of the classes' means is the "optimal" solution for the LSMM algorithm (assuming a normal distribution of pure pixels), this result illustrates that for this data set the linear, nonseparable SVM solution with model complexity parameter

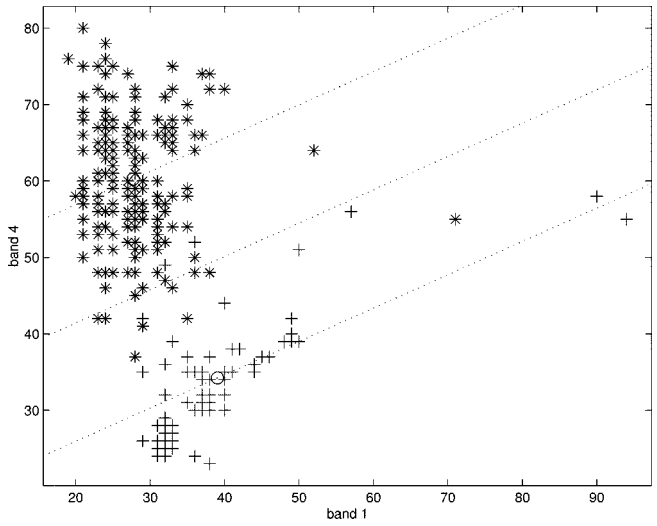


Fig. 12. CLS LSMM mixture model for Landsat bands 1 and 4. Circles are used to represent the mean spectral signatures for the two classes.

$C = 0.0001$ produces better mixture predictions than the CLS LSMM solution and, in addition, is able to automatically select the pure pixels without making assumptions of normality. The SVM approach minimizes 1-norm errors (sum of absolute errors), whereas the LSMM and the optimality criteria are both based on 2-norm errors (least squares). Hence, the SVM approach could be expected to be more robust to deviations from the normality assumption. This point is illustrated by the difference between the mixture regions shown in Figs. 11 and 12. The LSMM mixture region is sensitive to the pure pixel points that lie far from the classes' means, whereas the SVM mixture region appears to be less sensitive to these points. Given that normality assumptions are rarely met in practice, it can be argued that the linear, nonseparable SVM algorithm generally produces better mixture models than the LSMM algorithm.

### D. Nonlinear Mixtures

As well as considering the possibility that a single pure pixel may not be an appropriate assumption for the composite classes used in remote sensing, the assumption of linear mixing may not always be appropriate, particularly for spectral bands which lie outside the visible wavelengths. Therefore, nonlinear mixture regions may be appropriate in certain circumstances, and any of the kernel functions described in Section III-D can be used instead of a linear model.

A quadratic polynomial kernel was used to fit the nonseparable data set described in the previous section, and the result is shown in Fig. 13. As can be seen from the figure, the margin boundaries are no longer linear or parallel, and they appear to more closely represent the shape of the two class distributions. In this case the best model complexity parameter, $C$, was found to be $5 \times 10^{-8}$ (with the mixed pixel data set), and this selected 78 support vectors (24.9% of the training data). This took approximately 20 s to execute on a Pentium 166 MHz PC. The performance of this model on the mixed pixel data set was poorer than the linear, nonseparable SVM algorithm, but better than the CLS LSMM algorithm (SSE $= 78.091$, RMSE $= 0.314$ pixels). Given this result, it could be argued whether a quadratic

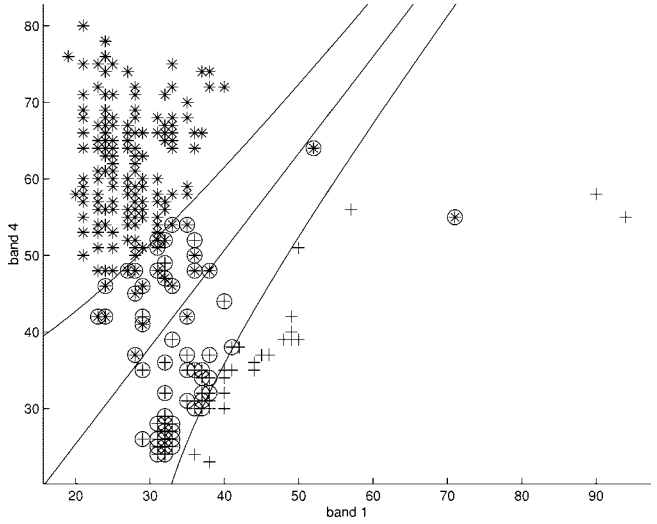


Fig. 13. Quadratic SVM mixture model for the two–class, two-input nonseparable problem with $C = 5 \times 10^{-8}$. The 78 data points selected as support vectors are circled.

mixture margin is appropriate in this case. Although the performance was convincing, compared to Fig. 11, there are probably too few data points that lie outside the two main data clusters to support this extra flexibility. Introducing nonlinear transformations of the measured spectral bands should only be done if the linear mixture assumptions are inappropriate and if there exists enough data to design a significantly better model.

## V. Conclusions

It has been shown that the CLS LSMM is related to the basic, linear SVM and under certain circumstances, both algorithms are identical. This is an important observation for the LSMM algorithms, as they have been shown to possess the "maximum margin" property, which always maximizes the size of the mixing margin. It also provides a method for automatic pure pixel selection, which is especially useful when there exists a large number of bands and the pure pixel sets contain more than a single element. In addition, it has been argued that the derivation of the algorithm from the normal LSMM constraints is non-unique, and a different, unique set of optimality constraints have been proposed which are obviously similar to the SVM constraints. In performing this work, the role of the bias term has been examined and an alternative representation has been proposed which ensures that the pure pixel matrix is full rank and can be used within the normal Lagrangian-derived expression. Finally, it has been argued that the SVM framework is more appropriate for empirical mixture modeling, as nonseparable distributions of pure classes can be handled appropriately, as well as nonlinear mixture modeling. These techniques have been illustrated with several simple examples, although it remains to fully quantify the performance of these algorithms with other nonlinear empirical modeling techniques.

## Acknowledgment

The authors gratefully acknowledge the discussions with Dr. S. Hobbs Cranfield University, Cranfield, U.K., and Dr. J. Settle, Reading University, Reading, U.K.

## References

[1] J. B. Adams, M. O. Smith, and P. E. Johnson, "Spectral mixture modeling: A new analysis of rock and soil types at the Viking lander 1 site," *J. Geophys. Res.*, vol. 91, no. B8, pp. 8098–8112, 1986.

[2] J. R. Anderson, E. E. Hardy, J. T. Roach, and R. E. Witmer, "A land use and land cover classification system for use with remotely sensed data," U.S. Geol. Survey, Professional Paper 964, Reston, VA, 1976.

[3] P. M. Atkinson, M. E. J. Cutler, and H. G. Lewis, "Mapping sub-pixel proportional land cover with AVHRR imagery," *Int. J. Remote Sensing*, vol. 18, no. 4, pp. 917–935, 1997.

[4] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ: Princeton Univ. Press, 1961.

[5] A. C. Bernard, G. G. Wilkinson, and I. Kanellopoulos, "Training strategies for neural network soft classification of remotely-sensed imagery," *Int. J. Remote Sensing*, vol. 18, no. 8, pp. 1851–1856, 1997.

[6] C. M. Bishop, *Neural Networks for Pattern Recognition*: Oxford Univ. Press, 1995.

[7] P. Bosdogianni, M. Petrou, and J. Kittler, "Mixture models with higher order moments," *IEEE Trans. Geosci. Remote Sensing*, vol. 35, pp. 341–353, Mar. 1997.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," in *Data Mining and Knowledge Discovery*, U. Fayyad, Ed. Boston, MA: Kluwer Academic, 1998, pp. 1–43.

[9] R. Fletcher, *Practical Methods of Optimization*. Chichester, U.K.: Wiley, 1987.

[10] G. M. Foody, R. M. Lucas, P. J. Curran, and M. Honzak, "Non-linear mixture modeling without end-members using an artificial neural network," *Int. J. Remote Sensing*, vol. 18, pp. 937–953, 1997.

[11] F. J. Garcia-Haro, M. A. Gilabert, and J. Melia, "Linear spectral mixture modelling to estimate vegetation amount from optical spectral data," *Int. J. Remote Sensing*, vol. 17, no. 17, pp. 3373–3400, 1996.

[12] S. R. Gunn, Support vector machine Matlab toolbox, http://www.isis.ecs.soton.ac.uk/resources/svminfo/, 1998.

[13] ——, "Support vector machines for classification and regression," Tech. Rep. ISIS-1-98, Dept. Electron. Comput. Sci., Univ. Southampton, Southampton, U.K., 1998.

[14] S. E. Hobbs, "Linear mixture modelling solution methods for satellite remote sensing," Tech. Rep. COA-9603, Cranfield Univ., Bedford, U.K., 1996.

[15] H. M. Horwitz, R. F. Nalepka, P. D. Hyde, and J. P. Morganstern, "Estimating the proportion of objects within a single resolution element of a multispectral scanner," Tech. Rep. NASA Contract NAS-9-9784, Univ. Michigan, Ann Arbor, MI, 1971.

[16] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Neural Information Processing Systems 11*. Cambridge, MA: MIT Press, 1999.

[17] N. Jewell, "An evaluation of multi-date SPOT data for agriculture and land use mapping in the United Kingdom," *Int. J. Remote Sensing*, vol. 10, pp. 939–951, 1989.

[18] I. Kanellopoulos, A. Varfis, G. G. Wilkinson, and J. Megier, "Land-cover discrimination in SPOT HRV imagery using an artificial neural network—A 20-class experiment," *Int. J. Remote Sensing*, vol. 13, pp. 917–924, 1992.

[19] J. T. Kent and K. V. Mardia, "Spatial classification using fuzzy membership models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 659–671, Sept. 1988.

[20] R. G. Krutchoff, "Classical and inverse methods of calibration," *Technometrics*, vol. 9, pp. 425–439, 1967.

[21] H. G. Lewis, M. Brown, A. R. L. Tatnall, M. Nixon, and J. Manslow, "Data analysis and empirical classification in FLIERS," Tech. Rep. ISIS-3-98, Dept. Electron. Comput. Sci., Univ. Southampton, Southampton, U.K., 1998.

[22] D. J. C. Mackay, "Bayesian Methods for Adaptive Models," Ph.D. dissertation, Calif. Inst. Technol., Pasadena, 1991.

[23] J. F. Manslow, M. Brown, and M. S. Nixon, "On the probabilistic interpretation of area based fuzzy land cover mixing proportions," in *Artificial Neuronal Networks: Application to Ecology and Evolution*, S. Lek, J.-F. Guégan, R. Allan, and U. Forstner, Eds. Berlin, Germany: Springer-Verlag, 2000.

[24] N. Matic, I. Guyon, J. Denker, and V. Vapnik, "Writer-adaptation for on-line handwritten character recognition," in *2nd Int. Conf. Pattern Recognition and Document Analysis*, Tsukuba, Japan, 1993, pp. 187–191.

[25] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 130–136.

[26] C. D. Rogers, "Characterization and error analysis of profiles retrieved from remotely sounding measurements," *J. Geophys. Res.*, vol. 95, no. D5, pp. 5587–5595, 1990.
[27] B. Scholkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
[28] A. Smola, "Learning with kernels," Ph.D. dissertation, Tech. Univ. Berlin, Berlin, Germany, 1998.
[29] J. Settle, "Private communication,", 1998.
[30] J. J. Settle and N. A. Drake, "Linear mixing and the estimation of ground cover proportions," *Int. J. Remote Sensing*, vol. 14, no. 6, pp. 1159–1177, 1993.
[31] G. Thomas, S. E. Hobbs, and M. Dufour, "Woodland area estimation by spectral mixing: Applying a goodness-of-fit solution method," *Int. J. Remote Sensing*, vol. 17, no. 2, pp. 291–301, 1996.
[32] E. J. van Koowijk, H. van der Voet, and J. J. M. Berdowski, "Estimation of ground cover composition per pixel after matching image and ground data with subpixel accuracy," *Int. J. Remote Sensing*, vol. 16, no. 1, pp. 97–111, 1995.
[33] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[34] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, Royal Holloway, Univ. London, London, U.K., 1998.
[35] B. Wright and B. Morrice, "Landsat TM spectral information to enhance the land cover of Scotland 1988 dataset," *Int. J. Remote Sensing*, vol. 18, pp. 3811–3834, 1997.

**Martin Brown** received the B.Sc. degree (applied mathematics) from the University of Bath, Bath, U.K., in 1989, and the Ph.D. degree form the University of Southampton, Southampton, U.K., in 1993 for his research into neural and fuzzy systems.

He became a Lecturer with the Department of Electronics and Computer Science, Southampton University, in 1994, and joined Unilever Research, Port Sunlight, Bebington, U.K., in 1999. His research interests currently include data preprocessing and empirical modeling using inductive learning algorithms and support vector machines, and these techniques are applied in a variety of domains such as environmental modeling, aluminum property modeling, and dynamical systems.

**Hugh G. Lewis** received the B.Eng. degree (Hons.) in control engineering in 1992 and the M.Sc. degree (Eng.) in control systems in 1993, both from The University of Sheffield, Sheffield, U.K., and the Ph.D. degree from the University of Southampton, Southampton, U.K., in remote sensing of clouds using neural networks, in 1998.

His research interests are in neural networks and empirical models for land area estimation from remotely sensed data.

**Steve R. Gunn** received the B.Eng. degree (electronic engineering) and the Ph.D. degree for his research into active contour methods in computer vision, both from the University of Southampton, Southampton, U.K., in 1992 and 1996, respectively.

From 1996 to 1998, he was a Research Fellow investigating intelligent modeling algorithms with the University of Southampton. He became a Lecturer in the Department of Electronics and Computer Science, University of Southampton, in 1998. His research interests currently include computer vision, active contours, support vector machines, and inductive learning algorithms.