Improving the Hyperspectral Linear Unmixing Problem with Unsupervised Cluster and Covariance Estimates

Eugene Brevdo[a], K. Kim Luu[b]

[a]Princeton University, Dept. of Electrical Engineering, Princeton, NJ, USA 08544
[b]Air Force Research Laboratory, Detachment 15, 535 Lipoa Pkwy Ste 200, Kihei, HI, USA 96753

ABSTRACT

The hyperspectral subpixel detection and classification problem has been intensely studied in the downward-looking case, typically satellite imagery of agricultural and urban areas. In contrast, the hyperspectral imaging case when "looking up" at small or distant satellites creates new and unforeseen problems. Usually one pixel or one fraction of a pixel contains the imaging target, and spectra tend to be time-series data of a single object collected over some time period under possibly varying weather conditions; there is little spatial information available. Often, the number of collected traces is less than the number of wavelength bins, and a materials database with imperfect representative spectra must be used in the subpixel classification and unmixing process. A procedure is formulated for generating a "good" set of classes from experimentally collected spectra by assuming a Gaussian distribution in the angle-space of the spectra. Specifically, Kernel K-means, a suboptimal ML-estimator, is used to generate a set of classes. Covariance information from the resulting classes and weighted least squares methods are then applied to solve the linear unmixing problem. We show with cross-validation that Kernel K-means separation of laboratory material classes into "smaller" virtual classes before unmixing improves the performance of weighted least squares methods.

1. INTRODUCTION

A survey of the literature will show that the hyperspectral subpixel detection and classification problem has been thoroughly addressed in the downward-looking case, e.g., imagery of terrestrial features taken from a space-borne sensor[1]. Several major sources of hyperspectral data, such as the AVIRIS dataset, are available for experimentation. In most of these cases, both supervised and unsupervised approaches benefit from the availability of pure pixels in the form of spatially contiguous and nearly homogeneous material representative spectra. Conversely, the hyperspectral imaging case when “looking up” at small or distant satellites creates new and unforeseen problems in characterizing the surface materials[2]. Usually one pixel or a fraction of the pixel contains the imaging target, and groups of spectra tend to be time-series data of the same object collected over some period of time under varying lighting and viewing geometries. Often, the number of collected traces is less than the number of wavelength bins available, and a materials database with imperfect representative spectra must be used in the subpixel classification and unmixing process; pure pixels with *a posteriori* truth data cannot be extracted from the observations.

In these conditions, two primary issues arise from the satellite material characterization problem. The first challenge is to detect and identify the material classes, also called superclusters, comprising the mixed pixel. The second issue is a marginally simpler problem, estimating the relative abundance of each class in the mixed pixel. In this work, we focus on the second issue solely in attempting to improve the solutions to the abundance estimation problem, while employing the linear mixing model. Specifically, assuming that the material classes are known in each case, these classes are broken into smaller clusters that obey certain angular separation properties. We then choose representative spectra from these clusters to solve the linear unmixing problem and obtain improved abundance estimates.

This paper is broken down into the following sections. We first state our assumptions about the spectral observations and the materials database. Second, the linear mixing model is presented with our method of solution. Next, the Kernel K-Means method for choosing physically intuitive clusters from a database of material spectra is described. Finally, we extend our solution to the mixing model with covariance information from the clusters. The last section contains simulations and cross-validation on lab-measured satellite material spectra.

2. APPROACH

We have N_R observed spectra $R = \{r_n\}_{n=1}^{N_R}$ over L wavelength bins. We also have a database of all detected material classes, or superclusters, $S = \{S_i\}_{i=1}^{N_S}$, where each supercluster S_i contains multiple spectra $S_i = \{S_{ik}\}_{k=1}^{|S_i|}$, over the same count of wavelength bins as the observed spectra. All spectra, including observation and database spectra, are assumed normalized to unit area: $s = \tilde{s} / \sum_{l=1}^{L} \tilde{s}_l$. Note that while all the spectra in one supercluster should span a "small" subspace of R^L, this is not necessarily so. Here we introduce the idea of intrinsic dimensionality and physical sources. Consider a highly reflective material such as aluminum. A laboratory reading of spectral data on an aluminum sample will return a wide variation of spectra depending on the angle of the light source, the observer, and the sample itself. A highly specular reflection causes the spectra to be a flat line, a reflection of the white light source, whereas at different lighting angles the specific colors of the surface material are visible. This distinction is apparent in Figure 1.

We now see that the "specular property", or the specular variation, of aluminum is a physical source for variation in the spectral bins, and there must exist a nonlinear embedding in R^L which captures the structure of this spectral variation. For example, if we consider 3 different types of aluminum, each with a specific embedding that spans its specular variation, then we can say that our aluminum supercluster has an intrinsic dimensionality of 3. Optimally there would be enough samples in our database to capture that embedding, and this variation can be extracted to improve the unmixing process.

2.1. Linear Mixing Model

According to the linear mixing model, all observations are linear combinations of spectra from some set of material traces. Given traces $\{m_k\}_{k=1}^{N_M}$, represented as column vectors in R^L, the observation spectra is modeled by

$$r = \alpha_1 m_1 + \alpha_1 m_2 + \cdots + \alpha_k m_k + n \qquad (1)$$

where n is a random vector that represents noise from the observation. Rewritten in matrix notation, we have

$$r_{L\times 1} = M_{L\times K}\alpha_{K\times 1} + n_{L\times 1} \qquad (2)$$

where the subscripts denote dimensions, M is a matrix of the column vectors m_k, and α is a column vector of the relative abundances for each material. For T observations, we can generalize the mixing process:

$$R_{L\times T} = M_{L\times K} A_{K\times T} + N_{L\times T} \qquad (3)$$

where A_{LxT} is a set of column vectors $\alpha^{(t)}$, with a column corresponding to each observation.

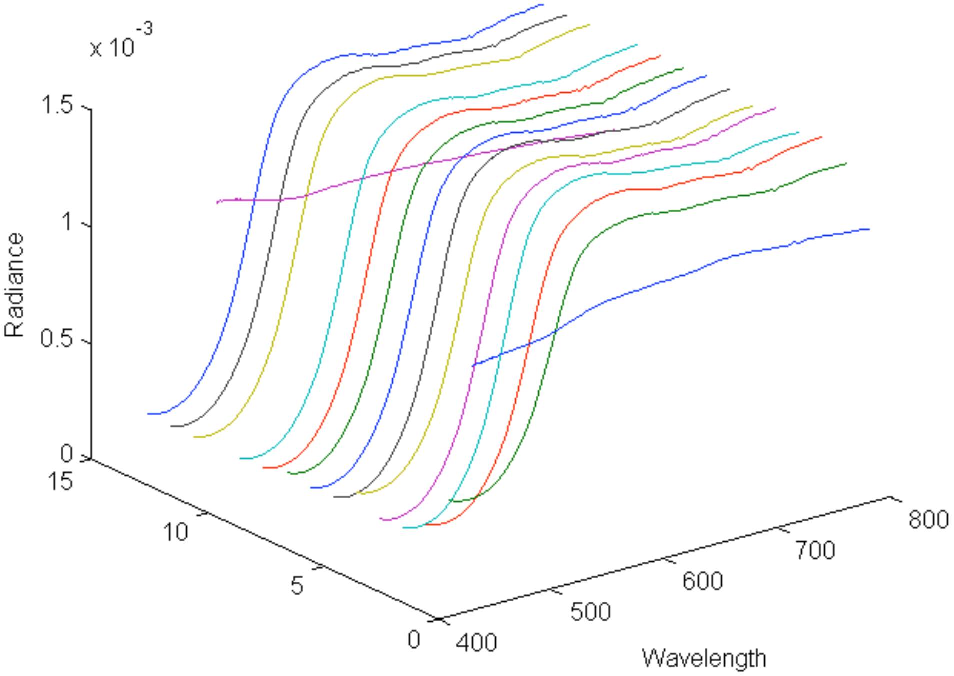

Figure 1: Example database spectra for aluminum with a yellow coating. Collected in 920 wavelength bins between 400 and 800nm.

A number of techniques have been developed to solve the inverse problem of estimating α in Eq. (2). These include a class of Least Squares Error solutions as described in Ref. 1. We solve the fully constrained least squares error formulation:

$$\begin{aligned} &\min_{\alpha} \| M\alpha - r \|_2^2 = \min_{\alpha} \alpha^T M^T M\alpha - 2\alpha^T M^T r + r^T r \\ &s.t. \quad \alpha_k \ge 0 \qquad k = 1,\ldots,N_M \\ &\qquad \sum_{k=1}^{N_M} \alpha_k = 1 \end{aligned} \qquad (4)$$

Letting $H = 2M^T M$, $f = -2M^T r$, $I_{P\times P}$ be the identity matrix with dimension P, and 1_P be a P-length column vector filled with the element 1, we can formulate this problem as a constrained quadratic programming problem

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T H\alpha + \alpha^T f \\ &s.t.\ \ I_{N_M \times N_M}\alpha \geq 0_{N_M} \\ &\quad\ \ 1_{N_M}^T \alpha = 1 \end{aligned} \tag{5}$$

This problem can be solved using Sequential Quadratic Programming with Active-Set Constraints, described as applied to this problem in Refs. 1,3-4, and for our purposes as implemented in the Matlab Optimization Toolbox.

2.2 Kernel K-Means

With the assumption that the detection problem has been solved and that N_S superclusters have been identified as the materials contributing to a given observation *r*, these superclusters can now be broken down into smaller clusters with less within-cluster variance. Given a supercluster S_i, the intrinsic dimensionality of the spectra within S_i depends on the physical variability in the data and on the noise caused by the laboratory data gathering process.

A number of methods are available for ascertaining the virtual dimensionality of a set of points[5], with Principal Component Analysis (PCA) being a prime example. Kernel PCA is an extension of PCA that can efficiently find separation of data points in some higher order nonlinear space[6]. The kernel trick is a well-known method for extending linear algorithms that work with "input spaces", in our case the space spanned by our database spectra, and which can be expressed in terms of dot products. The flexibility of the kernel trick stems from the fact that a nonlinear dot product function can be defined on the input space, thus mapping the input space into a higher dimensional "feature space" in which data is easier to separate. This property lends itself well to the supervised classification problem; the kernel method was first used to extend Support Vector Machines (SVM) to Kernel SVM[7].

More recently, kernel methods have been applied to the unsupervised classification problem of K-means; we will discuss K-means and Kernel K-means shortly. We now derive a kernel that allows us to generate a distance metric comparing the angle between two vectors (in our cases, the spectral angle between two database traces).

Consider the Euclidean distance metric $d(x_1,x_2)^2 = x_1^T x_1 - 2x_1^T x_2 + x_2^T x_2$. Assume that there is some operator $\Phi(\cdot)$ that maps points x_i into a space in which the Euclidean distance metric measures angular separation. A distance value of *1* implies that the two points are fully orthogonal, and a distance value of *0* implies that the two points are equal up to some scaling factor. We write the distance metric in terms of the function Φ on the space of the $x_i's$:

$$\begin{aligned} &d(\Phi(x_i),\Phi(x_j))^2 = \left\|\Phi(x_i) - \Phi(x_j)\right\|^2 = \\ &(\Phi_i - \Phi_j)^T(\Phi_i - \Phi_j) = \Phi_i^T\Phi - 2\Phi_i^T\Phi_j + \Phi_j^T\Phi \end{aligned} \tag{6}$$

We can now express this distance in terms of only dot products of the $\Phi's$. Consider the following kernel:

$$k_{\perp}(x_i,x_j) = \Phi_i^T\Phi_j = 1 - \frac{1}{2}\sin(\theta_{ij}) \tag{7}$$

where

$$\sin(\theta_{ij}) = \sin\left(\cos^{-1}\left(\frac{x_i^T x_j}{\sqrt{x_i^T x_i x_j^T x_j}}\right)\right) = \sqrt{1 - \frac{(x_i^T x_j)^2}{x_i^T x_i x_j^T x_j}} \tag{8}$$

Combining Eqs. (6) and (7), we have the distance metric

$$\begin{aligned} d_\perp(\Phi x_i, \Phi x_j)^2 &= 1 - \frac{1}{2}\sin(\theta_{ii}) - 2\left(1 - \frac{1}{2}\sin(\theta_{ij})\right) + 1 - \frac{1}{2}\sin(\theta_{jj}) \\ &= 2 - 2 + \sin(\theta_{ij}) = \sin(\theta_{ij}) \end{aligned} \tag{9}$$

where θ_{ii}, θ_{jj} are *0*. Note that the kernel $k_\perp$ meets the requirements we imposed on our distance metric.

We estimate the number of variance-maximizing directions in the orthogonality space of our data by running Kernel PCA with the kernel $k_\perp$ on each supercluster S_i. The eigenvalues returned by Kernel PCA, $\{\lambda_i\}_{i=1}^{|S_i|}$ sorted in descending order, for example, can be used for this estimate. Let D_i be the intrinsic dimensionality of supercluster S_i. Then we can estimate D_i:

$$D_i = \max_n \{\lambda_n > \tau\} \tag{10}$$

where $\tau > 0$ is some small threshold. The process of partitioning superclusters is the supervised part in the unmixing process; alternately τ may also be chosen via visual inspection of the respective spectra or eigenvalues for each supercluster. For example, after visual inspection one would expect that the intrinsic dimensionality of the supercluster shown in Figure 1 is 2.

2.2. K-Means and Kernel K-Means Algorithms

Given some cluster count *k* and a set of data points $x = \{x_i\}_{i=1}^{|x|}$, the goal of the K-means algorithm is to find a cluster membership $L : x \to \{v\}_{v=1}^{k}$ such that within-cluster distances between points are minimized and between-cluster distances are maximized. K-means is an iterative algorithm that always converges to some local minimum of this objective function. The result is a set of clusters $\{s^{(v)}\}_{v=1}^{k}$ with the group membership property $s^{(j)} = \{x_i : x_i \in x \cap L(x_i) = j\}$.

Note that the traditional K-means clustering algorithm is not directly suited for separating points (spectra) according to their angular separation. Consider the difference between the performance of regular K-means compared to that of Kernel K-means with the kernel $k_\perp$, as shown in Fig. 3.

The kernel method, as applied to K-means, replaces all distance calculations with functions of dot products between points, as in Eq. (6). This solves the problem of calculating distances between points. The problem of representing class means, however, remains. The following derivation closely follows the one in Ref. 6.

We wish to represent some cluster mean

$$\mu_v = \frac{1}{|s^{(v)}|} \sum_{x \in s^{(v)}} \Phi(x) \tag{11}$$

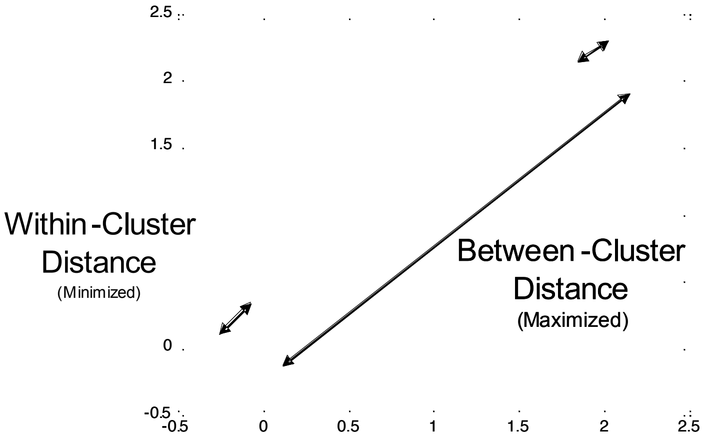

Figure 2: K-means clustering into *k=2* clusters

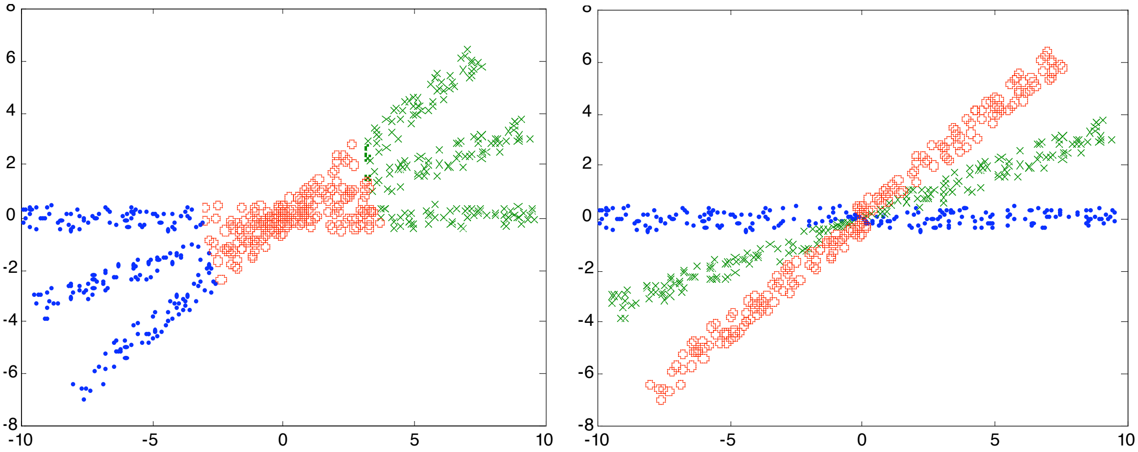

Figure 3: (left) Regular K-means on a dataset of points sampled from 3 randomly distributed variables, angles centered at 20° intervals; (right) Kernel K-means $(k_\perp)$ applied to the same dataset

However we need to represent this mean using dot products. From Eq. (11) it is easy to see that μ_v lies on the span of the $\Phi(x)'s$. We can therefore represent it as some linear combination:

$$\mu_v = \sum_{i=1}^{|S|} \gamma_{vi} \Phi(x_i) \qquad (12)$$

where $0 \le \gamma_{vi} \le 1$. Assuming the $\gamma' s$ are known, the distance from a cluster center to some point is

$$\left\| \Phi(x) - \mu_v \right\|^2 = \Phi(x)^T \Phi(x) - 2\sum_{i=1}^{|S|} \gamma_{vi} \Phi(x)^T \Phi(x_i) + \sum_{i=1,j=1}^{|S|} \gamma_{vi}\gamma_{vj} \Phi(x_i)^T \Phi(x_j) \qquad (13)$$

This distance can be rewritten in terms of kernel functions:

$$\left\| \Phi(x) - \mu_v \right\|^2 = k(x,x) - 2\sum_{i=1}^{|S|} \gamma_{vi} k(x,x_i) + \sum_{i=1,j=1}^{|S|} \gamma_{vi}\gamma_{vj} k(x_i,x_j) \qquad (14)$$

We still need an update rule for the coefficients γ once a point has changed membership. Consider some iteration t and some point x_t. Membership can be tested using Eq. (14):

$$\begin{aligned} &L \leftarrow \underset{v}{\arg\min}\ \left\|\Phi(x_t) - \mu_v\right\|^2 \\ &M_{t,v} = \delta_{L,v} \end{aligned} \tag{15}$$

where $\delta_{i,j}$ is the Kronecker delta function and $M_{t,v}$ is the membership function for centroid v at step t. An update rule that brings centroid μ_v closer to $\Phi(x_t)$ if $v = L$ (they are in the same group) and away from $\Phi(x_t)$ if $v \neq L$ (they are in different groups) is

$$\mu_v^t = \zeta\Phi(x_t) + (1-\zeta)\mu_v^{t-1} \tag{16}$$

where ζ is chosen to bring μ_v^t closer to $\Phi(x_t)$ when $v = L$. Choosing

$$\zeta_v = \begin{cases} M_{t,v} \Big/ \sum_{i=1}^{t} M_{t,v} & v = L \\ 0 & v \neq L \end{cases} \tag{17}$$

has the desired effect.

Replacing μ_v in Eq. (16) with the representation from Eq. (12) provides a way of updating just the terms γ:

$$\sum_{i=1}^{|S|} \gamma_{vi}^t \Phi(x_i) = \zeta_v \Phi(x_t) + (1-\zeta_v) \sum_{i=1}^{|S|} \gamma_{vi}^{t-1} \Phi(x_i) \tag{18}$$

Matching up the $\gamma's$, we get:

$$\gamma_{v,i}^t = \begin{cases} \gamma_{v,i}^{t-1}(1-\zeta_v) & i \neq t \\ \zeta_v + \gamma_{v,i}^{t-1}(1-\zeta_v) & i = t \end{cases} \tag{19}$$

While PCA and eigen-thresholding methods can provide an estimate of the dimensionality of a supercluster, we suggest an appropriate thresholding method when clustering. Define some maximum allowed cluster diameter $\Delta_{\max}$ and some maximum number of clusters *C*. Then one method of grouping is to make sure that for any cluster *k*, the diameter is smaller than the maximum.

Once a set of superclusters S has been broken down into smaller, more compact groups $\left\{s^{(v)}\right\}_{v=1}^{|s|}$, a representative must be chosen from each group for inclusion in the matrix *M* to use in the solution of Eq. (4). Letting m_v be the Euclidean mean of the spectra in $s^{(v)}$, we generate the *unmixing* matrix $M = \left[m_1 \quad \cdots \quad m_{|s|}\right]$.

2.3. Covariance Information

The quadratic programming problem as posed in Eq. (4) is a least squares problem that can be extended via a weighting matrix A:

$$\begin{aligned} &\min_{\alpha}\ (M\alpha - r)^T A(M\alpha - r) = \min_{\alpha}\ \alpha^T M^T A M\alpha - r^T A M\alpha - \alpha^T M^T A r + r^T A r \\ &s.t.\ \ \alpha_k \geq 0 \qquad k = 1,\ldots,N_M \\ &\qquad \sum_{k=1}^{N_M} \alpha_k = 1 \end{aligned} \tag{20}$$

Good results have been achieved[8] using the inverse of Fisher's within-class scatter matrix $A = C_W^{-1}$. Note that the scatter matrix is closely related to the covariance matrices for the clusters $s^{(v)}$. Given some set of clusters $\{s^{(v)}\}_{v=1}^{|s|}$, and assuming that each cluster is generated from a Gaussian distribution with center m_v, C_W can be calculated:

$$C_W = \sum_{v=1}^{|s|} C_v \qquad\qquad C_v = \sum_{x \in s^{(v)}} (x - m_v)(x - m_v)^T \tag{21}$$

The symmetric within-class scatter matrix C_W can be properly estimated when there are more than L traces in total within the clusters of laboratory spectra (where L is the number of wavelength bins). More specifically, if the count of linearly independent database spectra in $\{s^{(v)}\}_{v=1}^{|s|}$ is less than L, the scatter matrix C_W is singular and has no inverse.

Note that the objective function in Eq. (20) can be rewritten as

$$\begin{aligned} &\min_{\alpha}\ \alpha^T \left(A^{1/2}M\right)^T \left(A^{1/2}M\right)\alpha - \left(A^{1/2}r\right)^T \left(A^{1/2}M\right)\alpha - \alpha^T \left(A^{1/2}M\right)^T \left(A^{1/2}r\right) \\ =\ &\min_{\alpha}\ \alpha^T \tilde{M}^T \tilde{M}\alpha - 2\tilde{r}^T \tilde{M}\alpha \\ =\ &\min_{\alpha}\ \left\|\tilde{M}\alpha - \tilde{r}\right\|_2^2 \end{aligned} \tag{22}$$

where the root matrix $A^{1/2}A^{1/2} = A$, $\tilde{M} = A^{1/2}M$, and $\tilde{r} = A^{1/2}r$. Note that the reduced form of Eq. (22) has now been expressed as in the original problem and requires an eigenvalue decomposition of the matrix $A = C_W^{-1}$. Alternatively, the weighted least squares problem in Eq. (20) can be solved via traditional SQP methods by setting $H = 2M^T C_W^{-1} M$, and $f = -2M^T C_W^{-1} r$ in Eq. (5).

Note that the data points are the same whether K-means is applied to the superclusters, but the cluster centers m_v vary. As K-means is a reduced form of the Gaussian EM algorithm, we expect that after applying K-means, the new cluster centers are a better fit for the true probability distributions of the spectra within the superclusters; as such, the individual covariance matrices C_v then are better estimates of the second order variations within the laboratory spectra.

3. EXPERIMENTS AND RESULTS

Defining the performance of an unmixing method by the absolute error in its abundance estimates, we used cross-validation on two superclusters to compare the performance of abundance estimation. In the first experiment, two superclusters are employed, the material difference being the coatings on aluminum. One supercluster is yellow coating, and the second is red.

Choosing a distance metric $d(s_i, s_j) = \left\| s_i - s_j \right\|_2$, a maximum cluster count $C_{\max} = 3$, and a maximum cluster diameter $\Delta_{\max} = 1.5$, we broke down both superclusters into smaller clusters. That is, we iterated the kernel k-means algorithm by increasing c, breaking down the supercluster into smaller and smaller clusters $\{S_{i,k}\}_{k=1}^{c}$ until either $\max\limits_{k} \max\limits_{s_1, s_2 \in S_{i,k}} d(s_1, s_2) \le \Delta_{\max}$ or $c = C_{\max}$.

The results of the unsupervised classification are displayed in Fig. 4 in which cluster membership is denoted by different types of lines (e.g., no dashes, long dashes, and short dashes). Note that in each case, a cluster count $c = 3$ reduced the maximum diameter below $\Delta_{\max}$.

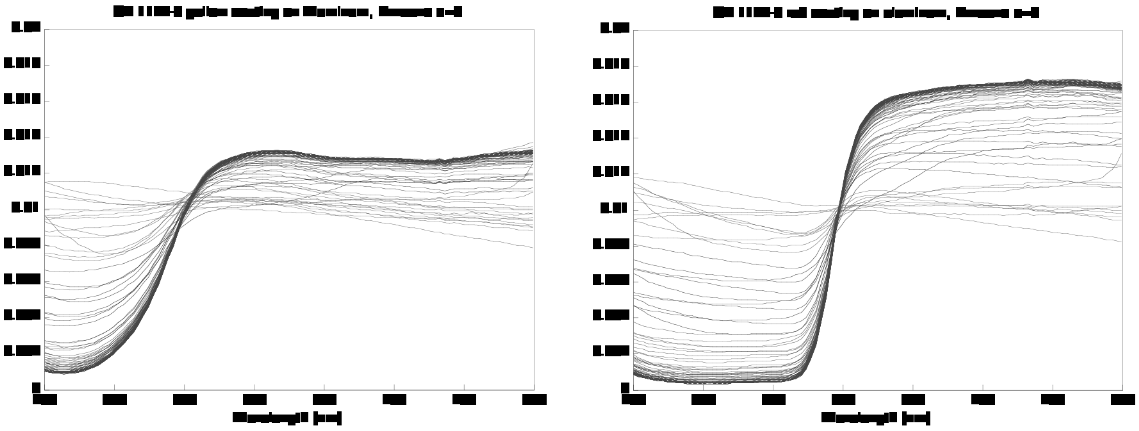

Figure 4: Results of Kernel K-means on (left) S_1 and (right) S_2

We then ran validation for each pair of spectra within S_1 and S_2. Superclusters S_1 and S_2 are each composed of 114 spectra; we therefore ran a total of 12996 simulations. For each simulation, the forward model mixing matrix was varied with $a = \{0, .1, .25, .5, .75, .9, 1\}$. The observation in each case was generated as

$$r = a_i S_1^{(n)} + (1 - a_i) S_2^{(p)} \tag{23}$$

for each $a_i \in a$ and for each pair of traces $S_1^{(n)} \in S_1$, $S_2^{(p)} \in S_2$. We first unmixed using traditional class means extracted from the superclusters (Standard case). Then a scatter matrix C_W was generated from the superclusters and unmixed using the WLS formulation (Covariance case). Finally, we selected the means from the six clusters generated via Kernel K-means (K-means Standard case) and generated a new scatter matrix from the six clusters and again unmixed using the WLS formulation (K-means Covariance case). To represent an estimate of the abundance in cluster S_1, we averaged the abundance α_i over each of the 12996 simulations; we also calculated the standard deviation. For K-means, the supercluster S_1 was spread over three clusters in the unmixing matrix; α_i is the sum of the estimated abundances of these three clusters. Table 1 contains the average error in the abundance estimates $\overline{\alpha}_i$, along with the aforementioned standard deviations.

Note in Table 1, that the Standard unmixing method is very inaccurate, with the standard deviation of abundance estimates at .30 when the true abundance was between 0 and 25%. Also note the significant reduction in error when the scattering matrix is used when unmixing (Covariance). K-means will usually, but not always, improve the abundance estimate over Standard; counter-examples include a higher standard deviation when the true abundance was between 75% and 100%. Finally, note that K-means with the scattering matrix (K-means Covariance) consistently outperformed K-means Standard. The average abundance estimated is somewhat closer to the true abundance, and the standard deviation of K-means Covariance estimates is consistently between ½ and ¾ of the standard deviation of regular Covariance estimates.

Table 1: Unmixing error in abundance estimates from spectra of CV 1144-5 (yellow coating)

	True a_i	**0%**	**10%**	**25%**	**50%**	**75%**	**90%**	**100%**
Standard	Error	0.136	0.048	-0.011	-0.008	-0.006	-0.023	-0.046
	Std Dev	0.305	0.309	0.284	0.197	0.124	0.070	0.027
Covariance	Error	0.033	0.002	0.000	0.000	0.000	-0.002	-0.035
	Std Dev	0.067	0.081	0.073	0.064	0.070	0.077	0.059
K-means Standard	Error	0.022	0.014	0.015	-0.003	-0.028	-0.040	-0.032
	Std Dev	0.100	0.127	0.150	0.144	0.139	0.139	0.095
K-means Covariance	Error	0.016	0.002	-0.001	-0.002	-0.005	-0.008	-0.019
	Std Dev	0.044	0.053	0.050	0.044	0.047	0.051	0.038

What happens if the superclusters are separated into more sets? Setting a smaller $\Delta_{\max} = .5$ and a higher $C = 6$ during the clustering process, we broke up S_1 into 4 clusters and S_2 into 5 clusters. Note from Table 2 that performance of both the K-means and K-means Covariance methods does not significantly change. At some point, creating smaller and smaller clusters does not significantly improve the estimation process.

Table 2: Higher Clustered K-means unmixing results on abundance from spectra of CV 1144-5 yellow coating on aluminum

	True a_i	**0%**	**10%**	**25%**	**50%**	**75%**	**90%**	**100%**
K-means(4,5)	Error	0.023	0.026	0.025	0.004	-0.024	-0.038	-0.030
	Std Dev	0.100	0.124	0.149	0.142	0.137	0.140	0.095
K-means(4,5) Covariance	Error	0.014	0.001	-0.001	-0.003	-0.007	-0.010	-0.015
	Std Dev	0.046	0.056	0.052	0.044	0.046	0.048	0.034

Is this performance consistent for other data sets? Consider two other superclusters, S_1 is very specular aluminum and S_2 is solar cell material. These two superclusters and the results of the Kernel K-means angular separation are shown in Fig. 5. With a total of 3420 simulations, each simulation for each of 7 a_i values of specular aluminum, the results are shown in Table 3. Note that in this case, K-means Standard consistently improves performance over Standard, and K-means Covariance performance is comparable to the regular Covariance method. Thus, the K-means algorithm may also be one alternative to the Covariance method, when there are less laboratory spectral traces available than there are wavelength bins.

Table 3: Unmixing results on abundance from spectra of specular aluminum

	True a_i	**0%**	**10%**	**25%**	**50%**	**75%**	**90%**	**100%**
Standard	Error	0.101	0.047	0.008	0.001	-0.001	-0.040	-0.108
	Std Dev	0.148	0.168	0.183	0.185	0.221	0.231	0.222
Covariance	Error	0.026	0.000	0.000	0.000	0.000	0.000	-0.020
	Std Dev	0.035	0.054	0.047	0.040	0.043	0.048	0.035
K-means Standard	Error	0.057	0.038	0.028	0.012	-0.005	-0.021	-0.035
	Std Dev	0.145	0.156	0.164	0.145	0.129	0.128	0.116
K-means Covariance	Error	0.019	0.003	0.002	0.002	0.002	0.000	-0.015
	Std Dev	0.042	0.052	0.046	0.039	0.041	0.043	0.030

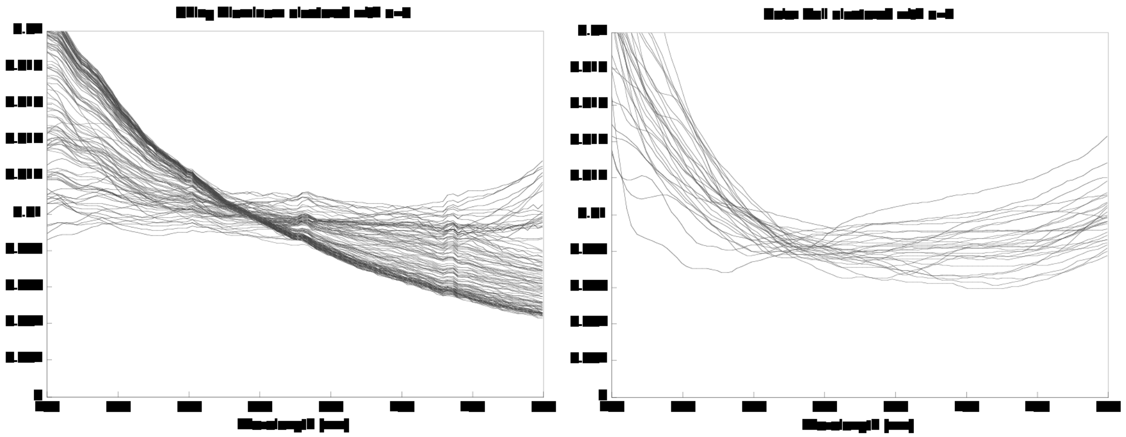

Figure 5: Two superclusters, (left) specular aluminum, 114 traces, (right) solar cell material, 30 traces

Finally, we compared the performance when mixing four different superclusters at random. The four materials were S_1, a tan/yellow target; S_2, a beige/yellow target; S_3, a very white coating on aluminum; and S_4, the red coating from the original experiments. Each supercluster was composed of 114 spectra. We ran 12^4 simulations (cross-mixing of 12 spectra from each supercluster). Selecting abundances for each simulation at random, the mixing percentage for each material was randomly distributed with a mean of 25% in each simulation. The fractional abundances always added up to 1. K-means with $\Delta_{\max} = 1$ and $C = 5$ resulted in cluster counts $k_i = \{3,3,2,4\}$ for the respective superclusters; the resulting unmixing matrix had 18 columns. Error estimates in the unmixing process are shown Table 4.

Table 4: Unmixing results on abundance from spectra of specular aluminum

	Cluster	S_1	S_2	S_3	S_4
Standard	Error	0.130878	0.166294	0.17307	0.073953
	Std Dev	0.124748	0.146678	0.172373	0.092332
Covariance	Error	0.080775	0.064552	0.077682	0.010172
	Std Dev	0.086918	0.065668	0.080253	0.008636
K-means Standard	Error	0.09983	0.117553	0.095194	0.040648
	Std Dev	0.117511	0.13149	0.116221	0.07716
K-means Covariance	Error	0.063274	0.057469	0.062983	0.009314
	Std Dev	0.075366	0.053667	0.081526	0.008321

4. CONCLUSIONS

By assuming that the detection problem has been solved at the supercluster level, we reduced the general problem of subpixel classification to calculating the abundances of the materials comprising these superclusters. While the detection problem, which is remarkably intractable for the case of small or distant satellites, has not been addressed, our efforts to improve solutions for the abundance estimation problem will hopefully motivate future work on the more difficult problem. Further, we theorize that unsupervised clustering methods can be extended to the semi-supervised detection problem.

We have demonstrated with several simulations using laboratory-measured spectra that using the Kernel K-means algorithm to further divide a large material class into smaller clusters with less within-cluster variance will improve the abundance estimation for the supercluster. In one case, we employ spectra from aluminum material with a yellow coating and mix those spectra with those from aluminum material with a red coating. We show that our results are consistent by varying the materials to include a mixture of specular aluminum and solar cell materials and finally increasing the number of superclusters to four materials. The Kernel K-means results are significantly more accurate under almost all conditions than the standard procedure. Our experiments also show that the results do not continue to improve by simply defining smaller and smaller clusters. Furthermore, we tested the impact of adding covariance information to extend the least square estimation problem with a weighting matrix. The covariance generally improves both standard estimation and estimation with Kernel K-means, but it requires a large number of spectra in each cluster.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Air Force Office of Scientific Research for their support of this work and the Directed Energy Scholars internship program. We also thank Dr. Chuck Matson for his interest and feedback.

REFERENCES

1. C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenum Publishers, New York, NY, 2003.

2. K.K. Luu, C.L. Matson, et al., "Object Characterization from Spectral Data," *Proceedings of the 2003 AMOS Technical Conference*, Maui Economic Development Board, Kihei, HI, 2003.
3. P.E. Gill, W. Murray, et al., "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Trans. Math. Software*, Vol. 10, pp 282-298, 1984.
4. Gill, P.E., W. Murray, and M.H. Wright, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison Wesley, 1991.
5. C.-I Chang and Q. Du, "Estimation of Number of Spectrally Distinct Signal Sources in Hyperspectral Imagery," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 42, No. 3, pp. 608-619, March 2004.
6. B. Schölkopf, A. Smola, and K.R. Müller, *Nonlinear Component Analysis as a Kernel Eigenvalue Problem,* Neural Computation, Vol. 10, pp. 1299-1319, 1998.
7. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, NY, 1995.
8. B. Ji, C-I. Chang, et al., "Weighted Least Squares Error Approaches to Abundance-Constrained Linear Spectral Mixture Analysis," *Proceedings of SPIE,* International Society for Optical Engineering, Vol. 5806, pp. 691-702, 2005.