Unsupervised Clustering and Active Learning of Hyperspectral Images with Nonlinear Diffusion

James M. Murphy Mauro Maggioni

***Abstract*—The problem of unsupervised learning and segmentation of hyperspectral images is a significant challenge in remote sensing. The high dimensionality of hyperspectral data, presence of substantial noise, and overlap of classes all contribute to the difficulty of automatically clustering and segmenting hyperspectral images. We propose an unsupervised learning technique called spectral-spatial diffusion learning (DLSS) that combines a geometric estimation of class modes with a diffusion-inspired labeling that incorporates both spectral and spatial information. The mode estimation incorporates the geometry of the hyperspectral data by using diffusion distance to promote learning a unique mode from each class. These class modes are then used to label all points by a joint spectral-spatial nonlinear diffusion process. A related variation of DLSS is also discussed, which enables active learning by requesting labels for a very small number of well-chosen pixels, dramatically boosting overall clustering results. Extensive experimental analysis demonstrates the efficacy of the proposed methods against benchmark and state-of-the-art hyperspectral analysis techniques on a variety of real datasets, their robustness to choices of parameters, and their low computational complexity.**

I. Introduction

A. Machine Learning for Hyperspectral Data

Hyperspectral imagery (HSI) has emerged as a significant data source in a variety of scientific fields, including medical imaging [1], chemical analysis [2], and remote sensing [3]. Hyperspectral sensors capture reflectance at a sequence of localized electromagnetic ranges, allowing for precise differentiation of materials according to their spectral signatures. Indeed, the power of hyperspectral imagery for material discrimination has led to its proliferation, making manual analysis of hyperspectral data infeasible in many cases. The large data size of HSI, combined with their high dimensionality, demands innovative methods for storage and analysis. In particular, efficient machine learning algorithms are needed to automatically process and glean insight from the deluge of hyperspectral data now available.

The problem of *HSI classification*, or supervised segmentation, is to label each pixel in a given HSI as belonging to a particular class, given a training set of labeled samples (pixels) from each class. A variety of statistical and machine learning techniques have been used for HSI classification, including nearest-neighbor and nearest subspace methods [4], [5], support vector machines [6], [7], neural networks [8], [9], [10] and regression methods [11], [12]. These methods are design to perform well especially when the number of labeled training pixels is large.

The process of labeling pixels typically requires an expert and it is costly. This motivates the design of machine learning techniques that require little or no labeled training data. So on the other end of the spectrum from classification, we have the problem of *HSI clustering*, or *unsupervised* segmentation, which has the same goal as HSI classification, but no labeled training data is available. This is considerably more challenging, and is an ill-posed problem unless further assumptions are made, for example about the distribution of the data and how it relates to the unknown labels. Recent techniques for hyperspectral clustering include those based on particle swarm optimization [13], Gaussian mixture models (GMM) [14], nearest neighbor clustering [15], total variation methods [16], density analysis [17], sparse manifold models [18], [19], hierarchical nonnegative matrix factorization (HNMF) [20], graph-based segmentation [21], and fast search and find of density peaks clustering (FSFDPC) [17], [22], [23].

Another interesting modality is *active learning* for HSI classification. This is a supervised technique where a small, automatically but carefully chosen set of pixels is labeled, as opposed to the standard supervised learning setting, in which the labels are usually randomly selected. Active learning can lead to high quality classification results with significantly fewer labeled samples than in the case of randomly selected training data. Since far fewer training points are available in the active learning setting, the structure of the data may be analyzed with unsupervised learning, in order to decide which data points to query for labels. Thus, active learning may be understood as a form of *semisupervised learning* that exploits both global structure of the data—learned without supervision—and a small number of supervised training data points. A variety of active learning methods have been successfully deployed in remote sensing [24], including those based on relevance feedback [25], region-based heuristics [26], exploration-based heuristics [27], belief propagation [28], support vector machines [29], and regression [30].

Machine learning for HSI suffers from several major challenges. First, the dimensionality of the data to be analyzed is high: it is not uncommon for the number of spectral bands in an HSI to exceed 200. The corresponding sampling complexity for such a high number of dimensions renders classical statistical methods inapplicable. Second, clusters in HSI are typically nonlinear in the spectral domain, rendering methods that rely on having linear clusters ineffective. Third,

J.M. Murphy is with the Department of Mathematics at Tufts University; email: JM.Murphy@tufts.edu

M. Maggioni is with the Department of Mathematics, Department of Applied Mathematics and Statistics, Institute of Data Intensive Engineering and Science, and the Mathematical Institute of Data Sciences at Johns Hopkins University; email: mauro.maggioni@jhu.edu

there is often significant noise and between-cluster overlap among HSI classes, due to the materials being imaged and poor sensing conditions. Finally, HSI images may be quite large, requiring machine learning methods with computational complexity essentially linear in the number of pixels.

This article addresses the problems of HSI clustering and, relatedly, active learning, which overcome these significant challenges. The methods we propose combine density-based methods with geometric learning through diffusion geometry [31], [32] in order to identify class modes. This information is then used to propagate labels on training data to all data points through a nonlinear process that incorporates both spectral and spatial information. The use of data-dependent diffusion maps for mode detection significantly improves over current state-of-the-art methods experimentally, and also enjoys robust theoretical performance guarantees [33]. The use of diffusion distances exploits low-dimensional structures in the data, which allows the proposed method to handle data that is high-dimensional but intrinsically low-dimensional, even when nonlinear and noisy. Moreover, the spectral-spatial labeling scheme takes advantage of the geometric properties of the data, and greatly improves the empirical performance of clustering when compared to labeling based on spectral information alone. In addition, the proposed unsupervised method assigns to each data point a measure of confidence for the unsupervised label assignment. This leads naturally to an active learning algorithm in which points with low confidence scores are queried for training labels, which then propagate through the remaining data. The proposed algorithms enjoy nearly linear computational complexity in the number of pixels in the HSI and in the number of spectral dimensions, thus allowing for its application to large scenes. Extensive empirical results, including comparisons with many state-of-the-art techniques, for our method applied to HSI clustering and active learning are in Sections III-E and III-F, respectively.

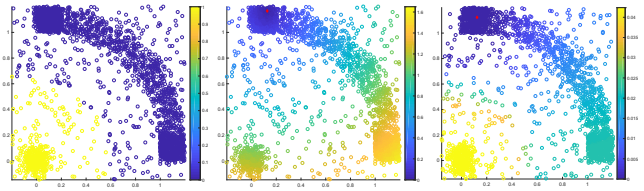
B. Overview of Proposed Method

The proposed unsupervised clustering method is provided with data $X = \{x_n\}_{n=1}^N \subset \mathbb{R}^D$ (for HSI, N = number of pixels and D = number of spectral bands) and the number K of classes, and outputs labels $\{y_n\}_{n=1}^N$, each $y_n \in \{1, \ldots, K\}$, by proceeding in two steps:

1. **Mode Identification**: This step consists first in performing density estimation and analyzing the geometry of the data to find K *modes* $\{x_i^*\}_{i=1}^K$, one for each class.
2. **Labeling Points**: Once the modes are learned, they are assigned a unique label. Remaining points are labeled in a manner that preserves spectral and spatial proximity.

By a mode, we mean a point of high density within a class, that is representative of the entire class. We assume K is known, but otherwise we have no access to labeled data; in Section V we discuss a method for estimating K.

One of the key contributions of this article is to measure similarities in the spectral domain not with the widely used Euclidean distance or distances based on angles (correlations) between points, but with *diffusion distance* [31], [32], which is a data-dependent notion of distance that accounts for the geometry—linear or nonlinear—of the distribution of the data. The motivation for this approach is to attain robustness with respect to the shape of the distributions corresponding to the different classes, as well as to high-dimensional noise. The modes, suitably defined via density estimation, are robust to noise, and the process we use to pick only one mode per class is based on diffusion distances. The labeling of the points from the modes respects the geometry of the data, by incorporating proximity in both spectral and spatial domains.



(a) *Nonlinear, multimodal example data.* (b) *Euclidean distance from* $(0,1)$. (c) *Diffusion distance from* $(0,1)$.

Fig. 1: *In this* 2*-dimensional example, data is drawn from two distributions* μ_1 *and* μ_2*.* μ_1 *is a mixture of two isotropic Gaussians with means at* $(0,1)$ *and* $(1,0)$*, respectively, connected by a set of points uniformly sampled from a nonlinear, parabolic shape.* μ_2 *is an isotropic Gaussian with mean at* $(0,0)$*. Samples of uniform background noise are added and labeled according to their nearest neighbor among the two clusters. The data is plotted and colored by cluster in subfigure (a). We plot the distances from the point* $(0,1)$ *in the Euclidean and diffusion distances in subfigures (b), (c), respectively. The "parabolic rectangle" acts as a "bridge" between the two Gaussians and causes the high density regions near* $(0,1)$ *and* $(1,0)$ *to be closer in diffusion distance than they would be in the usual Euclidean distance. The bridge is overcome efficiently with diffusion distance, because there are many paths with short edges connecting the high density regions across this bridge.*

We model X as samples from a distribution $\mu = \sum_{i=1}^K w_i \mu_i$, where each μ_i corresponds to the probability distribution of the spectra in class i, and the nonnegative weights $\{w_i\}_{i=1}^K$ correspond to how often each class is sampled, and satisfy $\sum_{i=1}^K w_i = 1$. More precisely, sampling $x \sim \mu$ means first sampling $Z \sim \text{Multinomial}(w_1, \ldots, w_K)$, then sampling from μ_i conditioned on the event $Z = i \in \{1, \ldots, K\}$.

1) Mode Identification: The computation of the modes is a significant aspect of the proposed method, which we now summarize for a general dataset X, consisting of K classes. The mode identification algorithm outputs a point x_i^* ("mode") for each μ_i. We make the assumption that modes of the constituent classes can be characterized as a set of points $\{x_i^*\}_{i=1}^K$ such that

1) the empirical density of each x_i^* is relatively high;
2) the diffusion distance between pairs $x_i^*, x_{i'}^*$, for $i \neq i'$, is relatively large.

The first assumption is motivated by the fact that points of high density ought to have nearest neighbors corresponding to a single class; the modes should thus produce neighborhoods of points that with high confidence belong to a single class. However, there is no guarantee that the K densest points will correspond to the K unique classes: some classes may have a multimodal distribution, meaning that the class has several modes, each with potentially higher density than the densest point in another class. The second assumption addresses this issue, requiring that modes belonging to different distributions are far away in diffusion distance.

Enforcing that these modes are far apart in diffusion distance has several advantages over enforcing they are far

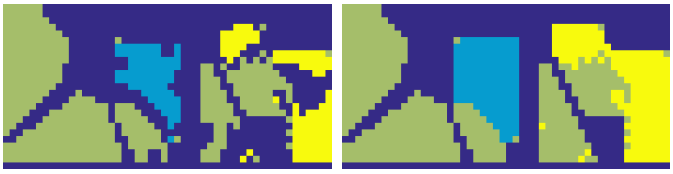
apart in Euclidean distance. Importantly, it leads, empirically, to a unique mode from each class. This is true even when certain classes are multimodal. Moreover, diffusion distances are robust with respect to the shape of the support of the distribution, and are thus suitable for identifying nonlinear clusters. An instance of these advantages of diffusion distance is illustrated in the toy example Figure 1, with the results of the proposed mode detection algorithm in Figure 3. We postpone the mathematical and algorithmic details to Section II-B.

2) Labeling Points: At this stage we assume that we found exactly one mode x_i^* for each class, to which a unique and arbitrary class label is assigned. The remaining points are now labeled in a two-stage scheme, which takes into account both spectral and spatial information. It is known that the incorporation of spatial information with spectral information has the potential to improve machine learning of hyperspectral images, compared to using spectral information alone [7], [28], [23], [34], [35], [36], [37], [38], [39], [40]. Spatial information is computed for each pixel by constructing a neighborhood of some fixed radius in the spatial domain, and considering the labels within this neighborhood. For a given point, let *spectral neighbor* refer to a near neighbor with distances measured in the spectral domain, and let *spatial neighbor* refer to a near neighbor with distances measured in the spatial domain.

In the first stage, a point is given the same label as its nearest spectral neighbor of higher density, unless that label is sufficiently different from the labels of the point's nearest spatial neighbors, in which case the point is left unlabeled. This produces an incomplete labeling in which we expect the labeled points to be far from the spectral and spatial boundaries of the classes, since these are points that are unlikely to have conflicting spectral and spatial labels. The first stage thus labels points using only spectral information, though spatial information may prevent a label from being assigned.

In the second stage we label each of the points left unlabeled in the first stage, by assigning the *consensus label* of its nearest spatial neighbors (see Section II-C), if it exists, or otherwise the label of its nearest spectral neighbor of higher density. In this way the yet unlabeled points, typically near the spatial and spectral boundaries of the classes, benefit from the spatial information in the already labeled points, which are closer to the centers of the classes. The second stage thus labels points using both spectral and spatial information. Figure 2 shows an instance of this two-stage labeling process.

This method of clustering combines the diffusion-based learning of modes with the joint spectral-spatial labeling of pixels and is called *spectral-spatial diffusion learning* (DLSS), detailed in Section II-C. We contrast it with another novel method we propose, called *diffusion learning (DL)*, in which modes are learned as in DLSS, but the labeling proceeds simply by enforcing that each point has the same spectral label as its nearest spectral neighbor of higher density. DL therefore disregards spatial information, while DLSS makes significant use of it, particularly in the second stage of the labeling. Our experiments show that while both DL and DLSS perform very well, DLSS is generally superior.



(a) *First stage labeling, using spectral information only.* (b) *Second stage, final labeling, using spectral and spatial information jointly.*

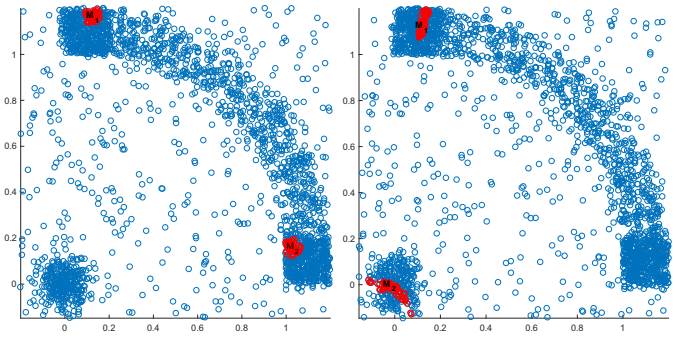
Fig. 2: *An example of the two-stage spectral-spatial labeling process, performed on the Indian Pines dataset used for experiments in Section III-E1. In subfigure (a), the partial labeling from the first stage is shown. After mode identification, points are labeled with the same label as their nearest spectral neighbor of higher density, unless that label is different from the consensus label in the spatial domain, in which case a point is left unlabeled. This leads to points far from the centers of the classes staying unlabeled after the first stage. In the second stage, unlabeled points are assigned labels by the same rule, unless there is a clear consensus in the spatial domain, in which case the unlabeled point is given the consensus spatial label; the results of this second stage appear in subfigure (b). For visual clarity, here and throughout the paper, pixels without ground truth (GT) labels are masked out.*

C. Major Contributions

We propose a clustering algorithm for HSI with several significant innovations. First, diffusion distance is proposed to measure distance between high-density regions in hyperspectral data, in order to determine class modes. Our experiments show that this distance efficiently differentiates between points belonging to the same cluster and points in different clusters. This correct identification of modes from each cluster is essential to any clustering algorithm incorporating an analysis of modes. Compared to state-of-the-art fast mode detection algorithms, the proposed method enjoys excellent empirical performance; theoretical performance guarantees are beyond the scope of the present article and will be discussed in a forthcoming article [33].

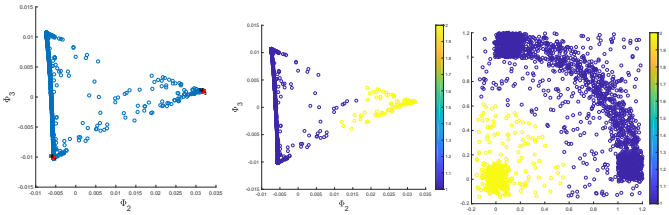
A second major contribution of the proposed HSI clustering algorithm is the incorporation of spatial information through the labeling process. Labels for points are determined by diffusing in the spectral domain from the labeled modes, unless spatial proximity is violated. By not labeling points that would violate spatial regularity, the proposed algorithm first labels points that, with high confidence, are close to the spectral modes of the distributions. Only after labeling all of these points are the remaining points, further from the modes, labeled. This enforces a spatial regularity which is natural for HSI, because under mild assumptions, a pixel in an HSI is likely to have the same label as the most common label among its nearest spatial neighbors [7], [28], [23], [34], [35], [36], [37], [38], [39], [40]. In both stages, DLSS takes advantage of the geometry of the dataset by using data-adaptive diffusion processes, greatly improving empirical performance. The proposed methods are $O(ND\log(N))$ in the number of points (N) and ambient dimension of the data (D) when the intrinsic dimension of the data is small, and thus have near optimal complexity, suitable for the big data setting.

A third major contribution is the introduction of an *active learning* scheme based on distances of points to the computed modes. In the context of active learning, the user is allowed to label only a very small number of points, to be chosen parsimoniously. We propose an unsupervised method for determining which points to label in the active learning setting. We



(a) *Euclidean modes* (b) *Diffusion modes*

Fig. 3: *Learned modes with Euclidean distances and diffusion distances. The Euclidean and diffusion distances from* $(0, 1)$ *are shown in subfigures (b), (c) of Figure 1, while the corresponding learned modes are labeled, with nearby points colored red in subfigures (a), (b), of the present figure. Notice that the proposed diffusion learning method, using diffusion distances, correctly learns* M_1, M_2 *from different clusters (b), while using Euclidean distances leads assigning both* M_1, M_2 *to the same cluster (a), which would lead to poor clustering results.*



(a) *Low-dimensional embedding and learned modes.* (b) *Labeling with diffusion distances and learned modes.* (c) *Learned labels projected on original data.*

Fig. 4: *In Subfigure (a), the data from Figure 1 is represented in a new coordinate system, given by the second and third eigenfunctions of a Markov transition matrix. In this coordinate system, the natural Euclidean distance is equal to the diffusion distance on the original image. It is seen that the two ends of the parabolic segment are much closer in this embedding than in the original data, owing to the many short paths connecting them. The learned modes are labeled in this low-dimensional embedding as in Figure 3, subfigure (b). In subfigure (b) of the present figure, points are labeled according to the proposed algorithm based on diffusion distance and the learned modes. Subfigure (c) shows the labels projected onto the original data, which conforms closely with the cluster structure in the data and the labels in Figure 1 (a).*

note that pixels that are equally far in diffusion distance from their nearest two modes are likely to be near class boundaries, and hence to be the most challenging pixels to label by the proposed unsupervised method. Our active learning method requires the labels of only the pixels whose distances to their nearest two modes are closest. The proposed active learning method builds naturally on the fully unsupervised method, since the computation of distances to nearest mode are already computed by the DL and DLSS algorithms, and hence the computational complexity of the proposed active learning method does not differ significantly from the fully unsupervised method. Our experiments show that this method can dramatically improve labeling accuracy with a number of labels $\ll 1\%$ of the total pixels. This work is detailed in Section III-F.

II. Unsupervised Learning Algorithm and Active Learning Variation

A. Motivating Example and Approach

A key aspect of our algorithm is the method for identifying the modes of the classes in the HSI data. This is challenging because of the high ambient dimension of the data, potential overlaps between distributions at their tails, along with differing densities, sampling rates, and distribution shapes.

Consider the simplified example in Figure 3, showing the same data set as that in Figure 1. The points of high density lie close to the center of μ_2, and close to the two ends of the support of μ_1. After computing an empirical density estimate, the distance between high density points is computed. If Euclidean distance is used to remove spurious modes, i.e. modes corresponding to the same distribution, then the learned modes M_1, M_2 both correspond to μ_2; see subfigure (a) of Figure 3. When diffusion distance is used rather than Euclidean distance, the learned modes M_1, M_2 correspond to two different classes; see subfigure (b) of Figure 3. This is because the modes on the opposite ends of the support of μ_2 are far in Euclidean distance but relatively close in diffusion distance. Furthermore, the substantial region of low density between the two distributions forces the diffusion distance between them to be relatively large. This suggests that diffusion distance is more useful than Euclidean distance for comparing high density points for the determination of modes, under the assumption that multimodal regions have modes that are connected by regions of not-too-low density. The results of the proposed clustering algorithm, as well a low-dimensional representation of diffusion distances, appears in Figure 4. In the low-dimensional embedding corresponding to diffusion distance coordinates, the parabolic segment is linear and compressed, enabling the correct learning of modes. Labels are then assigned according to these modes in the diffusion coordinates, which can be projected back onto the original data to yield a clustering of the original data.

B. Diffusion Distance

We now present an overview of diffusion distances. Additional analysis and comments on implementation appear in [31], [32]. Diffusion processes on graphs lead to a data-dependent notion of distance, known as *diffusion distance*. This notion of distance has been applied to a variety of application problems, including analysis of stochastic and dynamical systems [31], [41], [42], [43], semisupervised learning [44], [45], data fusion [46], [47], latent variable separation [48], [49], and molecular dynamics [50], [51]. *Diffusion maps* provide a way of computing and visualizing diffusion distances, and may be understood as a type of nonlinear dimension reduction, in which data in a high number of dimensions may be embedded in a low-dimensional space by a nonlinear coordinate transformation. In this regard, diffusion maps are related to nonlinear dimension reduction techniques such as isomap [52], Laplacian eigenmaps [44], and local linear embedding [53], among several others.

Consider a discrete set $X = \{x_n\}_{n=1}^N \subset \mathbb{R}^D$. The diffusion distance [31], [32] between $x, y \in X$, denoted $d_t(x, y)$, is a notion of distance that incorporates and is uniquely determined by the underlying geometry of X. The distance depends on a time parameter t, which enjoys an interpretation in terms of diffusion on the data. The computation of d_t involves constructing a weighted, undirected graph $\mathcal{G}$ with vertices

corresponding to the N points in X, and weighted edges given by the $N \times N$ weight matrix

$$W(x,y) := \begin{cases} e^{-\frac{\|x-y\|_2^2}{\sigma^2}}, & x \in NN_k(y) \\ 0, & \text{else} \end{cases}, \quad (1)$$

for some suitable choice of σ and with $NN_k(x)$ the set of k-nearest neighbors of y in X with respect to Euclidean distance. A fast nearest neighbors algorithm yields W in quasilinear time in N for k small (see Section IV-A for details). The degree of x is $\deg(x) := \sum_{y \in X} W(x,y)$.

A Markov diffusion, representing a random walk on $\mathcal{G}$ (or X) has $N \times N$ transition matrix $P(x,y) = W(x,y)/\deg(x)$. For an initial distribution $\mu \in \mathbb{R}^N$ on X, the vector μP^t is the probability over states at time $t \geq 0$. As t increases, this diffusion process on X evolves according to the connections between the points encoded by P. This Markov chain has a stationary distribution π s.t. $\pi P = \pi$, given by $\pi(x) = \deg(x)/\sum_{y \in X} \deg(y)$. The *diffusion distance at time* t is

$$d_t^2(x,y) := \sum\nolimits_{u \in X} (P^t(x,u) - P^t(y,u))^2 d\mu(u)/\pi(u). \quad (2)$$

The computation of $d_t(x,y)$ involves summing over all paths of length t connecting x to y, so $d_t(x,y)$ is small if x, y are strongly connected in the graph according to P^t, and large if x, y are weakly connected in the graph.

The eigendecomposition of P allows to derive fast algorithms to compute d_t: the matrix P admits a spectral decomposition (under mild conditions, see [32]) with eigenvectors $\{\Phi_n\}_{n=1}^N$ and eigenvalues $\{\lambda_n\}_{n=1}^N$, where $1 = \lambda_1 \geq |\lambda_2| \geq \cdots \geq |\lambda_N|$. The diffusion distance (2) can then be written as

$$d_t^2(x,y) = \sum\nolimits_{n=1}^N \lambda_n^{2t} (\Phi_n(x) - \Phi_n(y))^2. \quad (3)$$

The weighted eigenvectors $\{\lambda_n^t \Phi_n\}_{n=1}^N$ are new data-dependent coordinates of X, which are in fact close to being geometrically intrinsic [31]. Euclidean distance in these new coordinates is diffusion distance on $\mathcal{G}$.

Diffusion distances are parametrized by t, which measures how long the diffusion process on $\mathcal{G}$ has run when the distances are computed. Small values of t allow a small amount of diffusion, which may prevent the interesting geometry of X from being discovered, but provide detailed, fine scale information. Large values of t allow the diffusion process to run for so long that the fine geometry may be washed out. In this work an intermediate regime is typically when the diffusion geometry of the data is most useful; in all our experiments we set $t = 30$. The choices of σ, k, t in the construction of W are in general important, see Section III-G.

Note that under the mild condition that the underlying graph $\mathcal{G}$ is connected, $|\lambda_n| < 1$ for $n > 1$. Hence, $|\lambda_n^{2t}| \ll 1$ for large t and $n > 1$, so that the sum (3) may approximated by its truncation at some suitable $2 \leq M \ll N$. In our experiments, M was set to be the value at which the decay of the eigenvalues $\{\lambda_n\}_{n=1}^N$ begins to decrease; this is a standard heuristic for diffusion maps. The subset $\{\lambda_n^t \Phi_n\}_{n=1}^M$ used in the computation of d_t is a dimension-reduced set of diffusion coordinates. The truncation also enables us to compute only a few eigenvectors, reducing computational complexity, see Section IV-A. In this sense, the mapping

$$x \mapsto (\lambda_1^t \Phi_1(x), \lambda_2^t \Phi_2(x), \ldots, \lambda_M^t \Phi_M(x)) \quad (4)$$

is a dimension reduction mapping of the ambient space $\mathbb{R}^D$ to $\mathbb{R}^M$.

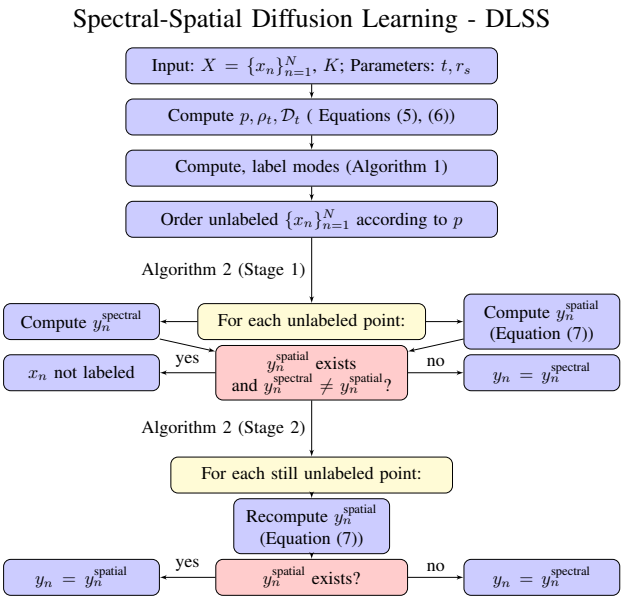


Fig. 5: *Diagram of proposed unsupervised clustering algorithm DLSS. First, modes are computed. Second, points are labeled in the two stage algorithm. Notice that in the first stage, a label may only be assigned based on spectral information, though spatial information may prevent a label from being assigned. In the second stage, a label may be assigned based on either spectral or spatial information.*

C. Unsupervised HSI Clustering Algorithm Description

We now discuss the proposed HSI clustering algorithm in detail; see Figure 5 for a flowchart representation. Let $X = \{x_n\}_{n=1}^N \subset \mathbb{R}^D$ be the HSI, and let K be the number of clusters. As described in Section I-B, our algorithm proceeds in two major steps: mode identification and labeling of points.

The algorithm for learning the modes of the classes is summarized in Algorithm 1. It first computes an empirical density for each point x_n with a kernel density estimator

$$p(x_n) = p_0(x_n) \Big/ \sum\nolimits_{m=1}^N p_0(x_m), \quad (5)$$

where $p_0(x_n) = \sum_{x_m \in NN_k(x_n)} e^{-\|x_n - x_m\|_2^2/\sigma_1^2}$. Here $\|x_n - x_m\|_2$ is the Euclidean distance in $\mathbb{R}^D$, and $NN_k(x_n)$ is the set of k-nearest neighbors to x_n, in Euclidean distance. The use of the Gaussian kernel density estimator is standard, enjoying strong theoretical guarantees [54], [55] but certainly other estimators may be used. In our experiments we set $k = 20$, though our method is robust to choosing larger k. The parameter σ_1 in the exponential kernel is set to be one twentieth the mean distance between all points (one could use the median instead in the presence of outliers). Once the empirical density p is computed, the modes of the HSI classes are computed in a manner similar in spirit to [22], but employing diffusion distances. We compute the time-dependent quantity $\tilde{\rho}_t$ that assigns, to each pixel, the minimum

diffusion distance between the pixel and a point of higher empirical density:

$$\tilde{\rho}_t(x_n) = \begin{cases} \min\limits_{\{p(x_m)\geq p(x_n)\}} d_t(x_n, x_m), & x_n \neq \arg\max_i p(x_i) \\ \max_{x_m} d_t(x_n, x_m), & x_n = \arg\max_i p(x_i) \end{cases},$$

where $d_t(x_m, x_n)$ is the diffusion distance between x_m, x_n, at time t. In the following we will use the normalized quantity $\rho_t(x_n) = \tilde{\rho}_t(x_n)/\max_{x_m} \tilde{\rho}_t(x_m)$, which has maximum value 1. The modes of the HSI are computed as the points $x_1^*, \ldots, x_K^*$ yielding the K largest values of the quantity

$$\mathcal{D}_t(x_n) = p(x_n)\rho_t(x_n)\,. \tag{6}$$

Such points should be both high density and far in diffusion distance from any other higher density points, and can therefore be expected to be modes of different cluster distributions. This method provably detects modes correctly under certain distributional assumptions on the data [33].

Algorithm 1: Geometric Mode Detection Algorithm

Input: X, K; t.
Compute the empirical density $p(x_n)$ for each $x_n \in X$.
Compute $\{\rho_t(x_n)\}_{n=1}^N$, the diffusion distance from each point to its nearest neighbor in diffusion distance of higher empirical density, normalized.
Set the learned modes $\{x_i^*\}_{i=1}^K$ to be the K maximizers of $\mathcal{D}_t(x_n) = p(x_n)\rho_t(x_n)$.
Output: $\{x_i^*\}_{i=1}^K, \{p(x_n)\}_{n=1}^N, \{\rho_t(x_n)\}_{n=1}^N$.

Once the modes are detected, each is given a unique label. All other points are labeled using these mode labels in the following two-stage process, summarized in Algorithm 2. In the first stage, running in order of decreasing empirical density, the *spatial consensus label* of each point is computed by finding all labeled points within distance $r_s \geq 0$ in the spatial domain of the pixel in question; call this set $NN_{r_s}^s(x_n)$. If one label among $NN_{r_s}^s$ occurs with relative frequency $> 1/2$, that label is the spatial consensus label. Otherwise, no spatial consensus label is given. In detail, let $L_n^{\text{spatial}} = \{y_m \mid x_m \in NN_{r_s}^s(x_n), x_m \neq x_n\}$ denote the labels of the spatial neighbors within radius r_s. Then the spatial consensus label of x_i is

$$y_i^{\text{spatial}} = \begin{cases} k, & \frac{|\{y_n | y_n = k,\ y_n \in L_n^{\text{spatial}}\}|}{|L_n^{\text{spatial}}|} > \frac{1}{2}, \\ 0 \text{ (no label)}, & \text{else.} \end{cases} \tag{7}$$

After a point's spatial consensus label is computed, its *spectral label* is computed as its nearest neighbor in the spectral domain, measured in diffusion distance, of higher density. The point is then given the overall label of the spectral label unless the spatial consensus label exists (i.e. is $\neq 0$ in (7)) and differs from the spatial consensus label. In this case, the point in question remains unlabeled in the first stage. Note that points that are unlabeled are considered to have label 0 for the purposes of computing the spatial consensus label, so in the case that most pixels in the spatial neighborhood are unlabeled, the spatial consensus label will be 0. Hence, only pixels with many labeled pixels in their spatial neighborhood can have a consensus spatial label. In this first stage, a label is only assigned based on spectral information, though the spatial information may prevent a label from being assigned.

Upon completion of the first stage, the dataset will be partially labeled; see Figure 2. In the second stage, an unlabeled point is given the label of its spatial consensus label, if it exists, or otherwise the label of its nearest spectral neighbor of higher density. Thus, in the second stage, a label is assigned based on joint spectral-spatial information.

Algorithm 2: Spectral-Spatial Labeling Algorithm

Input: $\{x_i^*\}_{i=1}^K, \{p(x_n)\}_{n=1}^N, \{\rho_t(x_n)\}_{n=1}^N$; r_s.
Assign each mode a unique label.
Stage 1: Iterating through the remaining unlabeled points in order of decreasing density among unlabeled points, assign each point the same label as its nearest spectral neighbor (in diffusion distance) of higher density, unless the spatial consensus label exists and differs, in which case the point is left unlabeled.
Stage 2: Iterating in order of decreasing density among unlabeled points, assign each point the consensus spatial label, if it exists, otherwise the same label as its nearest spectral neighbor of higher density.
Output: Labels $\{y_n\}_{n=1}^N$.

Points of high density are likely to be labeled according to their spectral properties. The reasons for this are twofold. First, these points are likely to be near the centers of distributions, and hence are likely to be in spatially homogeneous regions. Second, points of high density are labeled before points of low density, so it is unlikely for high density points to have many labeled points in their spatial neighborhoods. This means that the spatial consensus label is unlikely to exist for these points. Conversely, points of low density may be at the boundaries of the classes, and are hence more likely to be labeled by their spatial properties. The incorporation of spatial information into machine learning for HSI is justified by the fact that HSI images typically show some amount of spatial regularity, in that if a pixel's nearest spatial neighbors all have the same class label, it is likely that the pixel has this same label, compared to the case in which the pixel's nearest spatial neighbors have random labels [7], [28], [23], [34], [35], [36], [37], [38], [39], [40]. The spatial information regularizes and improves performance, but it cannot take the place of the spectral information, as shall be seen in Section III-G2: the spectral information is more discriminative than the spatial information, and is the more important of the two.

The proposed method, combining Algorithms 1, 2 is called *spectral-spatial diffusion learning (DLSS)*. In our experimental analysis, the significance of the spectral-spatial labeling scheme is validated by comparing DLSS against a simpler method, called *diffusion learning (DL)*. This method learns class modes as in Algorithm 1, but labels all pixels simply by requiring each point have the same label as its nearest spectral neighbor of higher density. The expectation is that DLSS will generally outperform DL, due to the former's incorporation of spatial data; this is confirmed by our experiments.

D. Active Learning DLSS Variation

Both the DL and DLSS methods are unsupervised. We now present a variation of the DLSS method for active learning of hyperspectral images, where a few well-chosen pixels are automatically selected for labeling. The DLSS method labels points beginning with the learned class modes, and mistakes tend to be made on points that are near the class boundaries; in the active learning scheme the algorithm will ask for the labels of the points whose distances from their nearest two modes are closest. That is, points whose nearest mode is ambiguous will be labeled using training data, and all other points will be labeled as in the DLSS algorithm.

More precisely, we fix a time t, and for each pixel x_n, let $x^*_{n_1}, x^*_{n_2}$ be the two modes closest to x_n in diffusion distance d_t. We compute the quantity

$$F_t(x_n) = |d_t(x_n, x^*_{n_1}) - d_t(x_n, x^*_{n_2})|. \tag{8}$$

If $F_t(x_n)$ is close to 0, then there is substantial ambiguity as to the nearest mode to x_n. Suppose the user is afforded the labels of exactly L points. Then the L labels requested in our active learning regime are the L minimizers of F_t. The proposed active learning scheme is summarized in Algorithm 3. To evaluate performance, we consider a range of L values in our experiments. The active learning setting is most interesting when $\alpha = L/N$ is very small, where N is the total number of pixels in the image.

Algorithm 3: Active Learning with DLSS

Input: X, K; t, r_s, L.
Compute the modes of the data using Algorithm 1.
Give each mode a unique label.
Compute, for each point x_n, $F_t(x_n)$ as in (8).
Label the L minimizers of F_t with ground truth labels.
Label the remaining, unlabeled points as in steps 3, 4 in Algorithm 2.
Output: Labels $\{y_n\}_{n=1}^N$.

Note that the active learning algorithm can be iterated, by labeling points then recomputing the quantity (8) to determine the most challenging points after some labels have been introduced [56].

III. Experiments

A. Algorithm Evaluation Methods and Experimental Data

We consider several HSI datasets to evaluate the proposed unsupervised (Algorithms 1, 2) and active learning (Algorithm 3) algorithms. For evaluation in the presence of ground truth (GT), we consider three quantitative measures, besides visual performance, namely:

1) *Overall Accuracy (OA)*: Total number of correctly labeled pixels divided by the total number of pixels. This method values large classes more than small classes.
2) *Average Accuracy (AA)*: The average, over classes, of the OA of each class. This method values small classes and large classes equally.
3) *Cohen's κ-statistic* (κ): A measurement of agreement between two labelings, corrected for random agreement [57]. Letting a_o be the observed agreement between the labeling and the ground truth and a_e the expected agreement between a uniformly random labeling and the ground truth, $\kappa = (a_o - a_e)/(1 - a_e)$. $\kappa = 1$ corresponds to perfect overall accuracy, $\kappa \leq 0$ corresponds to labels no better than what is expected from random guessing.

In order to perform quantitative analysis with these metrics and make consistent visual comparisons, the learned clusters are aligned with ground truth, when available. More precisely, let S_K be the set of permutations of $\{1, 2, \ldots, K\}$. Let $\{C_i\}_{i=1}^K$ be the clusters learned from one of the clustering methods, and let $\{C_i^{GT}\}_{i=1}^K$ be the ground truth clusters. Cluster C_i is assigned label $\hat{\eta}_i \in \{1, 2, \ldots, K\}$, with $\hat{\eta} = \arg\max_{\eta=(\eta_1,\ldots,\eta_K)\in S_K} \sum_{i=1}^K |C_{\eta_i} \cap C_i^{GT}|$. We remark that while this alignment method maximizes the overall accuracy of the labeling and is most useful for visualization, better alignments for maximizing AA and κ may exist.

We consider 4 real HSI datasets to shed light on strengths and weaknesses of the proposed algorithm. These datasets are standard, have ground truth, and are publicly available[1]. Experiments with active learning are performed for these same real HSI datasets with Algorithm 3. Additional experiments on synthetic and real HSI data are available, for conciseness, only in an appendix in the online preprint version.

Note that some images are restricted to subsets in the spatial domain, which is noted in their respective subsections. This is because unsupervised methods for HSI struggle with data containing a large number of classes, due to variation within classes and similarity between certain end-members of different classes [16]. Hence, the Indian Pines, Pavia, and Kennedy Space Center datasets are restricted to reduce the number of classes and achieve meaningful clusters. The Salinas A dataset is considered in its entirety. The ground truth, when available, is often incomplete, i.e. not all pixels are labeled. For these datasets, labels are computed for all data, then the pixels with ground truth labels are used for quantitative and visual analysis. The number of class labels in the ground truth images were used as parameter K for all clustering algorithms, though the proposed method automatically estimates the number of clusters; see Section V. Grayscale images of the projection of the data onto its first principal component and images of ground truth (GT), colored by class, for the Indian Pines, Pavia, Salinas A, and Kennedy Space Center datasets are in Figures 6, 8, 9, and 11, respectively. The projection onto the first principal component of the data is presented as a simple visual summary of the data, though it washes out the subtle information presented in individual bands.

Since the proposed and comparison methods are unsupervised, experiments are performed on the entire dataset, including points without ground truth labels. The labels for pixels without ground truth are not accounted for in the quantitative evaluation of the algorithms tested. Note that additional experiments, not shown, were performed, using only the data with ground truth labels. These experiments consisted

[1] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

in restricting the HSI to the pixels with labels, which makes the clustering problem significantly easier. Quantitative results were uniformly better for all datasets and methods in these cases; the relative performances of the algorithms on a given dataset remained the same.

B. Comparison Methods

We consider a variety of benchmark and state-of-the-art methods of HSI clustering for comparison. First, we consider the classic K-*means* algorithm [55] applied directly to X. This method is not expected to perform well on HSI data, due to the non-spherical shape of clusters, high dimensionality, and noise, all well-known problems for K-means. Several *dimension reduction* methods to reduce the dimensionality of the data, while preserving important discriminatory properties of the classes, as well as increasing the signal-to-noise ratio in the projected space, are also used as benchmarks for comparison with the proposed method. These methods first reduce the dimension of the data from D to $K_{GT} \ll D$, where K_{GT} is the number of classes, then run K-means on the reduced data. We consider linear dimension reduction via *principal component analysis (PCA)*; *independent component analysis (ICA)* [58], [59], using the fast implementation [60][2]; and *random projections* via Gaussian random matrices, shown to be efficient in highly simplified data models [61], [62].

We also consider more computationally intensive methods for benchmarking. *DBSCAN* [63] is a popular density-based clustering method, that although highly parameter-dependent, has proved useful for a variety of unsupervised tasks. *Spectral clustering* (SC) [64], [65] has been applied with success in classification and clustering HSI [36]. The spectral embedding consists of the top K_{GT} row-normalized eigenvectors of the normalized graph Laplacian L; in this features space K-means is then run (see Section III-G). We also cluster with *Gaussian mixture models* (GMM) [14], [66], [67], with parameters determined by expectation maximization (EM).

Finally we consider several recent, state-of-the-art clustering methods: *sparse manifold clustering and embedding (SMCE)* [18], [19][3], which fits the data to low-dimensional, sparse structures, and then applies spectral clustering; *hierarchical clustering with non-negative matrix factorization (HNMF)* [20][4], which has shown excellent performance for HSI clustering when the clusters are generated from a single endmember; a graph-based method based on the Mumford-Shah segmentation [68][21], related to spectral clustering, and called *fast Mumford-Shah (FMS)* in this article (we use a highly parallelized version[5]); *fast search and find of density peaks clustering* (FSFDPC) algorithm [22], which has been shown effective in clustering a variety of data sets.

[2]https://www.cs.helsinki.fi/u/ahyvarin/papers/fastica.shtml
[3]http://vision.jhu.edu/code/
[4]https://sites.google.com/site/nicolasgillis/code
[5]http://www.ipol.im/pub/art/2017/204/?utm_source=doi

C. Relationship Between Proposed Method and Comparison Methods

The FSFDPC method has similarities with the mode estimation aspect of our work, in that both algorithms attempt to learn the modes of the classes via a density-based analysis, as described in, for example, [17], [22]. Our method is quite different, however: the proposed measure of distance between high density points is not Euclidean distance, but diffusion distance [31], [32], which is more adept at removing spurious modes, due to its incorporation of the geometry of the data. This phenomenon is illustrated in Figures 1,3. The assignment of labels from the modes is also quite different, as diffusion distances are used to determine spectral nearest neighbors, and spatial information is accounted for in our DLSS algorithm. FSFDPC, in contrast, assigns to each of the modes its own class label, and to the remaining points a label by requiring that each point has the same label as its Euclidean nearest neighbor of higher density. This means that FSFDPC only incorporates spectral information measured in Euclidean distance, disregarding spatial information. The benefits of both using diffusion distances to learn modes, and incorporating spatial proximities into the clustering process are very significant, as the experiments demonstrate.

Both FSFDPC and the proposed algorithm have some similarities to DBSCAN which, however, performs poorly for data with clusters of differing densities, and is highly sensitive to its parameters. Note that FSFDPC was in fact proposed to improve on these drawbacks of DBSCAN [22].

The proposed DLSS and DL algorithms also share commonalities with spectral clustering, SMCE, and FMS in that these comparison methods compute eigenvectors of a graph Laplacian in order to develop a nonlinear notion of distance. This is related to computing the eigenvectors of the Markov transition matrix in the computation of diffusion maps. The proposed method, however, directly incorporates density into the detection of modes, which allows for more robust clustering compared to these methods, which work by simply applying K-means to the eigenvectors of the graph Laplacian. Moreover, our technique does not rely on any assumption about sparsity (unlike SMCE), and is completely invariant under distance-preserving transformations (it shares this property with SMCE), which could be useful if different imaging modalities (e.g. compressed modalities) were used.

Additionally, our approach is connected to semisupervised learning techniques on graphs, where initial given labels are propagated by a diffusion process to other vertices (points); see [45] and references therein. Here of course we have proceeded in an unsupervised fashion, replacing initial given labels by estimated modes of the clusters.

D. Summary of Proposed and Comparison Methods

The experimental methods are summarized in Table I. The two novel methods we proposed are the full spectral-spatial diffusion learning method (DLSS), as well as a simplified diffusion learning method (DL). We note that several algorithms were not implemented by the authors of this article: publicly

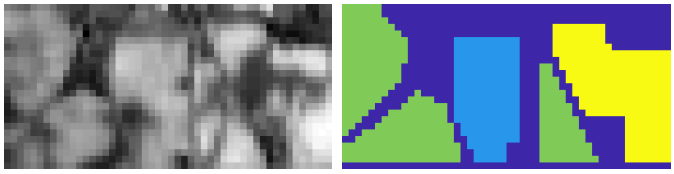


Fig. 6: *The Indian Pines data is a* 50×25 *subset of the full Indian Pines dataset. It contains 3 classes, one of which is not well-localized spatially. The dataset was captured in 1992 in Northwest IN, USA by the AVRIS sensor. The spatial resolution is 20m/pixel. There are* 200 *spectral bands. Left: projection onto the first principal component of the data; right: ground truth (GT).*

available libraries were used when available. Links to these libraries are noted where appropriate.

Method	D.R.	Metric
K-means on full dataset	No	Euclidean
K-means on PCA reduced dataset	Yes	Euclidean
K-means on ICA reduced dataset	Yes	Euclidean
K-means on data reduced by random projections	Yes	Euclidean
DBSCAN [63]	No	Euclidean
Spectral clustering [65]	Yes	Spectral
Gaussian mixture models	No	Euclidean
Sparse manifold clustering and embedding [18], [19]	Yes	Spectral
Hierarchical NMF [20]	No	Euclidean
Fast Mumford Shah [21]	No	Spectral
FSFDPC [22]	No	Euclidean
Diffusion learning (DL)	Yes	Diffusion
Spectral-spatial diffusion learning (DLSS)	Yes	Diffusion

TABLE I: *Methods used for experimental analysis, along with whether the method employs dimensionality reduction and which metric is used to compared points. The methods proposed in this article appear in bold. Note that the proposed methods employ dimension reduction, as illustrated in (4).*

All experiments and subsequent analyses, except those involving FMS, were performed in MATLAB running on a 3.1 GHz Intel 4-Core i7 processor with 16 GB of RAM; code to reproduce all results is available on the authors' website[6].

E. Unsupervised HSI Clustering Experiments

1) Indian Pine Dataset: The Indian Pines dataset used for experiments is a subset of the full Indian Pines datasets, consisting of three classes that are difficult to distinguish visually; see Figure 6. This dataset is expected to be challenging due to the lack of clear separation between the classes. Results for Indian Pines appear in Figure 7 and Table II.

The proposed methods, DL and DLSS, perform the best, with DLSS strongly outperforming the rest. For the average accuracy statistic, DBSCAN performs as well as DL, indicating that the clusters for this data are likely of comparable empirical density. The use of diffusion distances for mode detection and determination of spectral neighbors is evidently useful, as DL significantly outperforms FSFDPC, which has among the best quantitative performance of the comparison methods. Moreover, the use of the proposed spectral-spatial labeling scheme DLSS clearly improves over spectral-only labeling DL: as seen in Figure 7, DLSS correctly labels many small interior regions that DL labels incorrectly.

2) Pavia Dataset: The Pavia dataset used for experiments consists of a subset of the original dataset, and contains six classes, with one of them spread out across the image. As can be seen in Figure 8, the yellow class is small and diffuse, which is expected to add challenge to this example. Results

[6]http://www.math.jhu.edu/~jmurphy/

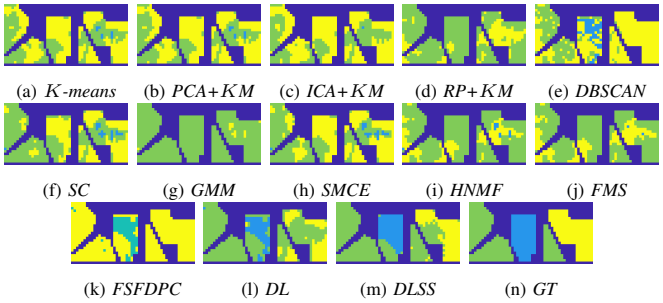


Fig. 7: *Clustering results for Indian Pines dataset. The impact of the spectral-spatial labeling scheme is apparent, as the labels for the DLSS method are more spatially regular than those of the DL method. Note that the regions of difference between DL and DLSS are primarily near boundaries of classes and in very small interior regions. Near the boundaries of classes, pixels are likely to be far from the spectral class cores, and hence are more likely to be labeled based on spatial properties. The small interior regions are unlikely to be formed under the DLSS labeling regime, since these regions consist of points whose spectral label differs from their spatial consensus label. The simplified DL method performs second best, and in particular outperforms FSFDPC, which performs well among the comparison methods.*

Fig. 8: *The Pavia data is a* 270×50 *subset of the full Pavia dataset. It contains 6 classes, some of which are not well-localized spatially. The dataset was captured by the ROSIS sensor during a flight over Pavia, Italy. The spatial resolution is 1.3 m/pixel. There are 102 spectral bands. Left: projection onto the first principal component of the data; right: ground truth (GT).*

appear in Table II. Visual results appear in the online preprint version of this article.

The proposed methods give the best results, which also provide evidence of the value of both the diffusion learning stage and the spectral-spatial labeling scheme. The proposed DLSS algorithm makes essentially only two errors: the yellow-green class is slightly mislabeled, and the blue-green class in the bottom right is labeled completely incorrectly. However, both of these errors are made by all algorithms, often to a greater degree. Among the comparison methods, SMCE performs best; classical spectral clustering also performs well.

3) Salinas A Dataset: The Salinas A dataset (see Figure 9) consists of 6 classes arrayed diagonally. Some pixels in the original images have the same values, so some small Gaussian noise (variance $< 10^{-3}$) was added as a preprocessing step to distinguish these pixels.

Clustering results for Salinas A appear in Figure 10. For this dataset, the proposed DLSS method performs best, with

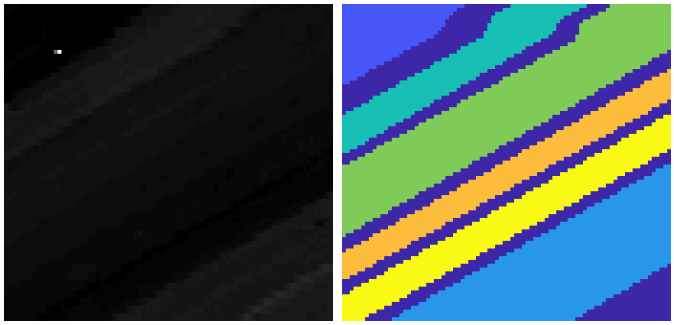


Fig. 9: *The Salinas A data consists of the full* 86×83 *HSI. It contains 6 classes, all of which are well-localized spatially. The dataset was captured over Salinas Valley, CA, by the AVRIS sensor. The spatial resolution is 3.7 m/pixel. The image contains 224 spectral bands. Left: projection onto the first principal component of the data; right: ground truth (GT).*

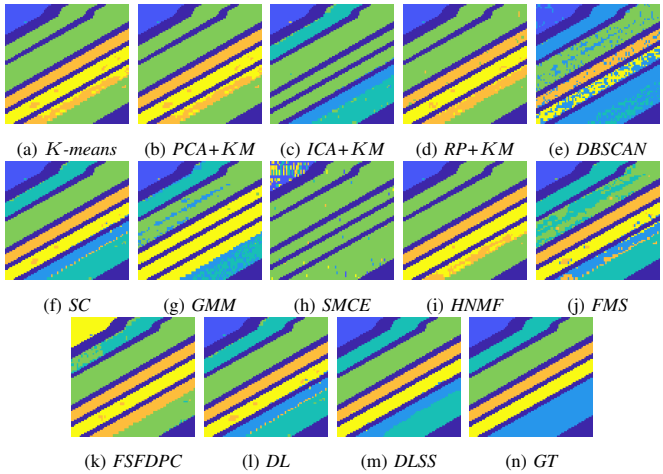


Fig. 10: *Clustering results for Salinas A dataset. The proposed method, DLSS performs best, with the simplified DL method and benchmark spectral clustering also performing well. Notice that the spectral-spatial labeling scheme removes some of the mistakes in the yellow cluster, and also improves the labeling near some class boundaries. However, it is not able to fix the mislabeling of the light blue cluster in the lower right. Indeed, all methods split the cluster in the lower right of the image, indicating the challenging aspects of this dataset for unsupervised learning.*

the only error made in splitting the bottom right cluster into two pieces, an error made by all algorithms. The simpler DL method also performs well, as does the benchmark spectral clustering algorithm. Comparing the labeling for DL and DLSS, the small regions of mislabeled pixels in DL are correctly labeled in DLSS, because these pixels are likely of low empirical density, and hence benefit from being labeled based on both spectral and spatial similarity, not spectral similarity alone. However, some pixels correctly classified by DL were labeled incorrectly by DLSS, indicating that the spatial proximity condition enforced in DLSS may not lead to improved results for every pixel. Details on this, and how to tune the size of the neighborhood with which spatial consensus labels are computed, are given in Section III-G2.

4) Kennedy Space Center Data Set: The Kennedy Space Center dataset used for experiments consists of a subset of the original dataset, and contains four classes. Figure 11 illustrates the first principal component of the data, as well as the labeled ground truth, which consists of the examples of four vegetation types which dominate the scene. Results appear in Table II. The proposed methods yield the best results, noting that the FMS method also performs well. Most linear methods, such as K-means with linear dimension reduction or NMF, perform poorly, suggesting that nonlinear methods are needed for this data. Spectral clustering performs much better than the linear methods. We note that spatial information for this dataset is

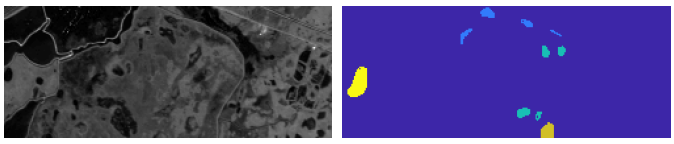


Fig. 11: *The Kennedy Space Center data is a* 250×100 *subset of the full Kennedy Space Center dataset. It contains 4 classes, some of which are not well-localized spatially. The scene was captured with the NASA AVIRIS instrument over the Kennedy Space Center (KSC), Florida, USA. The spatial resolution is 18 m. After removing low signal-to-noise-ratio and water-absorption bands, the dataset consists of 176 bands. Left: projection onto the first principal component of the data; right: ground truth (GT).*

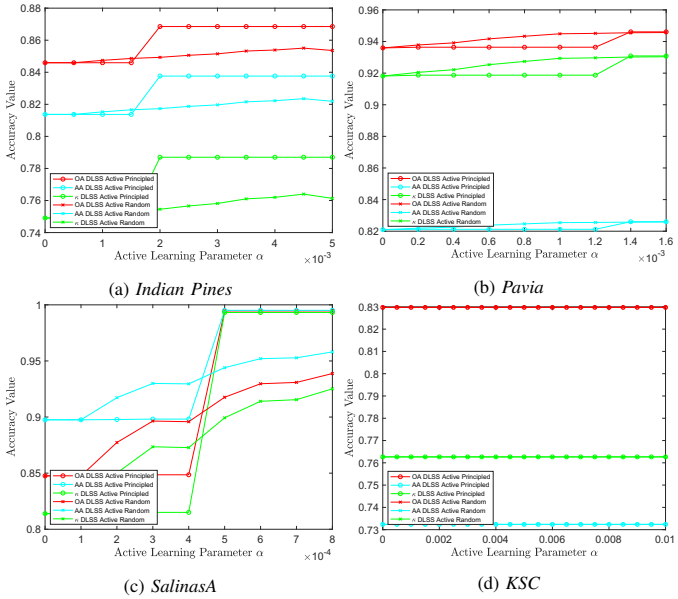


Fig. 12: *Active learning parameter analysis. The x-axis denotes the parameter α. As α increases, more labeled pixels are introduced. All measures of accuracy are monotonic increasing in α, and a small increase can lead to a huge jump in accuracy, as seen in the Indian Pines and Salinas A datasets. We see that randomly selecting points has a more incremental impact on improving accuracy than the principled approach, and may require a very large number of labels to achieve the performance achieved by active learning with a small number of labels. Many iterations of randomly selected points were used and averaged to produce the plots.*

less helpful than for the Indian Pines and Pavia datasets.

5) Overall Comments on Clustering: Quantitative results for the clustering experiments appear in Table II. We see that the DLSS method performs best among all metrics for all datasets. The DL method generally performs second best, though DBSCAN, spectral clustering, and SMCE occasionally perform comparably to DL. It is notable that DL outperforms FSFDPC, which uses a similar labeling scheme, but computes modes with Euclidean distances, rather than diffusion distances. This provides empirical evidence for the need to use nonlinear methods of measuring distances for HSI.

F. Active Learning

To evaluate our proposed active learning method, Algorithm 3, the same 4 labeled HSI datasets were clustered with increasing the percentage α of labeled points, chosen as in Algorithm 3. Note that $\alpha = 0$ corresponds the the unsupervised DLSS algorithm. The empirical results for this active learning scheme appear in Figure 12. We also consider selecting the labeled points uniformly at random; we hypothesize our principled approach will be superior to random sampling. The plots indicate that the proposed active learning can produce dramatic improvements in labeling with very few training labels. Indeed, an improvement in overall accuracy from 85% to 87% for Indian Pines can be achieved with only 3 labels. Even more dramatic is the Salinas A dataset, in which 3 labeled points improves the overall accuracy from 84% to 99.5%. The Pavia dataset enjoys some improved performance, though the random labels do about as well as the principled labels, and Kennedy Space Center dataset labelings are not affected by the small collection of labeled points. In the

Method	OA I.P.	AA I.P.	κ I.P.	OA P.	AA P.	κ P.	OA S.A.	AA S.A.	κ S.A.	OA K.S.C.	AA K.S.C.	κ K.S.C.
K-means	0.43	0.38	0.09	0.78	0.62	0.72	0.63	0.66	0.52	0.36	0.25	0.01
PCA+K-means	0.43	0.38	0.10	0.78	0.62	0.72	0.63	0.66	0.52	0.36	0.25	0.01
ICA+K-means	0.41	0.36	0.06	0.67	0.55	0.58	0.57	0.56	0.44	0.36	0.25	0.01
RP+K-means	0.51	0.51	0.26	0.76	0.61	0.70	0.63	0.66	0.53	0.60	0.50	0.43
DBSCAN	0.63	0.62	0.43	0.73	0.72	0.66	0.71	0.71	0.63	0.36	0.25	0.01
SC	0.54	0.45	0.24	0.82	0.76	0.77	0.83	0.88	0.80	0.62	0.52	0.44
GMM	0.44	0.35	0.02	0.64	0.59	0.55	0.64	0.61	0.55	0.42	0.31	0.10
SMCE	0.52	0.45	0.22	0.83	0.77	0.79	0.47	0.42	0.30	0.36	0.26	0.01
HNMF	0.41	0.32	-0.02	0.72	0.74	0.66	0.63	0.66	0.53	0.36	0.25	0.00
FMS	0.57	0.50	0.27	0.77	0.64	0.69	0.70	0.81	0.65	0.74	0.70	0.65
FSFDPC	0.58	0.51	0.26	0.78	0.75	0.73	0.63	0.61	0.54	0.36	0.25	0.00
DL	0.67	0.62	0.44	0.85	0.78	0.81	0.83	0.88	0.79	0.81	0.72	0.74
DLSS	**0.85**	**0.82**	**0.75**	**0.94**	**0.83**	**0.93**	**0.85**	**0.90**	**0.81**	**0.83**	**0.73**	**0.76**

TABLE II: *Summary of quantitative analyses of real HSI clustering; best results are in bold, second best are underlined. The datasets have been abbreviated as I.P. (Indian Pines), P. (Pavia), S.A. (Salinas A), and K.S.C. (Kennedy Space Center). Generally the proposed diffusion methods offer the strongest overall performance, particular DLSS. In all cases, DL outperforms FSFDPC, indicating the importance of using diffusion distances over Euclidean distances for HSI clustering.*

case of Pavia, however, the overall accuracy was already very large, so active learning seems not needed for this data set. Note that our principled scheme is generally superior to using randomly selected labeled points, which leads to a more gradual improvement in accuracy, compared to the huge gains that can be seen with the proposed principled method.

It is interesting to compare our active learning results to a state-of-the-art *supervised* method. We consider the supervised HSI classification with edge preserving filtering method (EPF) [69] algorithm, which combines a support vector machine with an analysis of spectral-spatial probability maps to label points. Using a publicly available implementation [7], we ran this algorithm using 1% and 5% of points as training data, generated as a uniformly random sample over all labeled points. 10 experiments were ran on each of the four datasets considered, with results averaged. Quantitative results are shown in Table III. The supervised results are generally superior to the results achieved by the unsupervised DL and DLSS method. The proposed active learning, however, is able to achieve the same performance on the Salinas A dataset using two orders of magnitude fewer points. This is because the proposed active learning method only uses training points for pixels that are considered especially important, whereas the EPF algorithm trains on a random subset of points. Moreover, when only 1% of training points are used, our active learning DLSS method with $.2\%$ of training points used outperforms the EPF method on the Indian Pines, Salinas A, and Kennedy Space Center datasets. This indicates the promise of the proposed active learning method, as it is able to outperform a state-of-the-art supervised method in the regime in which a low proportion of training points is available.

G. Parameter Analysis

We now discuss the parameters used in all methods, starting with those used for all comparison methods, and then discussing the two key parameters for the proposed method: diffusion time t and radius size r_s for the computation of the spatial consensus label in Algorithm 2. For experimental parameters except these, a range of parameters were considered, and those with best empirical results were used.

All instances of the K-means algorithm are run with 100 iterations, with 10 random initializations each time, and number of clusters K equal to the known number of classes in the ground truth. Each of the linear dimension reduction techniques, PCA, ICA, and random projection, embeds the data into $\mathbb{R}^K$, where K is the number of clusters. DBSCAN is highly dependent on several parameters, and a grid search was used on each dataset to select optimal parameters. Note that this means DBSCAN was optimized specifically for each dataset, while other methods used a fixed set of parameters across all experiments. Spectral clustering is run by computing a weight matrix as in (1), with $k = 100$ and $\sigma = 1$. The top K eigenvectors are then normalized to have Euclidean norm 1, then used as features with K-means.

Among the state-of-the-art methods, HNMF uses the recommended settings listed in the available online toolbox[8]. For FSFDPC, the empirical density estimate is computed as described in Section II-C, with a Gaussian kernel and 20 nearest neighbors. For SMCE, the sparsity parameter was set to be 10, as suggested in the online toolbox[9]. The FMS algorithm depends on several key parameters; grid search was implemented, and empirically optimal parameters with respect to a given dataset were used. Note that this means FMS was, like DBSCAN, optimized specifically for each dataset, while other methods used a fixed set of parameters across all experiments.

For the proposed algorithm, the same parameters for the density estimator as described above are used, in order to make a fair comparison with FSFDPC. Moreover, in the construction of the graph used to compute diffusion distances, we use the same construction as in spectral clustering and SMCE, again to make fair comparisons. The remaining parameters, diffusion time and spatial radius, were set to 30 and 3, respectively, for all experiments. We justify these choices and analyze their robustness in the following subsections.

1) Diffusion Time t: The most important parameter when using diffusion distances $d_t(x, y)$ is the time parameter $t \geq 0$, see eqn. (3). The larger t is, the smaller the contribution of the smaller eigenvalues in the spectral computation of d_t.

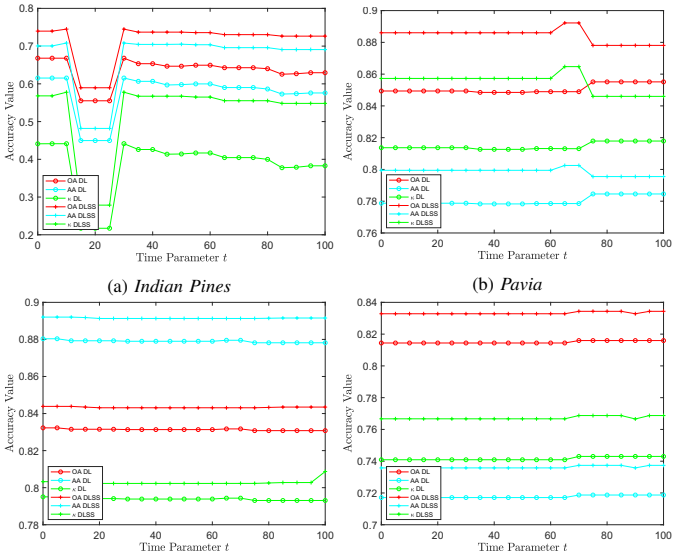
[7] http://xudongkang.weebly.com/

[8] https://sites.google.com/site/nicolasgillis/code

[9] http://www.vision.jhu.edu/code/

Method	OA I.P.	AA I.P.	κ I.P.	OA P.	AA P.	κ P.	OA S.A.	AA S.A.	κ S.A.	OA K.S.C.	AA K.S.C.	κ K.S.C.
DLSS (unsupervised)	0.85	0.82	0.75	0.95	0.83	0.93	0.85	0.90	0.81	0.83	0.73	0.76
Active learning, .2% training	0.87	0.84	0.79	0.95	0.83	0.93	1.00	1.00	1.00	0.83	0.73	0.76
EPF, 1% training	0.49	0.33	0.16	0.99	0.99	0.99	0.97	0.97	0.96	0.51	0.37	0.31
EPF, 5% training	0.82	0.86	0.72	0.99	0.99	0.99	1.00	1.00	1.00	0.98	0.98	0.98

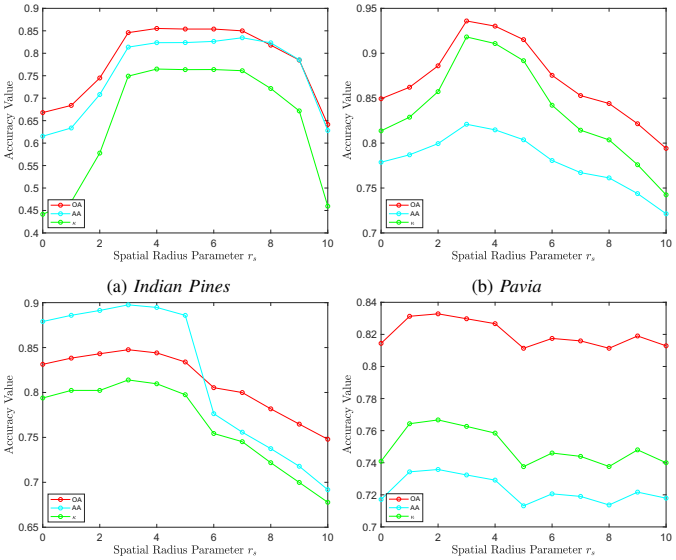
TABLE III: *We compare the state-of-the-art supervised classification algorithm, EPF, with the unsupervised DLSS algorithm and DLSS active learning variation. We see that the active learning method with only .2% of pixels used for training outperforms EPF with 1% training labels on the Indian Pines, Salinas A, and Kennedy Space Center datasets. Moreover, the active learning method with .2% labels performs comparably to or better than EPF with 5% training labels on the Indian Pines and Salinas A datasets.*



(a) *Indian Pines* (b) *Pavia*

(c) *SalinasA* (d) *KSC*

Fig. 13: *Time parameter analysis for the four real datasets. In general, the time parameter has little impact on performance of the proposed algorithm. As suggested by these plots, $t = 30$ is used for all ex periments.*



(a) *Indian Pines* (b) *Pavia*

(c) *Salinas A* (d) *KSC*

Fig. 14: *Space parameter analysis for DLSS. For each curve, increasing the radius of the neighborhood in which the spatial consensus label is computed improves quantitative performance up until a certain point, after which performance decays. The point at which the decay sets in differs for each example. These plots suggest a spectral-spatial tradeoff: spectral and spatial information must be balanced to achieve empirically optimal clustering.*

Allowing t to vary, connections in the dataset are explored by allowing the diffusion process to evolve. For small values of t, all points appear far apart because the diffusion process has not reached far, while for large values of t, all points appear close together because the diffusion process has run for so long that it has dissipated over the entire state space. In general, the interesting choice of t is moderate, which allows for the data geometry to be discovered, but not washed out in long-term.

In Figure 13, all the accuracy measures for t in $[0, 100]$ are displayed. The behavior is robust with respect to time. For Indian Pines, performance is largely constant, except for a dip from time $t = 15$ to $t = 25$. For the Pavia, Salinas A, and Kennedy Space Center examples, the performance is invariant with respect to the diffusion time. We conclude that a large range of $25 \leq t \leq 65$ or $t \geq 75$ would have led to the same empirical results as our choice $t = 30$.

2) Spatial Diffusion Radius: The spatial consensus radius r_s can also impact the performance of the proposed DLSS algorithm. Recall that this is the distance in the spatial domain used to compute the spatial consensus label (see Section II-C and definition (7)). If r_s is too small, insufficient spatial information is incorporated; if r_s is too large, the spectral information becomes drowned out. All measures of accuracy for each dataset for r_s in $[0, 10]$ appear in Figure 14. We see a trade-off between spectral and spatial information, suggesting that r_s should take a moderate value sufficiently greater than 0 but less than 10. This trade-off can be interpreted in the following way: empirically optimal results are achieved when both spectral and spatial information contribute harmoniously, and results deteriorate when one or the other dominates. We choose $r_s = 3$ for all experiments, though other choices would give comparable (or sometimes better) quantitative results for the datasets considered.

We note that the role of the spatial radius is analogous to the role of a regularization parameter for a general statistical learning problem. Taken too small, the problem is insufficiently regularized, leading to poor results. Taken too large, the regularization term dominates the fidelity term, leading also to poor results. In particular, the geometric regularity of the clusters in the spatial domain determine how large r_s may be taken while still preserving the spectral information. If the clusters are convex and not too elongated, then taking r_s large is reasonable. On the other hand, if the classes are very irregular spatially, for example highly non-convex or elongated, choosing r_s too large will wash out the spectral information which is generally more discriminative than the spatial information, resulting in inaccurate clustering.

H. Large Scale Experiments

The results of Section III-E analyzed subsets of larger images, in order to reduce the number of classes to allow for effective unsupervised learning [16]. In order to evaluate

the robustness of these results, we performed experiments in which the full HSI scenes were subdivided into small patches with fewer classes, then each patch—with a smaller number of classes than the total scene—were clustered. The results on individual patches may be used as the basis for a statistical evaluation of the performance of each clustering method. For the Indian Pines, Pavia, and Kennedy Space Center datasets, experiments for the entire dataset, suitably partitioned into smaller patches, were performed, with DLSS again performing best among all studied methods. Note that Salinas A had only 6 classes, and was considered in its entirety. The Indian Pines data set was partitioned into 24 rectangular patches of equal size; Pavia into 50 rectangular patches of equal size, and Kennedy Space Center into 25 patches of equal size. On each piece that contained some non-trivial ground truth, all clustering algorithms were ran. A series of statistical tests on the differences in performance were then executed as follows. For a pair of methods—denoted method i and j— let OA_k^i, OA_k^j be the overall accuracy of methods i and j on patch k, and let $\Delta_k^{i,j} = OA_k^i - OA_k^j$. The sample mean difference in error between methods i and j across the different patches is $\overline{\Delta^{i,j}} = \sum_{k=1}^{N_{\text{patches}}} \Delta_k^{i,j} / N_{\text{patches}}$, where N_{patches} is the total number of patches with ground truth. It is of interest to investigate whether $\overline{\Delta_{i,j}}$ can be inferred to be different from 0. In order to perform a statistical test, the sample standard deviation of difference between methods i, j is computed as $\sigma^{i,j} = \sqrt{\sum_k (\Delta_k^{i,j} - \overline{\Delta^{i,j}})^2 / (N_{\text{patches}} - 1)}$. Then, the null hypothesis that $\overline{\Delta_{i,j}} = 0$ may be tested against the alternative hypothesis that $\overline{\Delta_{i,j}} \neq 0$ by performing a two-sided t-test [70] with $N_{\text{patches}} - 1 = 72$ degrees of freedom. The normalized t-scores for the j corresponding to the DLSS method and i running through all other methods are reported in Table IV. The test confirms that for all methods i, the hypothesis that DLSS $(j = 13)$ does not significantly differ from method i $(\overline{\Delta_{i,13}} = 0)$ is rejected in favor of the alternative hypothesis that DLSS significantly differs from method i $(\overline{\Delta_{i,13}} \neq 0)$ at the 95% level. This provides evidence that DLSS performs competetively with benchmark and state-of-the-art HSI clustering algorithms across HSI with a variety of land cover types and complexity. Note that the values are $\Delta_{i,j}^k$ are not independent for different k, due to correlations across images. However, the t-test still provides a powerful method for inferring statistical significance in this case, despite this theoretical assumption not being satisfied.

In addition to providing the basis for a statistical evaluation of the proposed algorithm, splitting large, complicated HSI into patches for clustering allows to over-segment the image. Examples of the oversegmented maps, where we do not attempt to synchronize the labels across patches, appear in Figures 15, 16, 17. It is a topic of future research to combine these patches using the DLSS framework.

IV. Overall Comments on the Experiments and Conclusion

We proposed a novel unsupervised method for clustering HSI, using data-dependent diffusion distances to learn modes of clusters, followed by a spectral-spatial labeling scheme

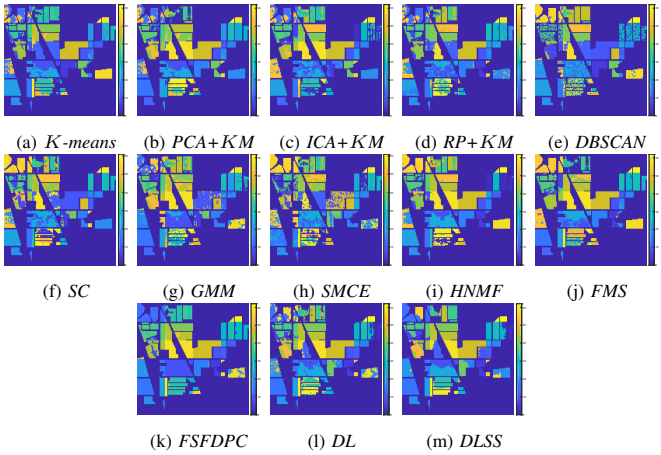


Fig. 15: *Results of clustering individual patches of the Indian Pines data, without synchronizing the labels.*

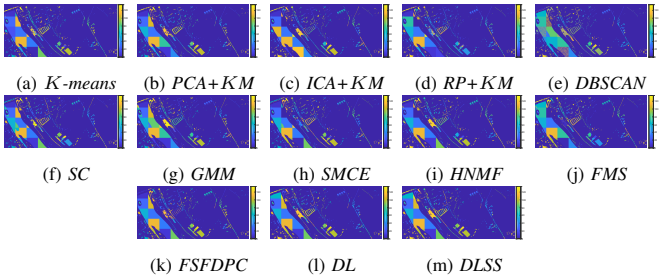


Fig. 16: *Results of clustering individual patches of the Pavia data, without synchronizing the labels.*

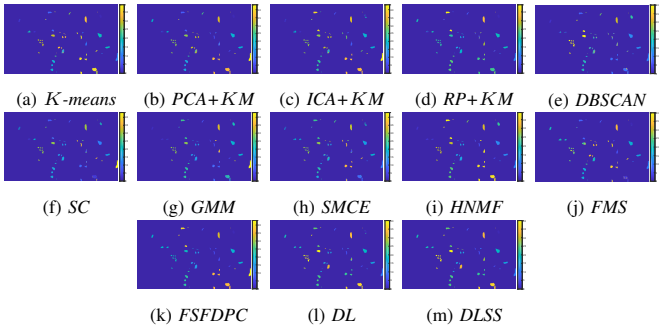


Fig. 17: *Results of clustering individual patches of Kennedy Space Center data, without synchronizing the labels.*

based on diffusion in both the spectral and spatial domains. We demonstrated on various data sets that the proposed DLSS algorithm performs well compared to state-of-the-art techniques, and that the DLSS algorithm outperforms DL thanks to the incorporation of spatial information. We remark that the methods which employ linear dimension reduction, including PCA, ICA, and random projections, generally outperform methods that use no dimension reduction, but do not perform as well as those which used nonlinear dimension reduction, including spectral clustering, SMCE, DL, and DLSS. This indicates that while HSI data does exhibit intrinsically low-dimensional structure, the data lies close not to subspaces, but manifolds, i.e. nonlinear sets of low dimensionality.

The proposed DL method, consisting of the geometric learning of modes but only spectral assignment of labels, largely outperforms all comparison methods (see Table II). In particular, it outperforms in all examples considered the very popular and recent FSFDPC algorithm. This indicates

Method	K-means	PCA+K-means	ICA+K-means	RP+K-means	DBSCAN	SC	GMM	SMCE	HNMF	FMS	FSFDPC	DL	DLSS
t-statistic	2.0163	2.1039	2.7710	3.1630	6.44774	2.6301	3.4461	4.1093	2.2160	2.1810	2.1219	2.1357	-

TABLE IV: *For each of the i comparison methods ($i = 1, \ldots, 12$), DLSS ($j = 13$) is compared against method i by computing the t-statistic score $\overline{\Delta^{i,j}}/(\sigma^{i,j}/\sqrt{N_{patches}-1})$. All of the t-statistics are significant enough to reject at the 95% level the two-sided null hypothesis that the results of DLSS do not differ significantly from those of method i, corresponding to the t-statistic exceeding 1.9934 when using $N_{patches} - 1 = 72$ degrees of freedom.*

that Euclidean distance is insufficient for learning the modes of complex HSI data. Moreover, the joint spectral-spatial labeling scheme DLSS improves over DL in all instances. In fact, DLSS gives the overall best performance for all datasets and all performance metrics.

The incorporation of active learning in the DLSS algorithm dramatically improves the accuracy of labeling of the Indian Pines, Pavia, and Salinas A datasets with very few label queries. This parsimonious use of training labels has the potential to greatly improve the efficiency of machine learning tasks for HSI, in which the number of labels necessary to label a significant proportion of the image is very high. The proposed active learning method can perform competitively with the state-of-the-art supervised EPF spectral-spatial classification algorithm, using a fraction of the number of labeled pixels.

A. Computational Complexity and Runtime

Let the data be $X = \{x_n\}_{n=1}^N \subset \mathbb{R}^D$. For the Indian Pines dataset, $N = 1250, D = 200$; for the Pavia dataset, $N = 13500, D = 102$; for the Salinas A dataset, $N = 7138, D = 224$; for the Kennedy Space Center dataset, $N = 25000, D = 176$. The most expensive step in DLSS is the construction of the nearest neighbor graph: we achieve near-linear scaling in N, $O(C_d DN \log N + k_1 DN)$, using the cover trees algorithm [71] with C_d a constant that depends exponentially on the intrinsic dimension d of the data, which is quite small in all the data sets considered. Once the nearest neighbors are found, the kernel density estimator, the random walk, and its eigenvectors can all be quickly constructed in time $O(N \log N)$, assuming that the number of nearest neighbors used in the density estimator is $O(\log N)$ and that the number of eigenvectors needed is $O(1)$. Computing the nearest spectral neighbor of higher empirical density, computing the spatial consensus labels, and active learning respectively have negligible computational complexity. We show empirical runtimes in Table V, which demonstrates that the proposed methods have superior runtimes to spectral clustering and DBSCAN, and are substantially faster than SMCE.

V. Future Research Directions

A drawback of many clustering algorithms, including the ones presented in this paper, is the assumption that the number of clusters, K, is known a priori. While unsupervised clustering experiments typically assume K is known, it is of interest to develop methods that allow efficient and accurate estimation of K, in order to make a truly unsupervised clustering method. Initial investigations suggest that looking for the "kink" in the sorted plot of $\mathcal{D}_t(x_i)$ could be used to detect K automatically. More precisely, we check if there is a prominent peak in the value $\mathcal{D}_t(x_{i+1}^{\text{sort}}) - \mathcal{D}_t(x_i^{\text{sort}})$, where where

Method	IP	Pavia	Salinas A	KSC
K-means	0.44	3.89	1.26	5.97
PCA+K-means	0.01	0.01	0.01	0.16
ICA+K-means	0.22	0.87	0.30	1.29
RP+K-means	0.11	0.79	0.13	0.58
DBSCAN	0.58	56.10	13.53	112.43
SC	0.77	101.20	13.87	483.56
GMM	0.40	3.07	1.91	2.16
SMCE	11.74	466.90	222.40	1315.21
HNMF	0.52	0.93	0.74	1.41
FMS	0.15	0.73	0.29	0.89
FSFDPC	1.48	33.64	10.05	69.46
DL	0.80	36.46	8.73	80.28
DLSS	1.40	73.77	12.24	106.05

TABLE V: *Run times for each method and each dataset, measured in seconds. The linear dimension reduction methods are extremely fast, as are NMF and GMM. The spectral clustering and FSFDPC algorithms are slower than DL, and DLSS is slightly slower is slightly slower than DL. The SMCE algorithm is substantially slower. Note that although FMS is quite fast, it is implemented in parallelized C++ code and ran on a machine with 24 cores and 48 threads.*

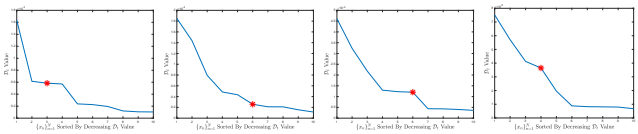
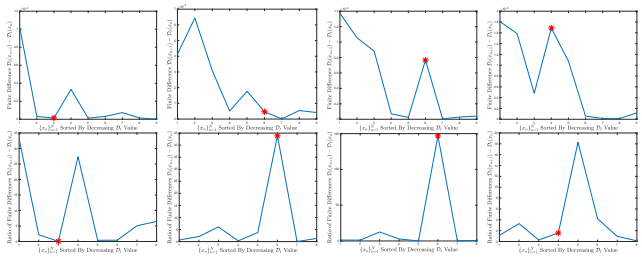


Fig. 18: *The sorted $\mathcal{D}_t(x_i^{sorted})$ values for each of the four datasets. From left to right: Indian Pines, Pavia, Salinas A, KSC. The estimate $\hat{K}$ is shown as a red star.*

the points $\{x_i^{\text{sort}}\}_{i=1}^n$ are the data, sorted in decreasing order of their $\mathcal{D}_t$ values. This is a discrete version of the gradient, so we are looking for a sharp drop-off in the sorted $\mathcal{D}_t$ curve. If there is a prominent such maxima in $\mathcal{D}_t(x_{i+1}^{\text{sort}}) - \mathcal{D}_t(x_i^{\text{sort}})$, precisely defined as a local maximum that is greater in magnitude than double the previous value, and also at least half the magnitude of the global maximum, we estimate $\hat{K}$ as this peak. If there is no such prominent peak, then we proceed to examine the second order information $(\mathcal{D}_t(x_{i+1}^{\text{sort}}) - \mathcal{D}_t(x_i^{\text{sort}}))/(\mathcal{D}_t(x_{i+2}^{\text{sort}}) - \mathcal{D}_t(x_{i+1}^{\text{sort}}))$. This is a discrete approximation to the second derivative of $\mathcal{D}_t$, to find when $\mathcal{D}_t$ begins to flatten. Initial analysis on the Indian Pines, Pavia, Salinas A and Kennedy Space Center datasets used in this article confirm the promise of analyzing the decay of $\mathcal{D}_t(x_i)$; results showing plots of $\{\mathcal{D}_t(x_i^{\text{sort}})\}$ values appear in Figure 18, while the corresponding statistics $\{\mathcal{D}_t(x_{i+1}) - \mathcal{D}_t(x_i)\}$ and $\{(\mathcal{D}_t(x_{i+1}^{\text{sort}}) - \mathcal{D}_t(x_i^{\text{sort}}))/(\mathcal{D}_t(x_{i+2}^{\text{sort}}) - \mathcal{D}_t(x_{i+1}^{\text{sort}}))\}$ are shown in Figure 19. The estimated number of clusters $\hat{K}$ appear in Table VI.

It is of interest to prove under what assumptions on the distributions and mixture model the plot $\mathcal{D}_t(x_n)$ correctly determines K. Moreover, in the case that one cluster is noticeably smaller or harder to detect than others, as in the case of the Indian Pines dataset, it may be advantageous to use a different statistic on the finite difference curve, rather than the proposed derivative conditions on $\mathcal{D}_t$. Initial mathematical results and more subtle conjectures are proposed in an upcoming article [33].



(a) *Indian Pines: first order estimate $\hat{K}$ inconclusive (top); second order $\hat{K} = 4$ incorrect (bottom).*

(b) *Pavia: first order estimate $\hat{K}$ inconclusive (top); second order estimate $\hat{K} = 6$ correct (bottom).*

(c) *Salinas A: first order $\hat{K} = 6$ correct (top); second order $\hat{K} = 6$ correct but not used (bottom).*

(d) *KSC: first order $\hat{K} = 4$ correct (top); second order $\hat{K} = 5$ incorrect but not used (bottom).*

Fig. 19: *We show the plots of $\{\mathcal{D}_t(x_{i+1}^{sort}) - \mathcal{D}_t(x_i^{sort})\}$ (top row) and the ratios $\{(\mathcal{D}_t(x_{i+1}^{sort}) - \mathcal{D}_t(x_i^{sort}))/(\mathcal{D}_t(x_{i+2}^{sort}) - \mathcal{D}_t(x_{i+1}^{sort})\})$ (bottom row) for each of the four HSI datasets considered in this article. The true number of classes is shown with a red star: the proposed method of estimating K is accurate for all the datasets except Indian Pines. We see that the first order information correctly determines that there are 6 clusters in Salinas A and 4 clusters in the KSC HSI, owing to the prominent peaks. The first order information is ambiguous for Indian Pines and Pavia, since there are no prominent peaks. The second order information correctly estimates that there are 6 clusters in the Pavia data, and incorrectly estimates 4 clusters for Indian Pines.*

Dataset	IP	Pavia	Salinas A	KSC
Estimated Number of Classes $\hat{K}$	4	6	6	4
Number of Labeled GT Classes K	3	6	6	4

TABLE VI: *We show the number of classes estimated by looking for the "kink" in the sorted plot of $\mathcal{D}_t(x_n)$. We see that for the Salinas A and Kennedy Space Center datasets, estimating based on $\{\mathcal{D}_t(x_{i+1}^{sort}) - \mathcal{D}_t(x_i^{sort})\}$ correctly estimates the number of labeled classes, while this statistic is inconclusive for Indian Pines and Salinas A. For these data, we move to second order information, namely estimating K by maximizing $(\mathcal{D}_t(x_{i+1}^{sort}) - \mathcal{D}_t(x_i^{sort}))/(\mathcal{D}_t(x_{i+2}^{sort}) - \mathcal{D}_t(x_{i+1}^{sort}))$. This estimator overestimates the number of classes for Indian Pines, estimating 4 instead of 3, while it correctly estimates the number of classes in Pavia. The Indian Pines dataset is the most challenging of the four labeled datasets analyzed, which suggests that the the proposed method for estimating K may be insufficient for challenging HSI data.*

Moreover, all unsupervised algorithms considered in this paper struggle with very large HSI scenes consisting of many classes. This is due to the large variation within clusters compared to the differences between clusters, which leads to genuine classes being split incorrectly; this is a well-known challenge for unsupervised clustering of HSI [16]. In Section III-H it is shown that DLSS is very effective at clustering on different patches of a large HSI. It remains an open question how to combine the results on these patches into a global clustering, which amounts to determining when to merge clusters learned in distinct patches. Automatically implementing such mergers with the DLSS framework is a direction of future research.

The present work is essentially empirical: it is not known mathematically under what constraints on the mixture model the method proposed for learning modes succeeds with high probability. Besides being of mathematical interest, this would be useful for understanding the limitations of the proposed method for HSI. To understand this phenomenon rigorously, a careful analysis of diffusion distances for data drawn from a non-parametric mixture model is required, which is related to investigating performance guarantees for spectral clustering and mode detection [72], [73].

It is also of interest to explicitly incorporate spectral-spatial features into the diffusion construction. It is known that use of spectral-spatial features is beneficial for supervised learning of HSI [69], [74], [75], and their use in unsupervised learning is an exciting research direction. Indeed, incorporating the spatial properties of the scene into the graph from which diffusion distances are generated may render the explicit spatial regularization step of the proposed algorithm redundant, thus improving runtime.

ACKNOWLEDGMENTS

The authors would like to thank Ed Bosch for his helpful comments on a preliminary version of this paper. This research was partially supported by NSF-ATD-1222567, NSF-ATD-1737984, AFOSR FA9550-14-1-0033, AFOSR FA9550-17-1-0280, NSF-IIS-1546392, and ARO subcontract to W911NF-17-P-0039. We would also like to thank the anonymous reviewers of this article, whose valuable comments significantly improved the presentation and content of this article.

REFERENCES

[1] G. Lu and B. Fei. Medical hyperspectral imaging: a review. *Journal of Biomedical Optics*, 19(1):010901–010901, 2014.

[2] Y. Wang, G. Chen, and M. Maggioni. High-dimensional data modeling techniques for detection of chemical plumes and anomalies in hyperspectral images and movies. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9):4316–4324, 2016.

[3] M.T. Eismann. *Hyperspectral remote sensing*. SPIE, 2012.

[4] L. Ma, M.M. Crawford, and J. Tian. Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4099–4109, 2010.

[5] W. Li, E.W. Tramel, S. Prasad, and J.E. Fowler. Nearest regularized subspace for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):477–489, 2014.

[6] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, 2004.

[7] M. Fauvel, J.A. Benediktsson, J. Chanussot, and J.R. Sveinsson. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11):3804–3814, 2008.

[8] F. Ratle, G. Camps-Valls, and J. Weston. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2271–2282, 2010.

[9] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[10] H. Liang and Q. Li. Hyperspectral imagery classification using sparse representations of convolutional neural network features. *Remote Sensing*, 8(2):99, 2016.

[11] Y. Qian, M. Ye, and J. Zhou. Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2276–2291, 2013.

[12] J. Li, J.M. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression. *IEEE Geoscience and Remote Sensing Letters*, 10(2):318–322, 2013.

[13] A. Paoli, F. Melgani, and E. Pasolli. Clustering of hyperspectral images based on multiobjective particle swarm optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 47(12):4175–4188, 2009.

[14] N. Acito, G. Corsini, and M. Diani. An unsupervised algorithm for hyperspectral image segmentation based on the gaussian mixture model. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 6, pages 3745–3747, 2003.

[15] C. Cariou and K. Chehdi. Unsupervised nearest neighbors clustering with application to hyperspectral images. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1105–1116, 2015.

[16] W. Zhu, V. Chayes, A. Tiard, S. Sanchez, D. Dahlberg, A. Bertozzi, S. Osher, D. Zosso, and D. Kuang. Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2786–2798, 2017.

[17] Y. Chen, S. Ma, X. Chen, and P. Ghamisi. Hyperspectral data clustering based on density analysis ensemble. *Remote Sensing Letters*, 8(2):194–203, 2017.

[18] E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. In *Advances in Neural Information Processing Systems*, pages 55–63, 2011.

[19] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[20] N. Gillis, D. Kuang, and H. Park. Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):2066–2078, 2015.

[21] Z. Meng, E. Merkurjev, A. Koniges, and A. Bertozzi. Hyperspectral video analysis using graph clustering methods. *Image Processing On Line*, 7:218–245, 2017.

[22] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[23] H. Zhang, H. Zhai, and L. Zhangand P. Li. Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3672–3684, 2016.

[24] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 2011.

[25] B. Demir and L. Bruzzone. A novel active learning method in relevance feedback for content-based remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2323–2334, 2015.

[26] A. Stumpf, N. Lachiche, J.-P. Malet, N. Kerle, and A. Puissant. Active learning in the spatial domain for remote sensing image classification. *IEEE transactions on geoscience and remote sensing*, 52(5):2492–2507, 2014.

[27] D. Tuia, E. Pasolli, and W.J. Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242, 2011.

[28] J. Li, J.M. Bioucas-Dias, and A. Plaza. Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):844–856, 2013.

[29] D. Tuia, F. Ratle, F. Pacifici, M.F. Kanevski, and W.J. Emery. Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2218, 2009.

[30] J. Li, J.M. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4085–4098, 2010.

[31] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005.

[32] R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[33] M. Maggioni and J.M. Murphy. Learning by unsupervised nonlinear diffusion. In preparation, 2018.

[34] Y. Tarabalka, J.A. Benediktsson, and J. Chanussot. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973–2987, 2009.

[35] J. Benedetto, W. Czaja, J. Dobrosotskaya, T. Doster, K. Duke, and D. Gillis. Integration of heterogeneous data for classification in hyperspectral satellite imagery. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII*, volume 8390, page 839027. International Society for Optics and Photonics, 2012.

[36] M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, and J.C. Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2013.

[37] N.D. Cahill, W. Czaja, and D.W. Messinger. Schroedinger eigenmaps with nondiagonal potentials for spatial-spectral clustering of hyperspectral imagery. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*, volume 9088, page 908804. International Society for Optics and Photonics, 2014.

[38] A. Cloninger, W. Czaja, and T. Doster. Operator analysis and diffusion based embeddings for heterogeneous data fusion. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 1249–1252. IEEE, 2014.

[39] Z. Wang, N.M. Nasrabadi, and T.S. Huang. Spatial–spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4808–4822, 2014.

[40] J.J. Benedetto, W. Czaja, J. Dobrosotskaya, T. Doster, and K. Duke. Spatial-spectral operator theoretic methods for hyperspectral image classification. *GEM-International Journal on Geomathematics*, 7(2):275–297, 2016.

[41] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[42] R.R. Coifman, I.G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[43] A. Singer and R.R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.

[44] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[45] A.D. Szlam, M. Maggioni, and R.R. Coifman. Regularization on graphs with function-adapted diffusion processes. *Journal of Machine Learning Research*, 9:1711–1739, 2008.

[46] S. Lafon, Y. Keller, and R.R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[47] W. Czaja, B. Manning, L. McLean, and J.M. Murphy. Fusion of aerial gamma-ray survey and remote sensing data for a deeper understanding of radionuclide fate after radiological incidents: examples from the fukushima dai-ichi response. *Journal of Radioanalytical and Nuclear Chemistry*, 307(3):2397–2401, 2016.

[48] R.R. Lederman and R. Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 2015.

[49] R.R. Lederman, R. Talmon, H. Wu, Y.-L. Lo, and R.R. Coifman. Alternating diffusion for common manifold learning with application to sleep stage assessment. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5758–5762. IEEE, 2015.

[50] M.A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(12):03B624, 2011.

[51] W. Zheng, M.A. Rohrdanz, M. Maggioni, and C. Clementi. Polymer reversal rate calculated via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(14):144109, 2011.

[52] J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[53] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[54] S.J. Sheather and M.C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[55] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[56] J.M. Murphy and M. Maggioni. Iterative active learning with diffusion geometry for hyperspectral images. In *9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2018. To appear.

[57] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha. Beyond kappa: A review of interrater agreement measures. *Canadian journal of statistics*, 27(1):3–23, 1999.

[58] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[59] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[60] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[61] S. Dasgupta. Experiments with random projection. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 143–151. Morgan Kaufmann Publishers Inc., 2000.

[62] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[63] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[64] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

[65] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[66] D. Manolakis, C. Siracusa, and G. Shaw. Hyperspectral subpixel target detection using the linear mixing model. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1392–1409, 2001.

[67] S. Kraut, L.L. Scharf, and L.T. McWhorter. Adaptive subspace detectors. *IEEE Transactions on Signal Processing*, 49(1):1–16, 2001.

[68] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[69] X. Kang, S. Li, and J.A. Benediktsson. Feature extraction of hyperspectral images with image fusion and recursive filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(6):3742–3752, 2014.

[70] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[71] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *International Conference on Machine Learning*, pages 97–104, 2006.

[72] C.R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Non-parametric inference for density modes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):99–126, 2016.

[73] G. Schiebinger, M.J. Wainwright, and B. Yu. The geometry of kernelized spectral clustering. *The Annals of Statistics*, 43(2):819–846, 2015.

[74] P. Ghamisi, R. Souza, J.A. Benediktsson, L. Rittner, R. Lotufo, and X.X. Zhu. Hyperspectral data classification using extended extinction profiles. *IEEE Geoscience and Remote Sensing Letters*, 13(11):1641–1645, 2016.

[75] X. Kang, X. Xiang, S. Li, and J.A. Benediktsson. PCA-based edge-preserving features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7140–7151, 2017.