Diffusion Geometric Methods for Fusion of Remotely Sensed Data

James M. Murphy[a], Mauro Maggioni[a,b,c,d]

Department of Mathematics[a], Department of Applied Mathematics and Statistics[b], Mathematical Institute of Data Sciences[c], Institute of Data Intensive Engineering and Science,[d] Johns Hopkins University, Baltimore MD, 21218

ABSTRACT

We propose a novel unsupervised learning algorithm that makes use of image fusion to efficiently cluster remote sensing data. Exploiting nonlinear structures in multimodal data, we devise a clustering algorithm based on a random walk in a fused feature space. Constructing the random walk on the fused space enforces that pixels are considered close only if they are close in both sensing modalities. The structure learned by this random walk is combined with density estimation to label all pixels. Spatial information may also be used to regularize the resulting clusterings. We compare the proposed method with several spectral methods for image fusion on both synthetic and real data.

Keywords: Multimodal data, remote sensing, diffusion maps, unsupervised learning, image fusion, clustering

1. INTRODUCTION

Unsupervised learning of data is a major problem in machine learning, and is necessary in the case that human guidance is impractical, particularly when the data set size renders human-guided training infeasible. One of the most significant unsupervised learning problems is the *clustering* problem, in which the data is to be partitioned and labeled according to their partition element. A wide variety of clustering techniques have been proposed in the literature, with differing theoretical guarantees and computational burdens. Classical methods include k-means,[1–3] hierarchical methods,[1,4] density-based methods,[5] and mode-based methods.[6–10] In order to achieve accurate and robust empirical performance, feature extraction is often combined with classical methods, which allows an algorithm to make clustering decisions based only on the most important features in the data, as determined by the feature extractor. In particular, *spectral methods*[11,12] construct graphs representing data, and use the spectral properties of the graph weight matrix or graph Laplacian to produce features that encode patterns and structure in the data. It is of crucial significance to develop unsupervised methods that not only perform well on real-world data, but enjoy low computational complexity with regards to the number of data points and number of dimensions in the dataset.

Unsupervised learning is of particular importance in remote sensing, where the volume of data acquired is rapidly exceeding human analysis capacity. Indeed, remote sensing sensors collect far more data than can be manually analyzed by humans. A particular instance of the clustering problem in remote sensing is when two or more sensors record data derived from the same scene. This allows for *data fusion* of signals that captures complementary aspects of the underlying scene. The use of multiple sensors can lead to significant improvement in a variety of remote sensing tasks,[13–19] by incorporating complementary information. One may consider data fusion as a preprocessing step to clustering in which the goal is to cluster data according to the features captured by multiple sensors. In order for such unsupervised learning to be meaningful, the fused clustering algorithm must synthesize disparate information from its input sensors in a principled manner.

Further author information:: J.M.M. (corresponding author): jmurphy@math.jhu.edu; M.M.: mauro@math.jhu.edu

In this article, we propose a clustering method for data realized as multiple measurements of the same underlying object, that is, a data-fusion clustering algorithm. The proposed method exploits non-linear structure in the fused data-space as well as its density, in order to identify modes, which are subsequently used for clustering. Our algorithm is demonstrated on synthetic and real remote sensing datasets, revealing the suitability of the proposed method and illustrating its competitive empirical performance against related spectral clustering algorithms which combine the two data sources in various ways. The article outline is as follows. In Section 2, we provide a detailed review of the *diffusion maps* construction, which is a crucial component of the proposed method. In Section 3, we present the main contribution of the paper, the *fused diffusion learning (FDL)* algorithm, which generalizes the recent diffusion learning algorithm[20] for data fusion. We conduct experiments with the FDL algorithm and comparison methods on synthetic and real remote sensing data in Section 4. Finally, we conclude and discuss related research directions in Section 5.

2. BACKGROUND ON DIFFUSION GEOMETRY

We propose an algorithm that compares distances between points with *diffusion distances*.[21–23] This nonlinear, graph-based distance has been applied to problems ranging from analysis of dynamical systems,[22,24–26] to semisupervised learning,[27,28] to data fusion,[29,30] to latent variable separation,[31,32] to the physical sciences.[33,34] Let $\{x_n\}_{n=1}^N = X \subset \mathbb{R}^D$ be a discrete set of data points. The diffusion distance between $x, y \in X$, denoted $d_t(x, y)$, is determined by the *underlying geometry of* X as computed by diffusion processes on X. It is particularly useful when the data lies close to a low-dimensional set (say, of dimension $d \ll D$), as it can be shown to depend uniquely on the intrinsic geometry and independent of the embedding in the ambient space. The parameter t enjoys an interpretation as the time length of the diffusion process. The computation of d_t requires constructing an undirected, weighted graph $\mathcal{G}$ with vertices corresponding to the points of X and weighted edges given by an $N \times N$ weight matrix $W(x, y) = e^{-\|x-y\|_2^2/\sigma^2}$ if $x \in NN_k(y)$ for some suitable choice of σ and $W(x, y) = 0$ otherwise, with $NN_k(x)$ the set of k-nearest neighbors of y in X with respect to Euclidean distance. Under mild assumptions, a fast nearest neighbors algorithm such as cover trees[35] yields W in time quasilinear in N for $k = O(\log(N))$. The *degree* of x is $\deg(x) := \sum_{y\in X} W(x, y)$. A Markov diffusion, which corresponds to a random walk on $\mathcal{G}$, has an $N \times N$ transition matrix $P(x, y) = W(x, y)/\deg(x)$. Given an initial probability 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, that is, according to the geometry of X. This Markov chain has a stationary distribution $\pi(x) = \deg(x)/\sum_{y\in X} \deg(y)$.

The *diffusion distance at time* t is $d_t^2(x, y) = \sum_{u\in X}(P^t(x, u) - P^t(y, u))^2 d\mu(u)/\pi(u)$. 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 when x, y are strongly connected in the graph, and large when x, y are weakly connected in the graph. The *spectral decomposition* of P allows to derive fast algorithms to compute d_t: the matrix P admits, under mild assumptions,[23] a spectral decomposition 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 can then be written as

$$d_t^2(x, y) = \sum_{n=1}^N \lambda_n^{2t}(\Phi_n(x) - \Phi_n(y))^2 . \tag{1}$$

The weighted eigenvectors $\{\lambda_n^t \Phi_n\}_{n=1}^N$ may consequently be interpreted as new data-dependent coordinates of X, which are almost geometrically intrinsic.[21,22] Euclidean distance in these new coordinates is diffusion distance on $\mathcal{G}$.

Assuming 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 (1) may truncated at some suitable $2 \leq M \ll N$, depending on the decay of the eigenvalues. In our experiments, M was set to be the value at which the decay of the eigenvalues $\{\lambda_n\}_{n=1}^N$ began to taper off, i.e. the "kink" in the eigenvalue plot; this is a standard heuristic for such metrics.[23] 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. In this sense, the mapping $x \mapsto (\lambda_1^t \Phi_1(x), \lambda_2^t \Phi_2(x), \ldots, \lambda_M^t \Phi_M(x))$ is a dimension reduction mapping of the ambient space $\mathbb{R}^D$ to $\mathbb{R}^M$.

Moreover, the truncation reduces the computational complexity by requiring the algorithm to compute only $M \ll N$ eigenvectors.

3. THE FUSED DIFFUSION LEARNING ALGORITHM

We propose to fuse two separate remotely sensed data sources as follows. Suppose there are two sensors which produce data $X_1 = \{x_n^1\}_{n=1}^N, \mathbb{R}^{D_1}, X_2 = \{x_n^2\}_{n=1}^N \subset \mathbb{R}^{D_2}$ from a common underlying scene. Here, N is the number of pixels and D_i are the number of dimensions in the data. We assume that the data sets X_i are registered, so that there is a bijective correspondence between $x_i^1 \leftrightarrow x_i^2$. While we assume N is constant across the datasets, we allow for $D_1 \neq D_2$. Allowing for the data to be of different dimensions is important in remote sensing, where multispectral and high dimensional data are much higher dimensional than optical or lidar digital elevation models.

Let K be the number of clusters in the data, assumed known or estimated a priori. Our algorithm proceeds in two major steps: mode identification and labeling of points. As in the recently proposed *diffusion learning algorithm*,[20] we consider diffusion processes on the data. However, in the method proposed in this article, the diffusion process used for learning is built on the *fused data space* corresponding to a weighted concatenation of X_1 and X_2. This allows for both datasets to contribute to the geometry of the underlying graph, thus leading to more precise learning of the scene.

We begin by concatenating the two datasets into a fused dataset. In order to ensure that the two datasets are valued equally in the fusion, we weight them by a balancing coefficient $\lambda = \|X_1\|_{\text{Fro}}/\|X_2\|_{\text{Fro}}$, where $\|A\|_{\text{Fro}}$ denotes the Frobenius norm of a matrix A. Let $X = \{x_n\}_{n=1}^N$ be the concatenated data matrix, where $x_n = (x_n^1, \lambda x_n^2)$. Note that this construction requires that the datasets to be fused have the same number of points, which may be a limit in practice for remote sensing data captured at very different spatial resolutions. This is could be addressed through image superresolution.

The algorithm for learning the modes of the classes is summarized in Algorithm 3.1. It first computes an empirical density for each point x_n with a kernel density estimator: $p(x_n) = p_0(x_n)/\sum_{m=1}^N p_0(x_m)$, where $p_0(x_n) = \sum_{x_m \in NN_k(x_n)} e^{-\|x_n - x_m\|_2^2/\sigma_1^2}$. The Gaussian kernel density estimator enjoys strong theoretical guarantees[1,36] but certainly other estimators may be used. Once the empirical density p is computed for every point, the modes of the HSI classes are computed by combining density with diffusion distances. Let $\tilde{\rho}_t$ be the time-dependent quantity 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 $\rho_t(x_n) = \tilde{\rho}_t(x_n)/\max_{x_m} \tilde{\rho}_t(x_m)$. The modes of the HSI are computed as the points $x_1^*, \ldots, x_K^*$ yielding the K largest values of the quantity $\mathcal{D}_t(x_n) = p(x_n)\rho_t(x_n)$. 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 distributions. This method provably detects modes correctly under suitable distributional assumptions on the data.[37] We note that all notions of distance here are in the fused data space.

Once the modes are detected, each is given a unique, arbitrary label. All other points are labeled using these mode labels in a two-stage process, described in Algorithm 3.2. 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 $> .5$, 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

Algorithm 3.1: Fused Mode Detection Algorithm

3.1.1 *Input*: $X_1 = \{x_n^1\}_{n=1}^N, X_2 = \{x_n^2\}_{n=1}^N, K$; t.
3.1.2 Compute scaling constant $\lambda = \|X_1\|_{\text{Fro}}/\|X_2\|_{\text{Fro}}$.
3.1.3 Compute concatenated dataset $\{x_n\}_{n=1}^N = X$, $x_n = (x_n^1, \lambda x_n^2)$.
3.1.4 Compute the empirical density $p(x_n)$ for each $x_n \in X$.
3.1.5 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.
3.1.6 Set the learned modes $\{x_i^*\}_{i=1}^K$ to be the K maximizers of $\mathcal{D}_t(x_n) = p(x_n)\rho_t(x_n)$.
3.1.7 *Output:* $\{x_i^*\}_{i=1}^K, \{p(x_n)\}_{n=1}^N, \{\rho_t(x_n)\}_{n=1}^N$.

within radius r_s. Then the spatial consensus label of x_i is

$$y_i^{\text{spatial}} = \begin{cases} k, & \frac{|\{y_n | y_n = k,\ y_n \in L_n^{\text{spatial}}\}|}{|L_n^{\text{spatial}}|} > .5, \\ 0 \text{ (no label)}, & \text{else.} \end{cases} \tag{2}$$

After a point's spatial consensus label is computed, it's *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 (2)) 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. 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.

Algorithm 3.2: Spectral-Spatial Labeling Algorithm

3.2.1 *Input:* $\{x_n\}_{n=1}^N, \{x_i^*\}_{i=1}^K, \{p(x_n)\}_{n=1}^N$; r_s.
3.2.2 Assign each mode a unique label.
3.2.3 *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.
3.2.4 *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.
3.2.5 *Output:* Labels $\{y_n\}_{n=1}^N$.

High density points are expected to be labeled according to their spectral properties, for two reasons. First, high density points are likely to be near the cores of distributions in real data, which are expected to correspond to spatially homogeneous regions. Second, points of high density are labeled before points of low density, so it is not likely for high density points to have many labeled points in their spatial neighborhoods. This means that the spatial consensus label is unlikely to even 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 according to their spatial neighbors. We remark that the incorporation of spatial information into HSI learning is justified by the fact that HSI typically show some amount of spatial regularity. Indeed, if a pixel's nearest spatial neighbors all have the same class label, it is likely that the pixel has this same label.[16,38–46] In practice, the spatial information regularizes and improves performance, but it cannot take the place of the specific information content of the sensor, which is generally more salient than the spatial properties of the image.

The proposed method, combining Algorithms 3.1, 3.2 is called *spectral-spatial fused diffusion learning (FDLSS)*. 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 3.1, but labels all pixels simply by requiring each point have the same label as its nearest spectral neighbor of higher density. In this sense, FDLSS is a smoothed version of FDL; indeed, the spatial regularization smoothes out some of the speckling that can occur in unsupervised learning of remotely sensed images.

4. EXPERIMENTAL ANALYSIS

In order to validate the efficacy of FDLSS, we qualitatively evaluate its performance on synthetic and real datasets. We also compare the clustering generated by the proposed method with those generated by related algorithms, which seek to cluster data using operators on graphs.

4.1 Experimental Data

We first consider a *synthetic dataset* in which there are four clusters to be learned. However, the two sets of measurements provided are both incomplete for accurate learning of the underlying classes, thus motivating the necessity of data fusion. We then consider two real remote sensing data sets, which capture an underlying scene with different sensing modalities. The 2013 IEEE GRSS data fusion contest data* consists of *lidar and hyperspectral data* capturing a scene of the University of Houston and neighboring urban areas in Houston, TX, USA. The lidar data is realized as a digitial elevation model (DEM). It was captured on June 22, 2012, and recorded at an average height of 2000 feet above ground. The HSI data consists of 144 spectral bands in the 380 to 1050 nm range. The 2000 IEEE GRSS data fusion contest data consists of *multispectral and panchromatic images* captured over Hasselt, Belgium in 1999. The multispectral image consists of 7 spectral bands at spectral resolution ranging from 450 to 2350 nm and a spatial resolution ranging from 30 to 60, while the panchromatic sensor has a spectral resolution of 520-900 and a fine spatial resolution of 15m.

4.2 Comparison Methods

In order to analyze the performance of FDL and FDLSS, we compare with several related methods. As a benchmark, we compute cluster labels on each of the two data sources individually using spectral clustering.[12] The results on the separate data sets are expected to be poorer than those on the fused data, thus validating the need to perform data fusion. We also consider clustering using the eigenvectors of a Markov transition matrix P_{AD}, which is constructed as the product of Markov transition matrices P_1, P_2 on data sources 1 and 2, respectively: $P_{\text{AD}} = P_1 P_2$. This method is known as *alternating diffusion*, and has been shown to be effective in learning latent variables in certain settings.[31,32] Unfortunately, the joint diffusion operator P is not stochastic, and in fact does not admit a spectral decomposition in general, which imposes computational limitations on its use. We also consider a method for combining Laplacian matrices constructed on separate data sources called the *power mean Laplacian*.[47,48] This method consists in computing graph Laplacians L_1, L_2 on the two data sources separately, then combining them by taking their power mean: $L_p = \left(\frac{1}{2}(L_1^p + L_2^p)\right)^{1/p}$, where typically $p < 0$. When $p \ll 0$, L_p can correctly learn labels in a modified stochastic block model (SBM) in which L_1, L_2 are individually insufficient.[48] While the SBM is a major simplification of the real-world data setting, it suggests that the power mean Laplacian could be a useful method for combining Laplacians generated from disparate sensors. A major drawback of using matrix power Laplacians is its high computational complexity, even in cases when L_1, L_2 are individually sparse. Indeed, computing L_p is generally $O(N^3)$ for $p < 0$. Finally, we consider spectral clustering with a kernel computed on the joint data space, balanced as in the FDLSS algorithm so that the two sources contribute equally.

*http://www.grss-ieee.org/community/technical-committees/data-fusion/2013-ieee-grss-data-fusion-contest/

4.3 Parameters

We set $\sigma = 1$ in the construction of the underlying graph for diffusion distances, and used 100 nearest neighbors. The diffusion time t was set to 30, and the cutoff M in the spectral expansion was set to be point at which the eigenvalues had the largest drop. In the density computation, 20 nearest neighbors were used and the scaling parameter σ was set to be half the mean nearest neighbor distance. We set $\sigma = 1$ for all spectral graph constructions in the comparison methods, and also used 100 nearest neighbors.

4.4 Experimental Results

4.4.1 Synthetic Data

Images of the synthetic data and subsequent clustering results appear in Figure 1. These data were generated pixel-wise as realizations of a Gaussian random variables. The data are 60×60, so $N = 3600, D_1 = D_2 = 1$. The means between each half of the respective sources differ substantially. In this example, we set $K = 4$, since there are evidently 4 unique clusters. Clearly the underlying four clusters cannot be learned from either data source individually, hence the poor labelings for running spectral clustering data sources 1 and 2 alone. We see that spectral clustering with a joint kernel and FDL both achieve mostly accurate clustering. However, certain points are anomalously labeled, due to high levels of noise at these pixels. Hence, the FDLSS with spatial regularity is able to achieve a smoother labeling, by replacing the labels of these noise points with the labels of their near spatial neighbors.

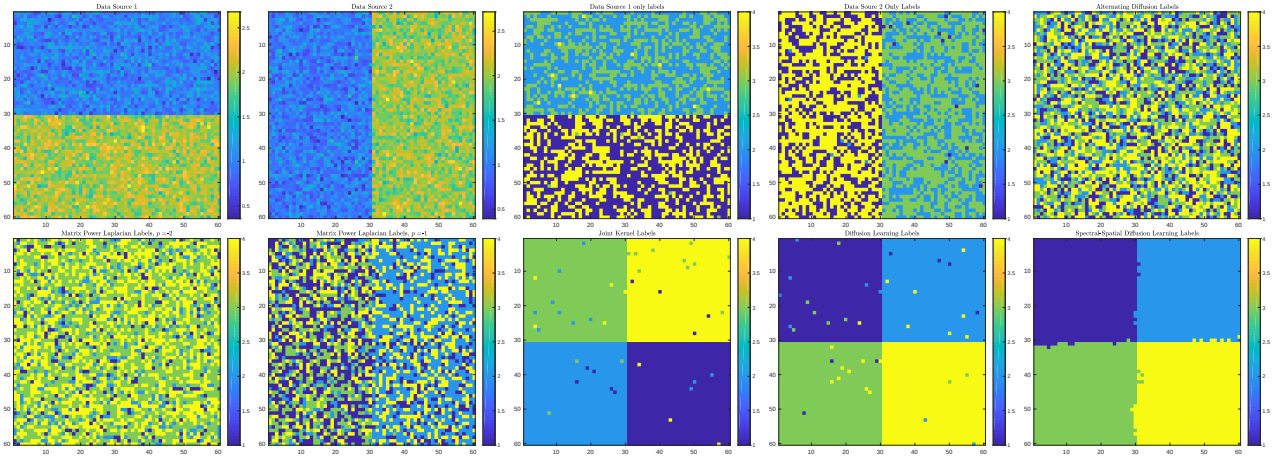

Figure 1: *First row, left to right: Data source 1; Data source 2; Spectral clustering on data source 1 alone; Spectral clustering on data source 2 alone; Alternating diffusion. Second row, left to right: Matrix power Laplacian, $p = -2$; Matrix power Laplacian, $p = -1$; Joint kernel; FDL; FDLSS. In the synthetic data, by construction it is not possible to learn the correct segmentation using either data source alone, as can be seen from the failure to segment. On the other other hand, spectral clustering with a joint kernel, diffusion learning on the joint data space, and spectral-spatial diffusion learning on the joint data space are all able to segment essentially correctly. Note that the spatial regularization fixes some errors in labeling high-noise points, at the expense of some errors on the class boundaries.*

4.4.2 HSI and Lidar Data

Images of the real HSI and lidar data and clustering results appear in Figure 2. We see that there are objects in one image not in the other, for example a road in the HSI scene and trees in the lidar scene. This is because the information in a hyperspectral sensor is chemical, while a lidar sensor captures height information, which suggests the merit in fusing these data sources for clustering. In order to make the data of tractable size for the

power mean Laplacian, a 100×100 subset of the full scene was used. So, $N = 10000, D_1 = 144, D_2 = 1$. We set $K = 10$ in order to allow for rich segmentation of the scene.

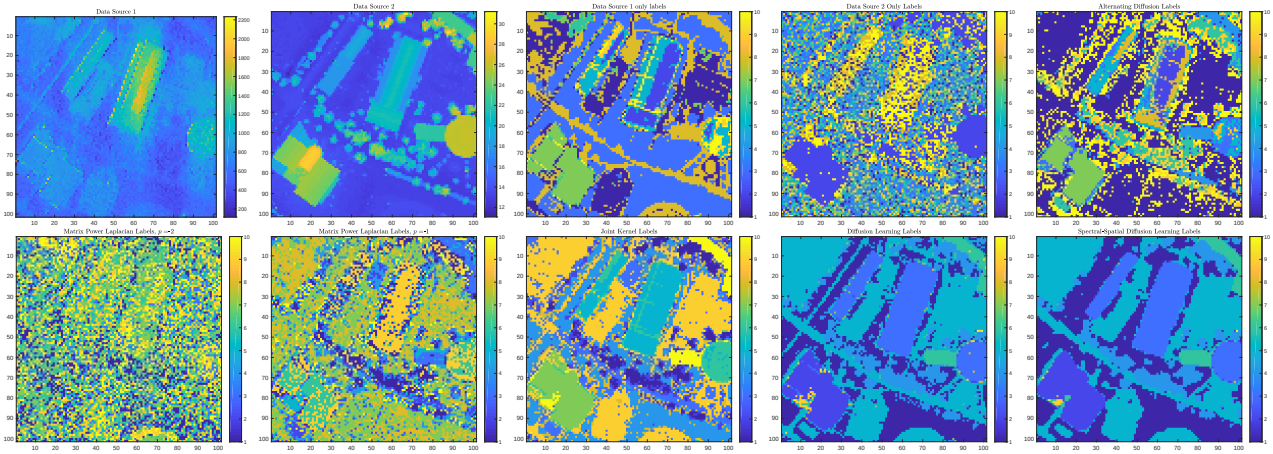

Figure 2: *First row, left to right: HSI; lidar; Spectral clustering on HSI alone; Spectral clustering on lidar alone; Alternating diffusion. Second row, left to right: Matrix power Laplacian, $p = -2$; Matrix power Laplacian, $p = -1$; Joint kernel; FDL; FDLSS. In this scene, the HSI and lidar show different objects. Using either alone to segment the scene yields insufficient segmentation. Of the fusion methods, only DL and DLSS on the fused data provide adequate segmentation, and these results are actually comparable in this case, indicating that spatial regularization is not especially necessary for this data. In fact, the spatial regularization may even be damaging, as can be seen from the disappearance of some highly elongated segments when using FDLSS, compared to DLSS.*

4.4.3 Multispectral and Panchromatic Data

Images of the real multispectral and panchromatic data and clustering results appear in Figure 3. The multispectral sensor enjoys a large range of electromagnetic resolution, but suffers from a low spatial resolution. On the other hand, the panchromatic sensor has lower spectral resolution, but improved spatial resolution. This suggests that more structure in the underlying scene can be learned through data fusion, than can be learned from a separate analysis of the data sources. In order to make the data of tractable size for the power mean Laplacian, a 100×100 subset of the full scene was used. So, $N = 10000, D_1 = 7, D_2 = 1$. We set $K = 10$ in order to allow for rich segmentation of the scene.

4.5 Computational Complexity

The computation of diffusion distances on the concatenated dataset is $O(C^d(D_1 + D_2)N \log(N))$ where N is the number of pixels and d is the intrinsic dimension of the concatenated dataset, thanks to the use of cover trees for computing Euclidean nearest neighbors and truncating the spectral computation after $M = O(1)$ eigenvectors. The computation of density is $O(C^d(D_1 + D_2)N \log(N))$, since a nearest neighbors search is required for this as well. The subsequent labelings and spatial regularizations are linear in N, so the overall complexity of the proposed algorithm is $O(C^d(D_1 + D_2)N \log(N))$. It is commonly the case that $d, D_1, D_2 = O(1)$ with respect to N, so that the overall complexity is $O(N \log(N))$. Spectral clustering can be made to run comparatively fast, though alternating diffusion and power mean Laplacian computations are considerably slower for large N.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this article, we propose a novel algorithm that fuses and clusters disparate data. Building on the DL method,[20] we proposed to cluster points by considering density and random walks in the joint data space. The proposed FDL method enjoys low computational complexity, and effectively clusters data in a variety of fusion settings.

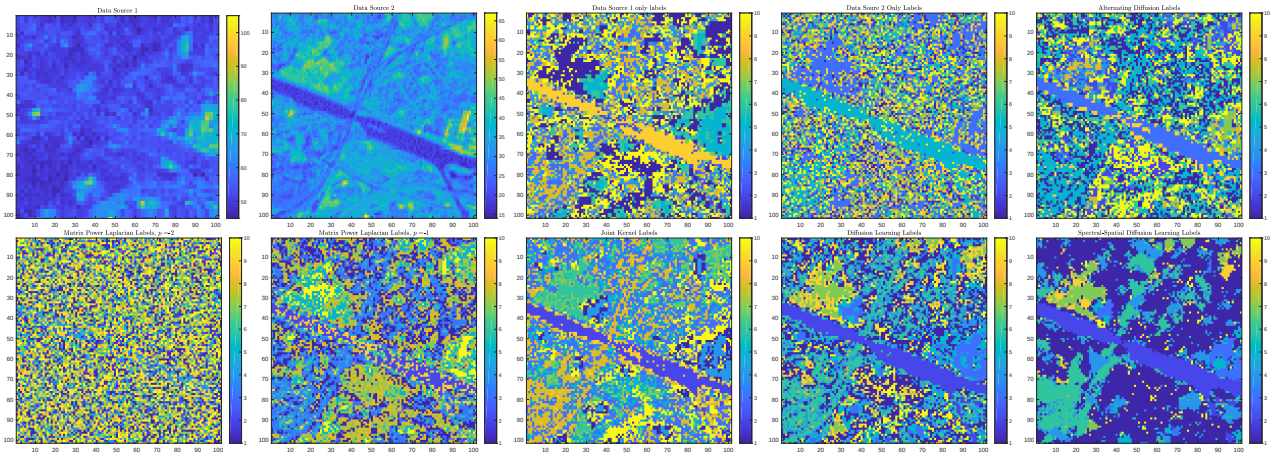

Figure 3: *First row, left to right: multispectral; panchromatic; Spectral clustering on multispectral alone; Spectral clustering on panchromatic alone; Alternating diffusion. Second row, left to right: Matrix power Laplacian, $p = -2$; Matrix power Laplacian, $p = -1$; Joint kernel; FDL; FDLSS. We see that for the multispectral and panchromatic data, many methods are able to segment the route flowing through the center of the scene. The DL method on the joint data space gives clustering results similar to the results for spectral clustering with the joint kernel. However, adding spatial regularization helps, as shown by the clearer segmentation results of DLSS.*

The incorporation of spatial information in the FDLSS algorithm further improves results by smoothing the clusters spatially.

The work in this article is essentially empirical, but an underlying mathematical model provides the basis for the proposed FDL algorithm, namely that data points should be considered similar only if they are similar in both of two different sensors. It is of interest to develop this theory, and in particular, to build a continuum of models that interpolate from the regime in which points are similar if they are similar in data source 1 *or* data source 2, and the regime in which points are similar if they are similar in data source 1 *and* data source 2. The and-or dichotomy will suggest which method of graphical fusion is appropriate for particular learning tasks. A related direction is the fusion of more than two data sources.

6. ACKNOWLEDGEMENTS

We thank the IEEE GRSS data fusion context committees in 2000 and 2013 for making the data for these contests publicly available. We thank Chris Kurcz of Intelligent Automation Incorporated for stimulating conversations on the topic of data fusion. We also thank Chae A. Clark of Cyberaics for many fruitful conversations and edits to the manuscript. The authors acknowledge support from NSF-ATD-1737984, AFOSR FA9550-17-1-0280, and NSF-IIS-1546392.

REFERENCES

1. Friedman, J., Hastie, T., and Tibshirani, R., [*The Elements of Statistical Learning*], vol. 1, Springer series in Statistics Springer, Berlin (2001).
2. Arthur, D. and Vassilvitskii, S., "k-means++: The advantages of careful seeding," in [*Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*], 1027–1035, Society for Industrial and Applied Mathematics (2007).
3. Park, H.-S. and Jun, C.-H., "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications* **36**(2), 3336–3341 (2009).
4. Hartigan, J., "Statistical theory in clustering," *Journal of Classification* **2**(1), 63–76 (1985).

5. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., "A density-based algorithm for discovering clusters in large spatial databases with noise.," in [*Kdd*], **96**, 226–231 (1996).
6. Fukunaga, K. and Hostetler, L., "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory* **21**(1), 32–40 (1975).
7. Comaniciu, D. and Meer, P., "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619 (2002).
8. Chacón, J., "Clusters and water flows: a novel approach to modal clustering through morse theory," *arXiv preprint arXiv:1212.1384* (2012).
9. Rodriguez, A. and Laio, A., "Clustering by fast search and find of density peaks," *Science* **344**(6191), 1492–1496 (2014).
10. Genovese, C., Perone-Pacifico, M., Verdinelli, I., and Wasserman, L., "Non-parametric inference for density modes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **78**(1), 99–126 (2016).
11. Ng, A., Jordan, M., and Weiss, Y., "On spectral clustering: Analysis and an algorithm," in [*NIPS*], **14**, 849–856 (2001).
12. Von Luxburg, U., "A tutorial on spectral clustering," *Statistics and Computing* **17**(4), 395–416 (2007).
13. Benediktsson, J. and Kanellopoulos, I., "Classification of multisource and hyperspectral data based on decision fusion," *IEEE Transactions on Geoscience and Remote Sensing* **37**(3), 1367–1377 (1999).
14. Dalponte, M., Bruzzone, L., and Gianelle, D., "Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas," *IEEE Transactions on Geoscience and Remote Sensing* **46**(5), 1416–1427 (2008).
15. Swatantran, A., Dubayah, R., Roberts, D., Hofton, M., and Blair, J., "Mapping biomass and stress in the sierra nevada using lidar and hyperspectral data fusion," *Remote Sensing of Environment* **115**(11), 2917–2930 (2011).
16. Cloninger, A., Czaja, W., and Doster, T., "Operator analysis and diffusion based embeddings for heterogeneous data fusion," in [*Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*], 1249–1252, IEEE (2014).
17. Czaja, W., Doster, T., and Murphy, J., "Wavelet packet mixing for image fusion and pan-sharpening," in [*Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*], **9088**, 908803, International Society for Optics and Photonics (2014).
18. Czaja, W., Manning, B., McLean, L., and Murphy, J., "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) (2016*).
19. Cloninger, A., Czaja, W., and Doster, T., "The pre-image problem for Laplacian Eigenmaps utilizingl^1 regularization with applications to data fusion," *Inverse Problems* **33**(7), 074006 (2017).
20. Murphy, J. and Maggioni, M., "Nonlinear unsupervised clustering of hyperspectral images, with applications to anomaly detection and active learning," *arXiv preprint arXiv:1704.07961* (2017).
21. Coifman, R. R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., and Zucker, S., "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).
22. Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., and Zucker, S., "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).
23. Coifman, R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis* **21**(1), 5–30 (2006).
24. Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I., "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis* **21**(1), 113–127 (2006).
25. Coifman, R., Kevrekidis, I., Lafon, S., Maggioni, M., and Nadler, B., "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *Multiscale Modeling & Simulation* **7**(2), 842–864 (2008).
26. Singer, A. and Coifman, R., "Non-linear independent component analysis with diffusion maps," *Applied and Computational Harmonic Analysis* **25**(2), 226–239 (2008).

27. Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation* **15**(6), 1373–1396 (2003).
28. Szlam, A., Maggioni, M., and Coifman, R., "Regularization on graphs with function-adapted diffusion processes," *Journal of Machine Learning Research* **9**, 1711–1739 (2008).
29. Lafon, S., Keller, Y., and Coifman, R., "Data fusion and multicue data matching by diffusion maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(11), 1784–1797 (2006).
30. Czaja, W., Manning, B., McLean, L., and Murphy, J., "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).
31. Lederman, R. and Talmon, R., "Learning the geometry of common latent variables using alternating-diffusion," *Applied and Computational Harmonic Analysis* (2015).
32. Lederman, R., Talmon, R., Wu, H., Lo, Y.-L., and Coifman, R., "Alternating diffusion for common manifold learning with application to sleep stage assessment," in [*Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*], 5758–5762, IEEE (2015).
33. Rohrdanz, M., Zheng, W., Maggioni, M., and Clementi, C., "Determination of reaction coordinates via locally scaled diffusion map," *The Journal of Chemical Physics* **134**(12), 03B624 (2011).
34. Zheng, W., Rohrdanz, M., Maggioni, M., and Clementi, C., "Polymer reversal rate calculated via locally scaled diffusion map," *The Journal of Chemical Physics* **134**(14), 144109 (2011).
35. Beygelzimer, A., Kakade, S., and Langford, J., "Cover trees for nearest neighbor," in [*International Conference on Machine Learning*], 97–104 (2006).
36. Sheather, S. and Jones, M., "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society. Series B (Methodological)* , 683–690 (1991).
37. Maggioni, M. and Murphy, J., "Learning by unsupervised nonlinear diffusion," (2018).
38. Fauvel, M., Benediktsson, J., Chanussot, J., and Sveinsson, J., "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing* **46**(11), 3804–3814 (2008).
39. Li, J., Bioucas-Dias, J., and Plaza, A., "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).
40. Zhang, H., Zhai, H., and Li, L. Z. P., "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing* **54**(6), 3672–3684 (2016).
41. Tarabalka, Y., Benediktsson, J., and Chanussot, J., "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing* **47**(8), 2973–2987 (2009).
42. Benedetto, J., Czaja, W., Dobrosotskaya, J., Doster, T., Duke, K., and Gillis, D., "Integration of heterogeneous data for classification in hyperspectral satellite imagery," in [*Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII*], **8390**, 839027, International Society for Optics and Photonics (2012).
43. Fauvel, M., Tarabalka, Y., Benediktsson, J., Chanussot, J., and Tilton, J., "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE* **101**(3), 652–675 (2013).
44. Cahill, N., Czaja, W., and Messinger, D., "Schroedinger eigenmaps with nondiagonal potentials for spatial-spectral clustering of hyperspectral imagery," in [*Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*], **9088**, 908804, International Society for Optics and Photonics (2014).
45. Wang, Z., Nasrabadi, N., and Huang, T., "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).
46. Benedetto, J., Czaja, W., Dobrosotskaya, J., Doster, T., and Duke, K., "Spatial-spectral operator theoretic methods for hyperspectral image classification," *GEM-International Journal on Geomathematics* **7**(2), 275–297 (2016).

47. Mercado, P., Tudisco, F., and Hein, M., "Clustering signed networks with the geometric mean of Laplacians," in [*Advances in Neural Information Processing Systems*], 4421–4429 (2016).
48. Mercado, P., Gautier, A., Tudisco, F., and Hein, M., "The power mean Laplacian for multilayer graph clustering," *arXiv preprint arXiv:1803.00491* (2018).