IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium | 978-1-7281-6374-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IGARSS39084.2020.9323091

PATCH-BASED DIFFUSION LEARNING FOR HYPERSPECTRAL IMAGE CLUSTERING

James M. Murphy [1]
[1]Tufts University, Department of Mathematics, Medford, MA 02155, USA

ABSTRACT

An algorithm for clustering hyperspectral images (HSI) based on diffusion geometry in the space of high-dimensional image patches is proposed. By using the patch structure of the HSI, robustness to noise is achieved in the clustering process. Results on real hyperspectral data indicate the effectiveness of working in the space of HSI patches, compared to working in the space of HSI pixels.

Index Terms— Hyperspectral images, unsupervised learning, clustering, image patches, diffusion geometry

1 INTRODUCTION

As remote sensors capture increasing quantities of high-dimensional hyperspectral imagery (HSI), the need to develop efficient machine learning algorithms for these data streams grows apace. The data captured is usually unlabeled, and it is infeasible for humans to produce sufficient labeled annotations to make use of traditional supervised learning on the entirety of this deluge of HSI data. There is consequently a pertinent need for efficient unsupervised and semisupervised machine learning algorithms to glean insight from these data streams.

This paper proposes a clustering algorithm for HSI based on diffusion geometry in the space of image patches. This method, a significant extension of the *diffusion learning* (DL) framework for HSI clustering [1, 2, 3, 4], gains robustness to noise and outliers by making comparisons not between individual pixels, but between averages across spatial patches of pixels. This is of particular value for HSI, which are often noisy and suffer from corruption due to poor atmospheric conditions. The proposed *spectral-spatial patch diffusion learning (DLSSP)* algorithm captures the intrinsic geometry of patch space, and efficiently labels all pixels in a manner that scales quasilinearly in the number of pixels in the image.

The structure of the rest of this article is as follows. Section 2 presents background on spaces of image patches and diffusion geometry. The proposed algorithm is presented in Section 3. Experimental results are given in Section 4, and conclusions and future work are discussed in Section 5.

2 BACKGROUND

In order to capture the latent structure of an HSI, we consider the *diffusion geometry [5, 6] of patch space [7, 8].* Let $X=\{x_i\}_{i=1}^n\subset\mathbb{R}^D$ be a point cloud representation of an $M\times N\times D$ hyperspectral image with $n=MN$ pixels and D spectral bands. For $p\in\{1,2,\dots\}$, let X_p be the space of $p\times p$ square image patches of X. Owing to size restrictions near the border, X_p has slightly fewer elements than X, but is substantially higher dimensional—X is D-dimensional, while X_p is p^2D dimensional. In this article, we perform analysis on X_p in order to leverage the smoothing properties of working in patch space.

Indeed, once the patches are constructed, we can *denoise* and *cluster* by applying a variant of the *non-local means denoising* (NLMD) algorithm [7, 8] followed by the *spectral-spatial diffusion learning* (DLSS) algorithm [2, 4]. More precisely, for $x\in X_p$, let $NN_k(x)$ be the set of k-nearest neighbors of x in X_p. We denoise X_p by replacing x with its local average over $NN_k(x)$ in patch space: $\tilde{x}=\frac{1}{k}\sum_{y\in NN_k(x)}y$. This has the effect of denoising x in a manner that respects the spatial properties of the local neighborhood around x.

Once the patch geometry has been used to smooth the data, the DLSS algorithm [2] is used to cluster it. At a high level, the DLSS algorithm learns modes in the underlying data that can be used to propagate labels to the remaining data points. In order to learn modes in a manner that is robust to the high dimensionality of the data and the nonlinear structure of its latent clusters, *diffusion distances* [5, 6] are used to make pairwise comparisons between pixels. More precisely, for any dataset

This research was partially supported by the U.S. National Science Foundation grants DMS 1912737 and DMS 1924513.

$\{x_i\}_{i=1}^n = X \subset \mathbb{R}^D$, let $\mathcal{G} = (X, W)$ be a weighted graph on X with symmetric weight matrix $W \in [0,1]^{n\times n}$; W is commonly constructed as $W_{ij} = \mathcal{K}(x_i, x_j)$ for an appropriate kernel function $\mathcal{K}$. The weights W generate a diffusion process on X by normalizing the rows of W to sum to 1. Indeed, let $P = D^{-1}W$, where D is the diagonal degree matrix with $D_{ii} = \sum_{j=1}^n W_{ij}$. Then as long as $\mathcal{G}$ is connected and aperiodic, P is an irreducible Markov matrix with a unique stationary distribution $\pi \in \mathbb{R}^{1\times n}$ satisfying $\pi P = \pi$. The diffusion distance between $x_i, x_j \in X$ at time $t \in [0, \infty)$ is

$$D_t(x_i, x_j) = \sqrt{\sum_{\ell=1}^{n} (P_{i\ell}^t - P_{j\ell}^t)^2 \frac{1}{\pi_\ell}}.$$

Intuitively, the diffusion distance between x_i and x_j at time t is small if the the diffusion process at time t prescribes similar transition profiles to x_i and x_j, that is, if the i^{th} and j^{th} rows of P^t are similar.

Diffusion distances are known to capture low-dimensional, nonlinear structure in X even when D is large [9, 10]. Unfortunately, computing $D_t(x_i, x_j)$ for fixed x_i, x_j has complexity $O(n)$, which is prohibitive for n large. However, one can compute the eigenvalues and right eigenvectors of P, $\{(\lambda_\ell, \psi_\ell)\}_{\ell=1}^n$, and show [6] that

$$D_t(x_i, x_j) = \sqrt{\sum_{\ell=1}^{n} \lambda_\ell^{2t} ((\psi_\ell)_i - (\psi_\ell)_j)^2}.$$

Moreover, when the underlying data is intrinsically low-dimensional, P is approximately low rank and a small subset of the spectrum of P is sufficient to well-approximate diffusion distances. Indeed, suppose without loss of generality that the eigenvalues of P (and their corresponding right eigenvectors) are sorted so that $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \geq 0$. Then for some $M \ll n$, we may truncate the spectral expansion to approximate diffusion distances:

$$D_t(x_i, x_j) \approx \sqrt{\sum_{\ell=1}^{m} \lambda_\ell^{2t} ((\psi_\ell)_i - (\psi_\ell)_j)^2}.$$

3 THE DLSSP ALGORITHM

The proposed DLSSP algorithm for clustering the HSI $\{x_i\}_{i=1}^n = X \subset \mathbb{R}^D$ proceeds in two major steps. First, the set of $p \times p$ patches in X, denoted X_p, is computed and smoothed according to the NLMD algorithm [7]. The collection of denoised patches $\tilde{X}_p = \{\tilde{x}\}_{x\in X_p}$ implicitly denoises the original HSI X by projecting an image patch onto its center coordinate; let $\tilde{X}$ be the image of $\tilde{X}_p$ under such a projection. To mitigate the very high dimensionality of patch space (p^2D), nearest neighbor calculations are performed on the projection of the patch space data onto its first d principal components. The construction of $\tilde{X}$ is detailed in Algorithm 1.

Algorithm 1: HSI Smoothing with NLMD

```
Input: $X$; $p, k, d$.
  1: Compute $X_p \subset \mathbb{R}^{p^2 D}$ the set of $p \times p$ patches in the HSI $X$.
  2: Project $X_p$ onto its first $d$ principal components, call the result $X_{p,d} \subset \mathbb{R}^d$.
  3: For each $x \in X_{p,d}$, compute the $k$-nearest neighbors $NN_k(x)$ in $X_{p,d}$.
  4: For each $x \in X_{p,d}$, compute the denoised patch $\tilde{x} = \sum_{y\in NN_k(x)} y$.
  5: Let $\tilde{X} \subset \mathbb{R}^d$ be the projection of the denoised patches onto their center pixel.
Output: $\tilde{X}$.
```

The second major stage of DLSSP is to cluster the smoothed HSI $\tilde{X}$ using DLSS [2]. This consists of detecting modes in the data as points that are both high-density and far in diffusion distance from other points of high density. Indeed, let $p : \tilde{X} \to (0,1)$ be a kernel density estimator for $\tilde{X}$; we use a Gaussian kernel on nearest neighbors with an adaptive scaling parameter. Let

$$\rho_t(x) = \begin{cases} \min\limits_{\{p(x_i)\geq p(x)\}} D_t(x_i, x), & x \neq \arg\max_i p(x_i) \\ \max_{x_i} D_t(x_i, x), & x = \arg\max_i p(x_i) \end{cases}$$

be the diffusion distance of each point to its D_t-nearest neighbor of higher density. The modes of the data are determined as the maximizers of $\mathcal{D}_t(x) = p(x)\rho_t(x)$; the $p(x)$ factor ensures that outlier points are not estimated as modes. The mode detection algorithm is summarized in Algorithm 2.

Once the modes are detected, points are labeled iteratively in a manner that preserves spatial regularity. The key notion to this labeling process is that of *spatial consensus neighbor*, which is the same as the most common label in a spatial neighborhood around a pixel, where both labeled and unlabeled pixels vote. The labeling procedure is summarized in Algorithm 3; details

Algorithm 2: Mode Detection Algorithm

```
Input: $\tilde{X}, K; t$.
    1: Compute $\{p(x_i)\}_{i=1}^n$, $\{\rho_t(x_i)\}_{i=1}^n$.
    2: Compute $\{x_k^*\}_{k=1}^K$, the $K$ maximizers of
       $\mathcal{D}_t(x_i) = p(x_i)\rho_t(x_i)$.
Output: $\{x_k^*\}_{k=1}^K, \{p(x_i)\}_{i=1}^n, \{\rho_t(x_i)\}_{i=1}^n$.
```

may be found in [2].

Algorithm 3: Spectral-Spatial Labeling Algorithm

```
Input: $\{x_k^*\}_{k=1}^K, \{p(x_i)\}_{i=1}^n$, $\{\rho_t(x_i)\}_{i=1}^n$
    1: Assign each mode a unique label.
    2: Iterating through the remaining unlabeled points in
       order of decreasing density among unlabeled
       points, assign each point the same label as its
       $D_t$-nearest spectral neighbor of higher density,
       unless the spatial consensus label exists and differs,
       in which case the point is left unlabeled.
    3: 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 $\{y_i\}_{i=1}^n$.
```

3.1 Computational Complexity

The proposed algorithm appears on first consideration to be highly penalized by the curse of dimensionality: even for small 3×3 patches, working in patch space increases the dimensionality (already very high) of the HSI by nearly an order of magnitude. However, the use of principal component analysis to project the p^2D-dimensional patch space into d dimensions significantly mitigates this complexity. Indeed, computing the projection onto the first d principal components has complexity $O(d^2p^4D^2 + np^4D^2)$, which is $O(nD^2)$ when $p, d = O(1)$. Once the patch-space has been projected onto the first d principal component directions, the subsequent application of the DLSS algorithm is, under mild assumptions [11, 2] $O(C^{d'} d \log^2(n) n)$, where $d' \leq d$ is the intrinsic dimension of the projected patch space data. This gives a total cost of $O(n(D^2 + C^{d'} \log^2(n)))$ when $d = O(1)$ with respect to n, D. Crucially, the dependence on n is quasilinear, so that the proposed method scales to large datasets. Code implementing DLSSP is publicly available [1].

4 EXPERIMENTAL RESULTS

We perform experiments on the Indian Pines dataset[2]; experimental data is shown in Fig. 1. The proposed DLSSP method is tested across a range of patch sizes p, and is also compared to the DLSS algorithm, which may be understood as the DLSSP algorithm in the special case $p = 1$. Results appear in Fig. 2, where labels for $p = 5$ are shown, along with the overall accuracy (OA), average accuracy (AA), and Cohen's κ score for DLSSP as a function of p; these accuracy metrics are formally defined in [2]. We see that when $p = 5$ is used, DLSSP strongly outperforms DLSS, with DLSSP achieving results of OA $= 0.4711$, AA $= 0.3846$, $\kappa = 0.4031$, compared to OA $= 0.4286$, AA $= 0.2243$, $\kappa = 0.3141$ for DLSS.

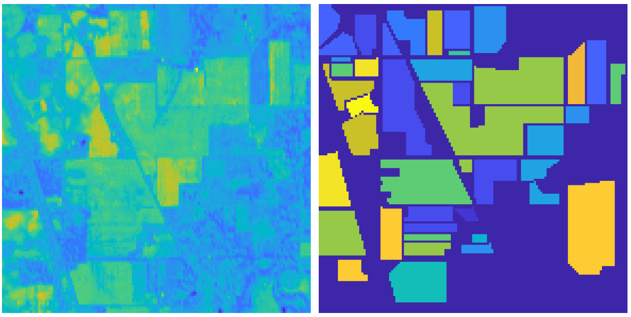

Fig. 1. The Indian pines dataset was recorded in 1992 in Northwest IN, USA by the AVRIS sensor. It has spatial dimensions 145×145 for a total of $n = 21025$ pixels, has spatial resolution 20m/pixel, and has spectral dimension $D = 200$. It contains 16 classes of varying sizes. *Left:* Sum of first 10 principal components. *Right:* ground truth labels.

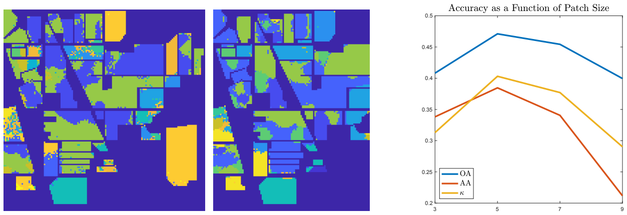

Fig. 2. Results on the Indian Pines data. *Left:* DLSS results. *Middle:* DLSSP results. *Right:* Impact of the patch size on DLSSP clustering accuracy.

We see that as a function of p, the proposed method

[1] https://jmurphy.math.tufts.edu/Code/

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

first improves, hitting an optimal value at $p = 5$, before declining. This can be interpreted in terms of regularization—the large p is, the more emphasis on the spatial regularization coming from considering spatial patches.

5 CONCLUSIONS & FUTURE DIRECTIONS

The results of the DLSSP algorithm on the Indian Pines dataset suggest its value for clustering HSI. While there is a computational cost to working in patch space (dimension D data becomes p^2D dimensional), projecting onto a small number of principal components before performing nearest neighbor searches and constructing the diffusion process on the data reduce this cost significantly.

It is of interest to develop unsupervised methods that involve clustering directly on the space of patches. There is a danger that such methods would be overwhelmingly influenced by spurious geometric properties of the patches (e.g. the orientation of edges), but such approaches may be feasible if a notion of distance on the space of patches that is rotation invariant is incorporated. It is also of interest to extend the proposed unsupervised framework to the semisupervised setting of *active learning*, where diffusion methods have shown strong theoretical and empirical results [12, 13, 14].

6 References

[1] J.M. Murphy and M. Maggioni, "Diffusion geometric methods for fusion of remotely sensed data," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIV*. International Society for Optics and Photonics, 2018, vol. 10644, p. 106440I.

[2] J.M. Murphy and M. Maggioni, "Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1829–1845, 2019.

[3] J.M. Murphy and M. Maggioni, "Spectral-spatial diffusion geometry for hyperspectral image clustering," *arXiv preprint arXiv:1902.05402*, 2019.

[4] M. Maggioni and J.M. Murphy, "Learning by unsupervised nonlinear diffusion," *Journal of Machine Learning Research*, vol. 20, no. 160, pp. 1–56, 2019.

[5] 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*, vol. 102, no. 21, pp. 7426–7431, 2005.

[6] R.R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[7] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2005, vol. 2, pp. 60–65.

[8] A.D. Szlam, M. Maggioni, and R.R. Coifman, "Regularization on graphs with function-adapted diffusion processes," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1711–1739, 2008.

[9] P.W. Jones, M. Maggioni, and R. Schul, "Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels," *Proceedings of the National Academy of Sciences*, vol. 105, no. 6, pp. 1803–1808, 2008.

[10] N.G. Trillos, M. Gerlach, M. Hein, and D. Slepčev, "Error estimates for spectral convergence of the graph laplacian on random geometric graphs toward the laplace–beltrami operator," *Foundations of Computational Mathematics*, pp. 1–61, 2019.

[11] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," in *Proceedings of the international conference on Machine learning*. ACM, 2006, pp. 97–104.

[12] J.M. Murphy and M. Maggioni, "Iterative active learning with diffusion geometry for hyperspectral images," in *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2018, pp. 1–5.

[13] M. Maggioni and J.M. Murphy, "Learning by active nonlinear diffusion," *Foundations of Data Science*, vol. 1, no. 3, pp. 271–291, 2019.

[14] J.M. Murphy, "Spatially regularized active diffusion learning for high-dimensional images," *Pattern Recognition Letters*, vol. 135, pp. 213–220, 2020.