A spectral graph based approach to analyze hyperspectral data

Blake Hunter
Department of Mathematics, UCLA
520 Portola Plaza
Los Angeles, CA 90095
Email: blakehunter@math.ucla.edu

Yifei Lou
Department of Mathematics, UCI
340 Rowland Hall
Irvine, CA 92697
Email: ylou1@uci.edu

Andrea L. Bertozzi
Department of Mathematics, UCLA
520 Portola Plaza
Los Angeles, CA 90095
Email: bertozzi@math.ucla.edu

Abstract—Hyperspectral imaging has emerged as a promising tool in the identification of objects and the state of objects, by their chemical and material composition. Hyperspectral imaging acquires spectral information at each pixel location across a wide range of the light spectrum. This enhanced spectrum information also comes with additional noise including spectral mixing, blurring and acquisition distortions. The analysis and processing of this high dimensional data requires efficient specialized techniques. We discuss a new novel graph based method for dimension reduction, image segmentation and classification based on the Ginzburg-Landau functional from classical PDE minimization. It aims to efficiently preserve as much spectral and structural information as possible. We show results from a field test of a wide field of view imaging spectrometer (WFIS) high performance hyperspectral imager designed for atmospheric chemistry and aerosols measurement from aircrafts and satellites.

I. Introduction

The hyperspectral data is obtained by a Wide Field-of-view Imaging Spectrometer (WFIS) [1], from Hamilton Sundstrand[1], a aerospace systems company. The imager is a high-performance pushbroom hyperspectral imager designed for atmospheric chemistry and aerosols measurement from an aircraft or satellite. Unlike the conventional hyperspectral dataset, WFIS produces images with high spatial resolution and a large number of wave bands. The spatial resolution is 350×1024, and the image cube contains 2048 bands. Three sample bands (image slices) are shown in Figure 1. In this paper, we apply two types of classification methods on this dataset; one is spectral clustering and the other is semi-supervised image segmentation.

To segment the image at a pixel-by-pixel level, we represent each pixel as a 2048-dimensional feature vector. This takes into account the complete observed spectrum and allows for measurements of differences across multiple bands. From the data, we construct a weighted graph whose vertices are the image pixels and edges are defined by the similarity between any two pixels. The similarity weight is computed based on a distance metric. Euclidean L_2 distance is commonly used in image processing for grayscale and 3 dimensional color images. As for higher dimensional images such as hyperspectral, the spectral angle (SA) [2] is often more appropriate.

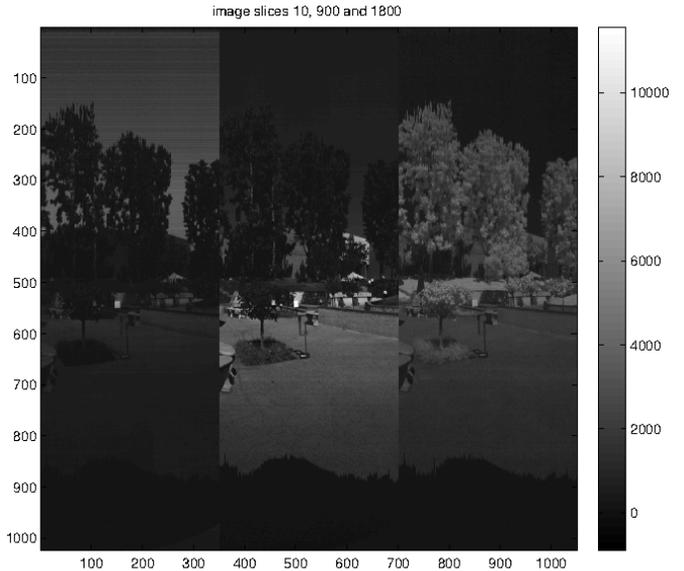[1]http://www.hamiltonsundstrand.com

Fig. 1. Three sample image slices of the hyperspectral image, slices 10, 900 and 1800.

The rest of the paper is organized as follows. In Section II-A, we discuss in details about the construction of the weighted graph with the choice of L^2 or SA as the distance metric. Based on the graph representation of the data, we apply two segmentation methods. Section II-B is devoted to unsupervised segmentation, or clustering, using spectral clustering. The second method in Section II-C is a semi-supervised binary classification using a diffuse interface model. Followed bt the conclusion is given in Section III.

II. Image Segmentation/Classification

A. Graph Representation of the Data

We represent each data point (each pixel of the hyperspectral image) as a vertex in a graph, which lies in a high dimensional space (each of length 2048). The edge weights are defined to be the similarity between two points, *i.e.*, a similarity weight
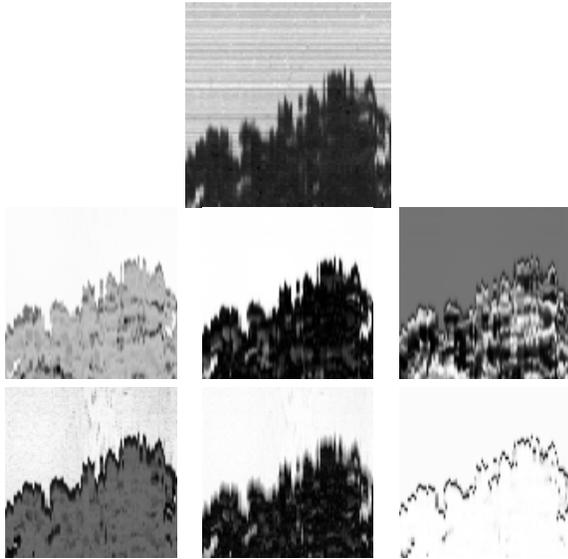
Fig. 2. Top: the original image (only one spectral band shown) with two groups of pixels: tree and sky. From left to right: the first three eigenvectors of the weight W by using L^2 (the second row) or using SA (the last row).
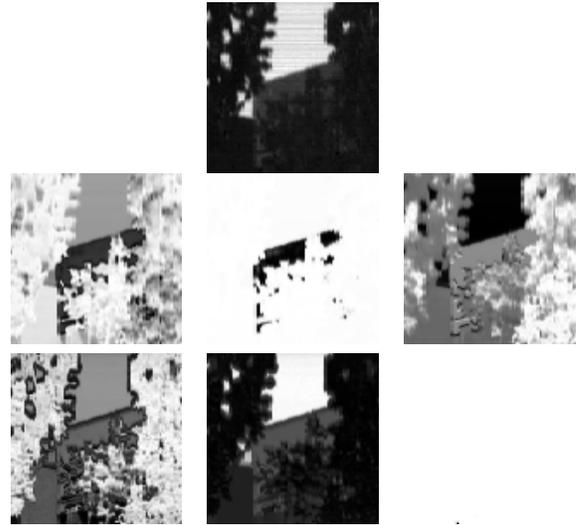

Fig. 3. Top: the original image (only one spectral band shown) with three groups of pixels: sky, building and tree. From left to right: the first three eigenvectors of the weight W by using L^2 (the second row) or using SA (the last row).

matrix $W \in \mathbb{R}^{n \times n}$ is computed as follows,

$$W_{i,j} = \exp\{-\frac{d(x_i, x_j)^2}{2\sigma^2}\} . \qquad (1)$$

The Gaussian kernel is a natural choice that describes the diffusion of heat or information through the data set (from pixel to pixel across the image). We consider two choices for the distance metric $d(x_i, x_j)$. One is the standard Euclidean L^2 distance,

$$d(x_i, x_j) = \|x_i - x_j\|_2. \qquad (2)$$

The other is the spectral angle, defined as

$$d(x_i, x_j) = \arccos \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}. \qquad (3)$$

Spectral embedding uses the eigenvectors of the adjacency matrix, the normalized weight matrix. Figure 2 and Figure 3 show the eigenvectors of the weight matrix W constructed by L^2 and then SA on two subimages from the raw data. Figure 2 shows the image with two groups of pixels, sky and tree (the group of tree pixels also has a hieratical group structure made up of subgroups), while Figure 3 is with three groups of pixels, tree, building, and sky points (each made up of their own hieratical group structure). Using either L^2 or SA, the eigenvectors naturally represent differences in pixel group structures in the data. For the sky-tree image, the third eigenvector using L^2 starts to select the structures within the tree, while the one by SA appears to choose the boundaries between two groups. In this sense, SA is better at distinguishing features of the same class than L^2.

B. Spectral Clustering

Spectral clustering is a popular and powerful modern clustering algorithm. It projects the data onto eigenvectors of a weighted graph, defined from local similarity, to enhance natural separation in the data. For a complete analysis and discussion of the theory behind spectral clustering see [3] by von Luxburg and [4] by Ng, Jordan, and Weiss.

Given a set of data $X = x_1, x_2, \ldots, x_n$, we want to cluster into k groups, where k is *a priori*. The algorithm goes as follows,

1) Construct a weight matrix $W \in \mathbb{R}^{n \times n}$ from Section II-A. Normalize each row of W to have unit length.
2) Find the top m largest eigenvectors of W, $v_1, v_2, \cdots, v_m$, and form the matrix $V = [v_1|v_2|...|v_m] \in \mathbb{R}^{n \times m}$ (We use the Nystrom method to approximates the eigenvectors of W from a small sub-sampled weight matrix.).
3) Each row of V as a point in $\mathbb{R}^m$, called the spectral coordinate, which are clustered into k clusters using k-means or other clustering methods.
4) Assign the original point x_i to cluster j if and only if row i of the matrix V was assigned to cluster j.

Figure 4 shows the results for spectral clustering using k-means. For the sky-tree example (the top row of Figure 4), the two clusters are mapped into two colors, black and white (the order has no meaning). No matter which distance is used to construct the weight, the two clusters are identified correctly here. Similarly on the bottom row of Figure 4, the three clusters found are mapped to three colors values, black, gray and white (again the order is arbitrary). In this example, SA gives more plausible clustering results than L^2.

Spectral clustering is an unsupervised method which can be extended to semi-supervised by adding partially labeled data. An advanced extension in this direction using the Ginzburg-Landau functional from classical PDE minimization, is the diffuse interface model.
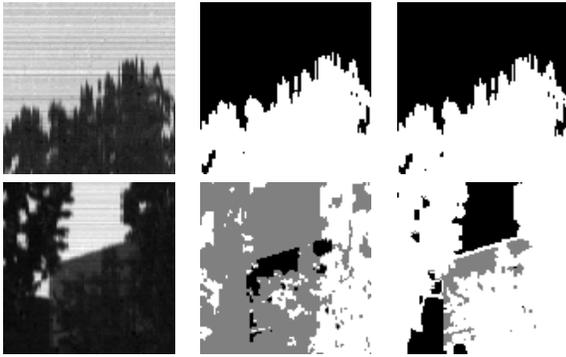
Fig. 4. Results of spectral clustering using 100 eigenvectors of the weight computed by L^2 (middle column) or SA (right column). The first column shows the original data. SA gives more plausible clustering results than L^2.

C. Diffuse Interface Model

The weighted graph is also used in the semi-supervised classification (SSC). Given the labels on a set of the vertices on a graph, the goal of SSC is to label the entire set of data points. Bertozzi and Flenner [5] formulate a binary segmentation based on the Ginzburg-Landau functional on the graph with a fidelity term. In particular, the following energy functional is minimized,

$$\hat{u} = \min_u \epsilon \langle u, Lu \rangle + \frac{1}{2\epsilon} \sum_{x \in X} (u^2(x)-1)^2 + \sum_{x \in X} \lambda(x)(f(x)-u(x))^2 .$$

(4)

The first term is a graph Laplacian, which is obtained by normalizing the matrix $D - W$ to be symmetric, *i.e.*,

$$L = Id - D^{-1/2}WD^{-1/2}$$

(5)

with D be the degree matrix. If we normalize each row of the weight W to sum up to one, then D is an identity matrix. The graph Laplacian L is an analogue to the standard Laplacian $\triangle$ for a function. The second term in (4) is a double well potential, which forces u to go to one of two classes, with value either one or negative one. The last term is a fidelity term for semi-supervised classification, where $\lambda(x)$ is set to 1 for the pixels with known labels and zero elsewhere. The Ginzburg-Landau fucntional minimizes a weighted graph cut between two regions. Thus the minimizer of (4), $\hat{u}$, approximately takes value one on a set of vertices and negative one on another with a smooth transition between them. The class labels are then determined by thresholding $\hat{u}$.

With the hand labeled image, the function u is initialized to one for one of the classes and negative one for the other. See the first column of Figure 5 for examples. In particular, the initial value of u is one on the rectangular region, and negative one on the rest. The graph Laplacian is constructed using (5) with the choice of metric, either L^2 (2) or SA (3). Then the optimal solution u of (4) is obtained by a convex splitting scheme. Please refer to [5] for more details. In the experiments we set the parameters as follows: $\epsilon = 1$, $c = 100$ and 400 iterations. For an image of size 100×100, we used a hand labeled 10x10 sub-image, the computational time in

Matlab for the weight matrix is 80 seconds, and computing the GL-min is about 60 seconds. The results are given in Figure 5. In the sky-tree example (the first row of Figure 5), the method is able to evolve the rectangle region with known label, "tree", to identify the entire tree region in the image. The weight using either L^2 or SA can produce reasonable labels, while SA gives more plausible segments than using L^2. The second and the third rows of Figure 5 deal with the same image, but with the different known regions: which are tree for the second row, and building for the third row. Again the choice of SA for similarity outperforms L^2 here.
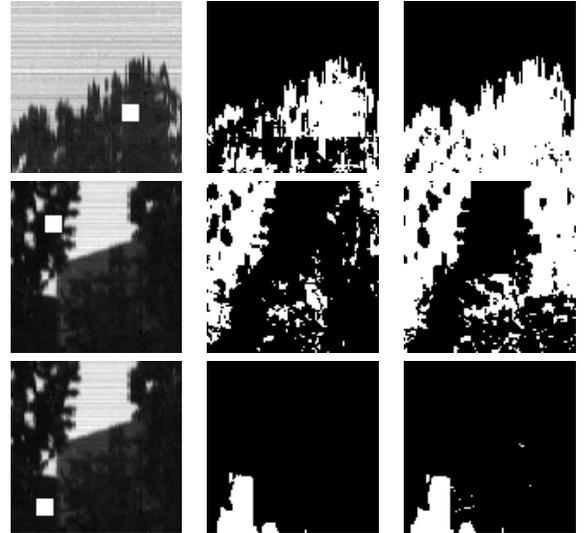

Fig. 5. Results of Ginzburg-Landau minimization of using L^2 (middle column) or SA (right column) to construct the weight W. The first column shows the original images with training patches, indicated by white rectangles.

III. CONCLUSION

This paper discusses graph-based classification methods on hyperspectral data. The difficulty of this new dataset obtained by WFIS is its huge dimensionality. We apply two classification methods on several 100×100 subimages, spectral clustering, a unsupervised multilabel classification method and the other uses Ginzburg-Landau minimization to achieve semi-supervised binary segmentation. Both methods are based a fully connected graph representation of the data, in which the weight is computed by using either L^2 or spectral angle as the distance metric. Experiments show that SA always give better results than L^2.

One future direction is to deal with the data of its original size. It takes about 3GB to store the data of size $350 \times 1024 \times 2048$, which causes a huge burden on the storage. A simple solution is to divide the entire data into thirty $100 \times 100 \times 2048$ sub-blocks and combine them together. Another alternative is to construct a very sparse weight matrix W. In this regard, it is helpful to investigate on computing eigenvector of a sparse matrix, such as [6]. Another possibilty is to use the Nystrom extension as recently applied in [7]. Also L^2 is a better choice to compute the weight, as it can produce a sparse weight by

using a relatively small σ in (1). SA, on the other hand, is bounded by 0 and 1, so the weight is either full or zero matrix.

REFERENCES

[1] R. Haring, R. Pollock, R. Cross, and B. Sutin, "Field testing the wide-field-of-view imaging spectrometer(wfis)," in *Proceedings of SPIE*, vol. 5543, 2004, pp. 283–292.

[2] F. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image-processing system (sips) - interactive visualization and analysis of imaging spectrometer data," *Remote Sensing of Environment*, vol. 2-3, no. 44, pp. 145–163, 1993.

[3] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems 14*, pp. 849–856, 2001.

[5] A. Bertozzi and A. Flenner, "Diffuse interface models on graphs for classification of high dimensional data," *Multiscale Modeling and Simulation*, vol. 10, no. 3, pp. 1090–1118, 2012.

[6] C. R. Anderson, "A Rayleigh-Chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse hermitian matrices," *J. Comput. Phys.*, vol. 229, no. 19, pp. 7477–7487, 2010.

[7] E. Merkurjev, T. Kostic, and A. L. Bertozzi, "An MBO scheme on graphs for segmentation and image processing," *submitted SIAM J. Imag. Proc.*, 2012.