Hyperspectral Imaging

Geoffrey Iyer

June 15, 2016

Contents

1 Introduction

A relatively recent development in remote sensing technology is the creation of hyperspectral cameras. Whereas the human eye primarily sees color in three wavelenghts (red, green, blue), a hyperspectral sensor collects data from a much larger range of the spectrum. Each wavelength collected is called a "band", and most hyperspectral sensors sample 100-200 bands, ranging from microwaves to UV rays. This data can be used to differentiate objects that would appear the same to a standard camera.

In this project we aim to perform unsupervised classification on hyperspectral images using a spectral clustering algorithm. This requires defining the Graph Laplacian matrix associated to our data (Section 2) and computing eigenvectors and eigenvalues of the matrix (Section 3). In Section 4 we show an application of our algorithm to a sample data set.

2 Graph Laplacian

We approach this problem via graph-based methods. A more detailed survey of the theory can be found in [6]. Here we state only the results necessary to implement our algorithm.

2.1 Similarity Graphs

We can represent a hyperspectral image using an undirected graph $G = (V, E)$. The nodes $v_i \in V$ of the graph correspond to the pixels in the image. We give each edge e_{ij} a *weight* $w_{ij} \geq 0$ representing the similarity between nodes v_i, v_j. This gives rise to a *similarity matrix* (also called the *weight matrix*)

$$W = (w_{ij})_{i,j=1}^{n}.$$

Since G is undirected, we require that $w_{ij} = w_{ji}$, which implies that W is a symmetric matrix. There are several popular means to measure "similarity" between nodes, which we shall discuss in more detail in 2.4.

2.2 Graph Min-Cut Problem

The problem of unsupervised clustering can be rephrased as a graph-cut-minimization problem of the similarity matrix W. Given a set $A \subseteq V$, we define the *graph-cut*

$$cut(A, A^c) = \frac{1}{2} \sum_{v_i \in A, v_j \in A^c} w_{ij}.$$

Or, more generally, for disjoint sets $A_1, A_2, \ldots, A_k$ such that $\cup_{j=1}^{k} A_j = V$ we define

$$cut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{v_i \in A_m, v_j \notin A_m} w_{ij}.$$

Each choice of sets $A_1, \ldots, A_k$ partitions our image into k classes. Finding an optimal $A_1, \ldots, A_k$ corresponds to minimizing the energy in the graph cut.

2.3 Definition of Graph Laplacian

Once we have formed the weight matrix W, we can use it to define the Graph Laplacian. For each node $v_i \in V$, define the *degree* of the node

$$d_i = \sum_j w_{ij}.$$

Intuitively, the degree represents the strength of a node. Let D be the diagonal matrix with d_i as the i-th diagonal entry. We then define the *unnormalized Graph Laplacian*

$$L = D - W,$$

and the *normalized Graph Laplacian*

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}.$$

For a thorough explanation of the properties of the Graph Laplacian, see [4]. In our algorithm we will use the following properties.

Theorem 2.1

- *For every $f \in \mathbb{R}^n$ we have*

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

- *L, L_{sym} are both symmetric and positive semidefinite*
- *The eigenvectors of L (or L_{sym}) corresponding to eigenvalues of small magnitude give an approximate solution to the graph min-cut problem posed in 2.2.*

2.4 Edge Weights and Sparsification

For $v_i, v_j \in V$, we define the weight of the edge e_{ij} via

$$w_{ij} = \exp(\|v_i - v_j\| / \tau).$$

Here τ is a scaling parameter, and $\|\cdot\|$ can be any method of measuring distance between nodes. In this project we considered both the Euclidean norm as well as the vector angle norm for our distance measure. Intuitively the vector angle norm (aka cosine similarity, aka angular distance) seems more relevant when working with hyperspectral data.

$$\begin{aligned} \|v_i - v_j\| &= \cos^{-1}\left(\frac{\langle x_i, x_j \rangle}{\|x_i\| \cdot \|x_j\|} \right). \\ &= \text{angle between } x_i, x_j \end{aligned}$$

This method ignores amplitude of a wave and focuses on shape, as can be seen in the example classification in Figures 1, 2, 3. However, when working with actual image we noticed very little difference between the two norms.

Using the method above will result in a dense Graph Laplacian matrix L, which is undesirable for finding eigenvalues and eigenvectors. Therefore we sparsify the matrix by thresholding relatively small weights to 0.

3 Spectral Clustering

Having created a Graph Laplacian matrix, we apply a spectral clustering algorithm from [5] to separate our data into classes.

Have $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ the smallest m eigenvalues of L. $f_1, \ldots, f_m$ the corresponding eigenvectors. For each pixel v_i we get a vector $y_i \in \mathbb{R}^m$ given by $y_i = ((f_1)_i, \ldots, (f_m)_i)$. Run k-means on the y_i to cluster the pixels v_i into classes $C_1, C_2, \ldots, C_k$.

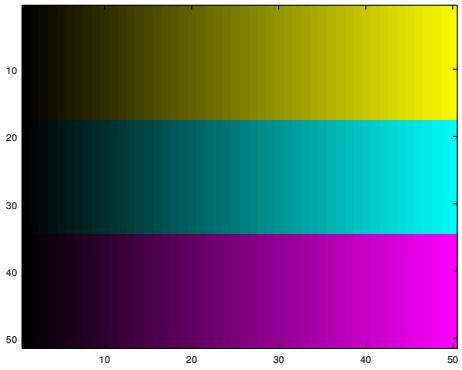

Figure 1: Original Image

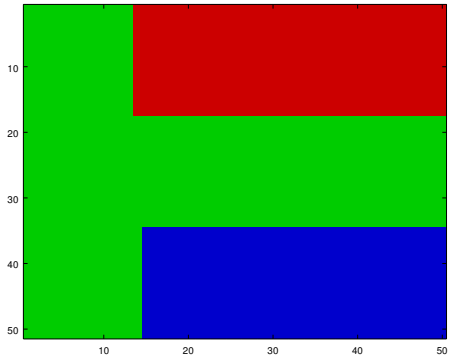

Figure 2: Classified with Euclidean Norm

Figure 3: Classified with Vector Angle

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

Table 1: 9 Classes of Pixel

3.1 Rayleigh-Chebyshev

To find the required eigenvectors and eigenvalues for spectral clustering we use the Rayleigh-Chebyshev method developed by Chris Anderson, explained in [7]. A simplified version of the algorithm is given below:

- If λ is an eigenvalue of A, and p is a polynomial, then $p(\lambda)$ an eval of $p(A)$.
- Pick p such that $p(\lambda)$ is large for λ near zero, and $p(\lambda)$ is small for λ far from zero.
  - Chebyshev polynomials have this property, hence the name of the algorithm.
- Apply the power method to $p(A)$ to find the eigenvectors corresponding to

$$\{\text{eigenvalues } \lambda \mid |\lambda| \text{ relatively small}\}.$$

4 Application to Pavia University Dataset

To test our algorithm, we use the a scene of Pavia University, Pavia, Italy[1] taken by the ROSIS sensor[2]. The image is 610×340 pixels, where each pixel represents roughly $1.3m^2$ of surface area. For each pixel, 103 different wavelengths are captured, covering values of the electromagnetic spectrum from microwaves to UV. Figures 5, 6, are example bands chosen from the full 103. They are placed alongside a grayscale image of the university (Figure 4) for easy comparison.

This dataset also comes with a hand-labeled "ground truth" (Figure 7, Table 4) which we will compare against our clustering results. The ground truth divides the image into 9 classes.

Unfortunately, our algorithm was not efficient enough to classify the entire Pavia University image, so instead we restrict our attention to a 100×100 sub-picture.

4.1 Sub-Picture

To test our clustering algorithm, we chose a section near the center of the image that contains a building, surrounded by a meadow. Figures 8, 9, 10, 11 show a grayscale image, a selected

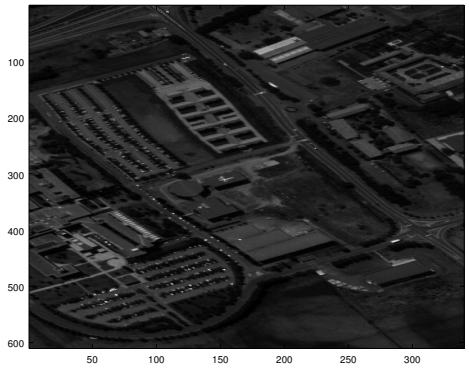

Figure 4: Grayscale Image

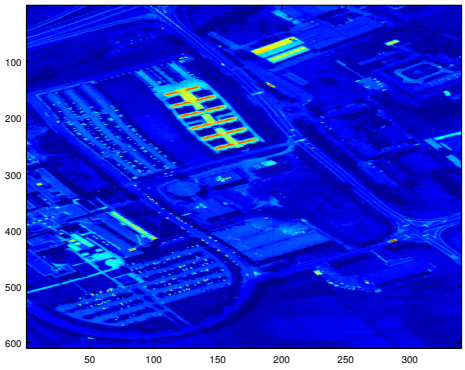

Figure 5: Low Wavelength

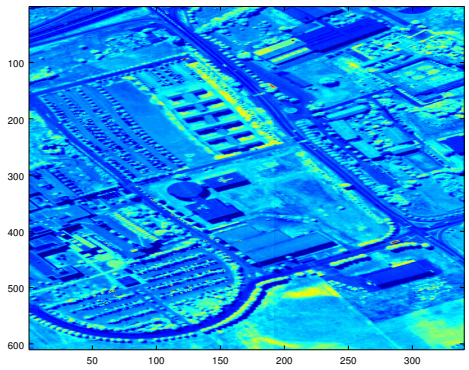

Figure 6: High Wavelength

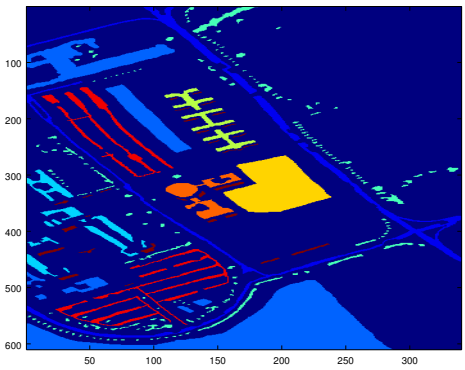

Figure 7: Ground Truth

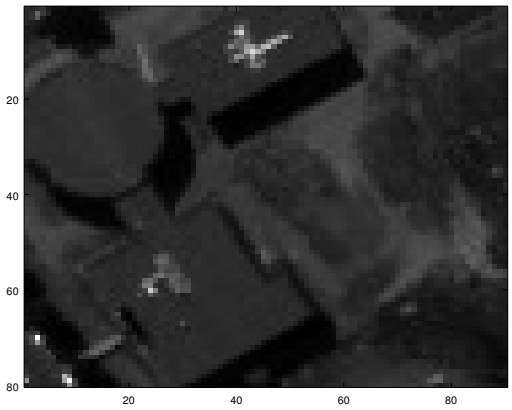

Figure 8: Grayscale Image

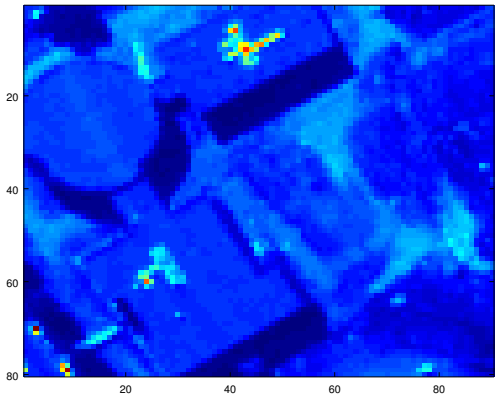

Figure 9: Mid Wavelength

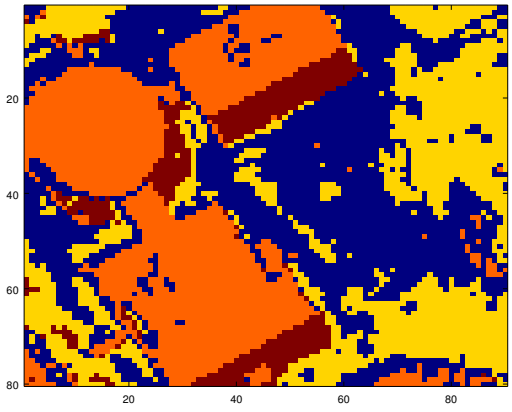

Figure 10: Spectral Clustering Result

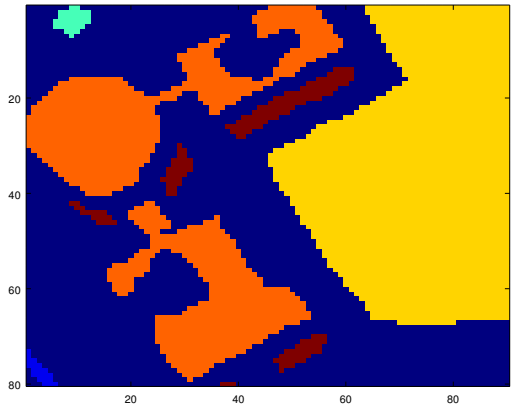

Figure 11: Ground Truth

band, the spectral clustering result, and the ground truth (respectively). Comparing the clustering result against the ground truth, we get that this method correctly matched 54.7% of pixels.

References

[1] ROSIS sensor. Hyperspectral image of Pavia University. Data can be obtained at `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes\#Pavia_University_scene`

[2] "Hyperspectral Systems, Airborne (ROSIS, HySpex)." Hyperspectral Systems, Airborne (ROSIS, HySpex). N.p., n.d. Web. 09 June 2016.

[3] Goldschmidt, O.; Hochbaum, D. S. (1988), Proc. 29th Ann. IEEE Symp. on Foundations of Comput. Sci., IEEE Computer Society, pp. 444451

[4] Mohar, B. (1991). The Laplacian Spectrum of Graphs. *Graph Theory, Combinatorics, and Applications* Vol 2. pp 871-898. Ed. Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk, Wiley.

[5] Ng, A., Jordan, M., & Weiss, Y. (2001). On Spectral Clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14.

[6] Luxburg, U. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing*, 17.

[7] C. Anderson, "A Rayleigh-Chebyshev Procedure for Finding the Smallest Eigenvalues and Associated Eigenvectors of Large Sparse Hermitian Matrices", *J. Computational Physics*, vol. 229, pp. 7477-7487, 2010