Rochester Institute of Technology

RIT Digital Institutional Repository

10-4-2016

Schroedinger Eigenmaps for Manifold Alignment of Multimodal Hyperspectral Images

Juan Emmanuel Johnson
jej2744@rit.edu

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation

Schroedinger Eigenmaps for Manifold Alignment of Multimodal Hyperspectral Images

by

JUAN EMMANUEL JOHNSON

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Applied Mathematics

School of Mathematical Sciences, College of Science

Rochester Institute of Technology

Rochester, NY

October 4, 2016

Committee Approval:

R·I·T

College of
SCIENCE

Nathan D. Cahill, D.Phil   Date

School of Mathematical Sciences

Thesis Advisor

Charles M. Bachmann, Ph.D.   Date

Center for Imaging Science

Committee Member

Paul Wenger, Ph.D.   Date

School of Mathematical Sciences

Committee Member

Matthew Hoffman, Ph.D.   Date

School of Mathematical Sciences

Director of MS Program in Applied and

Computational Mathematics

Abstract

Multimodal remote sensing is an upcoming field as it allows for many views of the same region of interest. Domain adaption attempts to fuse these multimodal remotely sensed images by utilizing the concept of transfer learning to understand data from different sources to learn a fused outcome. Semisupervised Manifold Alignment (SSMA) maps multiple Hyperspectral images (HSIs) from high dimensional source spaces to a low dimensional latent space where similar elements reside closely together. SSMA preserves the original geometric structure of respective HSIs whilst pulling similar data points together and pushing dissimilar data points apart. The SSMA algorithm is comprised of a geometric component, a similarity component and dissimilarity component. The geometric component of the SSMA method has roots in the original Laplacian Eigenmaps (LE) dimension reduction algorithm and the projection functions have roots in the original Locality Preserving Projections (LPP) dimensionality reduction framework. The similarity and dissimilarity component is a semisupervised component that allows expert labeled information to improve the image fusion process. Spatial-Spectral Schroedinger Eigenmaps (SSSE) was designed as a semisupervised enhancement to the LE algorithm by augmenting the Laplacian matrix with a user-defined potential function. However, the user-defined enhancement has yet to be explored in the LPP framework. The first part of this thesis proposes to use the Spatial-Spectral potential within the LPP algorithm, creating a new algorithm we call the Schroedinger Eigenmap Projections (SEP). Through experiments on publicly available data with expert-labeled ground truth, we perform experiments to compare the performance of the SEP algorithm with respect to the LPP algorithm. The second part of this thesis proposes incorporating the Spatial-Spectral potential from SSSE into the SSMA framework. Using two multi-angled HSI's, we explore the impact of incorporating this potential into SSMA.

Acknowledgments

My family, mentors and friends have my thanks for all of their guidance and support during my academic career. I especially would like to thank Dr. Nathan Cahill who helped cultivate my interest in the field as well as having the patience to mentor me throughout my graduate career. I would like to thank the School of Mathematical Sciences within the College of Science at RIT for their support during my time here. I would also like to thank Anna Fiorucci, Carrie Koneski and Corinne Teravainen for all of their assistance. Special thanks to my committee members: Dr. Charles Bachmann and Dr. Paul Wenger, and to the former and current director of graduate programs, Dr. Elizabeth Cherry and Dr. Matthew Hoffman.

Contents

I. Introduction

The next step for the remote sensing community to make the most of available data and provide higher quality image products is multisensor remote sensing. Performing data analysis on single sensor images can be limiting as the data interpretation is constrained to a specific set of acquisition factors. If there are multiple sensors via different satellites or multiple trials with the same sensor then there is suddenly an abundance of information of a remotely sensed scene. The multiple set of acquisition factors can give rise to different interpretations of the scene, enabling a greater understanding of earth observations. However, fusing information from multiple sensors can be challenging because it is highly unlikely that the satellite will observe the same area with the exact same acquisition factors such as sun elevation, sun-target-sensor angle and atmospheric conditions. With respect to multitemporal data, remotely sensed image models can be limited by the lack of observations from the same sensor over an area of interest. This can be rectified by data fusion techniques for multiple sensors.

The machine learning community have entire libraries of image processing and understanding algorithms that take into account the idea of 'Big Data'. Big data refers to large datasets that can be analyzed to reveal hidden factors and interactions between components. Hyperspectral images (HSI) definitely fall into this category of big data, however, the success of many land cover classification schemes are at the mercy of labeled data. Supervised learning algorithms tend to base their predictions on learned input factors so if the input factors are of poor quality, then the learned model will produce poor results. It is unfortunate but although most good classification algorithms are reliant on quality labeled data, the acquisition can be very expensive in the remote sensing field. In the context of multisensor output, it would be advantageous for domain adaption (transfer learning) techniques to reduce the cost of acquiring quality labeled data.

I.1 Research Objectives

My thesis work expands upon current concepts in domain adaption algorithms; in particular the Semisupervised Manifold Alignment (SSMA) method. This method is based on the Kernel Eigenmap Projection framework in order to preserve the inherent structure of each individual HSI whilst pulling like data points together and pushing unlike data points apart between different HSIs. This is a variation to the Kernel Eigenmap [1] approach as the Kernel Eigenmap Projection method produces explicit projection functions to map data points from one domain to a shared

latent space. The prime example of a method that uses this framework is the Locality Preserving Projection (LPP) [2] method; a derivative of the Laplacian Eigenmaps (LE) [3] algorithm. I introduce the Schroedinger Eigenmap Projection (SEP) algorithm which is a derivative of the Schroedinger Eigenmaps (SE) [4] method; both being expansions to the LPP and LE algorithms respectively. Using two iconic HSIs, I look at classification results for all four dimension reduction techniques and address the strengths and limitations of each. After exploring the use of the Schroedinger Operator within the Kernel Eigenmap and Kernel Eigenmap Projection frameworks, I analyze the effectiveness of the Schroedinger Operator within the SSMA schema using two HSIs. Wang [5] originally introduced the notion of Manifold Alignment with inspiration from transfer learning and the LPP methodology. Variations [5] of this method were subsequently introduced followed by Tuia et al. [6] using some of this framework on multimodal HSI fusion. I look at some of Wang's original work, Tuia's original work and the Schroedinger Operator framework [7] to compare and contrast the effectiveness of these three methodologies in multimodal HSI classification. Concise unique contributions within this thesis are as follows:

- Compare and contrast the HSI classification accuracy with Laplacian Eigenmaps, Locality Preserving Projections, Schroedinger Eigenmaps and Schroedinger Eigenmap Projections

- Compare and contrast the multimodal HSI classification accuracy with Wang's [5] original manifold alignment method, Tuia et al.'s [6] current Semisupervised Manifold Alignment method, and the Schroedinger Eigenmap Projection.

I.2 Organization of Thesis

The organization of this thesis is as follows:

- Chapter II gives a background of the remote sensing process and introduces general concepts such as dimensionality reduction, manifold learning and manifold alignment in the context of HSI classification.

- Chapter III gives an introduction to manifold learning and key manifold learning algorithms. It explores the three fundamental matrices used in this thesis including the Adjacency and the Laplacian matrix. This is followed by a step-by-step introduction to the Laplacian Eigenmaps algorithm and the Schroedinger Eigenmaps Algorithm. The Locality Preserving Projections and Schroedinger Eigenmap Projection algorithms are introduced following the Laplacian Eigenmaps and Schroedinger Eigenmap algorithms respectively. Discussions

including the strengths and limitations follow each of the listed algorithms.

- Chapter IV gives an introduction to the concept of Manifold Alignment by way of describing the step-by-step process of the Semisupervised Manifold Alignment algorithm. The matrix construction and projection functions obtained are highlighted in this section. The Schroedinger Operator in this Manifold Alignment framework is discussed and the final set of algorithms to be used in experiments are summarized.

- Chapter V introduces the HSI datasets used for the Eigenmap versus Projection classification results experiment as well as the HSI datasets used for the Multimodal Remote Sensing classification results experiment. The experiment setup and parameters used are described as well.

- Chapter VI shows the best results obtained from each of the experiments along with some sensitivity analysis to showcase how certain parameters affected the classification results. A table of best results is shown for each dataset and experiment to summarize the outcome of the classification experiments. This is followed by a detailed discussion of the results as well as some potential experiments to be done to expand upon this research.

- Chapter VII gives some concluding remarks about this thesis.

II.   Background

II.1   Preliminary Remote Sensing

Remote sensing is the science of acquiring information about the Earth's surface without physically being present at the scene. This is typically done through a long distance camera, via an aircraft or on-board satellites. In this particular application, I will be referring to satellite or airbourne remote sensing. These platforms have sensors that record the electromagnetic radiation on a platform, e.g. the satellite as shown in figure 1. A popular type of sensor is an passive optical sensor, which assumes that the sun is the primary source of radiation.
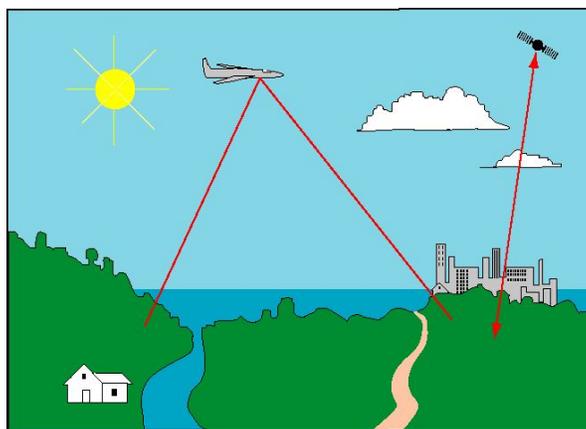
Figure 1: Remote sensing process using airbourne/satellite sensors.[1]

The sensor detects and records this solar radiation from the sun that is reflected or scattered as it makes contact with the earth. Different materials on the earth such as buildings, roads, soil and water reflect this solar radiation differently due to the chemical composition of the material as well as the angle of acquisition. We characterize the differences via a spectral reflectance signature which is a measure of the electromagnetic energy or reflectance as a function of wavelength.

An imaging spectrometer can be used to capture the spectral information of a material in relation to its spatial neighbours. Using this imaging spectrometer, we can acquire remote sensing images as digital images that represent the scene being observed. These digital images are two- or three-dimensional arrays where each pixel location contains spectral information of a specific area on the earth's surface. Each pixel has an 'intensity' range and a spatial location marker in this

---

[2]http://www.nasa.gov/centers/langley/images/content/69904main_RemoteSnsg-fig2.gif

[2]http://www.ces.fau.edu/nasa/images/Energy/VisibleLightSpectrum.jpg
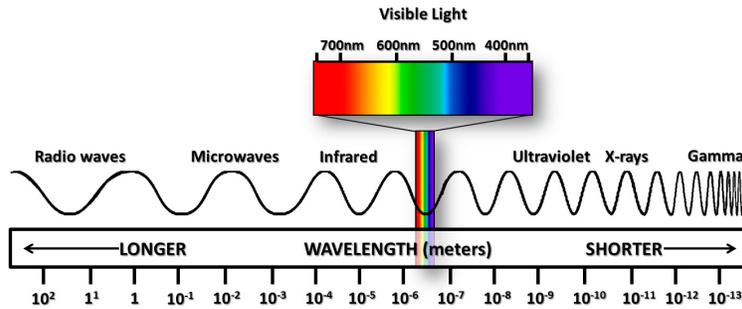
Figure 2: Electromagnetic Spectrum.[2]
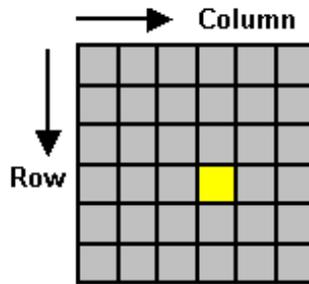
Figure 3: Digital Image shown to a user. [3]

digital image. Figure 3 is a good illustration of a single pixel in relation to it's neighbours. There is a one-to-one correspondence between the pixels location and its geographical coordinates such as longitude and latitude. According to figure 3, there is an (n_1, n_2) grid where each intensity value, i, is usually a measured physical quantity that is used to determine the reflected solar radiance of that spatial location relative to its neighbours, j where $i \neq j$ and $i, j \in \mathcal{R}^{n_1 \times n_2}$.

There are different types of sensors available for remote sensing. The defining characteristics of the sensors also characterize the acquired images. *Spatial resolution* is the scale of an individual pixel on the ground. The exact distance value between adjacent pixels within the digital image is the ground sample distance (GSD). This depends upon the altitude of the sensor as well as the instantaneous field of view (IFOV). An image with a high spatial resolution and a small GSD can detect smaller objects whereas an image with a low spatial resolution and large GSD can only detect large objects. As an example, an image with a high spatial resolution can detect objects up to 0.5 metres as opposed to a low spatial resolution image can detect objects up to 50 metres. In

---

[3]http://www.harrisgeospatial.com/docs/HyperspectralAnalysisTutorial.html

[3]http://www.crisp.nus.edu.sg/~research/tutorial/image.htm

the spatial dimension, pixels that are adjacent to one another are more likely to be the same type of material. This dimension is valuable because it provides information about the size, shape and structure of materials on the earth.
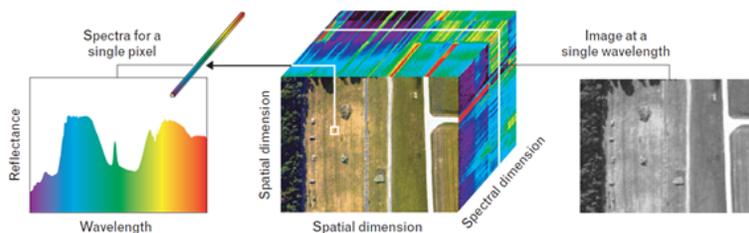
Figure 4: Hyperspectral Image [4]

The *radiometric resolution* is how sensitive the sensor is to the intensity levels. These intensity levels are recorded as a digital number (DN); a generic, pre-calibrated number. The higher the radiometric resolution the more accurately a sensor can discriminate varying intensity levels. Historically, multispectral sensors were very popular because they could detect and measure three to seven wavelength bands for each material. For example the SPOT satellite could measure three bands, the IKONOS satellite could measure four bands and the LANDSAT satellite could measure seven bands. In this case, the pixel value on the (n_1, n_2) grid in Figure 3 digital image would now be on an $(n_1, n_2, \bar{\lambda})$ grid where $\bar{\lambda} \leq 7$, where λ is a wavelength value and $\bar{\lambda}$ is a binned, discrete measure of the wavelength. This can be thought of as a stack of images with different spectral channels. As remote sensing technology got better, hyperspectral sensors became more popular because they could capture information at hundreds of spectral bands. This precision allowed for better characterization of areas on the Earth and thereby better identification of materials. This results in a hyperspectral data cube because each pixel value is within an $(n_1, n_2, \bar{\lambda})$ grid where $\bar{\lambda} \geq 100$.

There are a large number of image processing techniques that can be used to extract useful information about these scenes. Digital image processing usually attempts to enhance, correct, or restore the image. The objective of enhancement is to help with visually interpreting the scene, correction may occur to account for any oddities (e.g. noise removal, contrast enhancement, atmospheric compensation, or spatial registration) that may be in the image that could possibly hinder the image processing results, and restoration techniques enable a user to fill in any missing information that may have occurred during the acquisition process.

Another type of image processing is image interpretation or image characterization for applications

such as image segmentation and image classification. Originally, image processing techniques were reserved for multispectral images (MSI) but the techniques did not always transfer well to HSI. HSIs contained to nonlinearities within the data as larger hyperspectral cubes were constructed. This resulted in highly correlated adjacent spectral bands which led to a multitude of redundant information. This was problematic for some techniques which use a limited number of samples leading to the notion of the *curse of dimensionality* [8]; which is discussed in more detail later.

II.1.1 Applications

There are numerous applications of remote sensing using hyperspectral imagery. Below are a few examples from [5] [6] of how HSI have been used:

- to estimate parameters such as biomass, carbon, land cover changes and biodiversity of forest zones, useful in ecological applications,

- to retrieve physico-chemical mineral properties and parameters such as composition and spatial density, useful in geological science applications,

- to classify minerals as well as their neighbourhood relationships, useful in minerology,

- to monitor the coast for shoreline changes such as the sediment deposition, structural shoreline changes and map coastal features, which are useful for coastal mapping and geological applications to preventing soil erosion,

- to monitor ocean circulation, ocean temperature, wave heights, and sea ice for changes, useful for understanding how oceans function enabling us to manage the oceanic resources better,

- to perform target detection for military purposes because of the benefit from the high spatial and spectral resolution of remote sensing images,

- to classify agricultural classes such as vegetation and crops, used in precision agriculture,

- to track invasive species density measurements of certain vegetation types from temporal remotely sensed images, useful in ecological applications.

---

[5]http://gisgeography.com/multispectral-vs-hyperspectral-imagery-explained/
[6]http://oceanservice.noaa.gov/facts/remotesensing.html

II.2   Issues

The high spectral resolution of HSIs provides more wavelengths for the materials on the scene which is important for classification objectives. However, there are a few challenges that surface as the remote sensing technology starts to get better.

II.2.1   Big Data

Remote sensing has become a popular platform to collect hundreds of spectral bands of one region of interest (ROI) which can potentially be helpful when characterizing chemical compositions of different materials for the analysis of objects of interest. There has been a multitude of new satellites that are being deployed, resulting in a large supply of high dimensional and high resolution data. These data sources generate many airbourne or satellite images of different spatial, spectral and temporal resolutions which have increased over the last few decades. The increase in the volume, variation and speed of data distribution falls into the notion of *Big Data* [9]; large digital datasets which require high level programming skills in order to transform the data in a form that can convey meaningful information [7].

II.2.2   Intraband Correlation of HSI

The narrow and contiguous spectral bands in hyperspectral images may contain reflectance or radiance values that have a high correlation between similar classes of materials. The measured reflectance values for two material classes can be very similar in many spectral bands except for a small unknown portion of spectral bands. This leads to methods that require multivariate statistics to avoid difficulties distinguishing different classes from one another.

II.2.3   Curse of Dimensionality

We are in an era where massive automatic data collection is becoming simpler everyday so high resolution hyperspectral data is a victim of the curse of dimensionality. According to the Hughes phenomena [9], the accuracy of a classifier increases as the dimension of each sample increases up until an optimal dimension. After this point, the large number of features do not guarantee better results from the classifier because the number of samples become limited compared to the

---

[7]https://datascience.berkeley.edu/what-is-big-data/

volume of the feature space. One would need a very large number of samples to xharacterize the large dimensional space. In Hyperspectral remote sensing, this sample-to-dimension ratio is often impossible to obtain due to high labour costs, remote access and limited time to collect the ground truth with the same acquisition parameters as the sensor. This causes classification accuracies to decrease due to the low sample/dimension ratio. In addition, the large amount of hyperspectral data captures many features in a ROI with no immediate intuition about which features are the most important for which application. Consider the spectral domain, where each spectral channel is one dimension and each pixel is represented as a point in this domain. If we were to increase the number of spectral channels past the optimal number of channels, the amount of relevant information for classifier decreases. To combat this, the remote sensing communities use feature extraction methods to project the relevant information in the high dimensional space to a lower dimensional subspace to get a better separation between classes.

II.2.4 Nonlinearities

A few physical phenomena that cause a nonlinear spectral response include multiple scattering, the angular dependence of single scatter in the scene, and the nonlinear attentuation of enery in water bodies [10][11]. Nonlinear feature extraction methods such as manifold learning have successfully extracted important features in these high dimensional feature spaces. Linear feature extraction methods such as PCA and multidimensional scaling (MDS) can result in non-representative features which causes many classification schemes to have poor classification results [12].

II.2.5 Temporal Variations

Hyperspectral remote sensing seeks to capture scenes on the Earths surface. It is difficult to interpret these scenes temporally because the earth is dynamic and the environmental conditions can change in a short time span; causing spectral signatures of certain materials to change with time [13]. Vegetation and shoreline modeling are very popular applications but the dynamics associated with these areas are very apparent and so the spectral response of the same material can appear different with respect to time. Robust classifiers are needed in order to correctly classify ROIs despite the spectral signature variation with time. This can be more problematic when we account for the identical ground truth that needs to be captured during every remote sensing campaign in order to correctly classify these remotely sensed scenes. When limited ground truth is available, transfer learning or domain adaption can potentially solve this issue [14].

II.3 Multimodal Remote Sensing

The above section listed a few different applications of remotely sensed images. Among all applications, classification mapping is one of the most prominent applications. The image classification problem is necessary to solve because land-cover maps are essential in the development of multitemporal studies and modeling efforts in fields like climate change, ocean studies, arctic studies and vegetation analysis. The accuracies of the image classification schemes are well-studied in the machine learning community, but the remote sensing community has an ongoing problem associated with multitemporal campaigns: the analyst has to use multiple sensors at different spatial and spectral resolutions, coverage locations, angles of acquisition, and temporal scales. Therefore, there is a need to have image processing algorithms that can use multimodal data products to produce high quality classification outputs [15]. The need for multimodal image fusion techniques that are accurate, robust, reliable and fast is becoming paramount for handling big data problems associated with continuous streams of remotely sensed images. More specifically, these image processing regimes need to be able to jointly handle heterogeneous pieces of information. In remote sensing, the modality component stems from the fact that there are many types of remotely sensed images with different sources such as imaging spectrometry or light detection and ranging (LiDAR). The fusion problem stems from how these different modality components are pooled together to produce meaningful interpretations. These can be fused at a feature level, a classifier level or at an output level. Gomez-Chova et al. [14] review current methodologies for multimodal classification of remote sensing images and conclude that there are a wide range of different and successful schemes which reflect the efforts of the remote sensing community to solve this problem. Such schemes can be classified into:

- multiple kernel learning methods - perform fusion in the implicit high-dimensional feature representations of the original samples,

- sparse dictionary learning - perform fusion at the signal level

- Markov modeling - use global minimization energy concepts to formalize spatial and mulitmodal fusion,

- deep learning algorithms - these methods use neural network techniques to easily learn from different sources and unlabeled data,

- domain adaption and manifold alignment - fuse information from a geometrical perspective in a target space where the nature and dimensionality of the source domain space is

combined.

II.4   Domain Adaption
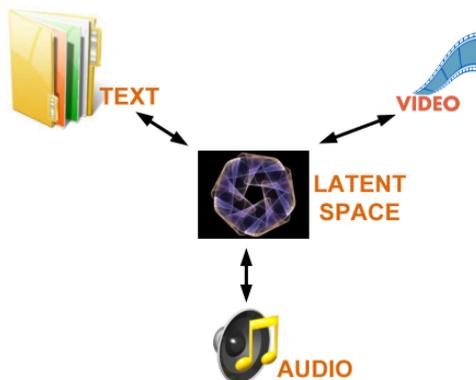
Figure 5: The general transfer learning paradigm. [5]

In the field of remote sensing, it is crucial to have good ground truth for applications such as classification. With the explosion of data sources over the past century, there is now an abundance of information available. However, the cost for acquiring good labeled data is still costly and may require extensive campaigns. There are many situations where multiple sources of information can prove useful in the remote sensing community [6] [14] [16] [17]. Often times there is a situation where multiple angles can give a better representation of a remote sensing scene. Other times the field-of-view (FOV) of a sensor can only capture a certain section of a ROI so there needs to be multiple run throughs. This can result in different acquisition parameters over a single ROI. Correction methods need to be implemented as well as image mosaicking techniques. The common case is the scenario of multitemporal remote sensing. The best multitemporal characterization of a ROI involves many data acquisitions with sensors. But it would be expensive and difficult to run a campaign to acquire ground truth with each acquisition. Using the same labeled samples from a single campaign for all sensor run-throughs would be beneficial to minimize the cost to produce quality classification results. Unfortunately, we cannot use existing ground truth for newly acquired images because the scene may actually change between acquisitions. Furthermore, certain acquisition parameters can change the outcome of the classification results. For example, composition, topography, illumination conditions, sun-target-sensor angles and sensor can effect characterizing vegetation [14] [18].

Domain adaption is a viable method for tackling the problem of domain adaption. One class of domain adaption methods that has recently surfaced in the literature is called feature-based domain adaption. Some examples include manifold alignment [5], kernel based methods [19] and transfer components analysis [20]. Under the assumption that each data source has its own feature space that are somewhat related to one another, the objective of domain adaption is to find a target feature space where the source feature spaces have the same attributes. See figure 5 for a generic view of domain adaption in terms of transfer learning. Notice how there are three different sources of information being conveyed to one single latent space. In the context of HSI, if there are two separate HSIs with labeled information, the two source images should have similar class labels in the target space. This ensures that the labeled instances in the source space can be used in the target space.

II.5   Manifold Alignment

Manifold Learning has been successfully applied to hyperspectral image classifcation [12]. It excels at addressing the curse of dimensionality as well as correctly modeling the nonlinearities that sensors capture in nature when sensors have narrow spectral bands and high spatial resolutions. As a data-driven methodology, many studies have used manifold learning as an approach to the domain adaption problem such as manifold Procustes analysis [21], manifold warping [22], class centroid alignment [23] and semisupervised manifold alignmemt [6]. There are many factors that can affect different manifold learning methods such as dimension of manifold coordinates, the selection of the free parameters, the number of classes and the specific scene characteristics.
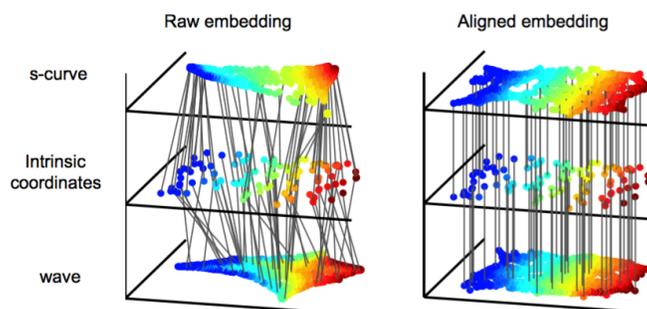
Figure 6: Manifold Alignment [8].

The manifold alignment approach attacks the domain adaption problem by reducing all data

sources to a common target space [24]. This is done by aligning different domains (or manifolds) from the source space to a manifold joint in the latent space (or target space). In this target space, the datasets will have similar behaviours to their respective source domains. Figure 6 depicts a high-level interpretation of this manifold alignment process. From the classification perspective, similar classes between the two source domains will be close together in the target domain. The target domain should have desirable properties such as local neighbourhood consistency, cross class correlation and class discrimination [23]. There should also be some sort of projection function in order to project data from a source domain into the target domain. Tuia et al. [6] found that by constructing this manifold alignment method using Laplacian Eigenmaps, they were able to get 90% classification accuracy with multiangluar remote sensing images. Using the Laplacian Eigenmaps (LE) framework for the similarity and dissimilarities between classes in each source domain, they were able to construct a cost function to minimize in order to pull similar classes apart and push dissimilar classes together in the target domain [25].

III.   Manifold Learning

We have established in the Chapter II the importance of dimension reduction in the RS field. Generally, dimensionality reduction seeks a transformation of the high dimensional data to a lower dimensional space whilst still preserving the important features present in the data.

The two main branches of dimensionality reduction are linear and nonlinear. Principal Components Analysis (PCA) [26] is the most famous method which finds the greatest variance between the data points. An excellent survey of some linear techniques with applications to remote sensing can be found here [27]. Nonlinear methods include some fundamental methods such as Kernel PCA (KPCA) [28], Isometric Mapping (Isomap) [29], Locally Linear Embedding (LLE) [30], and Laplacian Eigenmaps (LE) [3]. These were followed shortly by subsequent methods including Diffusion Maps (DM) [31], Schroedinger Eigenmaps (SE) [4], Local Tangent Space Alignment (LTSA)[32], and Hessian LLE (HLLE) [33]. Most of these methods can be described as Kernel Eigenmap Methods [1] as they attempt to construct some unique kernel matrix which provides information about the pairwise relationship between data points in order to discriminate important features.

Manifold learning is the nonlinear dimensionality reduction method of data. A manifold is a topological space that is locally Euclidean. A basic example is the earth which resides as a 2-D sphere in a 3-D space (when viewed from space) but it is a generally flat 3-D coordinate system from our point of view [34]. Simply put, the Earth appears to be 3D but there are places along the Earth's surface which are flat from our point of view. Thinking of high dimensional spaces in terms of manifolds allows for powerful techniques to emerge with an effort to reduce this dimensionality to an embedding space with valid approximations to the data in the higher space. An introduction to the necessary graph theory is presented in order to facilitate the introduction the manifold learning algorithms discussed. This chapter focuses on the Laplacian Eigenmaps (LE) algorithm and some derivatives including the Locality Preserving Projection (LPP) method and the Schroedinger Eigenmaps (SE) method. Furthermore, the Schroedinger Eigenmaps Projection (SEP) algorithm is introduced. This chapter will provide a step-by-step breakdown of the original LE algorithm and some key matrices and additions from the LPP, SE and SEP schemes.
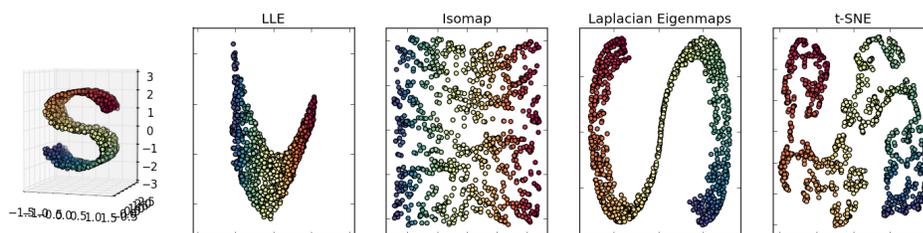
Figure 7: S-Curve example: The manifold learning techniques shown are Local Linear Embedding, Isometric Mapping, Laplacian Eigenmaps and t-distributed Stochastic Neighbourhood Embedding. [8]

III.1 Spectral Graph Theory

III.1.1 Overview

A graph $G :< V, E >$ is a set of vertices, V and a set of unordered pairs of vertices, E where $V = \{v_1, v_2, \ldots, v_N\}$ and $E = \{(u_1, v_1), (u_2, v_2), \ldots\}$. A single unordered pair of vertices, e is known as an edge where $e = (u, v)$. A simple graph will consist of vertices (points) that are connected to one another by edges (lines). Each vertex can represent an object and each edge will represent the relationship between one or more objects. We don't pay attention to the position of the vertices or to the length of the edges. We only care about how each vertex is related to one another. Figure 8 shows a simple example of a graph as a representation of an arbitrary set of data points. Graph-based models are popular in the computer vision and machine learning community due to their simplistic representation of complex structures whilst still maintaining important structural information. This is due to their emphasis on connectivity of the original structure rather than just comparing the similarity between data points.

If we are given a large graph as a representation of a scenario, it would be useful to extract information from the graph concisely. One way we could do this would be to study the spectra of the matrices used to represent these graphs. The core of spectral graph theory is the relationship between the eigenvalues and the structure of graphs. This is the culmination of basic graph theory and tools of linear algebra. In this section, I will introduce three common matrices associated with graphs; the adjacency matrix, the Laplacian matrix and the normalized Laplacian matrix. I will also give some examples of how the eigenvalues can be used to extract information from a graph. Throughout this section, I will use Figure 9 as a basic example and will present the key matrices that can be obtained from this graph.

---

[8] http://scikit-learn.org/stable/modules/manifold.html
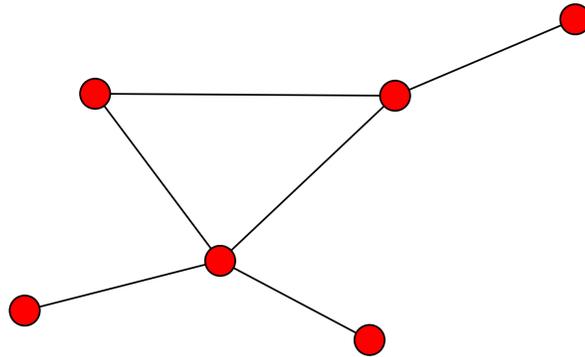
Figure 8: A graph with edges.

III.1.2 Adjacency Matrix
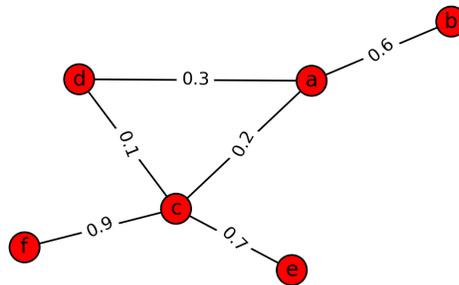
Figure 9: A simple graph, G with weighted edges.

The adjacency matrix A of graph G is the $|V| \times |V|$ matrix defined by:

$$
\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E; \\ 0 & \text{if } (v_i, v_j) \notin E;. \end{cases} \tag{III.1}
$$

An example of the adjacency matrix for Figure 9 is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{III.2}$$

From a spectral graph theory point of view, the ordering of vertices does not matter, as a reordering causes A to be transformed to a similarity matrix which preserves the eigenvalues. The eigenvalues can be thought of as functions on the set of vertices. The adjacency matrix is symmetric for undirected graph graphs, which also means that the eigenvalues are real. Also, because the matrix, $\mathbf{A}$ is symmetric, it follows that the matrix has a full set of eigenvectors, and that eigenvectors corresponding to distinct eigenvalues are mutually orthogonal. This is a nice property which enables many applications using undirected graphs.

In the adjacency matrix for 9, we use 0s and 1s to indicate the absences or presence of an edge. However, we could use different entries that specify how strongly connected a vertex is to another. Let $w(u, v)$ denote the weighting function for vertex u and v. For the graph in figure 9 get $\mathbf{W}_{u,v} = w(u, v)$ which results in the following matrix:

$$\mathbf{W} = \begin{bmatrix} 0 & 0.6 & 0.2 & 0.3 & 0 & 0 \\ 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0.1 & 0.7 & 0.9 \\ 0.3 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0 & 0 \end{bmatrix} \tag{III.3}$$

III.1.3   Laplacian Matrix

The second type of matrix that is important for our applications is the Laplacian matrix, $\mathbf{L}$, which is defined as:

$$\mathbf{L}_{i,j} = \begin{cases} d_i & \text{if } i = j; \\ -1 & i \text{ is adjacent to } j; \\ 0 & \text{otherwise,} \end{cases} \tag{III.4}$$

where d_i is the degree of the ith vertex, i.e. $d_i = \sum_j W_{ij}$. We can relate the Laplacian matrix to the adjacency matrix by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, or in the weighted case,

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{III.5}$$

where $\mathbf{D}$ is the diagonal matrix the degree on the diagonals and $\mathbf{W}$ is the adjacency matrix. An example of the Laplacian matrix for the graph of figure 9 is:

$$\mathbf{L} = \begin{bmatrix} 1.1 & -0.6 & -0.2 & -0.3 & 0 & 0 \\ -0.6 & 0.6 & 0 & 0 & 0 & 0 \\ -0.2 & 0 & 1.9 & -0.1 & -0.7 & -0.9 \\ -0.3 & 0 & -0.1 & 0.4 & 0 & 0 \\ 0 & 0 & -0.7 & 0 & 0.7 & 0 \\ 0 & 0 & -0.9 & 0 & 0 & 0.9 \end{bmatrix} \tag{III.6}$$

There are some important properties of the Laplacian matrix, $\mathbf{L}$ such as:

1. $\mathbf{L}$ is symmetric and positive semi-definite.

2. The smallest eigenvalue of $\mathbf{L}$ is $\lambda_0 = 0$, and $\mathbf{1} = (1, 1, \ldots, 1)^T$ is a correct eigenvector.

3. The algebraic multiplicity of the zero eigenvalue is equal to the number of connected components; where a connected component is the set of vertices in a graph that are linked to each other by paths.

III.2 Laplacian Eigenmaps

In machine learning, we often come across problems where the data is given to us in a high dimensional space but the actual intrinsic information of the data lies on a low dimensional manifold. Working in a lower dimension can make a classification problem easier because of the computational advantage as well as the better separation of classes.

To formally introduce the dimension reduction problem, consider m points $\{x_1, x_2, \ldots, x_m\} \in \mathcal{R}^N$, that lie in an an n dimensional manifold, $M \subseteq \mathbb{R}^N$. Our objective is to find a lower dimensional representation $\{y_1, y_2, \ldots, y_m\} \in \mathcal{R}^n$ where $n << N$. In this formulation, $\mathbf{y}_i$ would represent $\mathbf{x}_i$ in the new dimensional space. Linear algorithms such as PCA or MDS consider the variance between data points and construct the y_i's to be a linear transformation of the x_i's. LE considers the geometric relationship between the data points. There are other nonlinear dimensionality reduction techniques such as LLE or Isomap that use neighbourhood information of each point by considering its relationship within the graph structure.

In this section, we will to introduce the LE algorithm. I describe the algorithm followed by some possible pros and cons to the method.

III.2.1   Algorithm

Step 1. Construct the Adjacency Matrix

We build a graph G whose vertices i and j are connected if x_i is among k-Nearest Neighbours (k-NN) or ϵ-ball graph. The weights between the data points are measured in the euclidean distance metric. This adjacency matrix A represents the connectivity of the original data where k, the number of neighbours, is the free parameter for the $k - NN$ approach and ϵ is the free parameter for the ϵ-ball approach.

Step 2. Compute weights for each Edge

We want weighted edges on the graph to represent the proximity of the vertices to their adjacent neighbours. One common kernel to use is the diffusion weight matrix, W. We can define W as:

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}} & \text{if } i, j \text{ are connected} \\ 0 & \text{otherwise.} \end{cases} \tag{III.7}$$

where σ is a parameter that controls the width of the kernel.

Step 3. Solve the eigenvalue problem

Now, we want to minimize the following cost function

$$\min_{y^T D y = I} \frac{1}{2} \sum_{i,j} ||y_i - y_j||^2 W_{ij} \tag{III.8}$$

which is equivalent to minimizing the following

$$\min_{y^T D y = I} \text{tr} \left(y^T L y \right) \tag{III.9}$$

where $L = D - W$ is an mxm laplacian operator and I is the identity matrix.

The solution to this minimization problem is given by finding the first n eigenvalue solutions to the generalized eigenvalue problem:

$$L f = \lambda D f \tag{III.10}$$

We solve our generalized eigenvalue to obtain the solutions $f_0, f_1, \ldots, f_{n-1}$ eigenvectors which are ordered by the smallest $n-1$ eigenvalues $\lambda_0, \lambda_1, \ldots \lambda_{n-1}$. The smallest eigenvalue. λ_0 is trivial because of the normalization

$$\begin{bmatrix} d_1 - w_1 & \ldots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \ldots & d_n - w_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ \ldots \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \ldots \\ 0 \end{bmatrix} \tag{III.11}$$

This is the trivial solution $f_0 = \mathbf{1}$ associated with the $\lambda_0 = 0$ which results in the eigenfunction

$$f_0 : x \rightarrow (1, \ldots, 1) \tag{III.12}$$

This argument assumes that the graph is connected. This first vector represents the minimum distance necessary for the cluster which would result in the entire graph being one big cluster. So if there is only one connected component, then it would be the trivial solution of the entire graph. If this graph were not connected then the number of connected components would be equal to the number of zero eigenvalues. This rest of the eigenvectors will be orthogonal to f_0 because L is symmetric. If we take the first $d+1$ eigenvectors, we obtain an embedding from the original high dimensional space to the lower d dimensional space:

$$F : x_i \rightarrow (f_1(x_i), \ldots, f_k(x_i)) \tag{III.13}$$

III.2.2 Computational Complexity

The three main components of this problem are the weighted adjacency matrix construction, the graph Laplacian construction, and the partial eigenvalue decomposition. Assuming an approximate nearest neighbour method, the nearest neighbour search has a computational complexity of $\mathcal{O}\left(Dlog(k)Nlog(N)\right)$ where D is the original dimension of the data, N is the number of data points and k is the number of nearest neighbors chosen. The eigenvalue decomposition typically translates from a generalized eigenvalue problem of the form $Ax = \lambda Bx$ into the form $B^{-1}Ax = \lambda x$ of $\mathcal{O}\left(DN^3\right)$. Then solving for the d smallest eigenvalue results in $\mathcal{O}(dN^2)$, where d is the desired number of eigenvalues. This leaves us with the final order $\mathcal{O}\left(Dlog(k)Nlog(N)\right) + \mathcal{O}\left(DNk^3\right) + \mathcal{O}\left(dN^2\right)$.

III.2.3 Discussion

The LE method has many advantages and disadvantages over linear methods as well as other nonlinear methods. Some advantages include:

- The algorithm is easy to implement because it involves solving a generalized eigenvalue problem for which many packages exist. The nearest neighbour searches consist of the most computational effort, but there are many different packages available to optimize the search.

- The algorithm preserves geometric relationships the data. By reducing the dimension of our high dimensional data, we still maintain the local geometry.

- The locality-preserving principle of the LE algorithm means that the method is insenstive to modest noise and outliers [35]. So scale, rotation and translation does not effect the outcome of the final embedding.

Some disadvantages include:

- Requires out-of-sample extension, i.e. you cannot use new data points that was not in the original data space.

- The approximation assumes that the data is sampled uniformly from the high dimensional space. This rarely happens in real world applications.

- Some of the parameters are difficult to estimate and choose. The k, σ, d parameters require cross-validation methods in order to accurately reduce the dimension. The desired results also depend on the application so there is no one-size-fits all reduced dimension.

III.3   Locality Preserving Projections

The main disadvantages of LE are the computational cost and the requirement for out-of-sample extensions to project new samples into the embedding. A class of methods called Kernel Eigenmap Projection methods (similar to Kernel Eigenmap methods) [1] can be used to approximate dimension reduction methods required out-of-sample extensions. One member of this family, the Locality Preserving Projection (LPP) [36] algorithm, came forth as an approximation method to the original LE algorithm [2]. The purpose was to provide a linear dimensionality reduction technique that utilized the Laplacian operator but still retained the useful graph theoretic capabilities. By doing a simple transformation of letting $\mathbf{Y} = \mathbf{XF}$, we end up with the following eigenvalue decomposition equation

$$\mathbf{X^{T}LXF} = \lambda \mathbf{X^{T}DXF} \tag{III.14}$$

where $\mathbf{X}$ is the original data matrix, $\mathbf{L}$ is the Laplacian matrix and $\mathbf{D}$ is the degree matrix. The desired number of the top d eigenvalues are found to produce the projection functions for the dataset.

III.3.1   Discussion

A higher level explanation of a key difference between the Kernel Eigenmap and the Kernel Eigenmap projection methods entails the notion of feature-based versus instance-based methods dimension reduction techniques. An instance-based case is looking directly at the data instance-by-instance whereby the final eigenvalue decomposition finds likely clusters within these data points. The LE embedding is an example of a dimension reduction technique that pulls likely clusters together from the original instance-based space. A feature-based case transforms the data points into the feature-based space whereby the final eigenvalue decomposition finds likely clusters within these features. LPP is an example of a feature-based dimension reduction technique that translates the instance-based space to the feature-based space via the $\mathbf{X}$ matrix multiplied by the $\mathbf{L}$ and $\mathbf{D}$. Also, since this is a linear transformation, solving the eigenvalue decomposition produces

a projection function based off of the similar features. We can project the original data and new data from its original space to the new embedding from the solution to the LPP algorithm.

III.3.2 Computational Complexity

Similar to the LE method, the three main components of this problem is the weighted adjacency matrix construction, the graph Laplacian construction, and the partial eigenvalue decomposition. Assuming an approximate nearest neighbour method has a computational complexity of about $\mathcal{O}\left(Dlog(k)Nlog(N)\right)$ where D is the original dimension of the data, N is the number of data points and k is the number of nearest neighbors chosen. The eigenvalue decomposition typically translates a generalized eigenvalue of the form $Ax = \lambda Bx$ into the form $B^{-1}Ax = \lambda x$ of $\mathcal{O}\left(DN^3\right)$, an eigenvalue problem. Then solving for the d smallest eigenvalue results of the feature based matrix instead of the instance based matrix gives us $\mathcal{O}(d^3)$, where d is the desired number of eigenvalues. This leaves us with the final order $\mathcal{O}\left(Dlog(k)Nlog(N)\right) + \mathcal{O}\left(log(k_{sp})Nlog(N)\right) + \mathcal{O}\left(DNk^3\right) + \mathcal{O}\left(d^3\right)$.

III.4 Schroedinger Eigenmaps

The Schroedinger Eigenmaps method [4] has origins from the Laplacian Eigenmaps method with the addition of an extra term referred to as a "potential". The use of an additional potential matrix to augment the LE algorith have been studied in the context of hyperspectral remote sensing [37][38]. There are two types of potentials, the cluster potential and the barrier potential. Potentials that can force certain pixels that are similar to be close to one another in the final embedding are called 'cluster potentials'. Potentials that force certain pixels, and any pixels near to them, to stand out in the final embedding are called 'barrier potentials' because it tries to insert its own clustering centroid which draws other pixels to its origin. There have been many applications of SE to problems in remote sensing. Spatial-spectral cluster potentials have been shown to be very effective at classification [39]. Likewise, barrier potentials proved useful in target detection with attempts to highlight certain pixels [40][41][42]. In image segmentation applications, a generalization of the normalized cuts algorithm incorporates must-link and cannot-link constraints [43] in a similar manner to how SE incorporates cluster potentials.

III.4.1 Origin

The Laplacian operator has origins in the diffusion equation which describes the dynamics of particles that move from a region of higher concentration to a region of lower concentration. The diffusion operator is defined as

$$\partial_t \psi(x,t) = \triangle \psi(x,t) \tag{III.15}$$

Similarly, the Schroedinger operator is defined as:

$$\partial_t \psi(x,t) = \triangle \psi(x,t) + v(x)\psi(x,t) \tag{III.16}$$

where $v(x)$ is a potential function that is multiplied by a non-negative operator or constant. We can come up with a discrete analogue like so:

$$S \quad = L + V \tag{III.17}$$

where V is a nonnegative diagonal *mxm* potential matrix. The potential matrix can act as an enhancement or modification to the laplacian operator. So the problem given by the Laplacian eigenmaps method

$$\min_{y^T D y = I} \text{tr}(y^T L y) \tag{III.18}$$

now becomes

$$\min_{y^T D y = I} \text{tr}(y^T (L + \alpha V) y) \tag{III.19}$$

and subsequently

$$\min_{y^T D y = I} \left[\text{tr}(y^T L y) + \text{tr}\left(y^T \alpha V y\right) \right] \tag{III.20}$$

where the parameter α is greater than or equal to zero. This alpha parameter is used as a trade-off parameter between the Laplacian term and the potential term. So our final minimization problem can be expressed by adding a penalty term to [4]:

$$\min_{y^T D y = I} \frac{1}{2} \sum_{i,j} ||y_i - y_j||^2 W_{ij} + \alpha \sum_{i,j} V(i)||y_i|| \tag{III.21}$$

where $\mathbf{V}$ is a diagonal matrix. The end result is the same generalized eigenvalue problem as the original Laplacian Eigenmaps algorithm:

$$(L + \alpha V)f = \lambda D f \tag{III.22}$$

where we need to find the smallest $d + 1$ eigenvalues and eigenvectors. Since the potential matrix $\mathbf{V}$ is only used as a penalty term to the graph Laplacian, $\mathbf{L}$, there is some flexibility when using this term. In remote sensing applications, one could use this penalty term for semisupervised classification. By adding a potential, in the form of a cluster potential, to the graph Laplacian, one could try and force certain pixels with the same class to be closer to one another in the final embedding. Alternatively, one could use a potential, in the form of a barrier potential term, to try and push pixels, with the same or different class, apart in the final embedding.

III.4.2 Cluster Potentials

If we have a collection of nodes that represent the data samples, then we can define a cluster potential matrix, $\mathbf{V}$ which has zeros everywhere except for nodes that we want to link by adding a one in the off diagonal entries. For example, if we wanted to initiate a semisupervised connection between nodes 1 and 2, we would put a one in the off diagonal entries like so.

$$\mathbf{V} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{III.23}$$

Once we have put as many links between specific nodes as desired, we can find the laplacian matrix of the potential matrix which results in our final potential matrix.

$$\mathbf{V} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{III.24}$$

If we assume that the nonzero rows in equation III.24 are 'cluster potentials', then the generalized formulation for a single cluster potential can be shown as

$$\mathbf{V}_{k,l}(i,j) = \begin{bmatrix} m & -1 & -1 & \ldots & -1 \\ -1 & m & -1 & \ldots & -1 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ -1 & -1 & \ldots & m & -1 \\ -1 & -1 & -1 & \ldots & m \end{bmatrix}. \tag{III.25}$$

where each m represents the number of links that are added to each node in a semisupervised manner. In principle, the cluster potential acts attracts like vertices close together. Adding a cluster potential between two or more vertices forces those points to be closer together in the final embedding space. Practically speaking, this allows an analyst to use a semisupervised method on an unsupervised dimensionality reduction algorithm to force certain known pixels to be together with unknown pixels in the classification outcome. Ultimately, one can add as many cluster potentials $\mathbf{V}(i,j)$ as necessary within the full potential matrix, $\mathbf{V}$.

III.4.3   Barrier Potentials

If we want to make V a barrier potential, we could make a diagonal matrix with ones along the main diagonal to indicate everywhere there should be a barrier.

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

So the one located in the second row would be the location of the pixel where a barrier would be drawn around it in the high dimensional space. This potential matrix introduces a penalty term at

each nonzero entry. By forcing this term to zero, the nearest neighbours will also be pulled to zero because the local geometry is preserved in the embedding.

III.4.4   Computational Complexity

Similar to the LE and LPP method, the three main components of this problem is the weighted adjacency matrix construction, the graph Laplacian construction, and the partial eigenvalue decomposition. Assuming an approximate nearest neighbour method has a computational complexity of about $\mathcal{O}\left(Dlog(k)Nlog(N)\right)$ where D is the original dimension of the data, N is the number of data points and k is the number of nearest neighbors chosen. However, the SE method will also have a nearest neighbor search for the spatial domain as well. This has the computational complexity of $\mathcal{O}\left(log(k_{sp})Nlog(N)\right)$ where k_{sp} is the number of spatial nearest neighbors. The eigenvalue decomposition typically translates a generalized eigenvalue of the form $Ax = \lambda Bx$ into the form $B^{-1}Ax = \lambda x$ of $\mathcal{O}\left(DN^3\right)$, an eigenvalue problem. Then solving for the d smallest eigenvalue results in $\mathcal{O}(dN^2)$, where d is the desired number of eigenvalues. This leaves us with the final order $\mathcal{O}\left(Dlog(k)Nlog(N)\right) + \mathcal{O}\left(log(k_{sp})Nlog(N)\right) + \mathcal{O}\left(DNk^3\right) + \mathcal{O}\left(dN^2\right)$.

III.5   Schroedinger Eigenmap Projections

As a precursor to the manifold alignment regime, a new method will be briefly presented to augment the modifications of LPP on the LE method. The LPP method sought to reduce the computational complexity as well as get explicity projection functions for the data from the original space to the lower dimensional embedding. SE attempts to enhance the Laplacian matrix present in LE to get stronger cohesion between similar classes to be present in the embedding. The next logical outcome for the SE method is to enhance the LPP method with the same benefits as the LE method. The modification is simple: by adding the potential matrix, $\mathbf{V}$ to the LPP scheme, we end up with the following equation

$$\mathbf{X^T}(\mathbf{L} + \alpha\mathbf{V})\mathbf{XF} = \lambda(\mathbf{X^T}\mathbf{DXF}) \tag{III.26}$$

called the Schroedinger Eigenmap Projections (SEP) method.

III.5.1 Hypothesis

The LPP method has been shown to be effective in a wide range of applications including the remote sensing field. It successfully cuts down the computational time as well as give reasonably good results for classification purposes. There has also been a lot of work to further improve the LPP methodology by incorporating small additions to counter its disadvantages. The SE methodology successfully enhanced the LE methodology so I hypothesize that the SE methodology should also benefit the LPP methodology. The Laplacian matrix is a key component in all Kernel Eigenmap methods so it makes sense that strengthening the key component can enhance the outcome of the algorithm.

IV. Manifold Alignment
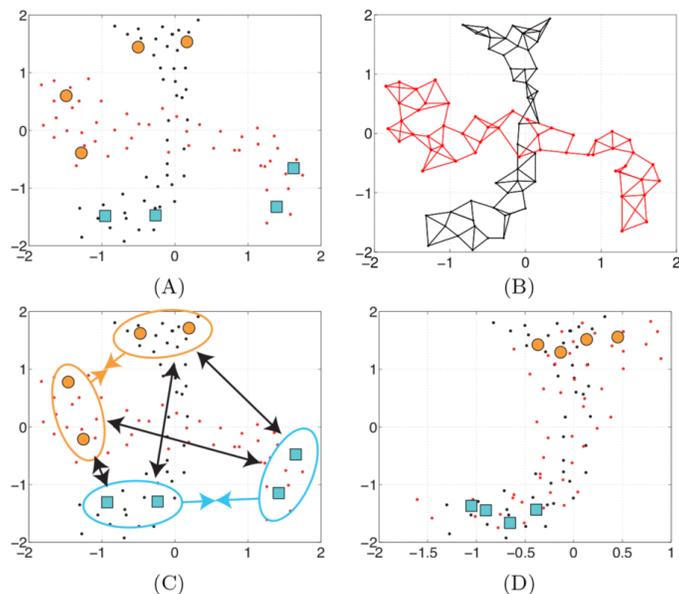
Figure 10: SSMA process: (a) two datasets, (b) geometric structures for each individual dataset, (c) similarity and dissimilarity between labeled samples, (d) aligned dataset. [6]

The main idea of the semisupervised manifold alignment (SSMA) method [6] is to align individual manifolds by projecting them into a joint latent space $\mathcal{F}$. See Figure 10 for a general overview of embedding process. With respect to remote sensing, the authors of [6] have provided evidence to support the notion that the local geometry of each image dataset is preserved and the regions with similar classes are brought together whilst the regions with dissimilar classes are pushed apart in the SSMA embedding. By using graph Laplacians for each of the similarity and dissimilarity terms, a Rayleigh Quotient function is minimized to extract projection functions. These projection functions are used to project the individual manifolds into a joint latent space.

IV.1 Outline

In this chapter, I introduce the semisupervsied manifold alignment (SSMA) method that was presented in [6]. I give a brief overview of the algorithm in section IV.1.1 followed by some notation declarations in 1.2. I proceed to talk about the cost function in section 1.3 followed by explicit construction of some cost function components in sections 1.4, 1.5 and 1.6. I give some

insight into the projection functions in section 1.7 followed by some advantages of this method with other alignment methods in section 1.8. I conclude this chapter with some insight into the computational complexity of the SSMA method in section 1.9.

IV.1.1 Notation

Suppose we have a series of M images with each data matrix $\mathbf{X}^m$, where $m = 1, 2, \ldots, M$. Let each matrix $\mathbf{X}^m$ be split into unlabeled samples $\{x_i^m\}_{i=1}^{u_m}$ and labeled samples $\{x_j^m, y_j^m\}_{j=1}^{l_m}$. Typically, there are many more unlabeled samples than labeled samples, so $l_m << u_m$. Let $\mathbf{X}^m$ be of d_mx n_m dimensions which says that d dimensional data by n_m labeled and unlabeled samples. Succinctly, we have $\mathbf{X}^m \in \mathbb{R}^{d_m \times n_m}$, where $n_m = l_m + u_m$. We can explicitly concatenate each individual data matrix into a block diagonal matrix X where $\mathbf{X} = diag(\mathbf{X}_1, \ldots, \mathbf{X}_M)$ and $\mathbf{X} \in \mathbb{R}^{d \times N}$.

$$
\mathbf{X} = \begin{bmatrix} \mathbf{X}^1 & 0 & 0 & 0 \\ 0 & \mathbf{X}^2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \mathbf{X}^M \end{bmatrix}.
\tag{IV.1}
$$

N is the total number of labeled samples for all of the images and d is the overall dimension of all of the images, i.e. $N = \sum_m^M n_m$ and $d = \sum_m^M d_m$.

Note that these images do not necessarily need to replicate the same spectral density, the same spatial dimension or even the same sensor. Practically, this means that we do not need to have the same m for each matrix $\mathbf{X}^m$, i.e. $m \neq \bar{m}$.

IV.1.2 Semisupervised Loss Function

Tuia et al. [6] construct a semisupervised loss function with the idea of aligning M images to a common representation. Ultimately, they created M projection functions f^m that mapped image $\mathbf{X}^m$ from $\mathbf{X}^m \in \mathbb{R}^{d_m \times n_m}$ to $f^m \in \mathbb{R}^{d_m \times d}$ where $m = 1, 2, \ldots, M$. They ensure that the projection functions f^m bring samples from the same class closer together and push samples from different classes apart whilst still preserving the geometry of each individual manifold. Their aim is to maximize the distance between the dissimilarities and minimize the distance between similarities. This can be expressed via the following equation:

$$\mathcal{C} = \frac{G + \mu S}{B} \tag{IV.2}$$

where G is the geometric term, S is the similarity term with an arbitrary parameter μ to control the trade off between G and S, and B is the dissimilarity term. If the denominator gets very large then this cost function will get very small. The Rayleigh Quotient is used because the problem has terms that need to be minimized and terms that need to be maximized.

$$F_{opt} = \underset{F}{argmin} \frac{F^T(G + \mu S)F}{F^T D F} \tag{IV.3}$$

$$= \underset{F}{argmin} \left\{ tr\left((F^T(G + \mu S)F)^{-1}F^T B F \right) \right\} \tag{IV.4}$$

F is a dxd matrix and the row blocks of F correspond to the domain specific projection functions $f^m \in \mathbb{R}^{d_m x d}$ that project the data matrix X^m into the joint latent space. The matrix $G + \mu S$ corresponds to the term that needs to be minimized and the matrix D corresponds to the term that needs to be maximized. Tuia et al. [6] describe the term $G + \mu S$ as two components, a geometric component G and a class similarity component, S. They want to preserve the geometric structure of each data matrix $\mathbf{X}^m$ and bring data from the same classes within each individual $\mathbf{X}^m$ closer together in the joint latent subspace manifold. They combine the geometric component, G, and the similarity component, S, to create the discriminant component A, i.e., $A = S + \mu G$, where μ is a free parameter to monitor the trade-off between the local manifold and the class similarity. The authors choose to use three graph Laplacian matrices to construct the terms in the Rayleigh quotient, G, S, and B as shown in subsequent sections.

IV.1.3 Geometric Spectral Similarity Term

The G term in equation IV.2 preserves the manifold of each individual data set $\mathbf{X}^m$ throughout the transformation as no inter-domain relationships are considered. Affinity matrices are constructed for each $\mathbf{X}^m$ and then put into a block diagonal matrix $\mathbf{W}_g^m \in \mathbb{R}^{n_m x n_m}$. There are many ways to construct affinity matrices including k nearest neighbour graphs (k-NN) and ϵ-neighbourhood graphs (ϵ-N). The graphs are assumed to be undirected so the affinity matrix should be symmetric, i.e. if x_i is connected to x_j then x_j is connected to x_i. An example of the final matrix $\mathbf{W}_g$ is:

$$\mathbf{W}_g = \begin{bmatrix} W_g^1 & 0 & 0 & 0 \\ 0 & W_g^2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & W_g^M \end{bmatrix} \qquad \text{(IV.5)}$$

Entries in the matrices $\mathbf{W}_g^m$ are $\mathbf{W}_g^m(i,j) = 1$ if x_i and x_j are connected and 0 otherwise. Using the k-NN weighted graph, this term ensures that the samples in the original m domain remains in the same proximity in the new latent projected space. A graph laplacian, $\mathbf{L}_g^m$ is constructed from each of the affinity matrices and put into a block diagonal matrix $\mathbf{L}_g$, where $\mathbf{L}_g^m = \mathbf{D}_g^m - \mathbf{W}_g^m$ and $\mathbf{D}_g^m$ is the degree matrix defined as $\mathbf{D}_g^m(i,i) = \sum_j^{n_m} \mathbf{W}_g^m(i,j)$. The laplacian matrix $\mathbf{L}_g^m$ is positive semi-definite and symmetric. An example of the final matrix $\mathbf{D}_g$ and $\mathbf{L}_g$ are below:

$$\mathbf{D}_g = \begin{bmatrix} D_g^1 & 0 & 0 & 0 \\ 0 & D_g^2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & D_g^M \end{bmatrix} \qquad \text{(IV.6)}$$

and

$$\mathbf{L}_g = \mathbf{D}_g - \mathbf{W}_g = \begin{bmatrix} \mathbf{L}_g^1 & 0 & 0 & 0 \\ 0 & \mathbf{L}_g^2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \mathbf{L}_g^M \end{bmatrix}. \qquad \text{(IV.7)}$$

The term G in equation IV.2 that needs to be minimized is now:

$$G = \sum_{m=1}^{M} \sum_{i,j=1}^{n_m} W_g^m(i,j) ||f^{mT} x_i^m - f^{mT} x_j^m||^2 \qquad \text{(IV.8)}$$

which is equivalent [44] to:

$$G = tr(F^T \mathbf{X} \mathbf{L}_g \mathbf{X}^T F). \qquad \text{(IV.9)}$$

So if two data points (or instances) are similar in the original space, then they should be similar in the shared latent space.

IV.1.4 Class Label Similarity Term

The S term in equation IV.2 ensures that the inter-domain classes are brought together in the joint latent space by pulling labeled samples together. Tuia et al. [6] uses a matrix of class similarities to act as an affinity matrix. Entries in the matrices $\mathbf{W}_s^{m,m'}(i,j) = 1$ if the samples between class m and m' have the same label and 0 if not, for $m, m' = 1, 2, \ldots, M$ because this affinity matrix only compares the labeled samples, we can be sure that this matrix will be very sparse.

A graph Laplacian, $\mathbf{L}_s$ is constructed from the affinities $\mathbf{W}_s^{m,m'}$. The term that needs to be minimized is now:

$$S = \sum_{m,m'=1}^{M} \sum_{i,j=1}^{l_m, l'_m} W_s^{m,m'}(i,j) ||f^{mT} x_i^m - f^{mT} x_j^m||^2 \tag{IV.10}$$

$$= tr(F^T \mathbf{X} \mathbf{L_s} \mathbf{X^T} F). \tag{IV.11}$$

IV.1.5 Class Label Dissimilarity Term

The B term in IV.2 ensures that the inter-domain classes are pushed apart in the joint latent space by pushing differently labeled samples apart. Tuia et al. [6] uses a matrix of class dissimilarities to act as an affinity matrix. Entries in the matrices $W_d^{m,m'}(i,j) = 1$ if the samples between class m and m' have different labels and if not, for $m, m' = 1, 2, \ldots, M$, because this affinity matrix is only comparing the labeled samples, we can be sure that this matrix will be very sparse.

A graph Laplacian, $\mathbf{L}_d$ is constructed in the same way as the geometric term is. The term that needs to be minimized is now:

$$B = \sum_{m,m'=1}^{M} \sum_{i,j=1}^{l_m, l'_m} W_d^{m,m'}(i,j) ||f^{mT} x_i^m - f^{mT} x_j^m||^2 \tag{IV.12}$$

$$= tr(F^T \mathbf{X} \mathbf{L_d} \mathbf{X^T} F). \tag{IV.13}$$

IV.1.6 Projection Functions

We can aggregate all of the graph Laplacians $\mathbf{L_g}, \mathbf{L_s}, \mathbf{L}_d \in \mathbb{R}^{n \times n}$ and then use the Rayleigh Quoitient formulation to get our cost function that needs to be minimized:

$$F_{opt} = \underset{F}{argmin} \left\{ tr \left((\mathbf{F^T X L_d X^T F})^{-1} \mathbf{F^T X} (\mu \mathbf{L_g} + \mathbf{L_s}) \mathbf{X^T F} \right) \right\}. \tag{IV.14}$$

We can find a solution to this minimization problem by looking for the smallest eigenvalues γ_i of the generalized eigenvalue problem:

$$\mathbf{X}(\mu \mathbf{L_g} + \mathbf{L_s}) \mathbf{X^T} \Gamma = \lambda \mathbf{X L_d X^T} \Gamma. \tag{IV.15}$$

The optimal solution to the generalized eigenvalue problem contains the projection functions necessary to project each $\mathbf{X}^m$ into the joint latent space. If we look at F_{opt}, we can see the matrix will be built as follows:

$$F_{opt} = \left[\sqrt{\lambda_1} \gamma_1 | \dots | \sqrt{\lambda_M} \gamma_M \right] \tag{IV.16}$$

$$F_{opt} = \begin{bmatrix} f_1^1 & \cdots & f_d^1 \\ f_1^2 & \cdots & f_d^2 \\ \vdots & \vdots & \vdots \\ f_1^M & \cdots & f_d^M \end{bmatrix}. \tag{IV.17}$$

To put it succinctly, we have a projection of $\mathbf{X}^m$ from domain m to the joint latent space F of dimension, d:

$$\mathcal{P}_f(\mathbf{X}^m) = f^{mT} \mathbf{X}^m$$

Furthermore, any data matrix X^m can be projected into the latent space of data matrix $\mathbf{X}^{m'}$. Let $\mathbf{X}^p = \mathbf{X}^m f^m$ be the projection of data matrix X^m into the joint latent space. This can be translated into the latent space of $\mathbf{X}^{m'}$ via the translation $\mathbf{X}^{m1} = \mathbf{X}^m f^m (f^{m'})^\dagger$ where $(f^{m'})^\dagger$ is the pseudoinverse of the eigenvectors of domain m'.

IV.2   Schroedinger Eigenmaps for Manifold Alignment

Tuia et al. [6] paved the way for using manifold alignment for remote sensing applications. As discussed in chapter IV, the SSMA framework is solid with evidence to explain the authors effectiveness with HSI fusion. Wang et al. [5] originally proposed the method that the SSMA framework is based off of. The original generalized eigenvalue decomposition proposed by Wang is

$$\mathbf{X^T}(\mathbf{L_g} + \mu\mathbf{L_s})\mathbf{XF} = \lambda\mathbf{X^T}\mathbf{D_g}\mathbf{XF}. \tag{IV.18}$$

The $\mathbf{L}_s$ matrix is defined exactly as the SSMA; however, the normalization term, $\mathbf{D}_g$, is still the degree matrix of the geometric Laplacian. This is very similar to the SE framework discussed in Chapter III where the potential term, $\mathbf{V}$ enables semisupervision. So we are setting up two comparisons: the first method implements the original manifold alignment equation IV.18 from Wang et al. [5] one method implements the original Wang method which utilizes a degree normalization term and the other method implements Wang et al.'s method

$$\mathbf{X^T}((\mathbf{L_g} + \alpha\mathbf{V}) + \mu\mathbf{L_s})\mathbf{XF} = \lambda\mathbf{X^T}\mathbf{D_g}\mathbf{XF} \tag{IV.19}$$

with an additional potential term, $\mathbf{V}$, to enhance the geometric component of the manifold alignment scheme.

IV.2.1   Normalization

In graph-based methods, the objective is to find a partition of the graph that minimizes some graph-cut; an arbitrary method which minimizes the quality of the partition [45]. A normalization term is a direct result of a continuous relaxation heuristic that approximately solves the graph-cut we use. The normalization term, typically $\mathbf{D}$, used in the Kernel Eigenmap methods has not been explored thoroughly in the community. In early clustering methods, the ratio cut [46] used a simple identity matrix for normalization. By using the Identity matrix, we optimize the objective function to the number of nodes in each cluster. The normalized cut metric [47] used the degree matrix which constrained the degrees of each vertex on the diagonal entries. Results typically seemed to be better when the degree of the vertex is used in the normalization, as it is the shifts the importance to vertices that have more connections. In the Kernel Eigenmap methods, Johnson et

al. [45] studied the affect of different normalization parameters and found that the identity matrix performed the least in classification experiments compared to the degree and a new implemented K-Scaling parameter.

A related issue comes up in the LPP scheme which stems from having more dimensions than samples. Since LPP is a kernel eigenmap projection method, it suffers from possibly having more dimensions than samples; more so than the embedding methods. Yu et al. [48] stated that the nullspace of the normalization matrix could be large whereas the nullspace of the Laplacian matrix could be small. Hence it is disadvantageous to disregard information from the Laplacian matrix because of the wealth of information it could hold. Yu et al. [48] and Kittler et al. [49] introduced a method that utilized a joint diagonalization scheme to preserve as much information as possible from the Laplacian component.

In both special cases of the spectral methods, ratio cuts and normalized cuts, the normalization parameter could have an important impact on the classification results of HSI. The SSMA scheme uses the difference in labeled instances between domains which inherently leaves the normalization very sparse with a large nullspace. The degree matrix is full rank so you will never divide by zero. However, the dissimilarity matrix from [6] is almost never full rank which means that you can get division by zero in the numerical computation. Furthermore, utilizing the degree matrix should ensure that information is preserved in the geometric and similarity terms. This equation is seen in IV.18 and also in the SE framework.

IV.2.2 Spatial-Spectral Fusion

There have been many attempts to incorporate spatial information into the LE methodology for HSI classification. This can be done by constructing heat kernels that combine the spatial and spectral information. Cahill et al. [39] successfully constructed a cluster potential matrix that would encode the proximity between the spatial components of the vertices. This was able to increase the classification accuracy of the Indian Pines and Pavia HSI. The cluster potential was constructed as:

$$\mathbf{V} = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathcal{N}_\epsilon^p(\mathbf{x}_i)} \mathbf{V}(i,j) \omega_{i,j} e^{-\frac{||x_i^p - x_j^p||^2}{\sigma_p^i}} \quad , \tag{IV.20}$$

where $\mathcal{N}_\epsilon^p(\mathbf{x}_i)$ is the set of points in the ϵ-neighborhood of $\mathcal{X}$, $\omega(i,j) = e^{-\frac{||x_i^f - x_j^f||^2}{\sigma_f^2}}$, and $V(i,j)$ can be defined as:

$$V(i,j)_{k,l} = \begin{cases} 1, & (k,l) \in \{(i,i),(j,j)\} \\ -1, & (k,l) \in \{(i,j),(j,i)\} \\ 0, & \text{otherwise} \end{cases} \quad . \tag{IV.21}$$

The $\mathbf{V}(i,j)$ term incorporated the spectral information of the pixels and One distinct advantage of the SSSE methodology is the ability to change the impact of the spatial-spectral information by the use of the parameter α or by the $\omega(i,j)$.

IV.2.3 New Cost Function with Spatial-Spectral Potential

The original Wang method [5] provides us with the final cost function to compare to is defined in equation IV.18. The SEMA method uses that normalization principle in the SSMA framework with the addition of the potential term, $\mathbf{V}$. Finally, the SEMA method solves:

$$\mathbf{F}_{opt} = \underset{F}{\operatorname{argmin}} \frac{\mathbf{F^T X}(\mathbf{L_s} + \mu\mathbf{L_g} + \alpha\mathbf{V})\mathbf{X^T F}}{\mathbf{F^T X L_d X^T F}} \quad . \tag{IV.22}$$

Critical points of the cost function IV.22 can be found by solving the following generalized eigenvalue decomposition in equation IV.19 where the only visible change is the included potential term $\mathbf{V}$ to augment the geometric term $\mathbf{L_g}$.

IV.2.4 Hypothesis

The SSSE method has demonstrated its effectiveness and is competitive with current state of the art Kernel Eigenmap methods. SSMA also uses the Laplacian framework and the LPP methodology to construct an effective method which fuses multiple HSIs with shared partial labels between them. By incorporating the spatial spectral term into manifold alignment, there should be an increase in classification accuracy over the traditional SSMA algorithm. The normalization parameter is present in [5] and in [4] as an important component of the spectral decomposition. The following experiments should show whether or not this normalization parameter is important for getting more accuracy classification results.

V. Experiment

V.1 Dataset I

(a) False Colour Composite Image   **(b)** Ground Truth
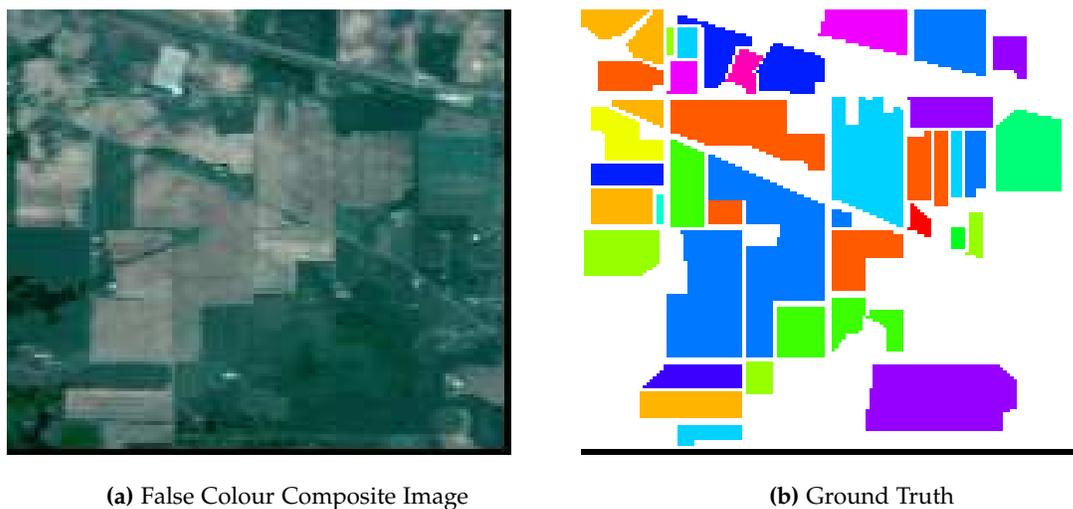
Figure 11: Indian Pines HSI

The this experiment, I have considered two data sets: a HSI of the well-known Indian Pines imagery as well as the also well-known Pavia University imagery. The Indian Pines imagery was acquired by the AVIRIS sensor and consists of 220 spectral bands. The Pavia Centre imagery was acquired from the ROSIS sensor and consists of 103 spectral bands. For illustrative purposes, figure 11 and 12 show a false color composition of the HSI data and the ground truth available for the respective datasets. Table 1 and 2 shows the details for the class names and available training and testing samples for both datasets respectively. These datasets are available online [9] as a community benchmark for comparing various machine learning algorithms in other applications a kin to this study.

V.2 Dataset II

In this experiment, I have considered a HSI dataset provided by the Virginia Commonwealth University [50]. This RS imagery covers the entire Hog Island area but a sub-sample was taken. The first image was captured at an altitude of 2000m above the ROI with 96 spectral bands collected. The second image was captured at an altitude of 400m above the ROI with 48 spectral

---

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

(a) False Colour Composite Image

(b) Ground Truth

Figure 12: Pavia HSI

(a) False Colour Composite Image
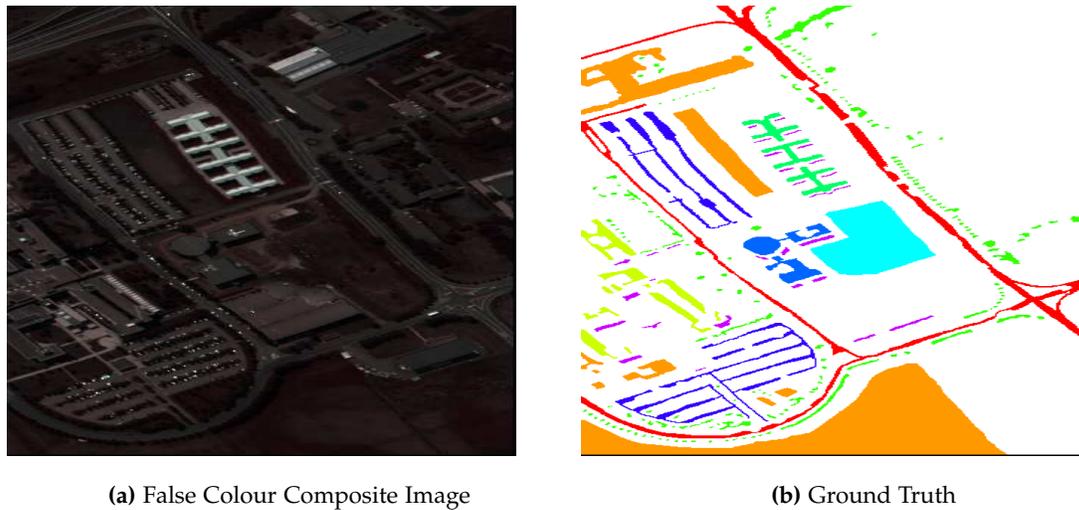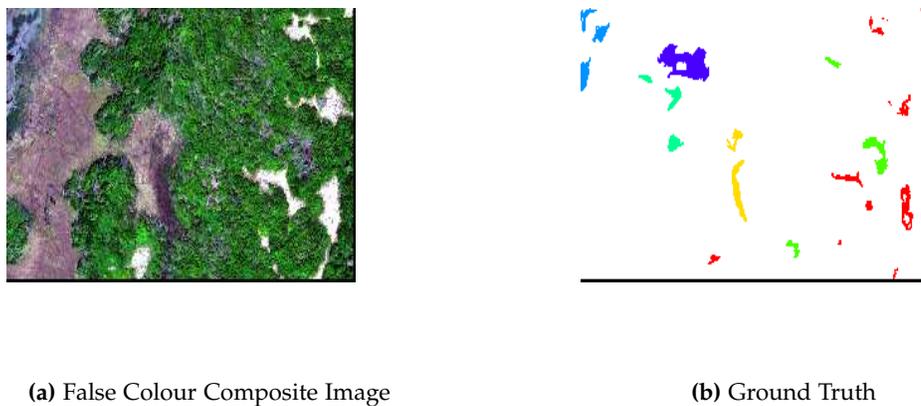
(b) Ground Truth

Figure 13: VCU Image I

bands collected. Both images are georeferenced to the same datum but they are not necessarily over the exact same ROI. There is partially labeled ground truth available with 7 classes.

V.3 Classifiers

This study aims to address the feasibility and effectiveness of the SEMA method in HSI. The manifold alignment techniques require labeled samples from either domain so standard supervised classifiers are used. The first classifier, Linear Discriminant Analysis (LDA), is known for its discriminative power. The second method, Support Vector Machine (SVM), is known for its

(a) False Colour Composite Image   **(b)** Ground Truth

Figure 14: VCU Image II

kernel nature. Although the exact details of these algorithms are beyond the scope of this thesis, please refer to [51] and [52] for more information regarding these techniques. In a nutshell, SVM is a binary classification method that kernelizes the data to project nonlinear data into a higher dimension where the classes are linearly separable whereas LDA is a linear transformation technique which maximizes the class-separation.

V.4 Experiment I

In this experiment, I compare the per class and overall classification accuracy of the two HSIs mentioned in section V.1 for the LE, LPP, SE and SEP algorithms. The objective of this experiment was to show the validity of the SEP algorithm as an extension to the original SE algorithm. A secondary objective is to see how LPP performs as a derivative of the original LE algorithm. The kernel eigenmap algorithms were used as the dimensionality reduction step to transform the data into a lower dimensional space. Subsequently, approximately 10% data was used for each class to train the SVM classifier and the remaining 80% was used to test how well the classifier was able to predict the remaining classes. Prior to dimensionality reduction, the data was normalized to a norm of 1. The k and σ parameter for the heat equation was kept constant at 20 and the suggested value from [39].

V.5 Experiment II

In this experiment, I compare and contrast the use of the degree matrix and dissimilarity matrix within the manifold alignment framework and I investigated how the spatial-spectral potential matrix in the SEMA algorithm affects the geometric matrix overall accuracy of the classification methods. The objective of this experiment was to see if normalization using the degree matrix will increase the classification accuracy per domain. The original method by Wang [5] used the degree matrix, whereas the new SSMA method from [6] used the dissimilarity matrix for the normalization. I choose 40% of the ground truth data to be used for labeled and unlabeled data. I then choose four cases of splitting that data between labeled and unlabeled data. Case 1 has 10% labeled data for both domains. Case 2 has 10% labeled and 50% unlabeled for domains 1 and 2 respectively. Case 3 has 50% labeled and 10% labeled for domains 1 and 2 respectively. Case 4 has 50% labeled and 50% labeled for domains 1 and 2 respectively. The values of μ and α have been chosen as arithmetically log spaced parameters between -2 and 1. Prior to dimensionality reduction, the data was also normalized. It is worth noting that this type of normalization is not appropriate to apply independently to different images when comparing classification results across a set of images. This is because the magnitudes of the spectral components of the data carry useful class-specific information that can be distored or lost when different images are normalized differently. The k and σ parameter for the heat equation was kept constant at 20 and the suggested value of 1 from [39].

VI. RESULTS

VI.1 Experiment I

VI.1.1 Best Results

(a) Laplacian Eigenmaps

(b) Schroedinger Eigenmaps

(c) Locality Preserving Projections

(d) Schroedigner Eigenmap Projections
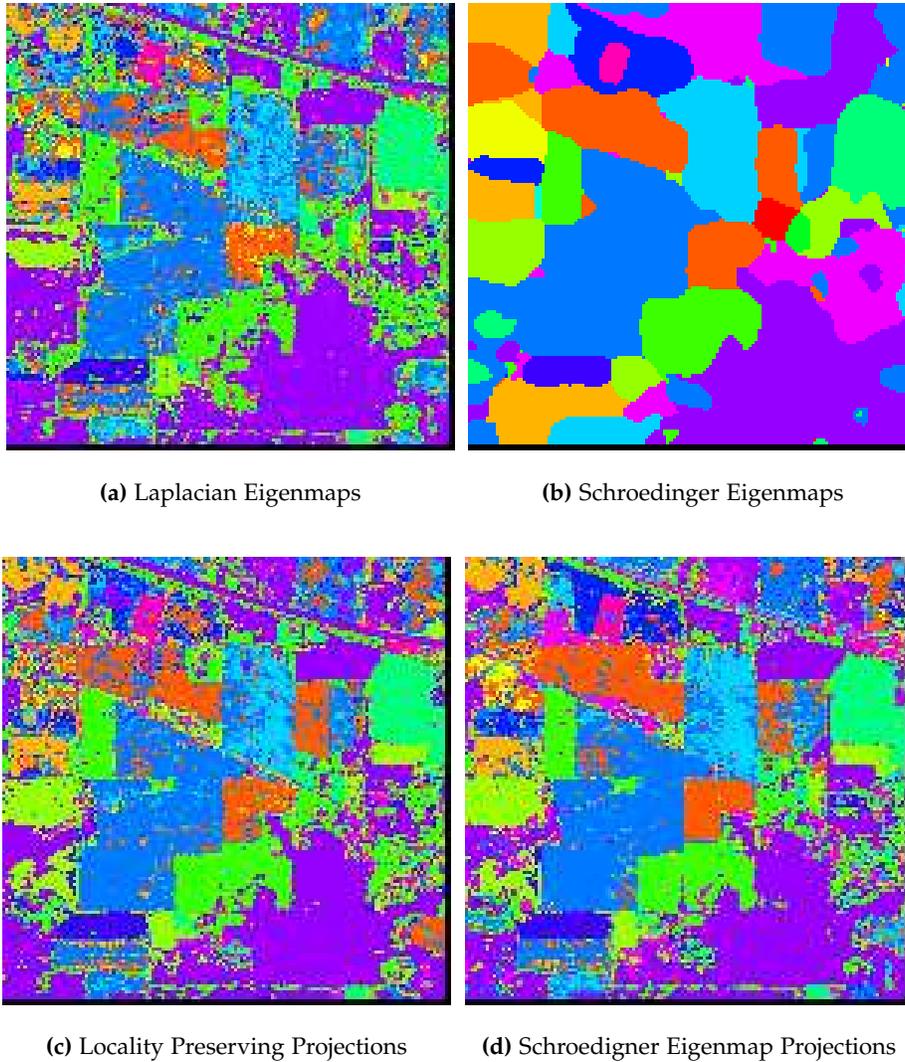
Figure 15: Classification Maps for Indian Pines Using SVM for each of the different Manifold Learning algorithms.

Table 1: Best Results for the Indian Pines HSI

Class	Train	Test	LE	SE	LPP	SEP
Alfalfa	5	41	**1.00**	**1.00**	**1.00**	**1.00**
Corn-notill	143	1285	0.87	**0.99**	0.89	0.93
Corn-mintill	83	747	0.92	**1.00**	0.94	0.95
Corn	24	213	0.97	**1.00**	0.98	0.98
Grass-Pasture	49	434	0.98	**1.00**	0.98	0.99
Grass-Trees	73	657	0.97	**1.00**	0.98	0.99
Grass-Pasture-Mowed	3	25	**1.00**	**1.00**	**1.00**	**1.00**
Hay-Windrowed	48	430	0.99	**1.00**	0.99	0.99
Oats	2	18	**1.00**	**1.00**	**1.00**	**1.00**
Soybean-notill	98	874	0.90	**0.99**	0.83	0.87
Soybean-mintill	246	2209	0.82	**0.99**	0.83	0.87
Soybean-clean	60	533	0.93	**1.00**	0.94	0.97
Wheat	21	184	0.99	**1.00**	**1.00**	**1.00**
Woods	127	1138	0.98	**1.00**	0.95	0.98
Buildings-Grass-Trees-Drives	39	347	0.96	**1.00**	0.97	0.98
Stone-Steel-Towers	10	83	**1.00**	**1.00**	**1.00**	**1.00**
Overall Accuracy			0.64	**0.98**	0.67	0.78
Average Accuracy			0.95	**1.00**	0.95	0.97
Average Precision			0.54	**0.97**	0.56	0.64
Average Specificity			0.97	**1.00**	0.98	0.98
κ			0.58	**0.98**	0.62	0.75
κ **Variance**			0.00	0.00	0.00	0.00
Dimensions			50	50	50	50
Time			38.4	69.1	**20.8**	21.1

Table 1 shows the best results for the Indian Pines HSI and figure 15 shows the different classification maps for the Indian Pines HSI. These are the best results found using the specified methods from experiment I. The SE algorithm had the best overall results and this is apparent because it has the most homogeneous regions of all the classmaps. The LE algorithm did the worst, which is evident from the presence of a lot of speckled regions. The LPP and the SEP method have better overall accuracy than the LE method but worse accuracy than the SE method. The SEP method was able get slightly greater accuracy than the LPP method at lower dimensions but the LPP method would get comparable or better accuracy with more dimensions in the embedding.
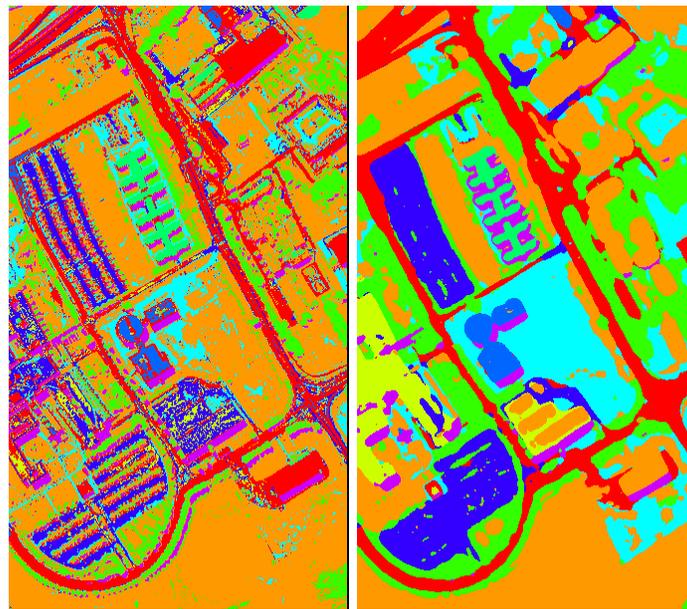
(a) LE   **(b)** SE
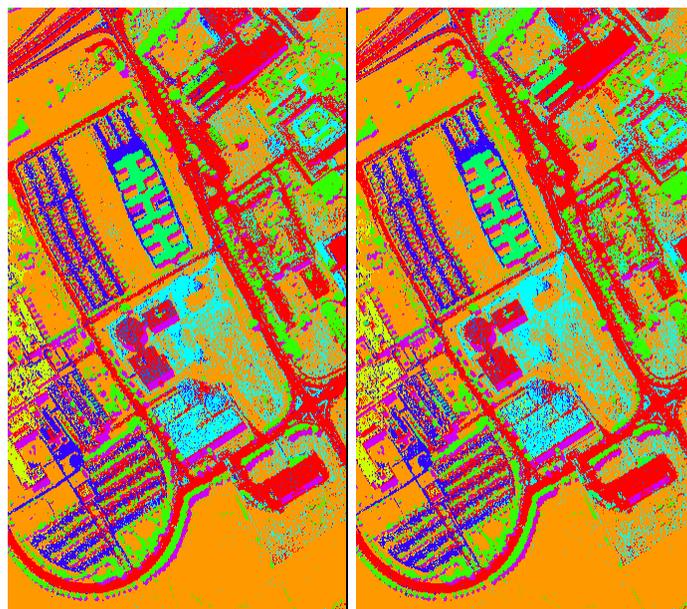
(c) LPP   **(d)** SEP

Figure 16: Classification Maps for Pavia Using SVM for each of the different Manifold Learning algorithms.

Table 2 shows the best results for the Pavia HSI and figure 16 shows the different classification maps

Table 2: Best Results for the Pavia HSI

Class Name	Train	Test	LE	SE	LPP	SEP
Trees	664	5967	0.96	**1.00**	0.95	0.94
Asphault	1865	16784	0.89	**1.00**	0.93	0.93
Bitumen	210	1889	0.95	**1.00**	0.97	0.97
Gravel	307	2757	0.98	**1.00**	0.99	0.99
Metal Sheets	135	1210	1.00	**1.00**	1.00	1.00
Shadow	503	4526	0.91	**1.00**	0.93	0.94
Bricks	133	1197	0.98	**1.00**	0.97	0.97
Meadows	369	3313	0.96	**1.00**	0.95	0.95
Bare Soil	95	852	1.00	**1.00**	1.00	1.00
Overall Accuracy			0.81	**1.00**	0.85	0.85
Average Accuracy			0.96	**1.00**	0.97	0.97
Average Precision			0.76	**0.99**	0.75	0.75
Average Specificity			0.98	**1.00**	0.98	0.98
κ			0.74	**0.99**	0.79	0.80
κ **Variance**			0.00	0.00	0.00	0.00
Dimensions			50	50	50	50
Time			852.6	1136.3	**591.4**	597.4

for the Pavia HSI. These are the best results found using the specified methods from experiment I. The LPP method had the best classification results at around 10 dimensions. However, after 15 dimensions the SE algorithm had the best overall results with every number of dimensions following. Similar to Indian Pines, the LE algorithm did the worst for all dimensions and the LPP and the SEP method has a better overall accuracy than the LE method. The SEP method has worse classification accuracy than the LPP method in earlier dimensions but the SEP method would get comparable for better classification accuracy in later dimensions. There also is less variability in the results of the SEP method as after 10 dimensions, the classification accuracy stays constant.

VI.1.2 Sensitivity Analysis

(a) Indian Pines **(b)** Pavia
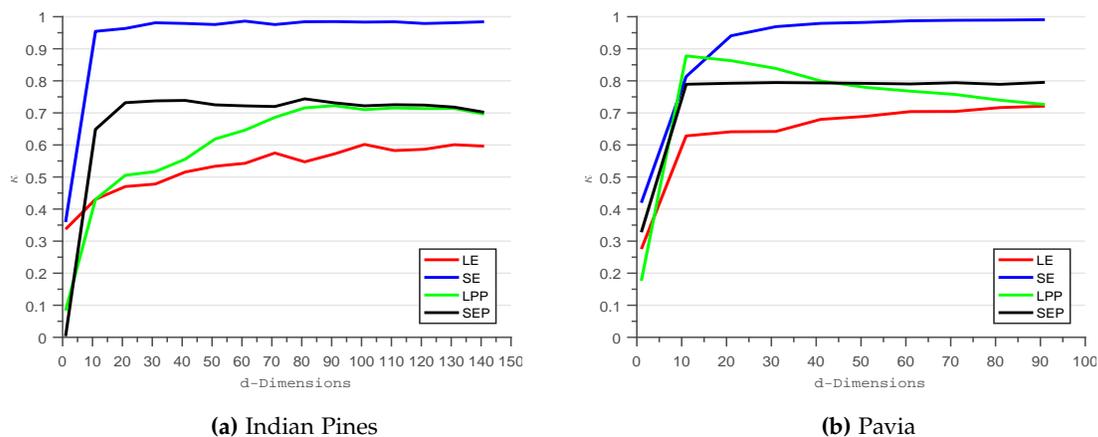
Figure 17: Sensitivity Analysis for the dimensions and κ coefficient.

Figure 17 compares the number of dimensions and the κ coefficient for the different manifold learning techniques. SE has the best overall results at the earliest dimensions for both Indian Pines and Pavia HSIs. The SE method also reaches a plateau very early; within the first 10 dimensions for Indian Pines and within the first 30 dimensions for Pavia. The SEP method reaches its peak κ coefficient of $\sim$0.73 at around 20 dimensions for the Indian Pines image whereas it reached its peak of $\sim$0.8 at around 10 dimensions. The LPP method does the worst for Indian Pines and does not reach its peak of $\sim$0.7 at around 80 dimensions. It does better for Pavia with a peak of $\sim$0.89 at around 10 dimensions. The LE method has the worst κ coefficient out of all the dimensionality reduction methods. It has a peak of $\sim$0.6 at 100 dimensions for indian pines versus a peak of $\sim$0.7 at 90 dimensions.

(a) Schroedinger Eigenmaps

(b) Schroedinger Eigenmap Projections
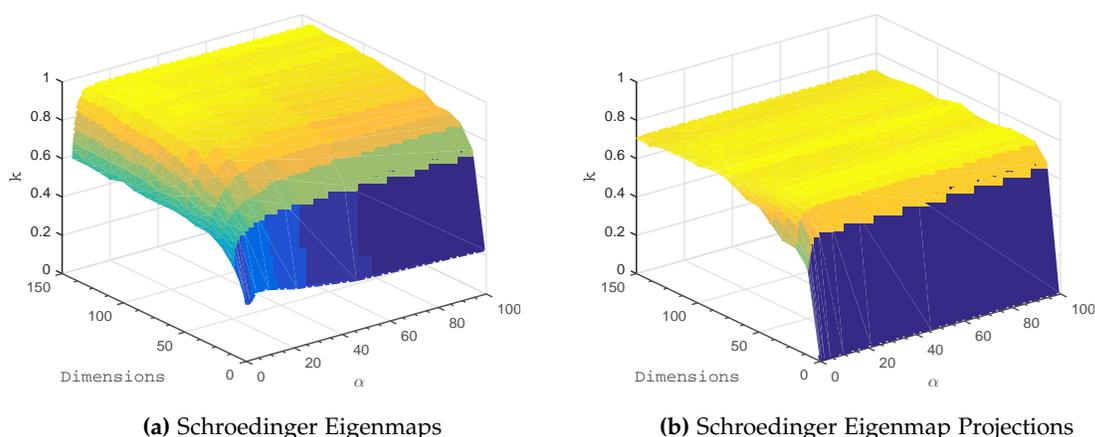
Figure 18: Sensitivity Analysis for the α parameter with respect to the number of dimensions and κ coefficient. The HSI is Indian Pines using the SVM classifier.

Figure 18 shows the α parameter with respect to the number of dimensions and the κ coefficient for the Indian Pines HSI. The SE method has the best LDA classification accuracy after 20 dimensions. The best α values are within 20. The κ coefficient starts to plateau within 50 dimensions and and an α value of 20. However, notice that there is still a steady increase in the κ coefficient as the α value and the κ coefficient increases. The SVM classification algorithm reaches its peak κ coefficient sooner than the SVM method at 25 dimensions with an α value of $\sim$20.

(a) Schroedinger Eigenmaps

(b) Schroedinger Eigenmap Projections
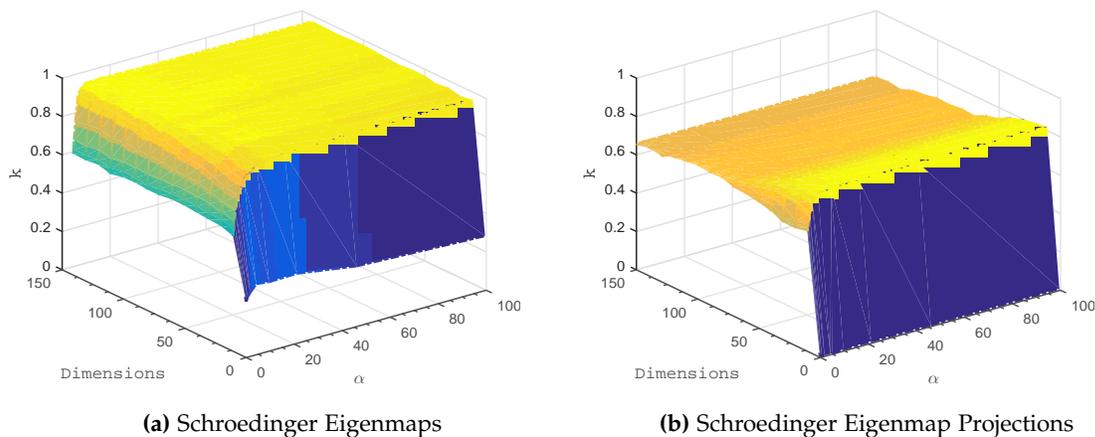
Figure 19: Sensitivity Analysis for the α parameter with respect to the number of dimensions and κ coefficient. The HSI is Indian Pines using the LDA classifier.

The SEP algorithm has the best κ coefficient after an α value of $\sim$10 and dimensions $\sim$20. Unlike the SE method, once the plateau is reached, there is no noticeable steady increase in the κ

coefficient. The SEP algorithm for the SVM classification method has the best κ value in earlier dimensions of 10 but decreases classification accuracy following 10 dimensions. The low α value does not increase the κ coefficient significantly after a value of ∼10.

(a) Schroedinger Eigenmaps

(b) Schroedinger Eigenmap Projections
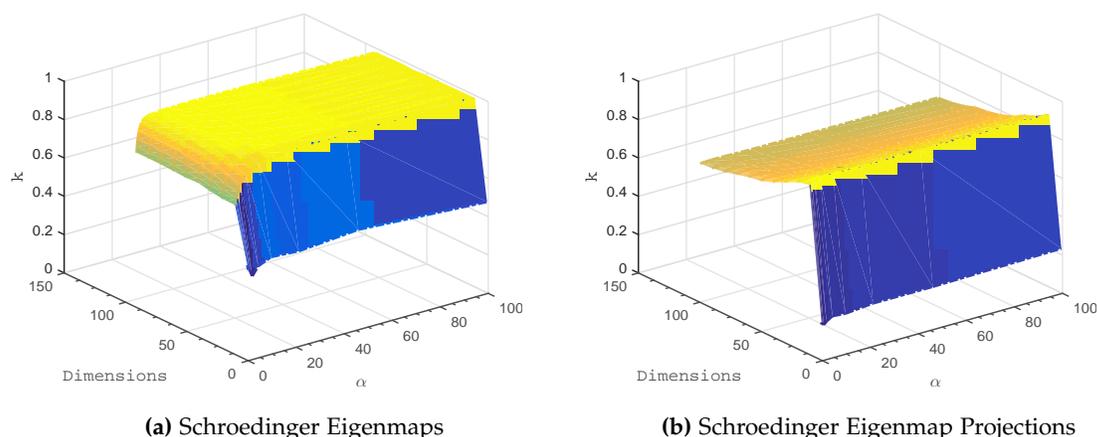
Figure 20: Sensitivity Analysis for the α parameter with respect to the number of dimensions and κ coefficient. The HSI is Pavia using the LDA classifier.

Figure 20 shows the κ coefficient with respect to the number of dimensions and the α coefficient for the Pavia HSI. The LDA method reaches a plateau κ coefficient at ∼20 dimensions. The α value does not significantly increase or decrease the κ coefficient if the number of dimensions are high enough. However, at lower dimensions there is a slight fluctuation in the κ coefficient depending upon the α value. The SEP method has a peak κ coefficient at ∼20 dimensions. However, after the peak there is a sharp decrease in the κ coefficient. The α parameter does not seem to cause the κ coefficient to fluctuate significantly at its peak.

VI.2 Experiment II

VI.2.1 Sensitivity Analysis

Figure 21: κ for different values of μ for the Wang implementation for each source domain. The values range between 0 and 1. The top row is case I with 10% labeled samples for both domains. The bottom row is case II with 10% labeled samples for domain I and 50% labeled samples for domain II. (a) and (c) are for Domain I. (b), and (d) are for Domain II.
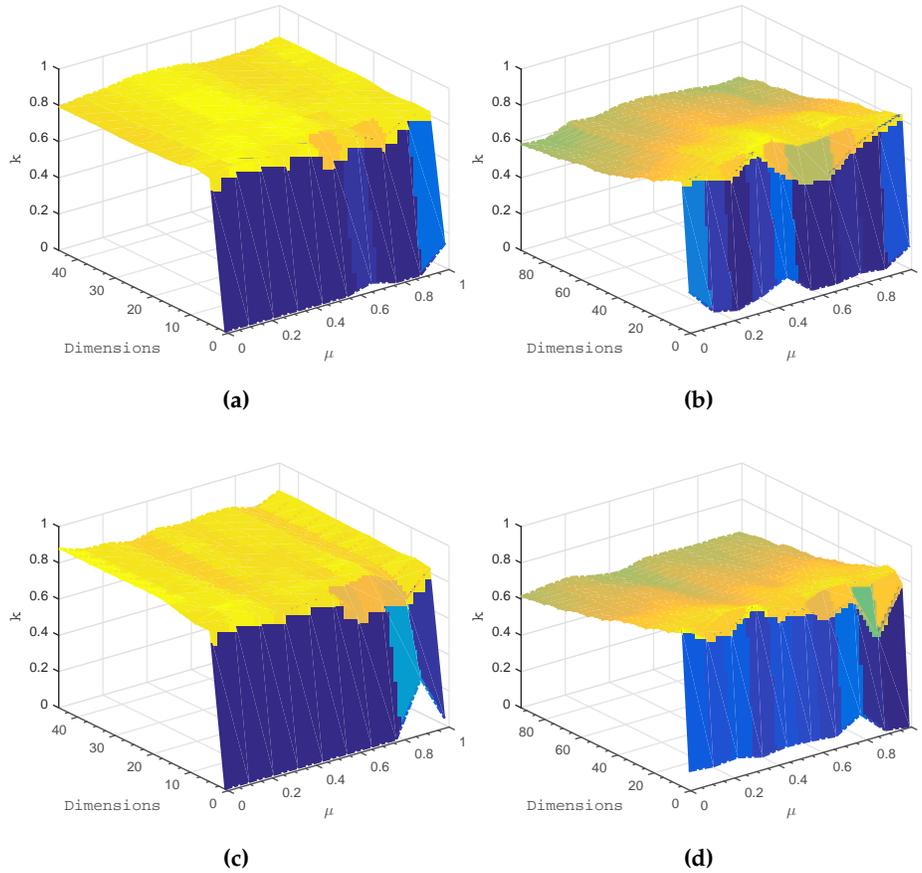
Figure 22: κ for different values of μ for the Wang implementation for each source domain. The values range between 0 and 1. The top row is case III with 50% labeled samples for domain I and 10% labeled samples for domain II. The bottom row is case IV with 50% labeled samples for both domains. (a) and (c) are for Domain I. (b), and (d) are for Domain II.

Figure 21 and 22 shows the results of different μ values for the κ coefficient versus the number of dimensions used in the latent space classification using the original Wang implementation. Similarly, Figure 23 and 24 shows the same results but for the SSMA algorithm. Immediately, one notices the difference in κ coefficient and the number of dimensions between the two domains. Domain I had a higher spatial resolution and domain II had a higher spectral resolution. As the number of dimensions increase for domain II, the κ coefficient decreases whereas the κ coefficient decreases as the number of dimensions increase for domain I. Otherwise, the results for both the Wang and the SSMA algorithm have very similar results with very little differences between values of μ. In general, there is no best μ parameter as there is no discernible highest κ coefficient, μ and

dimension combination that cannot be found in another figure. The key experimental parameter between subsequent figures is the amount of labeled samples used. For this dataset, increasing the number of labeled samples does not appear to have significant changes on the κ coefficient as there is some value of μ and number of dimensions to get a high classification accuracy.

(a)

(b)

(c)

(d)
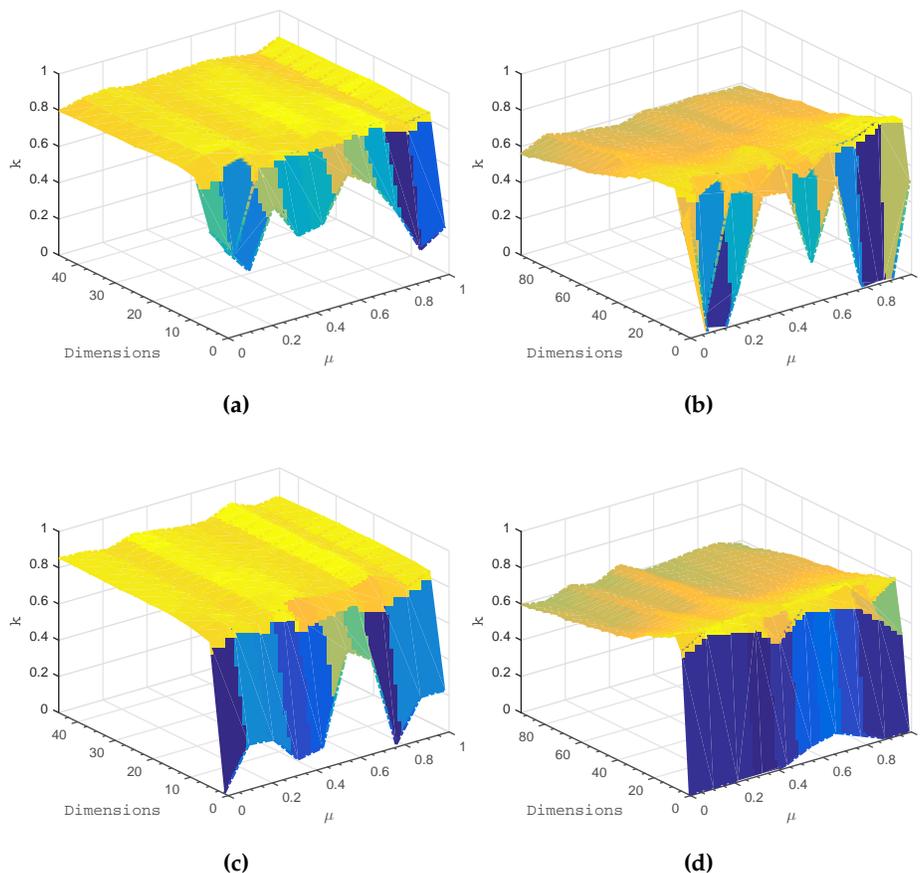
Figure 23: κ for different values of μ for the SSMA implementation for each source domain. The values range between 0 and 1. The values range between 0 and 1. The top row is case I with 10% labeled samples for both domains. The bottom row is case II with 10% labeled samples for domain I and 50% labeled samples for domain II. (a) and (c) are for Domain I. (b), and (d) are for Domain II.
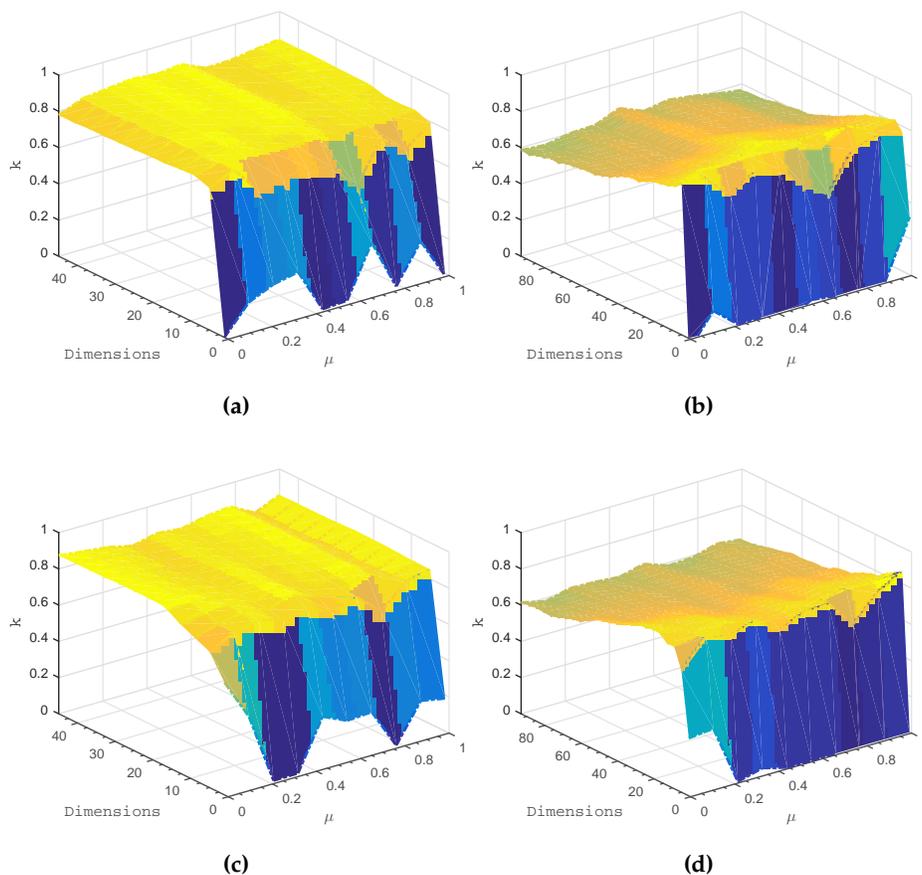
Figure 24: κ for different values of μ for the Wang implementation for each source domain. The values range between 0 and 1. The top row is case III with 50% labeled samples for domain I and 10% labeled samples for domain II. The bottom row is case IV with 50% labeled samples for both domains. (a) and (c) are for Domain I. (b), and (d) are for Domain II.

VI.2.2   Best Results

Table 3 shows the best results for the Manifold Alignment experiment. On a per class accuracy, the SSMA had the highest accuracy compared to the Wang and SEMA implementation. The SSMA method had the highest overall accuracy and kappa coefficient. Figures 25, 26, 27, and 28 show the sensitivity analysis when trying to find the best parameter for the SEMA algorithm. There are certain α values that provide a slight increase in the κ coefficient for some μ values. However, the α value that is common between all graphs with the best κ coefficient within VCU image I and II is $\sim$20.

Figure 25: κ for different values of α for the SEMA implementation for each source domain. The values range between 0 and 1. The x-axis represents the number of dimensions used for classification. The values range between 0 and 50 dimensions. (a), (c), and (e) are for Domain I. (b), (d), and (f) are for Domain II.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 26: κ for different values of α for the SEMA implementation for each source domain. The values range between 0 and 1. The x-axis represents the number of dimensions used for classification. The values range between 0 and 50 dimensions. (a), (c), and (e) are for Domain I. (b), (d), and (f) are for Domain II.

Figure 27: κ for different values of α for the SEMA implementation for each source domain. The values range between 0 and 1. The x-axis represents the number of dimensions used for classification. The values range between 0 and 50 dimensions. (a), (c), and (e) are for Domain I. (b), (d), and (f) are for Domain II.

(a) **(b)**

Figure 28: κ for different values of α for the SEMA implementation for each source domain. The values range between 0 and 1. The x-axis represents the number of dimensions used for classification. The values range between 0 and 50 dimensions. (a) is for Domain I. (b) is for Domain II.

Table 3: Best Results of the Manifold Alignment Experiment

Class	Train		Test		Wang		SSMA		SEMA	
1	78	78	5883	1013	0.96	**0.98**	0.95	**0.98**	0.96	0.98
2	127	127	8225	3094	0.93	0.93	0.93	0.94	0.94	**0.95**
3	85	85	5114	2450	**0.96**	0.92	**0.96**	0.94	0.95	0.94
4	88	88	5704	2130	0.97	0.94	**0.98**	0.95	0.96	0.96
5	102	102	6761	2259	0.95	0.94	**0.97**	0.95	0.95	0.96
6	101	101	6686	2311	0.96	0.96	**0.98**	0.97	0.97	0.97
7	46	46	2859	1256	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**
Overall Accuracy					0.86	0.83	**0.88**	0.85	0.86	0.87
Average Accuracy					0.96	0.95	**0.97**	0.96	0.96	0.96
Average Precision					0.85	0.85	0.87	0.86	0.85	**0.89**
Average Specificity					**0.98**	0.97	**0.98**	0.97	**0.98**	**0.98**
κ					0.83	0.80	**0.86**	0.83	0.83	0.85
κ **Variance**					0.00	0.00	0.00	0.00	0.0	0.0
Dimensions					10					
Time					4.6		8.61		4.76	

VI.3 Discussion

This study has addressed the notion of using the Schroedinger Operator in the LPP framework as well as in the manifold alignment framework. The LPP framework results dictate that the SEP method was more successful in accuracy than the traditional LE and LPP methods for both the Indian Pinesand Pavia but only better than the LPP method for the Indian Pines. This can be explained because the Indian Pines HSI appears to have more classes in larger portions whereas the Pavia HSI has less density in a class region. The SEP method proves to be effective at increasing the classification accuracy of a HSI with more dense class placement than the LPP method. However, this does not seem to be effective in a HSI with less densely placed classes. The alignment framework dictates that the methods do not necessarily increase or decrease the accuracy in a significant manner. The SEMA method did better in some specific regions but was unable determine a specific region where it had better classification accuracy.

The key difference between the SSMA method and the methods that I used was the exclusion of the dissimilarity matrix. There was no component in the methods I proposed that included a matrix that would try to push dissimilar classes apart whereas the SSMA method explicitly creates a matrix to do so. Experiment I showed that the LPP and SEP method were very similar in overall classification accuracy depending on the HSI dataset used. It is possible that the dataset used in Experiment II did not have the same homogeneous regions to ensure the additional Spatial-Spectral term in the SEMA method made a difference. Furthermore, these were all projection methods where there was no vast difference in classification accuracy. But Experiment II showed a vast difference in classification accuracy between the eigenmap methods and the eigenmap projection methods. The comparison between manifold alignment methods that use projection functions versus eigenmap functions would be a good aspect to explore in the future.

The LPP framework was originally introduced in order to obtain explicit projection functions and reduce the overall computational cost. The SE framework took the most computational time for each HSI image, but the accuracy was significantly greater than any other method. This study did not use embedding methods for the alignment stage, so there is no conclusion to be drawn from there. However, it is worth noting that there are alternative methods to cut down the computational cost of kernel eigenmap methods and produce explicit projection methods whilst still retaining accuracy [53] [54] [40] [37]. A good study is needed to showcase the difference speed-up techniques of the embedding methods versus the projection methods.

The strength of this study lies in the first look at the kernel eigenmap methods versus kernel

eigenmap projection methods. This allowed the justification to use the Schroedinger operator argument for the SSMA scheme. This moves the Schroedinger Eigenmaps literature forward and from a different perspective. Some limitations include the rudimentary computational complexity analysis. There could be alternative cases to looking at the entire dimension reduction - classification pipeline process; for example, by performing dimension reduction only on training data and then projecting the unreduced data into the embedded space. Finally, there are never enough practical datasets with good reference data to really validate the robustness of this method for industry.

VII. Conclusion

This objective of this thesis was to build upon the Tuia et al.'s [6] SSMA framework by leveraging the components from the SE algorithm in multimodal HSI classification. The modifications came from the SSSE algorithm where expert knowledge can increase the classification accuracy for HSIs. The SSMA algorithm used the Kernel Eigenmap Projection approach as there explicit are projection functions to map individual datasets to a shared latent space where similar data points are closer together and dissimilar data points are further apart. Therefore, it was necessary to explore the effectiveness of the SSSE algorithm within the LPP framework on HSI classification. Two HSI datasets, Indian Pines and Pavia, were used to assess the quality of the embedding produced by the LE, SE, LPP and SEP dimension reduction algorithms by looking at the κ coefficient obtained from the classifier. Some sensitivity analysis was done to look at how the variation in key parameters effect on the classification accuracy. The LE algorithm performed the worst and the SE algorithm performed the best. The SEP algorithm had a higher κ coefficient than the LPP algorithm for the Indian Pines dataset as it had more homogeneous regions. Conversely, the LPP algorithm performed better than SEP on the Pavia dataset as it had less homogeneous regions. The overall differences in classification accuracy between the LPP and SEP methods was relatively low compared to the difference between the SE algorithm and the other dimension reduction methods. Wang et al. [55] produced the original Manifold Alignment framework and Tuia et al. ultimately used a modified version for multimodal remote sensing classification. This modified version did not include the degree matrix for normalization and instead incorporated a dissimilarity component to ensure that dissimilar data points would be further away from each other in the latent space. Wang et al.'s original algorithm, Tuia et al.'s algorithm, and a modification using a Spatial-Spectral potential was studied by looking at the classification accuracy for two multimodal VCU HSIs. The results showed very little difference between the methods and the sensitivity analysis showed that parameter tuning is necessary to achieve the best κ coefficient. Ultimately, a future study would be to look at incorporating the Kernel Eigenmap framework into the SSMA framework for multimodal remote sensing classification.

VIII. Bibliography

References

[1] J. J. Benedetto and W. Czaja, "Dimension Reduction and Remote Sensing Using Modern Harmonic Analysis," in *Handbook of Geomathematics*, no. Section 2, pp. 1–22, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[2] X. He and P. Niyogi, "Locality preserving projections," *Neural information processing systems*, vol. 16, p. 153, 2004.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering.," *NIPS*, 2001.

[4] W. Czaja and M. Ehler, "Schroedinger eigenmaps for the analysis of biomedical data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1274–1280, may 2011.

[5] C. Wang, *A geometric framework for transfer learning using manifold alignment.* PhD thesis, 2010.

[6] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls, "Semisupervised manifold alignment of multimodal remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7708–7720, 2014.

[7] J. E. Johnson, C. M. Bachmann, and N. D. Cahill, "Manifold alignment with Schrödinger eigenmaps," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXII* (M. Velez-Reyes and D. W. Messinger, eds.), vol. 9840, p. 98401K, may 2016.

[8] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised alignment of manifolds," *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, vol. 10, pp. 120–127, 2005.

[9] G. Cavallaro, M. Riedel, M. Richerzhagen, J. A. Benediktsson, and A. Plaza, "On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, vol. 8, no. December, p. (Accepted for publication 2015), 2015.

[10] C. M. Bachmann, S. Member, T. L. Ainsworth, R. A. Fusina, M. J. Montes, J. H. Bowles, D. R. Korwan, and D. B. Gillis, "Bathymetric Retrieval From Hyperspectral Imagery Using Manifold Coordinate Representations," vol. 47, no. 3, pp. 884–897, 2009.

[11] D. Gillis and E. M. Winter, "Dimensionality Reduction in Hyperspectral Imagery," *Comportamento Espectral De Alvos/Mistura*, vol. 5093, no. 6, pp. 45–56, 2003.

[12] D. Lunga, S. Prasad, M. M. Crawford, O. Ersoy, and H. Data, "Manifold-Learning-Based Feature Extraction for Classification of Hyperspectral Data (A review of advances in manifold learning)," *IEEE Signal Processing Magazine*, vol. 31, no. January 2014, pp. 55–66, 2013.

[13] H. L. Yang and M. M. Crawford, "Manifold alignment for classification of multitemporal hyperspectral data," *Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing*, 2011.

[14] L. Gomez-Chova, D. Tuia, G. Moser, and G. Camps-Valls, "Multimodal Classification of Remote Sensing Images: A Review and Future Directions," *Proceedings of the IEEE*, vol. 103, pp. 1560–1584, sep 2015.

[15] J. M. Bioucas-dias, A. Plaza, G. Camps-valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral Remote Sensing Data Analysis and Future Challenges," *IEEE Geoscience and Remote Sensing Magazine*, no. June, pp. 6–36, 2013.

[16] D. Lahat, T. Adali, and C. Jutten, "Multimodal Data Fusion : An Overview of Methods , Challenges , and Prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[17] B. Du, L. Zhang, D. Tao, and D. Zhang, "Unsupervised transfer learning for target detection from hyperspectral images," *Neurocomputing*, vol. 120, pp. 72–82, nov 2013.

[18] H. L. Yang and M. M. Crawford, "Spectral and Spatial Proximity-Based Manifold Alignment for Multitemporal Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 51–64, jan 2016.

[19] D. Tuia, C. Persello, and L. Bruzzone, "Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 41–57, 2016.

[20] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised Transfer Component Analysis for Domain Adaptation in Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3550–3564, 2015.

[21] X. R. Wang, "Learning and Classification of Hyperspectral Images," no. August, 2008.

[22] H. Vu, C. Carey, and S. Mahadevan, "Manifold Warping: Manifold Alignment over Time," *Aaai*, 2012.

[23] L. Zhu and L. Ma, "Class centroid alignment based domain adaptation for classification of remote sensing images," *Pattern Recognition Letters*, vol. 000, pp. 1–9, jan 2016.

[24] W. Kim, *Manifold Learning for Robust Classification of Hyperspectral Data*. PhD thesis, Purdue University, 2011.

[25] M. Wang, "Atmospheric Correction for Remotely-Sensed Ocean- Colour Products," Tech. Rep. 1998, 2009.

[26] P. Bajorski, "Statistical inference in PCA for hyperspectral images," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 438–445, 2011.

[27] D. Gillis, J. Bowles, G. M. Lamela, W. J. Rhea, C. M. Bachmann, M. Montes, and T. Ainsworth, "Manifold learning techniques for the analysis of hyperspectral ocean data," *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, vol. 5806, pp. 342–351, jun 2005.

[28] X. Wang, B. Qian, and I. Davidson, "On Constrained Spectral Clustering and Its Applications," 2012.

[29] J. Tenenbaum, V. De Silva, and J. Langford, "A Global Geometric Framework for Nonlinear dimensionality Reduction," *Sciencemag.com*, vol. 290, 2000.

[30] L. Saul and S. Roweis, "An introduction to locally linear embedding," *unpublished. Available at: http://www. cs. toronto. . . .* , pp. 1–13, 2000.

[31] 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*, vol. 21, no. 1, pp. 113–127, 2006.

[32] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *Journal of Shanghai University (English Edition)*, 2004.

[33] D. L. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10, pp. 5591–5596, 2003.

[34] X. Huo, X. S. X. Ni, and a. A. K. Smith, "A Survey of Manifold-Based Learning Methods," *Mining of Enterprise Data*, pp. 06–10, 2004.

[35] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[36] X. He, D. Cai, and W. Min, "Statistical and computational analysis of locality preserving projection," in *Proceedings of the 22nd international conference on Machine learning - ICML '05*, (New York, New York, USA), pp. 281–288, ACM Press, 2005.

[37] W. Czaja, A. Hafftka, B. Manning, and D. Weinberg, "Randomized Approximations of Operators and their Spectral Decomposition for Diffusion Based Embeddings of Heterogeneous Data," in *Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, (Pisa), pp. 75–79, 2015.

[38] J. Benedetto, W. Czaja, J. Dobrosotskaya, T. Doster, K. Duke, and D. Gillis, "Semi-supervised learning of heterogeneous data in remote sensing imagery," in *Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X* (H. Szu and L. Dai, eds.), vol. 8401, pp. 840104–840104–12, may 2012.

[39] 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* (M. Velez-Reyes and F. A. Kruse, eds.), p. 908804, jun 2014.

[40] X. Zhang, S. E. Chew, Z. Xu, and N. D. Cahill, "SLIC Superpixels for Efficient Graph-Based Dimensionality Reduction of Hyperspectral Imagery," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, vol. 9472, p. 947209, 2015.

[41] L. P. Dorado Munoz, D. W. Messinger, and W. Czaja, "Assessment of Schrodinger Eigenmaps for target detection," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* (M. Velez-Reyes and F. A. Kruse, eds.), p. 908813, jun 2014.

[42] L. P. Dorado-Munoz and D. W. Messinger, "Schrodinger Eigenmaps for spectral target detection," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI* (M. Velez-Reyes and F. A. Kruse, eds.), vol. 9472, p. 947211, may 2015.

[43] N. D. Cahill, S. E. Chew, and P. S. Wenger, "Spatial-spectral dimensionality reduction of hyperspectral imagery with partial knowledge of class labels," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, vol. 9472, p. 94720S, 2015.

[44] S. J. Wang, H. L. Chen, X. J. Peng, and C. G. Zhou, "Exponential locality preserving projections for small sample size problem," *Neurocomputing*, vol. 74, no. 17, pp. 3654–3662, 2011.

[45] R. Johnson and T. Zhang, "On the effectiveness of Laplacian normalization for graph semi-supervised learning," *Journal of Machine Learning Research*, vol. 8, pp. 1489–1517, 2007.

[46] P. K. Chan, M. Schlag, and J. Zien, "Spectral K -Way Ratio-Cut Partitioning Part I : Preliminary Results," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1088 – 1096, 1992.

[47] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[48] H. Yu, J. Yang, and a. S. A, "A direct LDA algorithm for high-dimensional data * with application to face recognition," *Pattern Recognition*, vol. 34, no. February, pp. 2067–2070, 2001.

[49] J. Kittler, K. Messer, W. Shitong, X. J. Wu, J. Kittler, J. Y. Yang, K. Messer, and S. Wang, "A new direct LDA (D-LDA) algorithm for feature extraction in face recognition," *Proceedings - International Conference on Pattern Recognition*, vol. 4, no. 1, pp. 545–548, 2004.

[50] "Hyperspectral imagery for Hog Island, VA, 2013." `http://www.vcrlter.virginia.edu/cgi-bin/showDataset.cgi?docid=knb-lter-vcr.229`, 2013. [Online; accessed 6-April-2016].

[51] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[52] Q. Du, "Modified Fisher's linear discriminant analysis for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 503–507, 2007.

[53] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "ÂÍ m Method Spectral Grouping Using the Nystro," *Analysis*, vol. 26, no. 2, pp. 214–225, 2004.

[54] A. Talwalkar, S. Kumar, M. Mohri, and H. Rowley, "Large-scale Manifold Learning,"

[55] Y. Wang, "Quantitative Remote Sensing Inversion in Earth Science: Theory and Numerical-Treatment," 2010.