12-2018

Semi-Supervised Normalized Embeddings for Fusion and Land-Use Classification of Multiple View Data

Poppy Immel
pgi8114@rit.edu

Semi-Supervised Normalized Embeddings for Fusion and Land-Use Classification of Multiple View Data

by

Poppy Immel

A Thesis submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Golisano College of Computer and Information Sciences

Rochester Institute of Technology

Rochester, NY

December 2018

Semi-Supervised Normalized Embeddings for Fusion and Land-Use Classification of Multiple View Data

COMMITTEE APPROVAL :

Nathan D. Cahill, D.Phil., School of Mathematics, Advisor	Date
Zack Butler, Ph.D., Department of Computer Science, Advisor	Date
Richard Zanibbi, Ph.D., Department of Computer Science, Committee Member	Date

Abstract

Land-use classification from multiple data sources is an important problem in remote sensing. Data fusion algorithms like Semi-Supervised Manifold Alignment (SSMA) and Manifold Alignment with Schroedinger Eigenmaps (SEMA) use spectral and/or spatial features from multispectral, multimodal imagery to project each data source into a common latent space in which classification can be performed. However, in order for these algorithms to be well-posed, they require an expert user to either directly identify pairwise dissimilarities in the data or to identify class labels for a subset of points from which pairwise dissimilarities can be derived. In this paper, we propose a related data fusion technique, which we refer to as Semi-Supervised Normalized Embeddings (SSNE). SSNE is defined by modifying the SSMA/SEMA objective functions to incorporate an extra normalization term that enables a latent space to be well-defined even when no pairwise-dissimilarities are provided. Using publicly available data from the 2017 IEEE GRSS Data Fusion Contest, we show that SSNE enables similar land-use classification performance to SSMA/SEMA in scenarios where pairwise dissimilarities are available, but that unlike SSMA/SEMA, it also enables land-use classification in other scenarios. We compare the effect of applying different classification algorithms including a support vector machine (SVM), a linear discriminant analysis classifier (LDA), and a random forest classifier (RF); we show that SSMA/SEMA and SSNE robust to the use of different classifiers. In addition to comparing the classification performance of SSNE to SSMA/SEMA and comparing classification algorithm, we utilize manifold alignment to classify unknown views.

Acknowledgments

I would like to thank my thesis advisors and committee members for all of their guidance and feedback in completing this research. I would like to thank my coworkers at Ursa Space Systems for their encouragement and support in completing this thesis. I would also like to thank my family and friends for their endless support, I wouldn't be here without them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the field of remote sensing classification, the identification of specific objects or pixels in images based on spectral information, is an important task. Classification can be applied to remote sensing data in several ways including target detection, anomaly detection, and land-use classification. Target detection is used to identify where specific objects or spectra are in an image. Anomaly detection identifies where unusual patterns, such as camouflage, occur in image. Land-use classification is used to segment out areas of an image, either by region or pixelwise, and identify what type of use (i.e. forest or field) corresponds to these specific parts of the image. These tasks can be automated using machine learning so that an algorithm can learn which spectral pattern and other features correspond to a specific target or class.

However, these tasks present many challenges due to the high dimensionality of remote sensing data, the inherent properties of spectral data, and the prevalence of multi-modal and multi-source data. First, modern sensors are capable of both multispectral and hyperspectral imaging. While a standard image captures 3 bands (red, green, and blue), multispectral sensors typically detect 3 to 15 spectral bands and hyperspectral sensors can detect hundreds of bands. This, along with the use of very high-resolution images, gives a large amount of data points (i.e. an image that is $m \times n$ pixels with L bands can be thought of as mn data points each with L features). Second, remote sensing data is difficult to interpret due to spectral variations. These phenomena can be the results of temporal variations and nonlinearities in the spectral response due to atmospheric and other environmental conditions [27]. Spectral signatures of some materials can change over time [52, 27] and some physical phenomena, such as multiple scattering, can cause a nonlinear response [4, 20, 27]. Third, the same location or city may be recorded from different angles, from different sensors and at different times giving multi-modal views. The use of different modalities can provide complementary information; however, though the data is of the same distribution, it is unaligned.

In this thesis, we focus on the third issue: multi-view data. Many approaches have been taken to fully utilize all available data sources to improve classification while keeping the computation time manageable. This includes both reducing the amount of data and aligning multi-view data. Dimensionality reduction algorithms can be used to project the data into a lower-dimensional space while preserving the inherit properties of the data. Both principle component analysis (PCA) and linear discriminant analysis (LDA) are linear transformations that are commonly used as dimensionality reduction, preprocessing techniques prior to later classification. Many other algorithms

have been proposed for generating lower-dimensional data representations that are useful for classification of high dimensional datasets, including Local Linear Embedding (LLE) [28], Isometric Feature Mapping (ISOMAP) [4], Kernel Principal Components Analysis (KPCA) [18], Laplacian Eigenmaps (LE) [7, 22], Diffusion Maps [14], Stochastic Proximity Embedding (SPE) [2], Local Tangent Space Analysis (LTSA) [54], t-Distributed Stochastic Neighbor Embedding (t-SNE) [51], Schroedinger Eigenmaps (SE) [8] and Spatial Spectral Schroedinger Eigenmaps (SSSE) [10].

To perform land-use classification from multi-view data, one approach would be to simply perform dimensionality reduction on each individual view and then concatenate the results. However, this idea is only feasible if data from the individual views are spatially aligned and sampled in a common coordinate system. For use with multispectral and hyperspectral remote sensing data, a non-linear approach to dimensionality reduction is often appropriate due to the inherent, non-linear nature of the data. Manifold learning has been shown to successfully reduce the dimensionality of this data while preserving the nonlinearities that are captured in the data [30, 27]. Manifold alignment adapts manifold learning to compute transformations for each modality of multi-modal datasets that project the data into a common, low-dimensional embedding.

A well know algorithm for manifold learning is Laplacian Eigenmaps (LE) [7, 22]. LE is an unsupervised non-linear graph-based dimensionality reduction algorithm that uses manifold learning techniques. The goal of the algorithm is to project the original high dimensional data into a lower dimension manifold while preserving the local geometric structure of the spectral data. Schroedinger Eigenmaps (SE) [16] is a semi-supervised generalization of the Laplacian Eigenmaps algorithm which fuses both spectral and spatial information. These algorithms have been adapted to allow for manifold alignment of multi-modal data. Semi-Supervised Manifold Alignment (SSMA), like LE, preserves the local geometric structure of the spectral data from multiple sources [23, 50]. Manifold Alignment with Schroedinger Eigenmaps (SEMA) generalizes SSMA to include both spectral and spatial features [27]. Although these algorithms exhibit good general performance, they can be improved.

While SSMA/SEMA show good results when classification is performed using features extracted from the latent space, both algorithms require labeled similarities and dissimilarities in addition to the spectral and spatial features. These are points are pairwise similarity and dissimilarities between views that must be provided by an expert user. Due to this they both suffer from the same theoretical issue: the objective functions they propose to minimize diverge in the absence of any provided expert labeled dissimilarity information. This is because their objective functions rely on a normalization term vanishes when no dissimilarity information is provided. Ideally, the SSMA/SEMA formulations should appropriately handle situations having few or no pairwise dissimilarity constraints; instead, they become impossible to solve. In this thesis, we propose to resolve this problem by posing an objective function whose minimization that reduces to a feature-based multi-view version of LE/SSSE when no similarity or dissimilarity information is provided. Further, in spectral clustering, it has been shown that the use of normalized Laplacians improves clustering results [31]. Our proposed algorithm introduces a normalization factor for the Laplacian matrix

corresponding to the inherent spectral information of the image data. We refer to the latent space that results from the projection functions that optimize this new criterion as the Semi-Supervised Normalized Embedding (SSNE).

To validate the effectiveness of these proposed modifications, we preform land-use classification to determine if the modifications lead to improved performance. There are two inherent challenges to classification of remote sensing data. First, the underlying distribution of remote sensing data is often unknown. Second, the small number of available training samples makes it difficult to generate an accurate ground truth for remote sensing images [34, 43]. Manifold learning and manifold alignment algorithms have typically been evaluated with classification using linear discriminant analysis (LDA) [10, 17, 27, 50]. However, both support vector machines (SVM) and random forests (RF) have been utilized for classification of remote sensing imaging [6, 27, 34].

SEMA was tested using only LDA and SVMs using few images [27]. In this thesis, we propose to further evaluate SEMA along with the evaluation of our proposed algorithm, SSNE, using LDA, SVM, and RF classification algorithms. We will apply SEMA and SSNE to preprocess the multi-modal, multi-temporal, and multi-source data provided by the 2017 IEEE GRSS Data Fusion Contest [1]. We will then apply the several machine learning algorithms for land-use classification as evaluation methods. Our main results will be comparing the performance of SEMA and SSNE through various scenarios of the contest data. We compare the robustness of these algorithms through the application to different datasets in addition to the application of several classification algorithms.

The rest of this thesis will be structured as follows. Chapter 2 will give background information by giving an overview of remote sensing and spectral graph theory, detailing the construction of the LE, SE, SSMA, and SEMA algorithms, and describing LDA, SVM, and RF classification algorithms. Chapter 3 will introduce our proposed algorithm, SSNE. Chapter 4 will describe our data and experimental setup including which classification methods will be used. Chapter 5 will present the results of our experiments. Finally, Chapter 6 will provide some concluding remarks about the results of this thesis.

Chapter 2

Background

In this chapter, we give an overview of remote sensing and spectral graph theory, a description of LE, SE, SSMA, and SEMA algorithms, and an introduction to several classification algorithms. Section 2.1 gives a preliminary introduction to remote sensing and remote sensing data. In Section 2.2, we discuss spectral graph theory and spectral clustering and their relation to manifold learning. In Section 2.3, we describe the construction of LE and SE. Section 2.4, describes the how manifold learning concepts can be applied to manifold alignment via SSMA and SEMA. Finally, in Section 2.5, we describe LDA, SVM, and RF algorithms and their application to remote sensing.

2.1 Remote Sensing and Data Fusion

Remote sensing is the field of gathering information about the surface of the Earth without contact with the area of interest. Typically, remote sensing data is gathered using airborne or space sensors as show in Figure 2.1. The most common type of sensors used are optical sensors gather imagery with light from the sun.

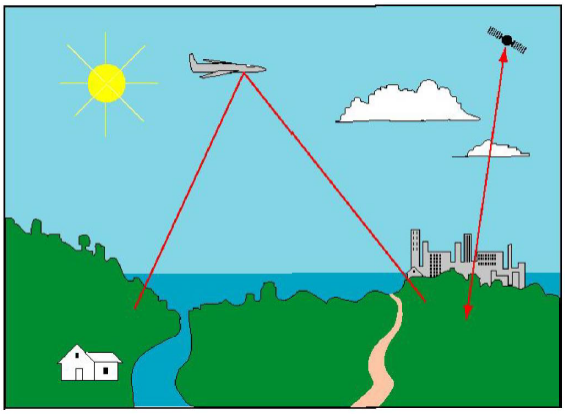

Figure 2.1: An example of airborne and satellite collection of remote sensing [36].

Standard imagery typically captures a single, panchromatic band or three bands representing red, blue, and green wavelengths. However, some optical sensors can be multispectral (which typically detect 3 to 15 spectral bands) and hyperspectral (which can detect hundreds of bands). Figure 2.2 demonstrates how each spectra is represented as a layer in an image.

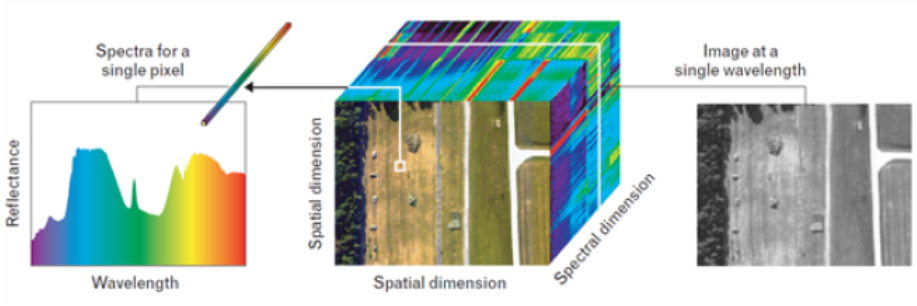

Figure 2.2: A hyperspectral image [48].

Different types of material have different spectral reflectance; for example, in Figure 2.3, we can see the spectral reflectance of three different materials. The recognition, or classification of different materials and features can be based on these spectral reflectance properties. In a n-band image, the spectral reflectance curve is represented as a n-dimensional feature vector at each pixel. For pixel-wise classification we can learn the representation of different materials in order to classify, for example, land cover. Figure 2.4 shows an example of a land cover map.

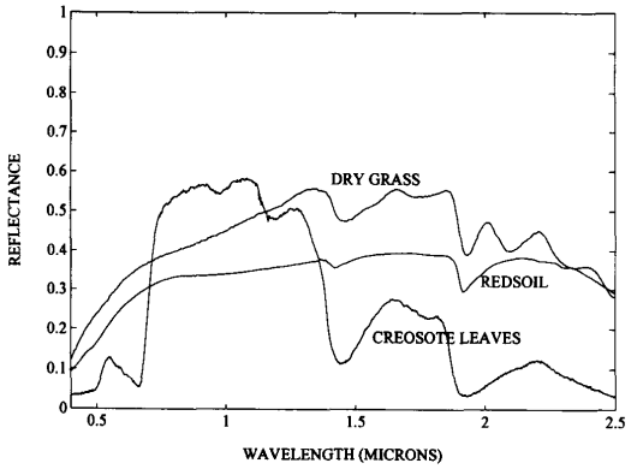

Figure 2.3: The spectral reflectance of vegetation and terrain. [25].

There are several classification systems that have been developed for mapping land cover and use. The local climate zone system (LCZ) [47] has been developed to provide a standardized scheme for classifying natural and urban landscapes based on climate surface properties. This system provides ten build types (i.e. compact high-rise, open high-rise, sparsely built, etc.) and seven land cover types (i.e. dense trees, scattered trees, bush/scrub, etc.). Another system commonly used is Open Street Maps [39] which include different, but similar building and land types.

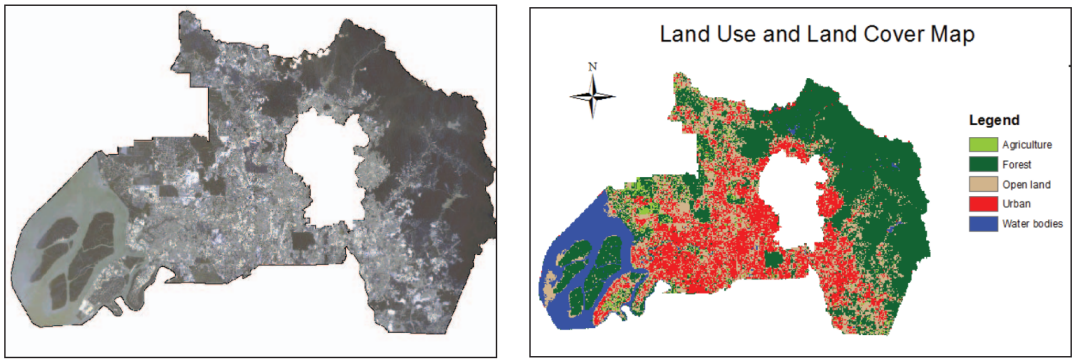

Figure 2.4: Satellite image of an area of interest (left) and the corresponding land use/land cover map. [32]

2.2 Spectral Graph Theory

An important observation to make is the relationship between spectral graph theory and the manifold learning. A graph $G = \{V, E\}$ is a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $E \subseteq V \times V$. A pair of vertices $(v_i, v_j) \in E$ if there is a connection or edge between v_i and v_j. In this application, we assume that G is undirected, such that $(v_i, v_j) \in E$ if and only if $(v_j, v_i) \in E$. Spectral graph theory studies the properties of matrices representing graphs, including adjacency matrices and Laplacian matrices. An adjacency matrix $\boldsymbol{A}$ is defined as an $n \times n$ matrix where each element $A_{i,j} = 1$ if $(v_i, v_j) \in E$ and $A_{i,j} = 0$ otherwise.

A weighted adjacency matrix $\boldsymbol{W}$ is defined elementwise by $W_{i,j} = w_{i,j}$ where $w_{i,j}$ represents how strongly v_i and v_j are connected. If $(v_i, v_j) \notin E, W_{i,j} = 0$. Note that if G is undirected then $\boldsymbol{W}$ must be symmetric. The degree matrix of $\boldsymbol{W}$, $\boldsymbol{D}$ is defined as $D_{i,j} = \sum_j W_{i,j}$. The Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ also characterizes many useful properties of a graph; these properties are described in [31]. A graph Laplacian can be normalized in several ways, most commonly as

$$\boldsymbol{L_{sym}} = \boldsymbol{D^{-1/2} L D^{-1/2}} = \boldsymbol{I - D^{-1/2} W D^{-1/2}} \ , \quad \text{or} \tag{2.1}$$

$$\boldsymbol{L_{rw}} = \boldsymbol{D^{-1} L} = \boldsymbol{I - D^{-1} W} \ , \tag{2.2}$$

where $\boldsymbol{L_{sym}}$ is symmetric matrix, but $\boldsymbol{L_{rw}}$ is not; however, $\boldsymbol{L_{rw}}$ represents the Laplacian of a graph whose weights are given by the transition probabilities of a random walk on the graph. In the next section we discuss methods of spectral clustering and graph partitioning.

2.2.1 Spectral Clustering and Graph Partitioning

Spectral clustering studies similarity graphs, a graph in which an edge v_i, v_j exists if some similarity metric between the two vertices is above a certain threshold. Graph partitioning uses weighted similarity graphs to determine an optimal partitioning of the graph into subgraphs; this is the method that we focus on in this thesis. From a differing point of view, spectral clustering can be

thought of as a clustering such that a random walk stays within the same cluster instead of moving between clusters. In contrast, another justification for spectral clustering is from a perturbation theory point of view; we can consider the Laplacian matrices to be perturbations of the ideal case where the between-class similarly is zero [31].

In the domain of imagery, the vertices in G represent pixel values and the similarity metric is commonly defined on some distance metric between two data points in feature space. Feature space can be representative of both spectral and spatial properties. Some method used to define connectivity include mutual k-nearest neighbors, ϵ-neighborhoods, and fully connected graphs [31]. A k-nearest neighbor graph is constructed such that each vertex is connected the k closest vertices; in mutual k-nearest neighbors two vertices are connected if they both in each other's nearest neighbor set. A ϵ-neighborhood graph is constructed such that pairs of vertices are connected if the distance between them is less than ϵ. In a fully connected graph all vertices are connected, and each edge is weighted according to some decreasing function of distance.

Clustering is then defined as finding the best partitions of these similarity graphs such that points in the same cluster are similar and in different clusters are dissimilar [31]. For the binary case, the goal is to partition this graph into two disjoint clusters A and B such that $A \cup B = V$. The optimal partitioning is found by minimizing

$$cut(A, B) = \sum_{v_i \in A, v_j \in V} w_{i,j}. \tag{2.3}$$

However, this minimization favors weakly connected outliers in the feature set. To counter this, in [45], [46] Shi and Malik propose an algorithm for instead minimizing the normalized cut

$$Ncut(A, B) = cut(A, B) \left(\frac{1}{vol(A)} + \frac{1}{vol(B)} \right) \tag{2.4}$$

where $vol(A) = \sum_{v_i \in A, v_j \in B} w_{i,j}$. Essentially, the cut is normalized so that the resulting subgraphs do not have do not have hugely different degrees allowing for a more balanced partitioning. Figure 2.5 compares a minimum cut and a normalized cut on the same graph.

Equation (2.4) has been extended for multiway partition [45], [46], [53]. Supposed that we want to partition a graph V into k clusters $A_1 \cup \ldots \cup A_k = V$; then, the k-way normalized cut can be defined by:

$$kNcut(A_1, \ldots, A_k) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{cut(A_i, A \setminus A_i)}{vol(A_i)} \right). \tag{2.5}$$

Note that $kNcut(A_1, A_2) = Ncut(A_1, A_2)$.

Further, Equation (2.5) can be rewritten in terms of the matrix Laplacian. The matrix Laplacian is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ where $\boldsymbol{W}$ is the edge weight matrix and $\boldsymbol{D}$ is a diagonal degree matrix given by $D_{i,i} = \sum_j W_{i,j}$. By forming Equation 2.5 as

$$kNcut(A_1, \ldots, A_k) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{\boldsymbol{c_i}^T \boldsymbol{L} \boldsymbol{c_i}}{\boldsymbol{c_i}^T \boldsymbol{D} \boldsymbol{c_i}} \right) \tag{2.6}$$

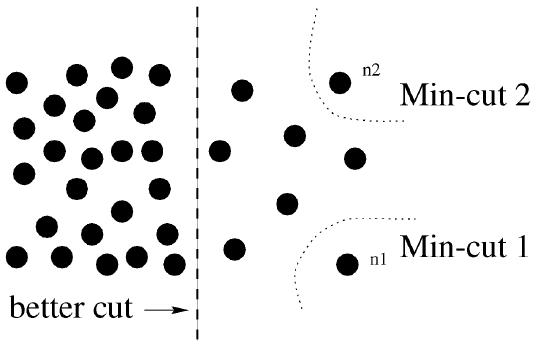

Figure 2.5: A toy example showing that minimum cut gives a sub-optimal partitioning [46]

where $c_i = v \in A_i$. It can be see that the stationary points for each of the k clusters can be computed from the generalized eigenproblem: $\boldsymbol{LC} = \lambda \boldsymbol{DC}$, where $c_{i,j} = 1$ if $c_i \in A_j$ and $c_{i,j} = 0$ otherwise.

It has been shown that in spectral clustering, utilizing a normalization factor such as described above or Equation 2.1 and 2.2 improves clustering [31], as opposed to using the unnormalized Laplacian. This directly motivates our proposed algorithm described in Chapter 3.

2.3 Manifold Learning

Manifold learning is class of non-linear methods for dimensionality reduction that seeks to preserve properties of the non-linear spectral responses that are captured in high dimensional data that implicitly lies on a manifold.

Let $\boldsymbol{X} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_k}\}$ be points in $\mathbb{R}^n$ that represent, for example, the spectral features of pixels in a remote sensing image. The goal of manifold learning is to create a mapping $\boldsymbol{Y} = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_k}\}$ in $\mathbb{R}^m$ of the points in $\boldsymbol{X}$ where $m << n$ while preserving the geometric structure of $\boldsymbol{X}$.

2.3.1 Laplacian Eigenmaps

Laplacian Eigenmaps (LE) is an unsupervised manifold learning technique proposed by Belkin and Niyogi in [7]. The LE algorithm preserves local neighborhoods by penalizing when neighboring points in $\boldsymbol{X}$ are mapped such that their respective points in $\boldsymbol{Y}$ are far apart.

The LE algorithm consists of the following three steps:

1. Construct a graph G where the vertices are points in $\boldsymbol{X}$ and the edges are defined based on some proximity measure such as ϵ-neighborhoods or mutual k-nearest neighbors.

2. Define the graph Laplacian as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \tag{2.7}$$

where $\boldsymbol{W}$ is an edge weight matrix and $\boldsymbol{D}$ is a diagonal degree matrix given by $D_{i,i} = \sum_j W_{i,j}$. Typically, $\boldsymbol{W}$ is defined by the heat kernel such that $W_{i,j} = \exp\left(-||\boldsymbol{x_i} - \boldsymbol{x_j}||^2/\sigma\right)$ if there exists an edge between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ in G and $W_{i,j} = 0$ otherwise.

3. The mapping $\boldsymbol{Y}$ is given by the minimization of the objective function

$$\Phi(\mathbf{F}) = \mathrm{tr}\Big(\left(\mathbf{F^T X D X^T F}\right)^{-1}\left(\mathbf{F^T X L X^T F}\right)\Big), \tag{2.8}$$

where tr is the trace of a matrix. One solution to Equation (2.8) can be found by solving the generalized eigenproblem

$$\boldsymbol{Lf} = \lambda \boldsymbol{Df} \tag{2.9}$$

for the smallest m smallest non-trivial eigenvector. The points $\boldsymbol{y_1^T}, \ldots, \boldsymbol{y_m^T}$ are defined by the rows of $\boldsymbol{F} = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_m}]$ where $\boldsymbol{f_0}, \boldsymbol{f_1}, \ldots \boldsymbol{f_m}$ are ordered so that $0 = \lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_m$. We note that Equation (2.8) is similar to Equation (2.4). From this we can observe that Ncuts and the LE algorithm have a similar form.

2.3.2 Spatial-Spectral Schroedinger Eigenmaps

Schroedinger Eigenmaps (SE), proposed by Czaja and Ehler in [16] is a semi-supervised generalization of LE that takes into consideration not just the spectral features of the image data but also the knowledge of particular classes. Spatial-Spectral Schroedinger Eigenmaps (SSSE), proposed by Cahill et al. in [10], gives an instance of SE in which the spatial proximity between pixels, i.e the spatial features, are used rather than knowledge of the class labels. SSSE follows the same procedure as LE but incorporates a potential matrix $\boldsymbol{V}$. The potential matrix can be constructed to cause specific points to be mapped close to the origin or close to each other. These two types of potential matrices are called barriers and clusters, respectively.

A barrier potential is defined with a non-negative diagonal matrix $\boldsymbol{V}$. Each non-negative $V_{i,i}$ pulls the corresponding ith point toward the origin. A cluster potential is the sum of non-diagonal matrices $\boldsymbol{V^{(i,j)}}$ defined as $V_{i,i}^{(i,j)} = V_{j,j}^{(i,j)} = 1$, $V_{i,j}^{(i,j)} = V_{j,i}^{(i,j)} = -1$, and $V_{k,l}^{(i,j)} = 0$ otherwise. In this construction, the corresponding ith and jth points are pulled closer together in the embedding. Both constructions are further described in [8], [10], and [16].

SSSE follows the same procedure as LE with a modification to Equation (2.8). The inclusion of $\boldsymbol{V}$ gives the following objective function

$$\Phi_{\mathrm{SSSE}}(\mathbf{F}) = \mathrm{tr}\Big(\left(\mathbf{F^T X D X^T F}\right)^{-1}\left(\mathbf{F^T X}(\mathbf{L} + \alpha\mathbf{V})\mathbf{X^T F}\right)\Big) \tag{2.10}$$

where α is a chosen weight parameter defining the contribution of $\boldsymbol{V}$. Like LE, a solution to Equation (2.10) can be found by solving the generalized eigenproblem

$$(\boldsymbol{L} + \alpha\boldsymbol{V})\boldsymbol{f} = \lambda\boldsymbol{Df}. \tag{2.11}$$

This is similar to Equation 2.9 with the addition of the weighted V potential matrix on the left side. As before in LE, we solve for the m smallest non-trivial eigenvectors of Equation (2.11) which give the m vectors used to construct the projection matrix to the latent space. Specifically, the points $\boldsymbol{y_1^T}, \ldots, \boldsymbol{y_m^T} \in \mathbf{Y}$ are defined by the rows of $\boldsymbol{F} = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_m}]$ where $\boldsymbol{f_0}, \boldsymbol{f_1}, \ldots \boldsymbol{f_m}$ are ordered so that $0 = \lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_m$.

2.4 Manifold Alignment

Laplacian Eigenmaps and Spatial-Spectral Schroedinger Eigenmaps are both dimensionality reduction algorithms that can be applied to data from a single modality. However, Ham et al. [23] uses similar techniques for manifold alignment. Manifold alignment allows for multiple datasets to be fused together. Semi supervised alignment of manifolds [23] requires partially labeled data which map to intrinsic coordinates for multiple datasets. Using these intrinsic coordinates, we can create a Laplacian matrix that encodes the labeled and unlabeled data points for each dataset. Like in LE, this method minimizes a cost function to create an optimal projection function which projects each dataset into an aligned latent space. Figure 2.6 illustrates a comparison between the raw unaligned embedding and the aligned embedding of two datasets with known correspondences. Manifold alignment seeks to project the source data into a common latent space.

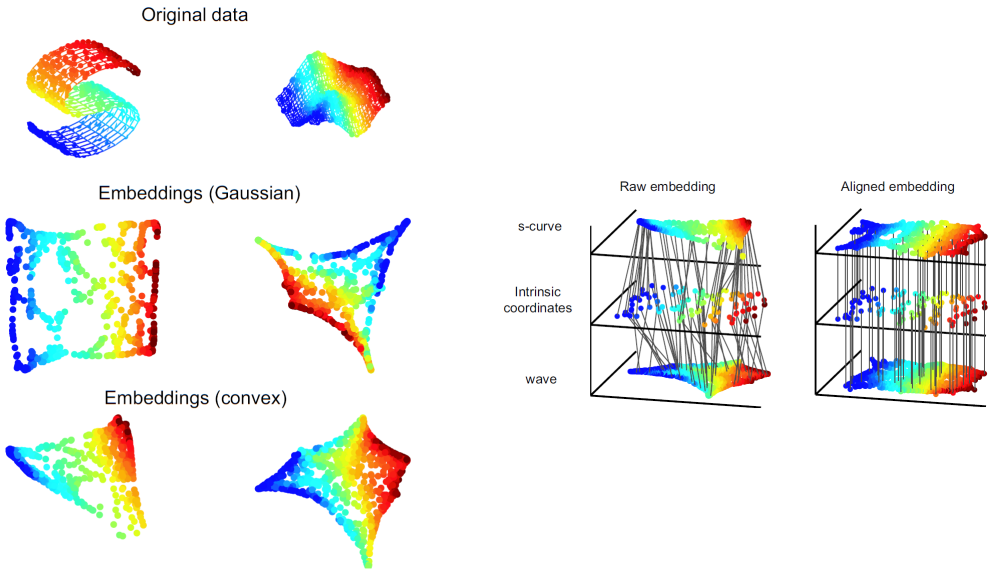

Figure 2.6: On the left, 2-dimensional embeddings of 3-dimensional surfaces (a s-curve and a wave). On the right, manifold alignment applied to two raw embeddings where the lines connecting points between each dataset represent known correspondences. The raw embeddings are projected into a latent space in which the correspondences are aligned, represented by the now straight lines from the first dataset to the section though the intrinsic coordinates [23].

2.4.1 Semi-supervised Manifold Alignment

Tuia et al. [50] applied Ham's [23] manifold alignment method to sets of hyperspectral images captured with varying angles of acquisition with an algorithm called Semi-Supervised Manifold Alignment (SSMA). These sets of images have the same underlying objects but are distorted from one another. Figure 2.7 shows a toy example of the SSMA algorithm in which two distorted datasets drawn from the same underlying distribution are aligned.

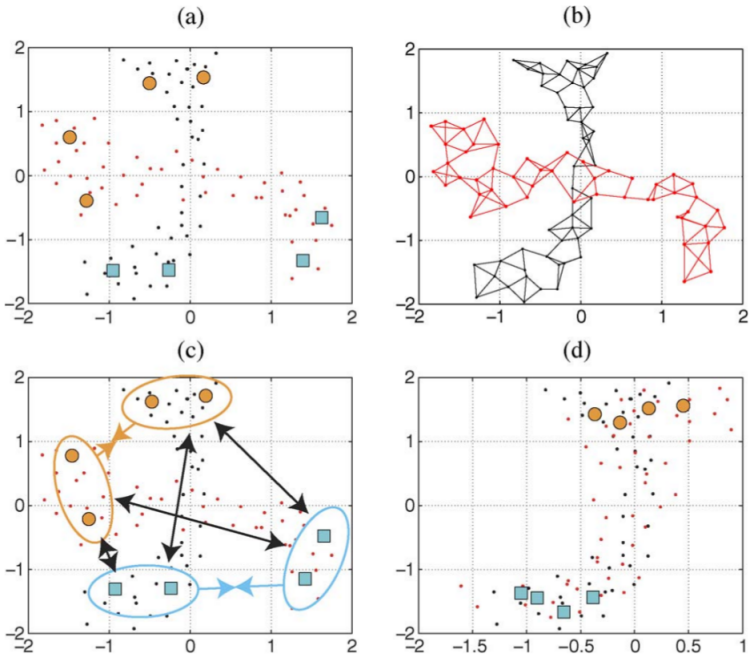

Figure 2.7: A toy example of SSMA: (a) two data sets with the same underlying distribution represented as black and red dots with labeled (blue and yellow) points; (b) the geometric structure of each dataset as a graph; (c) the similar classes are pulled together, and dissimilar classes pushed apart; (d) the aligned embedding. [50].

Consider M datasets where each dataset $\boldsymbol{X^m} \in \mathbb{R}^{d_m}$ contains the spectral features of the pixel in the mth image. Each $\boldsymbol{X^m}$ is constructed of a set of l_m labeled pairs of samples and classes $\{\boldsymbol{x_i^m}, \boldsymbol{y_i^m}\}$ and u_m unlabeled samples $\{\boldsymbol{x_j^m}\}$ where $l_m << u_m$. All M datasets can be represented using a block diagonal matrix $\boldsymbol{X}$ where each $\boldsymbol{X^m}$ are placed as blocks along the diagonal of the matrix and all other elements are zero. For $\boldsymbol{X}$ we construct three graphs, each having a different set of weights. We refer to the Laplacian matrices of these graphs as the geometric preserving term $\boldsymbol{L_g}$, the similarity term $\boldsymbol{L_s}$, and the dissimilarity term $\boldsymbol{L_d}$.

1. For each dataset $\boldsymbol{X^m}$, the spectral geometric preserving term is constructed as

$$\boldsymbol{L_g^m} = \boldsymbol{D_g^m} - \boldsymbol{W_g^m} \tag{2.12}$$

where each $\boldsymbol{W_g^m}$ is a weight matrix constructed using some distance metric on all the points in $\boldsymbol{X^m}$ and $\boldsymbol{D_g^m}$ is a diagonal degree matrix with $D_g^m(i,i) = \sum_j W_g^m(i,j)$. This graph Laplacian matrix for each domain is then used to construct a block diagonal matrix $\boldsymbol{L_g}$.

2. Next, for each dataset the similarity Laplacian matrix is constructed as

$$\boldsymbol{L_s^m} = \boldsymbol{D_s^m} - \boldsymbol{W_s^m} \tag{2.13}$$

where $W_s^m(i,j) = 1$ if $\boldsymbol{x_i^m}$ and $\boldsymbol{x_j^m}$ have the same label and $W_s^m(i,j) = 0$ otherwise, and $D_s^m(i,i) = \sum_j W_s^m(i,j)$. This graph Laplacian matrix for each dataset is used to construct a block diagonal matrix $\boldsymbol{L_s}$.

3. Finally, for each data set the dissimilarity Laplacian matrix is constructed as

$$\boldsymbol{L_d^m} = \boldsymbol{D_d^m} - \boldsymbol{W_d^m} \tag{2.14}$$

where $W_d^m(i,j) = 1$ if $\boldsymbol{x_i^m}$ and $\boldsymbol{x_j^m}$ have different labels and $W_d^m(i,j) = 0$ otherwise, and $D_d^m(ii) = \sum_j W_d^m(ij)$. This graph Laplacian matrix for each dataset is used to construct a block diagonal matrix $\boldsymbol{L_d}$.

Now, the goal of SSMA is to create a projection matrix $\boldsymbol{f^m} \in \mathbb{R}^{d_m \times d_m}$ for each dataset that will project the points in $\boldsymbol{X^m}$ to a latent space $\mathcal{F}$. This is done by simultaneously minimizing the distances between similar classes and maximizing the distance between dissimilar classes with the objective function:

$$\Phi_{\text{SSMA}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X L}_d \mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_s + \mu \mathbf{L}_g)\mathbf{X^T F}\big)\Big) \tag{2.15}$$

where $\boldsymbol{F}$ is our $d_{max} \times d_{max}$ projection matrix with $d_{max} = \sum_m d_m$ and μ is a chosen parameter that determines how much the geometric term is considered.

The minimization can be solved using a generalized eigenvalue problem:

$$\boldsymbol{X}(\boldsymbol{L_s} + \mu \boldsymbol{L_g})\boldsymbol{X^T f} = \lambda \boldsymbol{X}\left(\boldsymbol{L_d}\right)\boldsymbol{X^T f} \tag{2.16}$$

by computing the d_{max} smallest non-trivial eigenvalues. Specifically, the points $\boldsymbol{y_1^T}, \ldots, \boldsymbol{y_m^T}$ are defined by the rows of $\boldsymbol{F} = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_{d_{max}}}]$ where $\boldsymbol{f_0}, \boldsymbol{f_1}, \ldots \boldsymbol{f_{d_{max}}}$ are ordered so that $0 = \lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_{d_{max}}$. The projection for each source domain to the latent space is given by

$$\boldsymbol{P_{\mathcal{F}}(X_m)} = \boldsymbol{f_m^T X_m}. \tag{2.17}$$

A kernelization of this method has been introduced in [49] which enables nonlinear projections.

2.4.2 Manifold Alignment with Schroedinger Eigenmaps

SSMA creates a projection function based solely on the spectral properties of the imagery. Manifold Alignment with Schroedinger Eigenmaps (SEMA) is an algorithm proposed by Johnson, et al. [27] that builds on SSMA by considering the spatial properties of the images in the optimal cost function. As with the SSSE generalization of LE, this is done by constructing a potential term $\boldsymbol{V}$ that preserves the spatial neighbors of each image.

SEMA generalizes the cost function in Equation (2.15) with the inclusion of the potential term $\boldsymbol{V}$ in the numerator. The new cost function is given as

$$\Phi_{\text{SEMA}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X L}_d\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_s + \mu(\mathbf{L}_g + \alpha\mathbf{V}))\mathbf{X^T F}\big)\Big) \tag{2.18}$$

The projection can be computed using the same procedure as SSMA: by solving the modified generalized eigenvalue problem

$$\boldsymbol{X}\left(\boldsymbol{L_S} + \mu\left(\boldsymbol{L_g} + \alpha\boldsymbol{V}\right)\right)\boldsymbol{X^T}\gamma = \lambda\boldsymbol{X}\left(\boldsymbol{L_d}\right)\boldsymbol{X^T}\gamma \tag{2.19}$$

for the d_{max} smallest eigenvalues. As before, the corresponding eigenvectors are used to construct a projection matrix for each dataset. Specifically, the points $\boldsymbol{y_1^T}, \ldots, \boldsymbol{y_m^T}$ are defined by the rows of $\boldsymbol{F} = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_{d_{max}}}]$ where $\boldsymbol{f_0}, \boldsymbol{f_1}, \ldots \boldsymbol{f_{d_{max}}}$ are ordered so that $0 = \lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_{d_{max}}$. The projection for each source domain to the latent space is given by

$$\boldsymbol{P_{\mathcal{F}}}(\boldsymbol{X_m}) = \boldsymbol{f_m^T X_m}. \tag{2.20}$$

This method allows for the fusion of spatial and spectral information of the datasets to be used for manifold alignment.

2.5 Classification

The algorithms described in Sections 2.3 and 2.4 consider the problem of dimensionality reduction and data fusion. We now consider when the data is representative of multiple categories or classes. Classification is the problem of using characteristic of the data to predict the classification of each sample in a dataset. A classifier can be trained on a set of known sample and class pairs such that it can be applied predictively to determine the class of unknown samples in the same feature space. Preprocessing techniques, such as dimensionality reduction and data fusion, can be performed on a dataset prior to classification in order to reduce computation time and allow for consistent data.

In Section 2.1 we introduced the idea that remote sensing images can be classified based on the spectral reflectance of different material type. There are many options for classification and many of these techniques have been successfully used for the task of land-use classification in remote sensing. The nature of hyperspectral and multispectral images leads to two inherent difficulties

for classification: typically, there are only a small number of training samples available, and the distribution of each band is often unknown. We discuss an overview of Linear Discriminant Analysis (LDA), support vector machines (SVM), and random forests (RF) and previous applications of these classifiers to remote sensing data.

2.5.1 Linear Discriminant Analysis

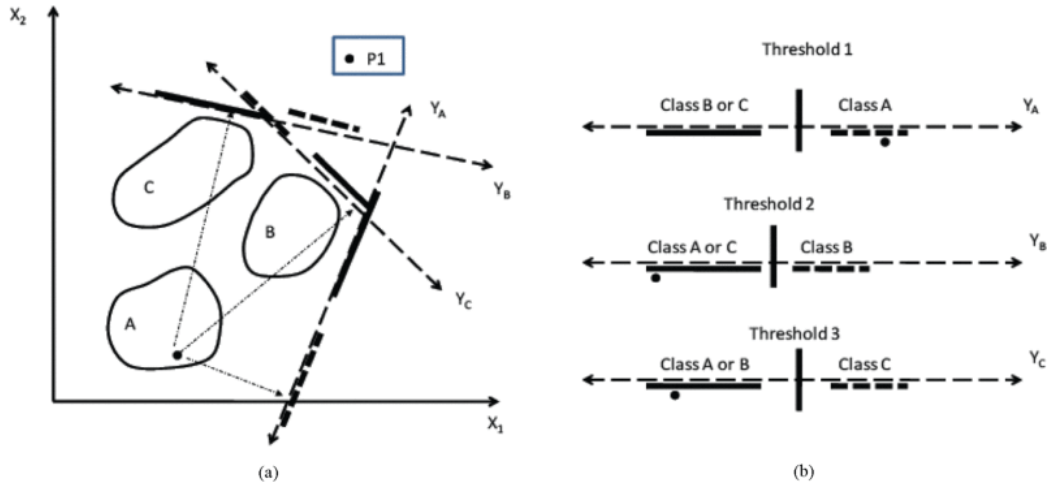

Figure 2.8: LDA can be applied to a dataset to project the data down to a lower dimensional embedding that can then be linearly separable. This example shows how a point would be classified depending on the linear discriminate [24]

Linear Discriminant Analysis (LDA), a generalization of Fisher's linear discriminant, is a method of linear transformation that maximizes class separation based on the conditional probability densities of the class distributions. This algorithm is commonly used for both dimensionality reduction and classification. For this we discuss LDA as a classification algorithm. LDA computes the linear discriminants that represent the best directions that maximizes the separation of classes. These linear discriminants are computed from the eigenvalues and eigenvectors of the scatter matrices, which characterize the between-class and within-class scatter. These scatter matrices are defined by the covariance of the class means.

A full overview of LDA's application to remote sensing is described in [17]. This paper describes a modified LDA algorithm for remote sensing applications which relaxes the requirements for the training samples and the knowledge of the class distributions in the image. LDA has been used in conjunction with the image preprocessing algorithms described in several papers including [10], [27], and [50].

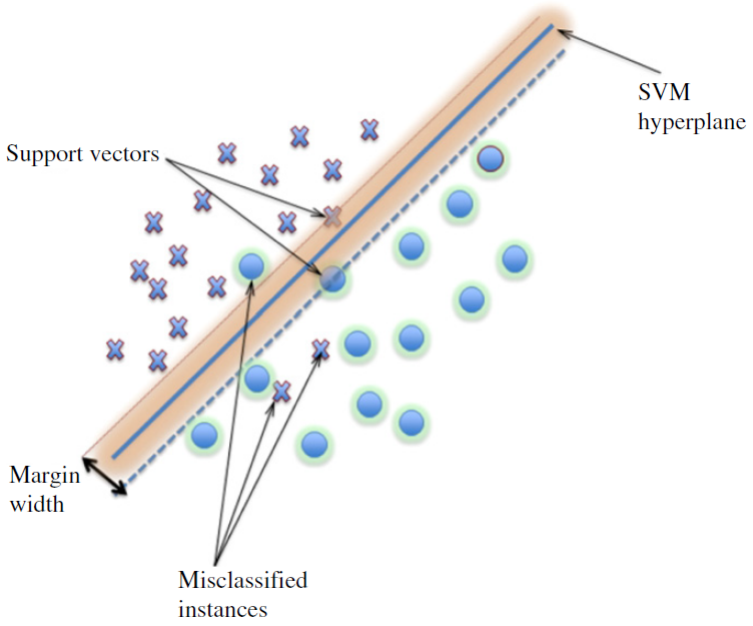

Figure 2.9: A linear support vector machine [34]

2.5.2 Support Vector Machines

A support vector machine (SVM) is a supervised machine learning technique. The goal of an SVM is to find a hyperplane or decision boundary that separates data into discrete classes; see Figure 2.9. The optimal decision boundary is learned by iteratively minimizing misclassification of labeled training samples. An SVM is classically a binary classifier, but in practice an SVM can be used for multi-class problems by combining multiple binary SVMs. Further, a linear SVM assumes that the classes are linearly separable, but in cases where class clusters overlap one another or are not linearly separable, a kernel function can be incorporated. This will project nonlinear data into a higher dimension in which the classes can be separated linearly.

Support vector machines are ideal for classification problems in the remote sensing field for two reasons: SVMs makes no assumptions about the underlying probability distribution and have been shown to have high accuracy on small training sets [34], [43]. SVMs have been used for classification in several remote sensing applications including [27], [33] and [56].

2.5.3 Random Forests

Unlike binary SVM, which create a single hyperplane decision boundary that separates the data into two classes, decision trees create decision boundaries based on hierarchical rules. This essentially partitions the feature space with multiple decision boundaries.

A decision tree is more formally defined as a tree-like structure where each node represents a decision rule based on a specific attribute of the feature space. The tree is constructed recursively

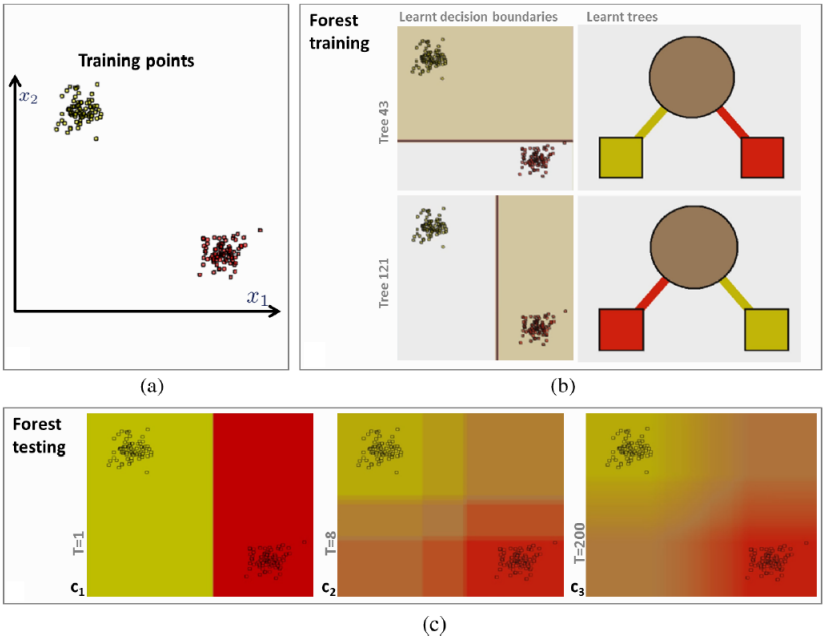

Figure 2.10: Example of a RF classifier. (a) The training samples plotted in 2-d space. (b) Two decision trees and their corresponding learned decision boundaries for two bootstrap samples, and (c) the effect of the number of decision trees, T on the decision regions for $T = 1, 8, 200$ [15].

such that at each node, the best split is computed, where a split is a partitioning on a single attribute. The value of the split is computed as the sum of the squared error of the resulting attribute split. This procedure is usually done using a greedy algorithm, adding nodes until splitting no longer improves prediction. Decision trees are subject to overfitting, but there are several ways to overcome this including pruning, where nodes are iteratively removed from the tree if they do not affect prediction, and ensemble methods, where multiple decision trees are created. Random Forests (RF) are one such ensemble method.

RF classifiers create several decision trees each based on a bootstrap sample of the training data. A bootstrap sample is a random sampling with replacement of the dataset. An RF classifier preforms the following procedure k times. 1) Draw a bootstrap sample from the data. 2) Train a decision tree using this sample set. A prediction is then made with a majority vote among the k trees. Figure 2.10 shows an example of a RF classifier.

In the domain of remote sensing, RF is an appropriate method of classification for several reasons. Unlike SVMs, random forests are less sensitive to feature selection [6]. Additionally, they have been shown to outperform other classification algorithms [6]. Random forests have previously been used for classification problems in [21], [29], and [41].

Chapter 3

Semi-Supervised Normalized Embeddings

Both SSMA and SEMA are manifold alignment algorithms that have been shown to successfully fuse multi-modal remote sensing image data and reduce dimension as a preprocessing step for classification. However, we propose an improved algorithm. In Section 3.1, we discuss some problems with SSMA/SEMA. In Section 3.2, we propose our algorithm as a solution to these problems. In Section 3.3 we discuss the computational complexity of our algorithm.

3.1 Problems with SSMA/SEMA

The cost functions, Equations (2.16) and (2.19), described in Section 2.4 are minimized to compute projections function for projecting each dataset into the latent, aligned space. These cost functions are based on the use of Laplacian graph matrices to encode the spectral, spatial, and expert-labeled class information. An objective function $\Phi(\mathbf{F})$ for both SSMA and SEMA are defined as:

$$\Phi_{\text{SSMA}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X L}_d\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_s + \mu\mathbf{L}_g)\mathbf{X^T F}\big)\Big) \quad , \quad \text{or} \tag{3.1}$$

$$\Phi_{\text{SEMA}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X L}_d\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_s + \mu(\mathbf{L}_g + \alpha\mathbf{L}_p))\mathbf{X^T F}\big)\Big) \quad , \tag{3.2}$$

where $\mathbf{X}$ is the raw unaligned data, $\mathbf{F}$ is the matrix used to project this data into the aligned latent space, $\mathbf{L_g}$ is the Laplacian encoding of the spectral information, $\mathbf{L_p}$ is the Laplacian encoding of the spatial information, $\mathbf{L_d}$ is the Laplacian encoding of the provided dissimilarity pairs, $\mathbf{L_s}$ is the Laplacian encoding of the provided similarity pairs, and μ and α are chosen parameters that weight the influence of $\mathbf{L_g}$ and $\mathbf{L_p}$.

Both Φ_{SSMA} and Φ_{SEMA} can be expressed in the form:

$$\Phi(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X L_B X^T F}\big)^{-1}\big(\mathbf{F^T X L_A X^T F}\big)\Big), \tag{3.3}$$

where $\mathbf{L_A}$ and $\mathbf{L_B}$ are graph Laplacians, and so under the assumption that $\mathbf{XL_BX^T}$ is invertible, $\Phi(\mathbf{F})$ has a family of minima characterized by a set of $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_d$ and $\lambda_1, \ldots, \lambda_d$ be the generalized eigenvectors and corresponding generalized eigenvalues of $\big(\mathbf{XL_AX^T}, \mathbf{XL_BX^T}\big)$ (i.e., they satisfy $\mathbf{XL_AX^T}\boldsymbol{\mu}_i = \lambda_i\mathbf{XL_BX^T}\boldsymbol{\mu}_i$), sorted such that $\lambda_1 \leq \ldots \leq \lambda_d$ and normalized so that $\boldsymbol{\mu}_i^{\mathbf{T}}\mathbf{XL_BX^T}\boldsymbol{\mu}_i = 1$, $i = 1, \ldots, d$. Then, (3.3) has a local minimum value of $\sum_{i=1}^{q}\lambda_i$ for any $\hat{\mathbf{F}}$ of the form $\hat{\mathbf{F}} = \mathbf{MQ^T}$, where $\mathbf{M} = \big[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_q\big]$ and $\mathbf{Q} \in \mathbb{R}^{q\times q}$ is orthogonal.

Upon examination, they are clearly unsuitable if no similarity or dissimilarity pairs are provided. In this case, $\mathbf{L_d} = \mathbf{0}$, where $\mathbf{L_d}$ is the encoding of dissimilarity pairs as a Laplacian matrix, and so $\mathbf{F^T X L}_d\mathbf{X^T F} = \mathbf{0}$, in which case $\Phi(\mathbf{F})$ diverges and cannot be minimized. Even if some dissimilarity pairs are provided, this may still be problematic. We note that by construction, $\text{rank}(\mathbf{L}_d) \leq |\mathcal{D}|$, where D is the degree matrix corresponding to the weighted adjacency matrix W, then $\text{rank}\big(\mathbf{F^T X L}_d\mathbf{X^T F}\big) \leq |\mathcal{D}|$. Therefore, if fewer than q, where q is the number of dimensions in the embedding, dissimilarity pairs are provided, $\mathbf{F^T X L}_d\mathbf{X^T F}$ is guaranteed to be singular.

Ideally, we would like to pose an objective function that exhibits similar behavior to Φ_{SSMA} or Φ_{SEMA} when $\mathbf{F^T X L}_d\mathbf{X^T F}$ is invertible, but that also gives meaningful projections into a latent space when $\mathbf{F^T X L}_d\mathbf{X^T F}$ is singular. Furthermore, if no expert information is provided at all (i.e., when $|\mathcal{S}| = |\mathcal{D}| = 0$), such an objective function should revert to one that behaves like an objective function from some "standard" dimensionality technique that is applied independently to each view,

To focus on this last point first, we argue that if no expert information is provided at all, then appropriate objective functions to minimize would be either:

$$\Phi_{\text{LE}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X D}_g\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X L}_g\mathbf{X^T F}\big)\Big) \quad , \quad \text{or} \tag{3.4}$$

$$\Phi_{\text{SSSE}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X D}_g\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_g + \alpha\mathbf{L}_p)\mathbf{X^T F}\big)\Big) \quad , \tag{3.5}$$

which are essentially feature-based multi-view versions of the objective functions minimized in the Laplacian Eigenmaps [7] and Spatial-Spectral Schroedinger Eigenmaps [10] methods. Note that $\mathbf{X D}_g\mathbf{X^T}$ is positive definite and hence invertible (assuming $\mathbf{X}$ is full rank), and so the minima of Equations (3.4)–(3.5) can be found by using the solution of the generalized eigenvector problems.

3.2 Proposed Solution: SSNE

Next, we introduce a normalization term to these objective functions. As described in Section 2.2, in the domain of spectral clustering it has been shown that the use of normalized Laplacians improves clustering results. The most common way to normalize Laplacian is to normalize the Laplacian matrix by is corresponding degree matrix. Hence, in order to incorporate a normalization term into Equations (3.1)–(3.2) we add there D_g into objective functions. Both types of new objective function take the form:

$$\Phi_{\text{SSNE}}(\mathbf{F}) = \text{tr}\Big(\big(\mathbf{F^T X}(\mathbf{D}_g + \gamma_d\mathbf{L}_d)\mathbf{X^T F}\big)^{-1}\big(\mathbf{F^T X}(\mathbf{L}_g + \gamma_s\mathbf{L}_s + \gamma_p\mathbf{L}_p)\mathbf{X^T F}\big)\Big) \quad , \tag{3.6}$$

where γ_d, γ_s, and γ_p are chosen parameters that weight the influence of $\mathbf{L_s}$, $\mathbf{L_d}$ and $\mathbf{L_p}$. We can see that Φ_{SSNE} reduces to Φ_{LE} when $\gamma_d = \gamma_s = \gamma_p = 0$ and to Φ_{SSSE} when $\gamma_d = \gamma_s = 0$ and $\gamma_p = \alpha$. Since $\mathbf{D}_g + \gamma_d\mathbf{L}_d$ is guaranteed to be positive definite for $\gamma_d \geq 0$, the minimum of (3.6) can be found by solving a generalized eigenvector problem defined as

$$\boldsymbol{X}\left(\mathbf{L}_g + \gamma_s\mathbf{L}_s + \gamma_p\mathbf{L}_p\right)\boldsymbol{X^T f} = \lambda\boldsymbol{X}\left(\mathbf{D}_g + \gamma_d\mathbf{L}_d\right)\boldsymbol{X^T f}, \tag{3.7}$$

by computing the d_{max} smallest non-trivial eigenvalues. The projection for each source domain corresponds to the row blocks of $\mathbf{F}$ such that $\mathbf{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, ..., \boldsymbol{f_{d_{max}}}]$.

Using this optimal solution to project into the latent space yields the *Semi-Supervised Normalized Embedding* (SSNE).

3.3 Computational Complexity

As with SSMA and SEMA, the algorithm for computing SSNE depends mainly on the construction of weighted adjacency matrixes, the construction of graph Laplacian matrices and then computing the generalized eigenvalue decomposition. As described in [27], the complexity of SEMA is given by four parts. First, the computation of the spectral similarities and dissimilarities: $O(DN\log(k)\log(N))$; second, the computation of the spatial similarities: $O(N\log(k_{sp})\log(N))$; third, the solution of the generalized eigenvalue: $O(DNk^3)$; and fourth, obtaining the d smallest non trivial eigenvalues: $O(DN^2)$, where D is the original dimension of the data, N is the number of data points, k is the number the number of spectral nearest neighbors, k_{sp} is the number of spatial nearest neighbors, and d is the desired embedded dimensionality. The full derivation of these computation complexity is described in [27]. Since SSNE only includes the addition of a single term, D_g that must already be computed to construct L_g, the complexity will be that same as SEMA.

Chapter 4

Datasets and Experiment Methodology

In [26, 27] it has been shown that, as a preprocessing step, SSMA and SEMA improve accuracy for classification of remote sensing imagery. Throughout the experiments posed in the thesis, we further demonstrate the effectiveness of SEMA and evaluate our proposed algorithm SSNE. We test on the multi-modal, multispectral data set provided by the 2017 IEEE GRSS Data Fusion Contest. Using the data provided we evaluate both SEMA and SSNE through exploring the parameter space of these algorithms. Further we evaluate the robustness of SSNE by using several subsets of the contest data and application of several classification algorithms.

In this chapter, we first describe the contest data in full in Section 4.1, then we describe the basic system that we will follow for all experiments in Section 4.2, and finally we describe the specific experiments we will perform in Section 4.3.

4.1 Dataset

The 2017 IEEE GRSS Data Fusion Contest [1] provides a multi-modal, multi-temporal, multi-source dataset for exploring land-use classification algorithms. The dataset is split into a set of five training cities: Berlin, Hong Kong, Paris, Rome, and Sao Paulo, and a set of four testing cities: Amsterdam, Chicago, Madrid, and Xian. Several datasets are provided for each city which include: 2-4 Landsat-8 with 9 bands provided, resampled to 100m resolution; a Sentinel-2 with 10 bands provided, resampled to 100m resolution; and three Open Street Map (OSM) layers with land use and building information provided as raster layers at 20m resolution. For the training cities there are ground truth maps for land-use, using classes defined in terms of Local Climate Zones (LCZ) [47, 5], which include ten urban classes and seven rural classes. These ground truth maps are provided as rasters with 100m resolution. For each city, all data has been spatially registered.

For this thesis, we utilize only the data for the five training cities, due to the availability of ground truth. Table 4.1 shows the number of ground truth samples for each city. Figure 4.1 shows two Landsat-8 images, the Sentinel-2 image and the corresponding ground truth map for each of these five cities.

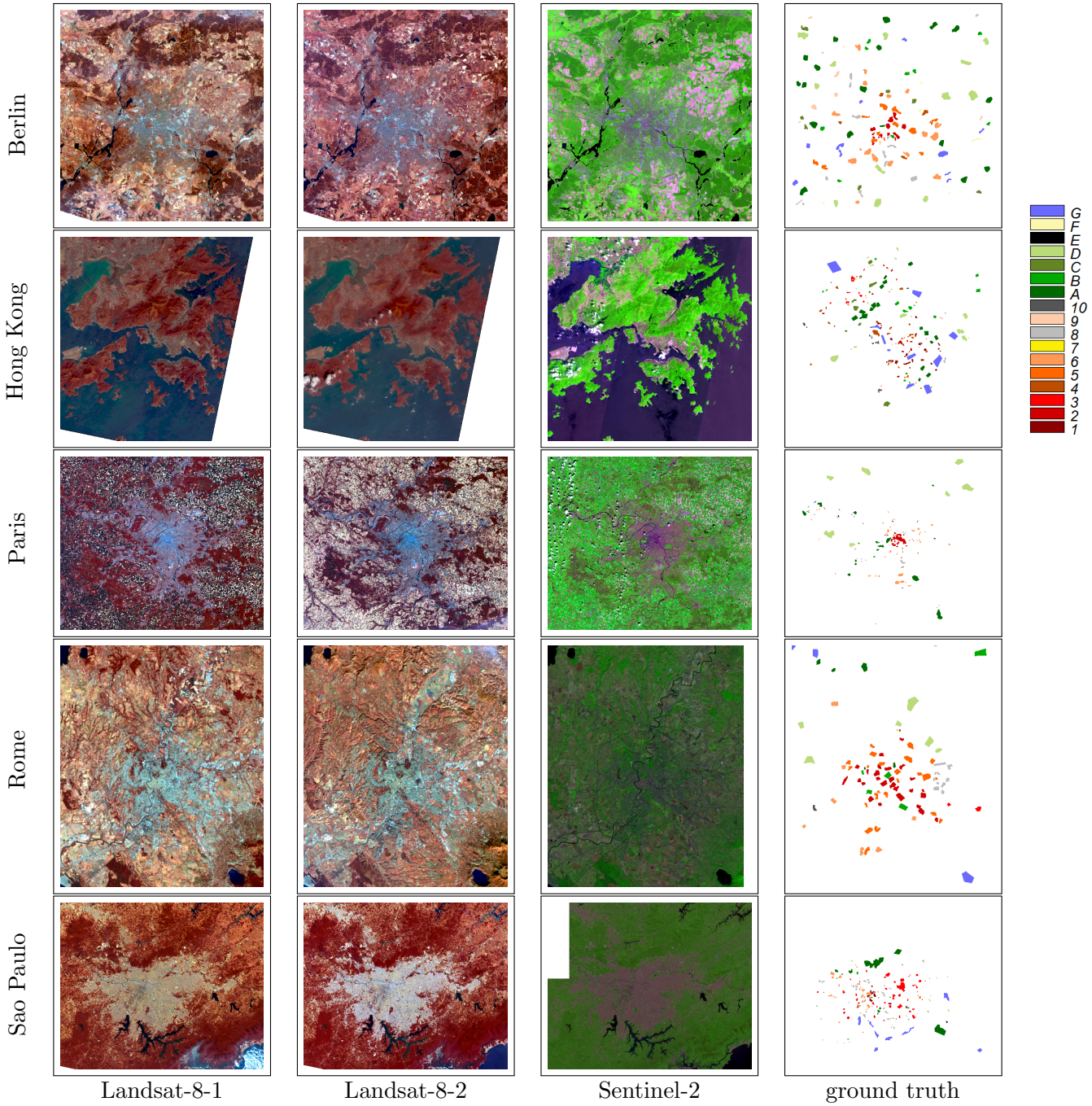

Figure 4.1: **Training cities:** The Landsat-8 (L8-1 and L8-2) images have been rendered by selecting red, green, and blue channels to correspond to the bands having wavelengths 10.9μm, 1.6μm, and 655nm, respectively, and then by adjusting brightness, contrast, and gamma for visualization. The Sentinel-2 (S2) image has been rendered so that the red, green, and blue channels correspond to the bands having wavelengths 2.2μm, 835nm, and 665nm.

Table 4.1: The seventeen types of Local Climate Zones (LCZs), sixteen of which are present in the training data. The number of ground-truth pixels for each class represents the number on the original grid.

LCZ	Type	# GT Pixels				
		Berlin	Hong Kong	Paris	Rome	Sao Paulo
1	Compact High-rise	−−	631	56	−−	955
2	Compact Midrise	1534	179	2705	1551	134
3	Compact Low-rise	−−	326	−−	104	5308
4	Open High-rise	577	673	366	−−	482
5	Open Midrise	2448	126	446	1495	244
6	Open Low-rise	4010	120	2419	480	1862
7	Lightweight Low-rise	−−	−−	−−	−−	−−
8	Large Low-rise	1654	137	748	435	1915
9	Sparsely Built	761	−−	60	−−	335
10	Heavy Industry	−−	219	−−	51	179
A	Dense Trees	4960	1616	4497	284	6359
B	Scattered Trees	1028	407	394	555	302
C	Bush, Scrub	1050	691	−−	−−	−−
D	Low Plants	4424	568	7688	984	376
E	Bare Rock or Paved	−−	−−	214	−−	109
F	Bare Soil or Sand	359	−−	−−	−−	144
G	Water	1732	2379	234	500	3492

4.2 System

The provided ground truth for each image in the 2017 IEEE GRSS Data Fusion Contest data will be divided into a training and testing set. Using the labeled points given by the training set, the images will be aligned using a manifold alignment algorithm. The datasets will be projected into a latent space using a manifold alignment technique in which a classifier will be trained. The points in the testing set, projected into the latent space, will be classified using this classier and compared to the ground truth values. The manifold alignment algorithms used are LE/SSSE, SSMA/SEMA, and SSNE. For classification we use a linear discriminant analysis classifier (LDA), a support vector machine (SVM), and a random forest (RF) classifier. Figure 4.2 shows a flowchart of this process.

Specifically, the pixels in L8-1, L8-2, and S2 corresponding to the training set form the three-view data for each city in $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$, respectively, and the pixels in each image corresponding to the test set are stored in $\widetilde{\mathcal{X}}^{(1)}$, $\widetilde{\mathcal{X}}^{(2)}$, and $\widetilde{\mathcal{X}}^{(3)}$. Since all images are registered, $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$ share a common set of class labels $\mathcal{Y}$, and $\widetilde{\mathcal{X}}^{(1)}$, $\widetilde{\mathcal{X}}^{(2)}$, and $\widetilde{\mathcal{X}}^{(3)}$ share a common set of class labels $\widetilde{\mathcal{Y}}$.

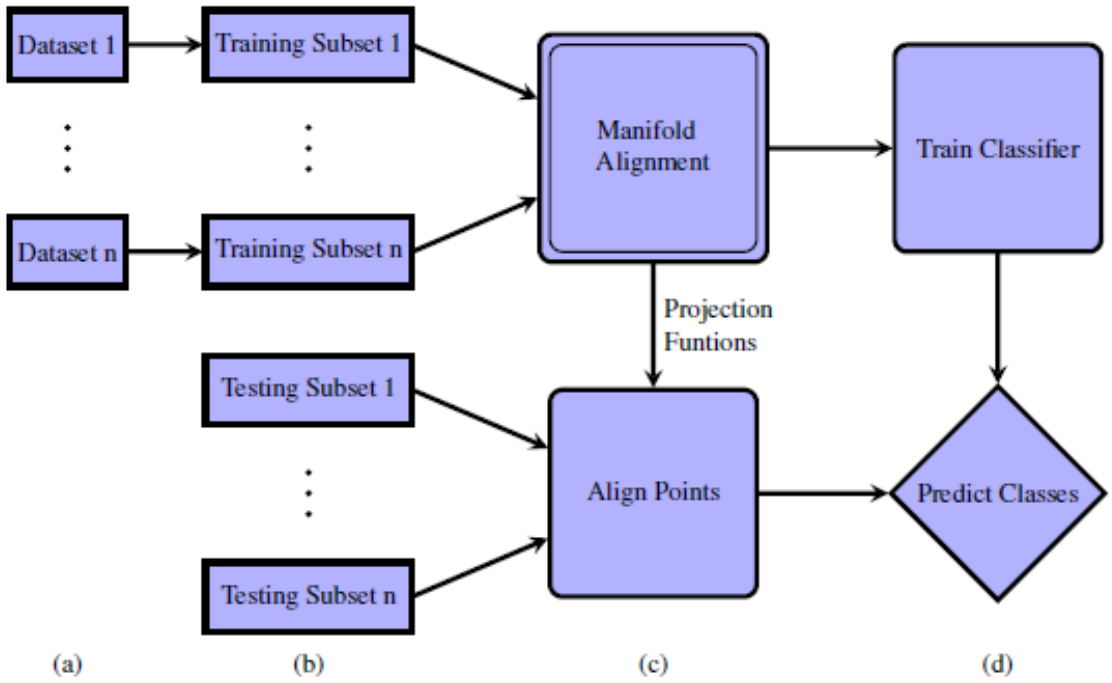

Figure 4.2: **Flowchart of General Experiment Pipeline:** (a) The original n datasets also referred to as views. (b) The split of labeled ground truth points into independent training and testing sets. (c) Manifold alignment is applied to the raw data and the training and testing data is projected into an aligned latent space. (d) A classifier is then trained on the now aligned training data. The trained classifier is used to predict the classes of the testing data.

4.2.1 Classifiers

In addition to comparing manifold alignment algorithms we show their robustness to different classification algorithms. We compare three classifiers: a linear discriminant analysis classifier (LDA), a support vector machine (SVM), and a random forest (RF) classifier.

The LDA classifier was implemented and used for analysis of manifold alignment [10, 26, 27]. The SVM is one-versus all; it uses Gaussian RBF kernels as implemented in MATLAB's `fitcsvm` function, and it employs the heuristic procedure available in `fitcsvm` in order to automatically determine the kernel scaling. The implementation of the SVM classifier used is identical to that used in [26, 27]. The random forest was implemented using scikit-learns's `sklearn.ensemble.RandomForestClassifer`. We set $n\ estimators = 100$, $max\ depth = 10$, $random\ state = 0$, $criterion =$ entropy, and $class\ weight = \{0 : .9, 1 : 1\}$, and utilize the default for all other parameters. These parameters we chosen bases on a parameter search over a small parameter space, we use the parameters that show the best results. However, since we are not trying to optimize the classifiers for our experiments, only show that the SSME/SEMA and SSNE are robust across different classification techniques we do not show these experiments in this thesis.

4.3 Experiments

We propose two sets of experiments: the first shows the performance of SEMA and SSNE for four scenarios and compares classifiers, and the second applies manifold learning to the task of classifying unknown views.

4.3.1 Experiment 1

For this experiment, we utilize the data from a single city, Berlin. We use versions of the two Berlin Landsat-8 images (denoted *L8-1* and *L8-2*) and the Sentinel-2 image (denoted *S2*) that are down sampled versions of the originals in which every other row and column are removed. (This is done for computational expediency.) Furthermore, we normalize the pixel values so that the spectra at each pixel has unit norm. The down sampled version of the ground truth is split into training/testing sets by randomly sampling 50% of the ground truth pixels in each class for training and reserving the remaining pixels for testing. This yields a training/testing split of 3061/3060 pixels across 12 classes. The pixels in L8-1, L8-2, and S2 corresponding to the training set form the three-view data in $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$, respectively, and the pixels in each image corresponding to the test set are stored in $\widetilde{\mathcal{X}}^{(1)}$, $\widetilde{\mathcal{X}}^{(2)}$, and $\widetilde{\mathcal{X}}^{(3)}$. Since all images are registered, $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$ share a common set of class labels $\mathcal{Y} = \{y_j | j = 1, \ldots, 3061\}$, and $\widetilde{\mathcal{X}}^{(1)}$, $\widetilde{\mathcal{X}}^{(2)}$, and $\widetilde{\mathcal{X}}^{(3)}$ share a common set of class labels $\widetilde{\mathcal{Y}} = \{\tilde{y}_j | j = 1, \ldots, 3060\}$.

We describe a number of possible scenarios we will investigate for exploiting multiple view data for land use classification. Across all scenarios, we use the same training/testing split, and we use only the training data to compute projections into a latent space using SSNE, SSMA, or SEMA. Once the training data is projected, it is used to train one-versus-all support vector machine (SVM) classifiers. The testing data is then projected into the latent space, and class labels are predicted using our three classifiers: LDA, SVM, RF.

The scenarios we explore are as follows:

- **Scenario A: Baseline:** All views are treated independently, and no dimensionality reduction / data fusion is performed. Separate classifiers are trained directly on the data in $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, and $\mathcal{X}^{(3)}$, respectively.
- **Scenario B: Independent Views with Dimensionality Reduction:** All views are treated independently, so that $\mathcal{S} = \mathcal{D} = \emptyset$. Dimensionality reduction is performed via SSNE with $\gamma_s = \gamma_d = 0$. If $\gamma_p = 0$, this is roughly equivalent to performing a feature-based version of Laplacian Eigenmaps independently on each view; if $\gamma_p > 0$, this is roughly equivalent to performing independent feature-based versions of SSSE on each view. (The "rough" equivalence is because SSNE would project the data from each image into a 9- or 10-dimensional subspace of $\mathbb{R}^{28}$, whereas feature-based LE or SSSE would project the data directly into $\mathbb{R}^9$ or $\mathbb{R}^{10}$.) Neither SSMA nor SEMA are possible in this scenario.

- **Scenario C: Labeled Pairwise Similarities/Dissimilarities:** For each of the 12 classes, one pair of pixels (in different spatial locations) is randomly selected and assigned as a similarity in $\mathcal{S}_{1,2}$, one pair in $\mathcal{S}_{1,3}$, and one pair in $\mathcal{S}_{2,3}$. This simulates a scenario in which an expert analyst would manually identify, for each class, a point in $\mathcal{X}^{(i)}$ and a different point in $\mathcal{X}^{(j)}$ that both belong to the same class. The analyst would therefore be required to manually identify 36 pairs of points. The dissimilarity set is automatically constructed by populating it with pairs of identified points that are from different classes. SSNE, SSMA, and SEMA are all possible in this scenario.

- **Scenario D: Similarities via Alignment:** Registration and resampling of the images into a common coordinate system is exploited by defining similarities between pixels from each view that share the same spatial location. Specifically, $\mathcal{S}_{i,j} = \left\{ \left(\mathbf{x}_s^{(i)}, \mathbf{x}_s^{(j)} \right) \middle| \, s = 1, \ldots, 3061 \right\}$, $1 \leq i, j \leq 3$, $i \neq j$. No dissimilarities are provided, so neither SSMA nor SEMA are possible in this scenario. Note that this is the only scenario in which the spatial alignment of the multiple view data is exploited.

4.3.2 Experiment 2

Next, we expand our dataset to include all five training cities from the contest data. We use versions of the two Landsat-8 images (denoted *L8-1* and *L8-2*) and the Sentinel-2 image (denoted *S2*) for each city (totaling 15 views). Each view is a down sampled version of the originals in which every other row and column are removed, and the pixel values have been normalized so that the spectra at each pixel has unit norm.

We perform two scenarios. First, we use the data from four cities: Hong Kong, Paris, Rome, and Sao Paulo, to align the three views (L8-1, L8-2, S2) and train an SVM classifier. With this classifier, we predict the classes of the pixels in the Berlin image that correspond to points with a class label provided by the contest. Second, we use the data from four cities: Berlin, Hong Kong, Paris, and Rome to align the views and train an SVM classifier in order to predict the classes of the pixels in the Sao Paulo image that correspond to points with a class label provided by the contest. We run this experiment of two cities since the class distributions, and consequently the training/testing split, vary between each city.

For this experiment, we utilize a baseline similar to the one described in Scenario A, in which each view is classified independently with no fusion. We then compare the application of SEMA and SSNE to the baseline. For SEMA, $\alpha = \mu = 100$ and for SSNE, $\gamma_s = \gamma_d = \gamma_p = 100$.

Chapter 5

Results and Discussion

For each scenario in each experiment, we report two classification performance measures for the test data: overall accuracy (OA) and kappa coefficient (κ). Overall accuracy is defined as ratio of the total number of correctly-predicted class labels to the total number of test points evaluated. Kappa coefficient is defined in Senseman et al. [44] and measures the improvement of a classification result over the result that would be achieved by random assignment of class labels.

5.1 Experiment 1: Classifying Berlin

5.1.1 Scenario A: Baseline

When training classifiers independently on each data set with no dimensionality reduction, the resulting OA values for the test data are shown in Table 5.1. It is clear from these baseline values that the raw data available from the Landsat images are better able to predict land use than the raw data from the Sentinel image.

Table 5.1: **Experiment 1, Scenario A:** Classification results for SVM, LDA, and RF when training classifiers independently on each data set with no dimensionality reduction with the Berlin data split. This illustrates both the baseline OA as well as a baseline comparison between our three classifiers.

Classifier	OA			κ		
	L8-1	L8-2	S2	L8-1	L8-2	S2
SVM	0.8268	0.8072	0.7428	0.8000	0.7769	0.7009
LDA	0.7327	0.7314	0.6997	0.6925	0.6895	0.6508
RF	0.7324	0.7418	0.6840	0.6867	0.6985	0.6271

In addition, we can see that using an SVM yields higher OA and κ values for all three views. Figure 5.1 shows the results of k-fold cross-validation for each classifier. Again, we see that the Landsat-8 images have slightly higher OA values than the Sentinel-2 image. Comparing classification methods, SVM consistently preforms the best.

5.1.2 Scenario B: Independent Views with Dimensionality Reduction

There are two possible ways to execute this scenario. One way is to apply feature-based versions of Laplacian Eigenmaps and Spatial-Spectral Schroedinger Eigenmaps individually to each

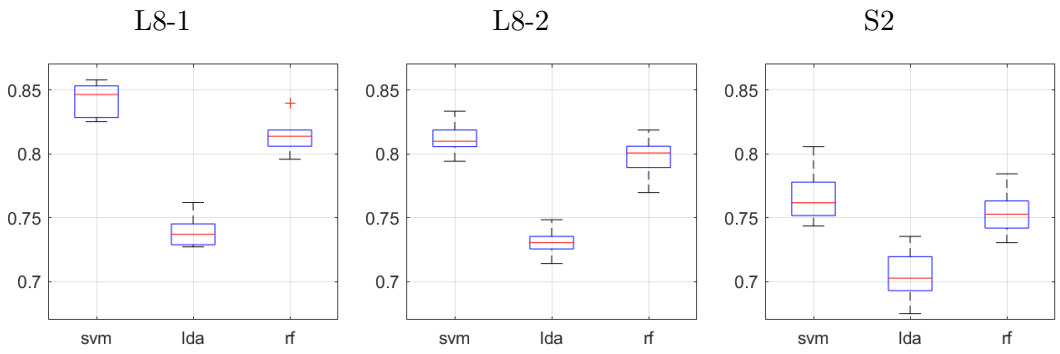

Figure 5.1: **Experiment 1, Scenario A, OA:** K-fold cross-validation OA of Berlin to test sensitivity of each classifier, $k = 10$. Note that this plot shows the results of a 90/10 training/testing data split. We see that SVM and LDA yield $2 - 3\%$ higher OA values across all views with the larger training set. However, RF yield $6 - 9\%$ higher OA values across all views with the larger training set.

view, yielding three separate generalized eigenvector problems. The second is to perform SSNE under the assumption that there are no similarities or dissimilarities, yielding one block diagonal generalized eigenvector problem. From a theoretical point of view, the only difference between these two approaches is that the first will yield three separate "latent" spaces, whereas the second will yield one larger-dimensional latent space which contains three orthogonal subspaces representing the embeddings of each view.

Figures 5.2–5.3 illustrate classification results from both approaches. Further, these figures compare the sensitivity over different γ_p values, which weights the contribution of spatial feature, and compare the effect of applying each of our three classifiers. Again, the use of the SVM classifier yields slightly higher OA values; however, when the random forest classifier is used, there is a greater improvement from the baseline for both feature-based LE and SSNE. Also, the random forest classifier appears to enable good classification performance for lower dimensionality of the latent space (lower values of q) than with the other two classifiers.

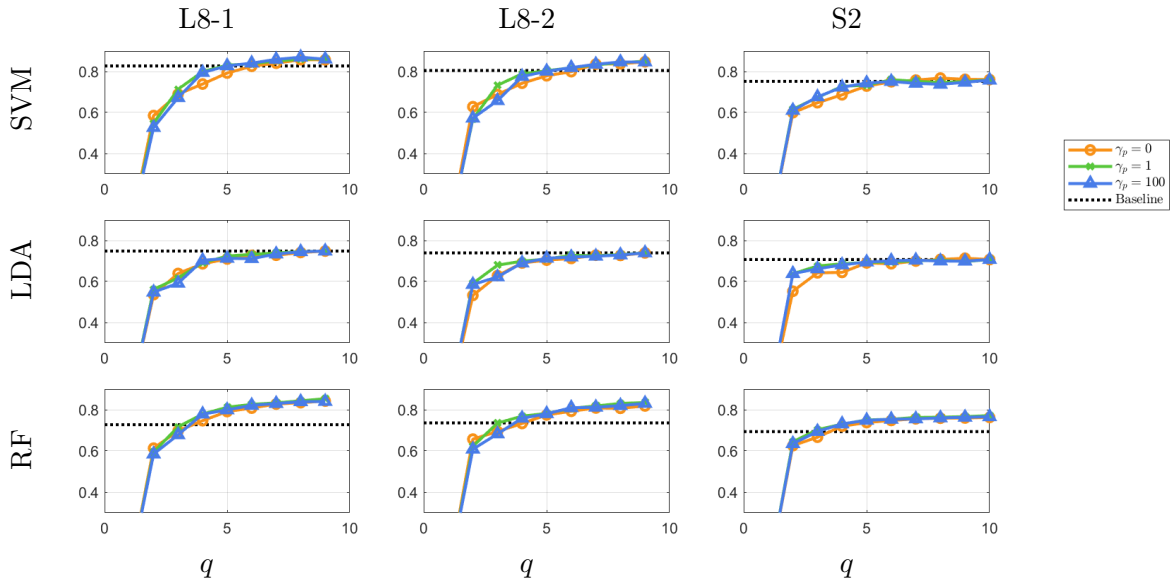

Figure 5.2: **Experiment 1, Scenario B, LE:** Classification performance (OA) on the test set for each image, after the training sets for each image have been used individually to perform feature-based LE ($\gamma_p = 0$) or feature-based SSSE ($\gamma_p = 1$, $\gamma_p = 100$). The horizontal axes represent the feature dimension q, and the baseline results from Scenario A are added to the plots for comparison. For all cases, these feature representations yield classifiers that outperform the baseline when $q > 6$.

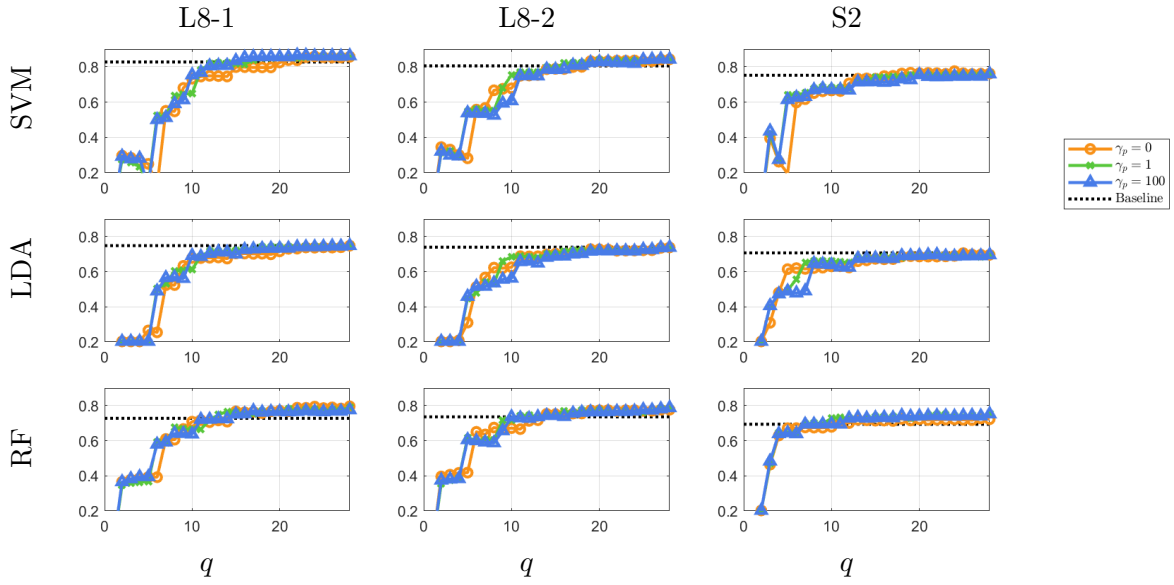

Figure 5.3: **Experiment 1, Scenario B, SSNE, OA:** Classification performance (OA) on the test set for each image, after the training sets for each image have been used to perform SSNE with $\mathcal{S} = \mathcal{D} = \emptyset$. The horizontal axes represent the feature dimension q, and the baseline results from Scenario A are added to the plots for comparison. For all cases, these feature representations yield classifiers that outperform the baseline when the latent space has dimension $q > 19$.

Table 5.2: **Experiment 1, Scenario B, SVM:** Classification results versus baseline (Scenario A) with the use of our SVM classifier. Each performance measure is based on the maximum possible embedding dimension q. For feature-based LE/SSSE, $q = 9$ for both Landsat images and $q = 10$ for the Sentinel image. For SSNE, $q = 28$ for all images. The OA and *kappa* values remain consistence when varying $gamma_p$; this shows that for this data set the inclusion of spatial features does not necessarily improve classification.

		Feature-based LE/SSSE			SSNE with $\mathcal{S} = \mathcal{D} = \emptyset$			Baseline (Scenario A)		
Measure	γ_p	L8-1	L8-2	S2	L8-1	L8-2	S2	L8-1	L8-2	S2
OA	0	0.8621	0.8382	0.7582	0.8611	0.8386	0.7627	0.8268	0.8072	0.7428
	1	0.8650	0.8451	0.7611	0.8647	0.8425	0.7660			
	100	0.8670	0.8474	0.7572	0.8663	0.8425	0.7634			
κ	0	0.8409	0.8133	0.7190	0.8398	0.8136	0.7239	0.8000	0.7769	0.7009
	1	0.8443	0.8213	0.7225	0.8440	0.8182	0.7277			
	100	0.8466	0.8240	0.7179	0.8458	0.8182	0.7247			

Table 5.3: **Experiment 1, Scenario B, SVM:** Per-class and overall classification results versus baseline (Scenario A) with the use of our SVM classifier. Each performance measure is based on the maximum possible embedding dimension q. For feature-based LS/SSSE, $q = 9$ for both Landsat images and $q = 10$ for the Sentinel image. For SSNE, $q = 28$ for all images.

			Baseline (Scenario A)			LE/SSSE $\gamma_p = 100$			SSNE with $\mathcal{S} = \mathcal{D} = \emptyset, \gamma_p = 100$		
class	train	test	L8-1	L8-2	S2	L8-1	L8-2	S2	L8-1	L8-2	S2
2	191	190	0.82	0.67	0.62	0.80	0.80	0.65	0.81	0.82	0.66
4	72	72	0.26	0.06	0.07	0.47	0.36	0.06	0.43	0.29	0.06
5	304	303	0.59	0.38	0.42	0.55	0.49	0.35	0.60	0.48	0.33
6	499	501	0.86	0.88	0.86	0.90	0.91	0.86	0.90	0.93	0.87
8	207	206	0.71	0.53	0.57	0.81	0.77	0.62	0.80	0.74	0.66
9	94	96	0.66	0.24	0.21	0.76	0.63	0.16	0.74	0.63	0.11
A	621	621	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	1.00
B	128	128	0.67	0.34	0.41	0.80	0.83	0.45	0.79	0.78	0.46
C	133	132	0.48	0.44	0.45	0.63	0.60	0.48	0.61	0.56	0.42
D	550	549	0.99	0.98	0.94	0.99	1.00	0.98	0.99	0.99	0.99
F	46	46	0.45	0.34	0.26	0.62	0.53	0.43	0.66	0.55	0.40
G	216	216	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Overall Accuracy			0.83	0.80	0.75	0.86	0.85	0.76	0.86	0.84	0.76
Average Accuracy			0.71	0.66	0.57	0.78	0.74	0.58	0.78	0.73	0.58
Average Precision			0.77	0.72	0.66	0.81	0.81	0.67	0.83	0.80	0.68
Average Recall			0.71	0.66	0.57	0.78	0.74	0.58	0.78	0.73	0.58
kappa			0.80	0.77	0.71	0.84	0.82	0.72	0.84	0.82	0.72
Dimensions			9	9	10	9	9	10	28	28	28

In Table 5.2 we see that when the maximum possible embedding dimension is chosen,

Scenario B outperforms the baseline (Scenario A) by margins of $4-5\%$ in OA and κ for both Landsat images, and by $2-3\%$ in OA and κ for the Sentinel image. Table 5.3 illustrates that the classes with more training samples provided yield higher accuracy values, specifically: compact open low-rise (LCZ 6), dense trees (LCZ A), and low plants (LZC D) have the three largest training sets and yield higher accuracies by a margin of $2-5\%$. Though water (LCZ G) has an average amount of samples, it is almost always classified correctly. For all classes the application of LE and SSNE yield higher accuracy values than the baseline scenario.

5.1.3 Scenario C: Labeled Pairwise Similarities/Dissimilarities

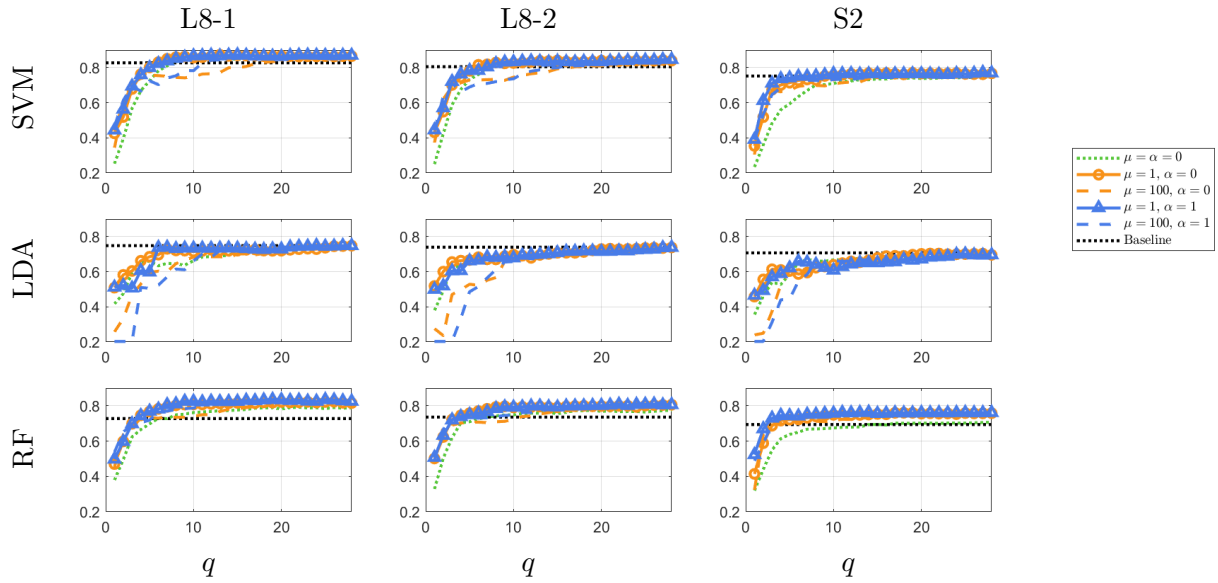

Figure 5.4: **Experiment 1, Scenario C, SSMA/SEMA:** Classification performance (OA) on the test set for each image, after the training sets for each image have been used to perform SSNE with many pairwise similarities/dissimilarities. The horizontal axes represent the feature dimension q, and the baseline results from Scenario A are added to the plots for comparison. The use of SEMA (when $\alpha > 0$) appears to enable lower choices for q than SSMA (when $\alpha = 0$).

From Figures 5.4–5.5, we see that when 36 pairs of corresponding labels are provided that span all of the classes, both SSNE and SEMA appear to enable good classification performance for lower dimensionality of the latent space (lower values of q) than is possible from SSMA or from SSNE in Scenario B, when the SVM and RF classifiers are used. The use of the LDA classifier shows similar results to the best cases in Scenario B.

To determine whether there is any actual increase in classification performance, we consider Tables 5.4–5.5. For SSNE, it appears that each performance measure can be slightly improved (by a margin of $1-2\%$) by incorporating spatial information ($\gamma_p \neq 0$). It is difficult to gauge whether the incorporation of spatial information in SSMA ($\mu, \alpha \neq 0$, yielding the SEMA algorithm) significantly

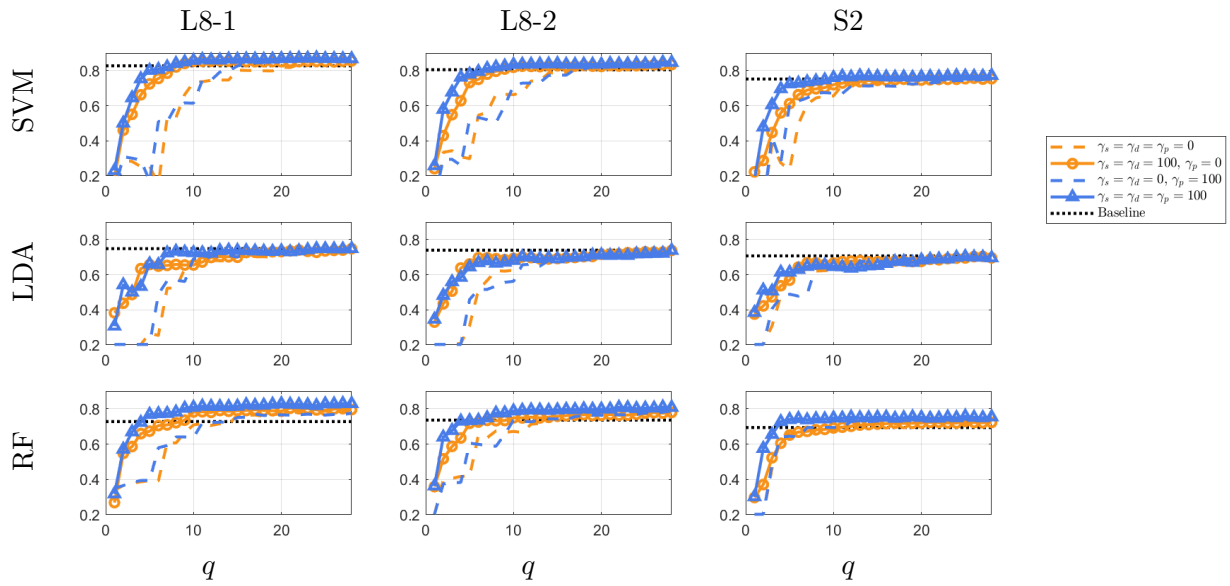

Figure 5.5: **Experiment 1, Scenario C, SSNE:** Classification performance (OA) on the test set for each image, after the training sets for each image have been used to perform SSNE with many pairwise similarities/dissimilarities. The horizontal axes represent the feature dimension q, and the baseline results from Scenario A are added to the plots for comparison. It is clear that the inclusion of similarities/dissimilarities (when $\gamma_s, \gamma_d > 0$) enables much lower choices for q than Scenario B (when $\gamma_s = \gamma_d = 0$).

improves performance measures in this scenario.

Of particular interest, however, is the relationship between Scenario C and Scenario B: when compared to the results generated by performing dimensionality reduction independently on each view, the use of similarities/dissimilarities appears to improve classification performance measures by $1-2\%$ for the Sentinel-2 image; however, it does not appear to significantly improve classification performance on the Landsat-8 images. This suggests that the information present in the Landsat-8 images is useful in improving classification of the Sentinel-2 image, but perhaps the converse may not hold. In Tables 5.4, 5.5 , we see the results for OA and κ for the maximum possible embedding dimension $q = 28$ for both SSMA/SEMA and SSNE. Aside from the first row in Table 5.4, when $\alpha = \mu = 0$, the OA and κ values are consistent across parameter selection.

Figures 5.6 and 5.7 show the results of cross validation of each classifier for both LE and SSNE. We see the same trends when comparing the OA accuracies of different classifiers as in the baseline; SVM consistently preforms better than the other two classifiers. We see that the application of SSMA and SSNE yield the same or better OA as the baseline across all classifiers.

Table 5.4: **Experiment 1, Scenario C, SSMA/SEMA:** Classification results for various choices of μ and α for SSMA/SEMA with the SVM classifier. Each performance measure is based on the maximum possible embedding dimension q.

		OA			κ		
μ	α	L8-1	L8-2	S2	L8-1	L8-2	S2
0	0	0.8497	0.8330	0.7474	0.8263	0.8069	0.7058
1	0	0.8605	0.8408	0.7650	0.8389	0.8161	0.7266
100	0	0.8624	0.8399	0.7732	0.8412	0.8150	0.7363
1	1	0.8611	0.8444	0.7745	0.8397	0.8203	0.7380
100	1	0.8667	0.8412	0.7745	0.8462	0.8166	0.7380
1	100	0.8673	0.8402	0.7729	0.8469	0.8154	0.7360
100	100	0.8667	0.8418	0.7729	0.8462	0.8173	0.7361

Table 5.5: **Experiment 1, Scenario C, SSNE:** Classification results for various choices of γ_s, γ_d, and γ_p for SSNE with the SVM classifier. Each performance measure is based on the maximum possible embedding dimension q.

			OA			κ		
γ_s	γ_d	γ_p	L8-1	L8-2	S2	L8-1	L8-2	S2
0	0	0	0.8611	0.8386	0.7627	0.8398	0.8136	0.7239
100	0	0	0.8601	0.8350	0.7650	0.8386	0.8094	0.7268
0	100	0	0.8631	0.8382	0.7641	0.8420	0.8131	0.7253
100	100	0	0.8523	0.8392	0.7618	0.8294	0.8141	0.7228
0	0	100	0.8663	0.8425	0.7634	0.8458	0.8182	0.7247
100	0	100	0.8637	0.8389	0.7680	0.8428	0.8140	0.7302
0	100	100	0.8647	0.8405	0.7706	0.8440	0.8158	0.7330
100	100	100	0.8598	0.8448	0.7765	0.8382	0.8207	0.7402

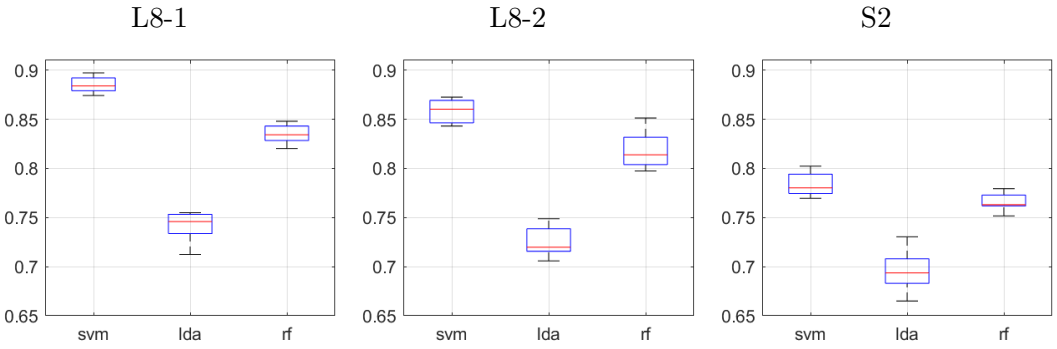

Figure 5.6: **Experiment 1, Scenario C, SEMA, OA:** K-fold cross-validation OA of Berlin to test sensitivity of each classifier, $k = 10$. Note that this plot shows the results of a 90/10 training/testing data split for $\alpha = \mu = 100$.

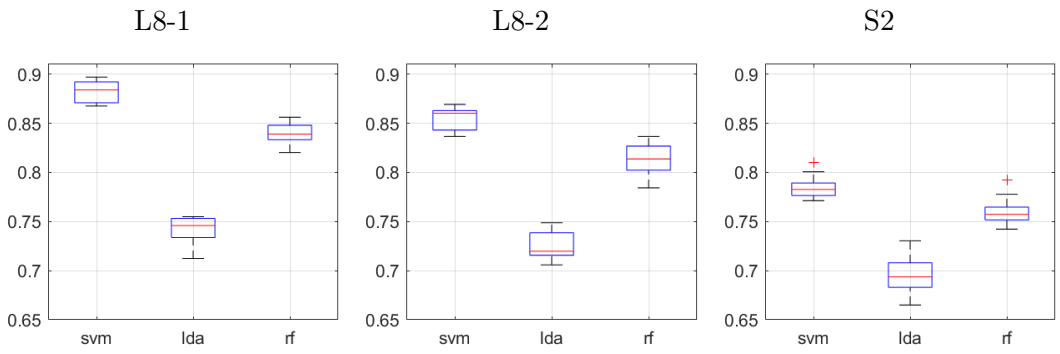

Figure 5.7: **Experiment 1, Scenario C, SSNE OA:** K-fold cross-validation OA of Berlin to test sensitivity of each classifier, $k = 10$. Note that this plot shows the results of a 90/10 training/testing data split for $\gamma_s = \gamma_d = \gamma_p = 100$.

Table 5.6: **Experiment 1, Scenario C, SVM:** Per-class and overall classification results with the use of our SVM classifier. Each performance measure is based on the maximum possible embedding dimension $q = 28$.

			SSMA/SEMA $\mu = \alpha = 100$			SSNE $\gamma_s = \gamma_d = \gamma_p = 100$		
class	train	test	L8-1	L8-2	S2	L8-1	L8-2	S2
2	191	190	0.80	0.74	0.67	0.81	0.78	0.70
4	72	72	0.47	0.28	0.07	0.36	0.28	0.07
5	304	303	0.64	0.51	0.44	0.63	0.51	0.41
6	499	501	0.93	0.90	0.90	0.93	0.91	0.89
9	207	206	0.83	0.78	0.68	0.84	0.81	0.67
8	94	96	0.72	0.61	0.07	0.72	0.60	0.09
A	621	621	1.00	1.00	0.99	1.00	1.00	0.99
B	128	128	0.74	0.79	0.50	0.71	0.79	0.49
C	133	132	0.59	0.59	0.42	0.59	0.61	0.42
D	550	549	0.99	0.99	0.99	0.99	0.99	0.99
F	46	46	0.70	0.52	0.37	0.67	0.54	0.35
G	216	216	1.00	1.00	1.00	1.00	1.00	0.99
Overall Accuracy			0.87	0.84	0.78	0.87	0.85	0.77
Average Accuracy			0.78	0.73	0.59	0.77	0.74	0.59
Average Precision			0.84	0.79	0.71	0.83	0.81	0.70
Average Recall			0.78	0.73	0.59	0.77	0.74	0.59
kappa			0.85	0.81	0.74	0.85	0.82	0.73
Dimensions			28	28	28	28	28	28

5.1.4 Scenario D: Similarities via Alignment

From Figure 5.8, we see that when SSNE exploits the spatial alignment of the three views in the form of pairwise similarities, it enables good classification performance for lower values of q than

when it does not exploit this information (as in Scenario B). Likewise, from Table 5.7, the achieved classification performance measures rival those of Scenario C. Similarly, to that scenario, it appears that the fusion involved in this scenario improves classification performance in the Sentinel-2 image but does not seem to significantly improve classification performance in the Landsat-8 images. Again, in Table 5.8, we see the same trends for classification accuracy of each class as in Tables 5.3-5.6.

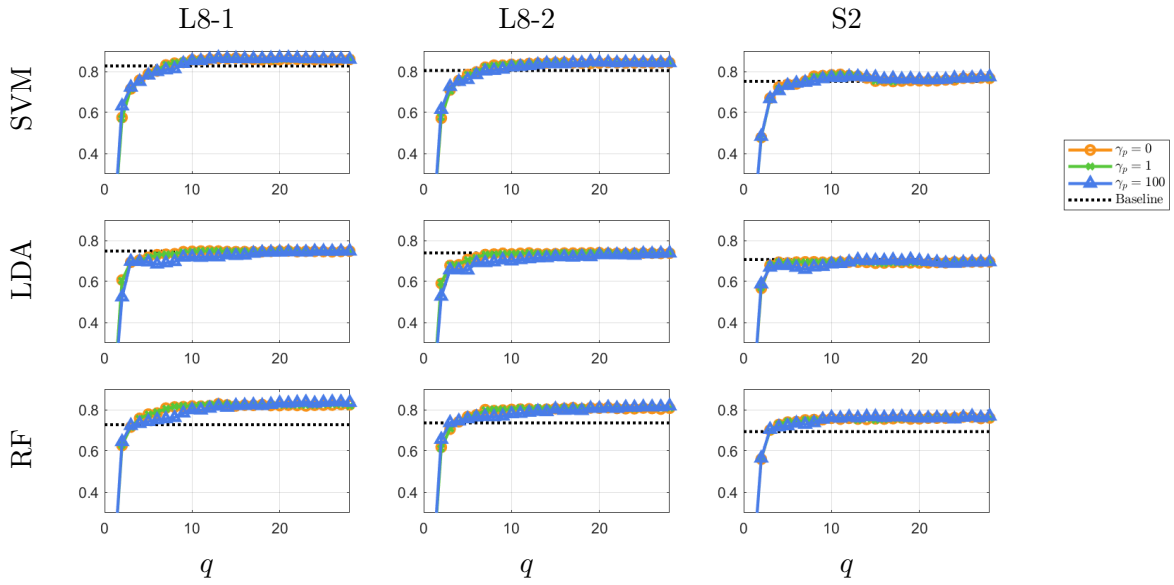

Figure 5.8: **Experiment 1, Scenario D:** Classification performance (OA) on the test set for each image, after the training sets for each image have been used to perform SSNE with similarities provided across views for pixels in the same location and $\gamma_s = 100$. The horizontal axes represent the feature dimension q, and the baseline results from Scenario A are added to the plots for comparison. For all three images and classifiers, these feature representations yield classifiers that outperform the baseline when the latent space has dimension $q > 9$.

Table 5.7: **Experiment 1, Scenario D, SVM:** Classification results for $\gamma_s = 100$ and various choices of γ_p. Each performance measure is based on the maximum possible embedding dimension $q = 28$.

Measure	γ_p	L8-1	L8-2	S2
OA	0	0.8624	0.8382	0.7690
	1	0.8624	0.8389	0.7690
	100	0.8670	0.8418	0.7703
κ	0	0.8413	0.8132	0.7313
	1	0.8413	0.8140	0.7313
	100	0.8466	0.8175	0.7328

To put these results in the context of the workflow of an analyst using multiple view data to

Table 5.8: **Experiment 1, Scenario D, SVM:** Per-class and overall classification results with the use of our SVM classifier. Each performance measure is based on the maximum possible embedding dimension $q = 28$.

			SSNE $\gamma_s = \gamma_p = 100$		
class	train	test	L8-1	L8-2	S2
2	191	190	0.81	0.82	0.68
4	72	72	0.43	0.29	0.08
5	304	303	0.58	0.49	0.39
6	499	501	0.90	0.93	0.88
8	207	206	0.80	0.73	0.67
9	94	96	0.73	0.64	0.21
A	621	621	0.99	0.99	0.99
B	128	128	0.77	0.78	0.49
C	133	132	0.62	0.56	0.45
D	550	549	0.99	0.99	0.99
F	46	46	0.62	0.55	0.40
G	216	216	1.00	1.00	1.00
Overall Accuracy			0.86	0.84	0.77
Average Accuracy			0.77	0.73	0.60
Average Precision			0.83	0.80	0.72
Average Recall			0.77	0.73	0.60
kappa			0.84	0.82	0.74
Dimensions			28	28	28

perform land-use classification, we note that if the multiple views of the data are registered, SSNE can exploit the alignment to yield similar classification results to SSMA/SEMA in Scenario C. However, Scenario D shows that this can be done *without the analyst having to provide any extra pairs of class labels.*

5.2 Experiment 2: Classifying an Unknown View

In Experiment 1, we demonstrated the performance of SEMA and SSNE in several scenarios. In that experiment we focused on the classification of a single city from multiple views; in addition, we used ground truth points from all views in both the training and testing of the algorithms. However, it is often the case that we need to classify an unknown image.

In this section, we evaluate the performance of SEMA and SSNE compared to the baseline for the classification of an unknown city. For this experiment we utilize two Landsat-8 (L8-1 and L8-2) images, the Sentinel-2 (S2) image and the ground truth provided for all the training cities in the contest training data. The dataset is formed by simply concatenating the spectral features from each city into a single dataset for each view, i.e. each group of Landsat-8 images and the Sentinel-2 image.

Table 5.9: **Experiment 2, Berlin:** The class training and testing counts and classification results for SEMA and SSNE compared to the baseline. The emphasized values show improvement from the baseline.

			Baseline			SSMA/SEMA $\mu = \alpha = 100$			SSNE $\gamma_s = \gamma_d = \gamma_p = 100$		
class	train	test	L8-1	L8-2	S2	L8-1	L8-2	S2	L8-1	L8-2	S2
1	412	-	-	-	-	-	-	-	-	-	-
2	1148	381	0.58	0.95	0.23	0.00	0.88	0.21	0.00	0.89	**0.39**
3	1437	-	-	-	-	-	-	-	-	-	-
4	395	144	0.06	0.00	0.26	0.00	0.00	**0.28**	0.00	**0.01**	**0.34**
5	583	607	0.01	0.09	0.37	**0.23**	0.01	0.27	**0.34**	0.02	0.24
6	1219	1000	0.00	0.37	0.38	0.00	**0.61**	**0.39**	0.00	**0.65**	0.46
7	-	-	-	-	-	-	-	-	-	-	-
8	820	413	0.29	0.43	0.15	**0.37**	0.11	**0.20**	**0.30**	0.09	**0.17**
9	98	190	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**
10	113	-	-	-	-	-	-	-	-	-	-
A	3192	1242	0.00	0.59	0.90	0.00	**0.96**	**0.99**	0.00	**0.92**	**0.99**
B	414	256	0.00	0.00	0.16	0.00	0.00	0.09	0.00	0.00	0.10
C	175	265	0.00	0.08	0.14	0.00	0.10	0.09	0.00	0.05	0.08
D	2403	1099	0.55	0.78	0.18	0.34	0.42	0.13	0.35	0.49	0.16
E	79	-	-	-	-	-	-	-	-	-	-
F	38	92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
G	1651	432	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Overall Accuracy			0.23	0.49	0.43	0.18	**0.51**	0.43	0.19	**0.52**	**0.45**
Average Accuracy			0.21	0.36	0.31	0.16	0.34	0.30	0.17	0.34	0.33
Average Precision			0.15	0.26	0.28	0.08	0.24	0.27	0.06	0.25	0.29
Average Recall			0.21	0.36	0.31	0.16	0.34	0.30	0.17	0.34	0.33
kappa			0.15	0.42	0.35	0.09	0.43	0.35	0.09	0.44	0.37
Dimensions			9	9	10	28	28	28	28	28	28

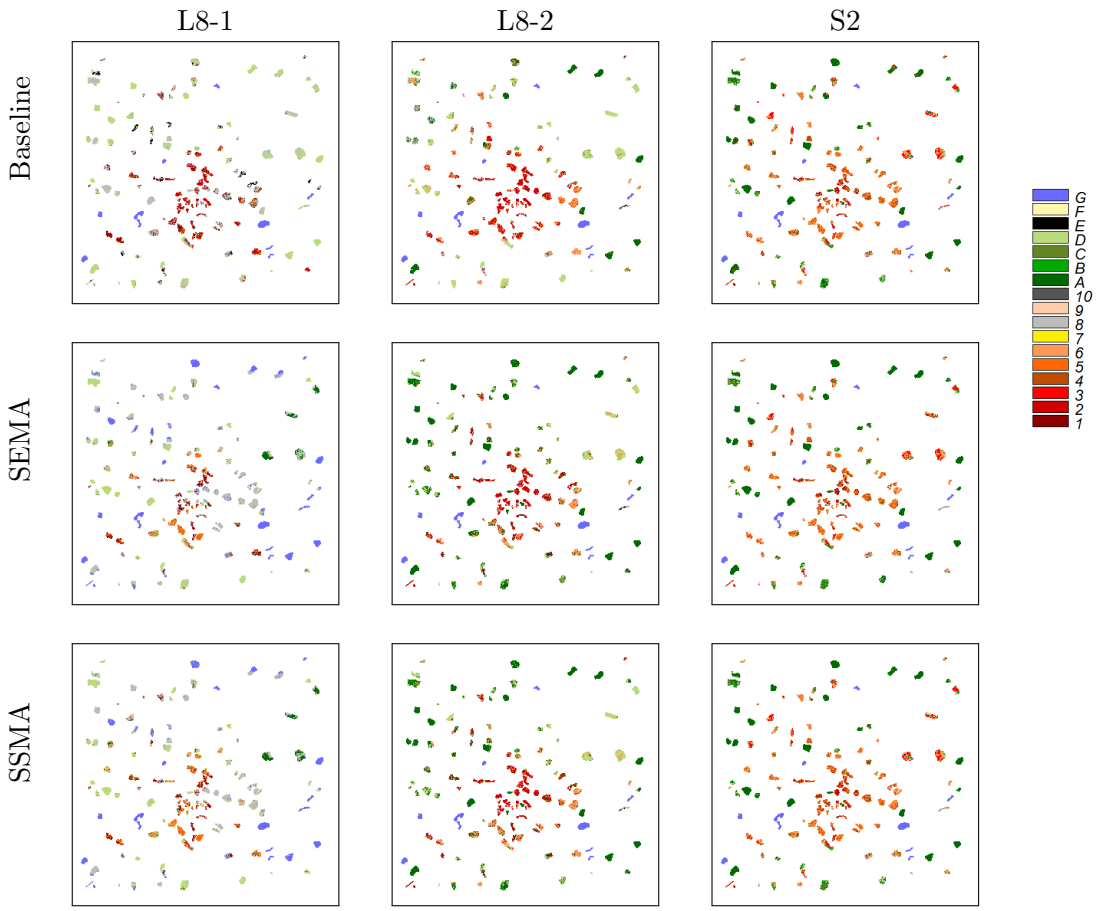

Figure 5.9: **Experiment 2, Berlin:** Classification map of predicted classes using the baseline method, SEMA, and SSNE for each view.

We train and SVM classifier using the three views for each Hong Kong, Paris, Rome, and Sao Paulo and test the classifier on the three views for Berlin and w e train and SVM classifier using the three views for each Berlin, Hong Kong, Paris, and Rome and test the classifier on the three views for Sao Paulo. We compare the testing accuracy when no manifold alignment is applied (the baseline), SEMA, and SSNE. The baseline application is akin to Scenario A in Experiment 1 in which each view is classified independently. For SEMA, $\alpha = \mu = 100$ and for SSNE, $\gamma_s = \gamma_d = \gamma_p = 100$.

In Table 5.9 the first three columns show the number of training and testing points for each class. We see that for this experiment the class training/testing sets are unbalanced with the testing data varying from $0 - 194\%$ of the training data per class. The remaining columns illustrate the classification accuracy of the baseline and with the application of SEMA and SSNE. We see that for many classes the per-class accuracy is very low or even almost 0% accurate. We note that the κ values for all cases demonstrate that our results are still better than the random assignment of class labels.

Table 5.10: **Experiment 2, Sao Paulo:** The class training and testing counts and classification results for SEMA and SSNE compared to the baseline. The emphasized values show improvement from the baseline.

			Baseline			SSMA/SEMA $\mu = \alpha = 100$			SSNE $\gamma_s = \gamma_d = \gamma_p = 100$		
class	train	test	L8-1	L8-2	S2	L8-1	L8-2	S2	L8-1	L8-2	S2
1	171	241	0	0	0.008	0	0	0.00	0.00	0	0.00
2	1493	36	0.33	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00
3	107	1330	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.01
4	414	125	0.01	0.00	0.00	**0.02**	0.00	0.00	**0.08**	0.00	0.00
5	1131	59	0.10	0.00	0.05	0.00	0.00	0.03	0.00	0.00	0.05
6	1751	468	0.14	0.00	0.09	0.00	0.00	0.03	0.00	0.00	0.03
7	-	-	-	-	-	-	-	-	-	-	-
8	744	489	0.55	0.11	0.22	0.49	**0.19**	**0.23**	0.38	**0.47**	**0.24**
9	204	84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	69	44	0.00	0.00	0.00	0.48	0.00	0.00	**0.57**	0.00	0.00
A	2844	1590	0.51	0.00	0.59	0.00	0.00	0.54	0.00	0.00	0.44
B	598	72	0.56	0.00	0.03	0.51	0.00	0.00	0.13	0.00	0.00
C	440	-	-	-	-	-	-	-	-	-	-
D	3406	96	0.54	0.99	0.69	0.38	0.84	0.61	0.36	0.88	0.60
E	51	28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F	92	38	0.00	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.03
G	1209	874	0.98	0.99	0.97	0.98	0.99	**0.98**	0.98	0.99	**0.99**
Overall Accuracy			0.38	0.18	0.37	0.21	**0.19**	0.35	0.20	**0.21**	0.32
Average Accuracy			0.25	0.14	0.19	0.19	0.14	0.17	0.17	0.16	0.16
Average Precision			0.32	0.22	0.22	0.25	0.18	0.19	0.26	0.19	0.18
Average Recall			0.25	0.14	0.19	0.19	0.14	0.17	0.17	0.16	0.16
kappa			0.32	0.15	0.30	0.18	0.14	0.27	0.17	0.16	0.24
Dimensions			9	9	10	28	28	28	28	28	28

In Table 5.10 the first three columns show the number of training and testing points for each class. We see that for this experiment the class training/testing sets slightly more balanced than Scenario E, however the testing data still varies from $5 - 140\%$ of the training data per class, ignoring LCZ C, this class is not represented in the Sao Paulo image, and LCZ 3 where there are almost 12 times as many samples in the testing set. The remaining columns illustrate the classification accuracy of the baseline and with the application of SEMA and SSNE. We see that for many classes the per-class accuracy is very low or even almost 0% accurate. Again, we note that the κ values for all cases demonstrate that our results are still better than the random assignment of class labels.

For the classification of Berlin, the application of SEMA does not improve classification OA compared to the baseline case for any view, however SSNE improves classification accuracy for the

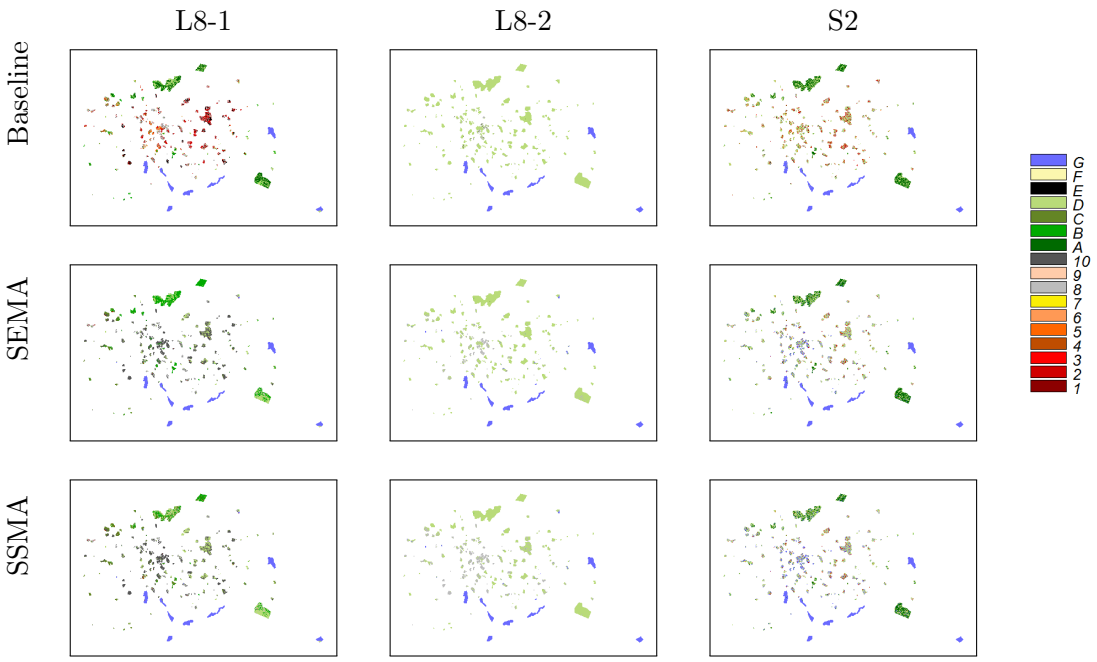

Figure 5.10: **Experiment 3, Sao Paulo:** Classification map of predicted classes using the baseline method, SEMA, and SSNE for each view.

L8-2 and S2 views by a margin of $2-3\%$ accuracy compared to the baseline. In Figure 5.9, we can visually see that the classification of Berlin improves slightly with the use of SEMA and SSNE. Specifically, near the center of the city there is improved separation for LCZ 1-6 for L8-2 and S2 when SEMA is applied and for L8-2 when SSNE is applied; near edges of the image the separation between LCZ A-D are more defined for L8-2 when SSNE is applied.

For the classification of Sao Paulo, the application of SEMA and SSNE only improves slightly. In Figure 5.10, we cannot see much improvement visually in the classification of Sao Paulo with the use of SEMA and SSNE. There seems to be fewer classes identified for all views when SEMA and SSNE are applied.

Now, we look at some specific results of testing these two classifiers. In Figure 5.11 the confusion matrices for the classification results for each view in each scenario are show. We see that in the classification of Berlin, the classification of L8-1 is much worse than of L8-2. We do see that in general, the classification of most pixels tends to be LCZ D. Looking at the classification result of Berlin in Table 5.9, we see that specifically LCZ 4 has 0% OA for L8-1 but 59% OA for L8-2. We see similar trends in Scenario F; for the classification of Sao Paulo the results for L8-2 are much worse than of L8-1 and pixels are commonly misclassified as LCZ D.

The classification of an unknown view is a difficult problem and manifold alignment only improves overall classification results by a small margin.

In Figure 5.14, we illustrate the class distributions for each city across the provided spectral

bands for each class. In Figures 5.15– 5.17, we illustrate the class distributions for each city across all classes for each spectral band for each of the three views. In these figures, we see that there is overlap of the distributions for most classes in each band and further. It is worth noting here, the number of class samples for each image is uneven. Further, the spectral signatures for most classes are not consistent between cities or even Landsat-8 views. The differences in spectral signatures between classes could be caused by temporal, atmospheric, and environmental factors in the imagery.

In Figures 5.12 and 5.13 the training and testing spectra for the two Landsat views for each class are shown. By examining these spectral distributions, we can see that there is significant variation in the spectral signatures for all classes; further, the training and testing spectra for each class are not consistent. It is clear from this analysis that the variation in spectra across class, within each class overshadows the differences between spectral responses for each class. This variation in spectra affected the Baseline classification negatively. When SEMA and SSNE are applied, some of the classification results improve. despite the aligning the L8-1 and L2-2 views this technique relies on the consistency of the spectral responses across each city, it is a poor assumption. A better way to apply manifold alignment to this task would be to treat each image (by image as well as by view) as an independent view that must be aligned.

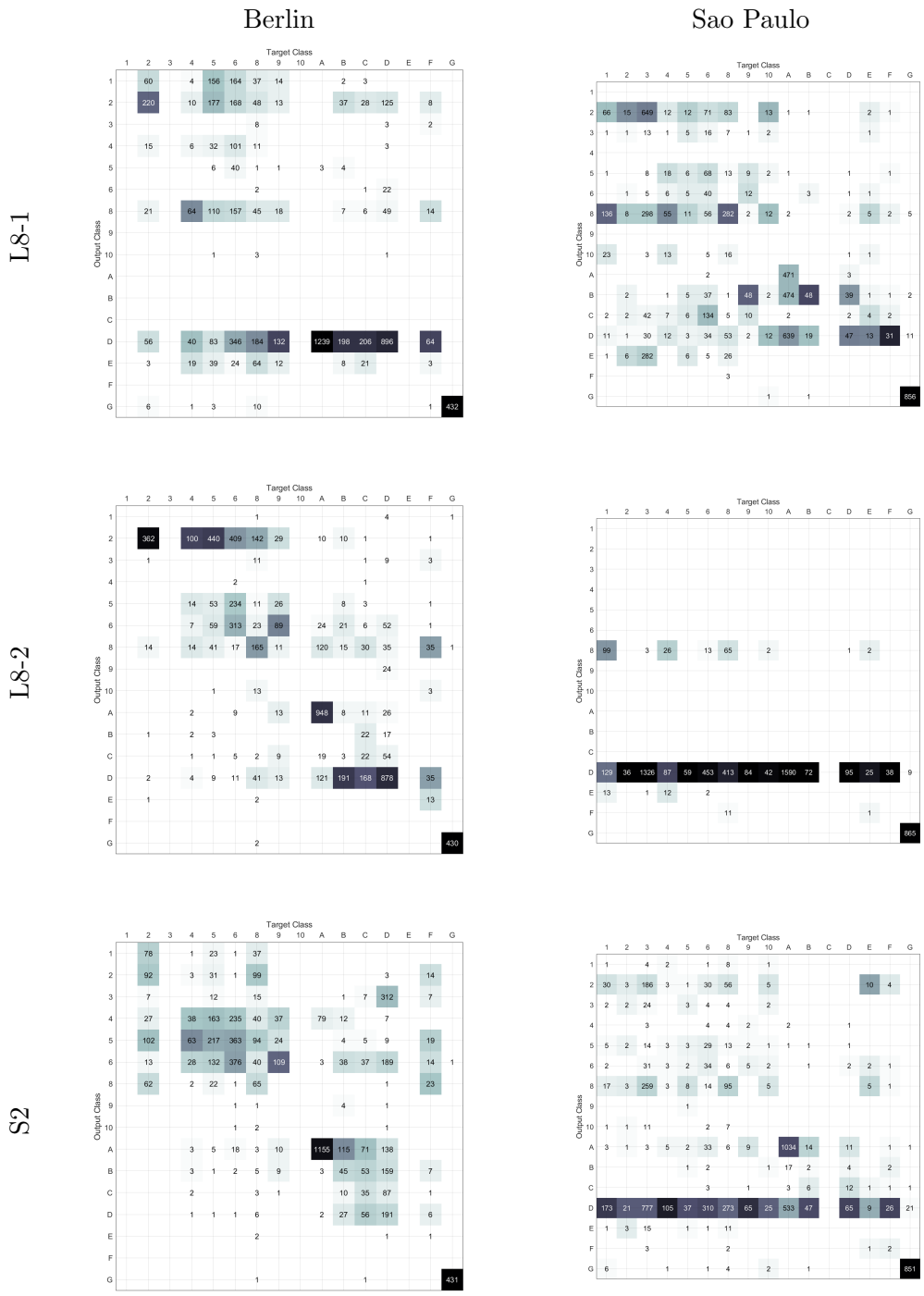

Figure 5.11: **Confusion matrices of the Baseline results:** We show the confusion matrices for both the Berlin and Sao Paulo classification for each view. The x-axis displays the target class; the y-axis displays the output class for all of the 16 classes present in the dataset. The lighter to darker shading represents a 0-1 classification rate of the target class for the output class. The number is the number of target class pixel classified as the output class.

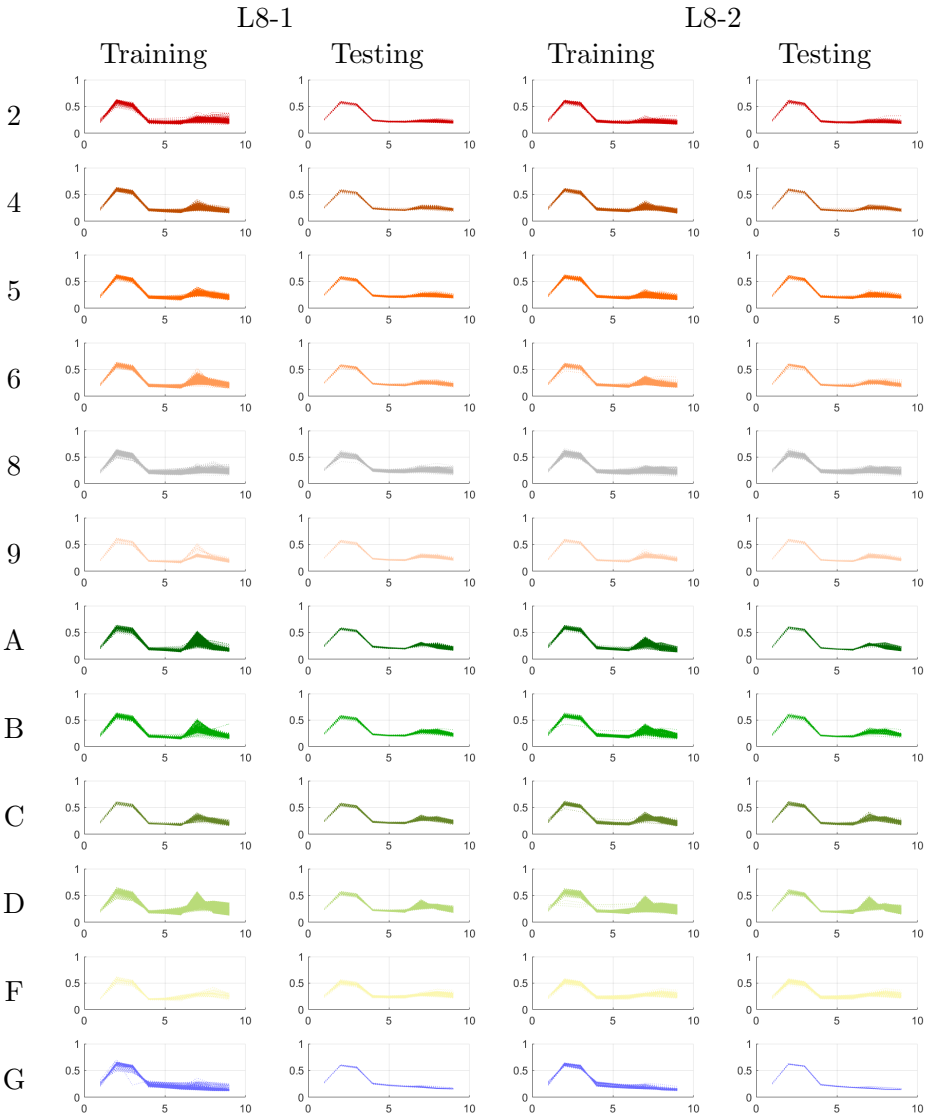

Figure 5.12: **Spectral Signatures of the training and testing sets for Berlin for L8-1 and L8-2:** Each plot illustrates the spectral distributions for each LCZ class that is in the testing city across the training of testing cities. The training set is the combined spectral from Hong Kong, Paris, Rome, and Sao Paulo. Across the x-axis, 1 corresponds to Band 1, 2 corresponds to Band 2, etc. We can see that there is slightly less variation in the spectra signatures between the training and testing sets for L8-2 than there is for L8-1. Specifically, when looking at LCZ A.

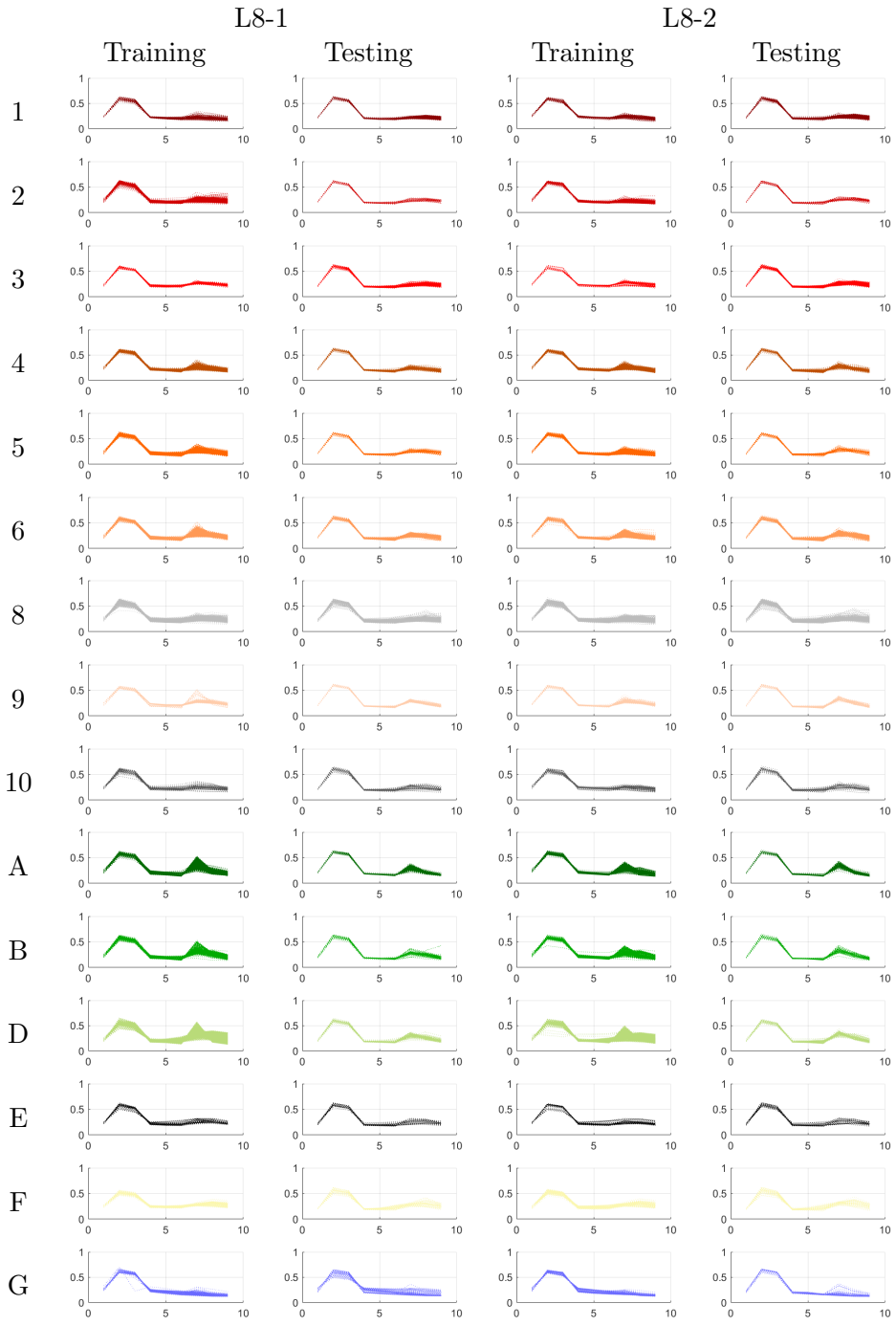

Figure 5.13: **Spectral Signatures of the training and testing sets for Sao Paulo for L8-1 and L8-2:** Each plot illustrates the spectral distributions for each LCZ class that is in the testing city across the training of testing cities. The training set is the combined spectral from Berlin, Hong Kong, Paris, and Rome. Across the x-axis, 1 corresponds to Band 1, 2 corresponds to Band 2, etc. We can see that there is slightly less variation in the spectra signatures between the training and testing sets for L8-1 than there is for L8-2.

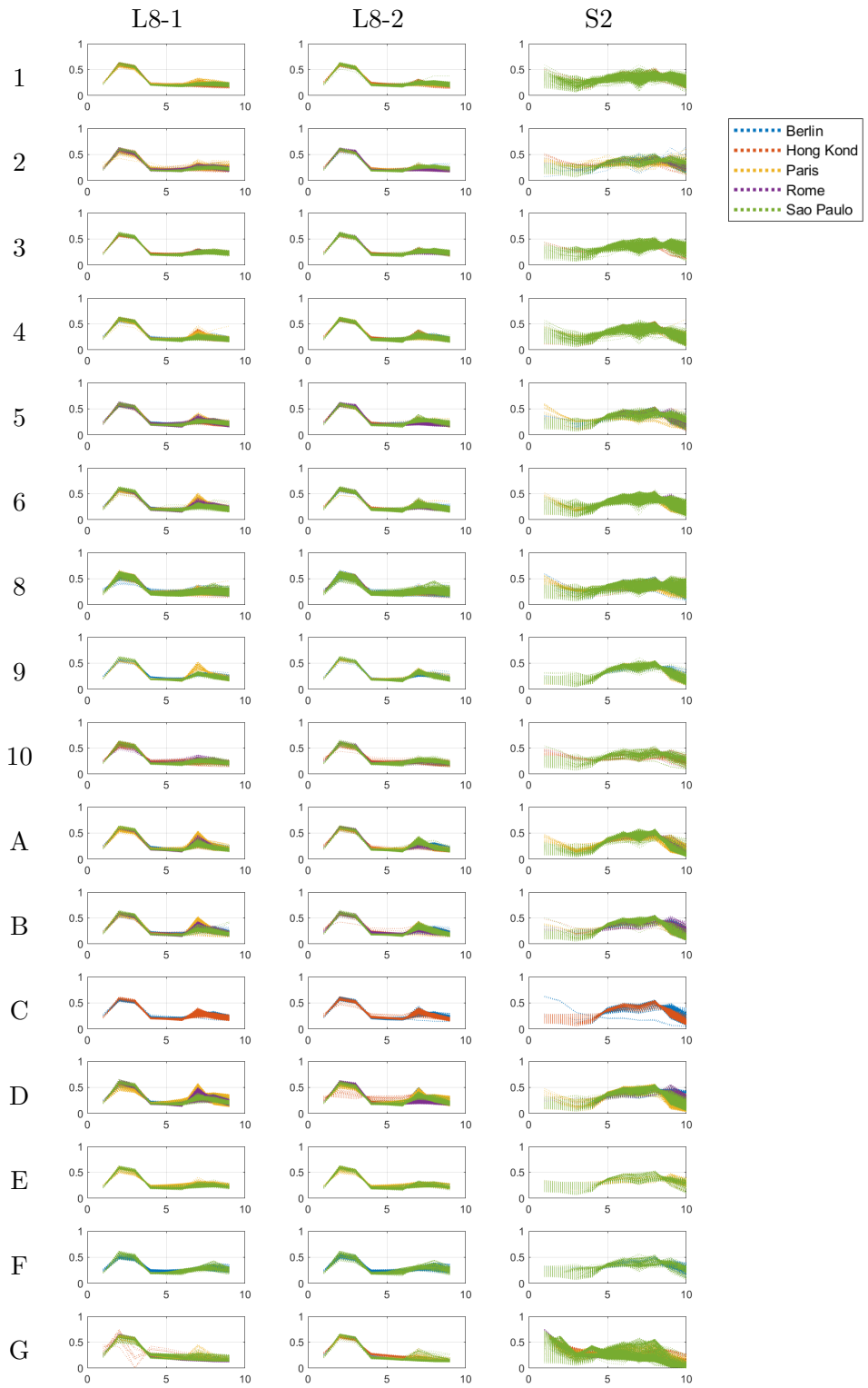

Figure 5.14: **Spectral Signatures for each view:** Each plot illustrates the spectral distributions for each LCZ class across the five cities. Across the x-axis, 1 corresponds to Band 1, 2 corresponds to Band 2, etc.

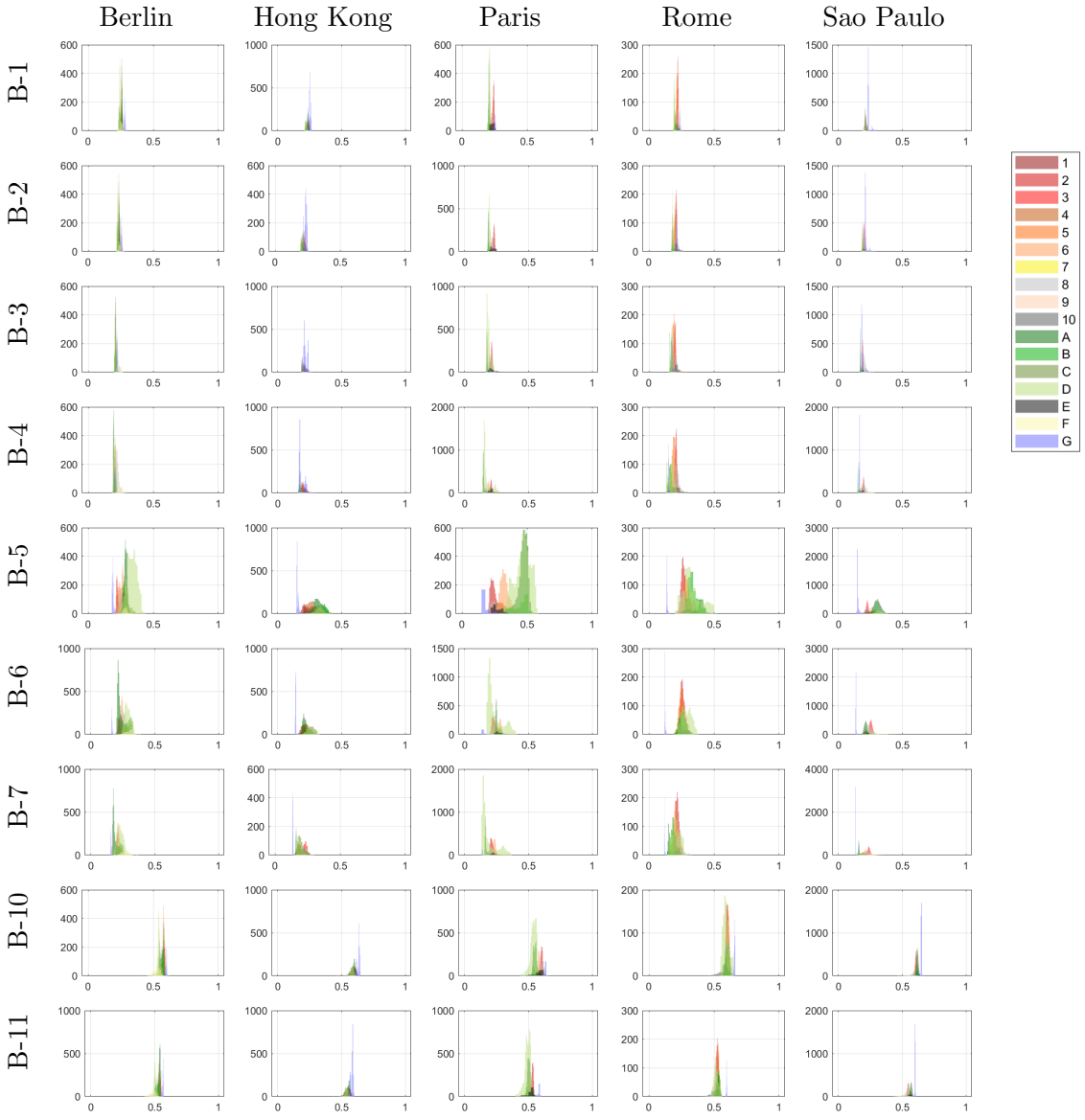

Figure 5.15: **Class Distributions for L8-1:** Each histogram plot illustrates the class distributions for each LCZ class. Across the columns the distribution for each city is shown. Across the rows, in the distribution for each band in the Landsat-8, view 1 images. The x-axis of each histogram shows the spectral value of each pixel, the images have been normalized to have values between 0 and 1. The y-axis represents the number of counts across a small range of spectral values using MATLAB's `histogram` function. The counts are not normalized across all plots.

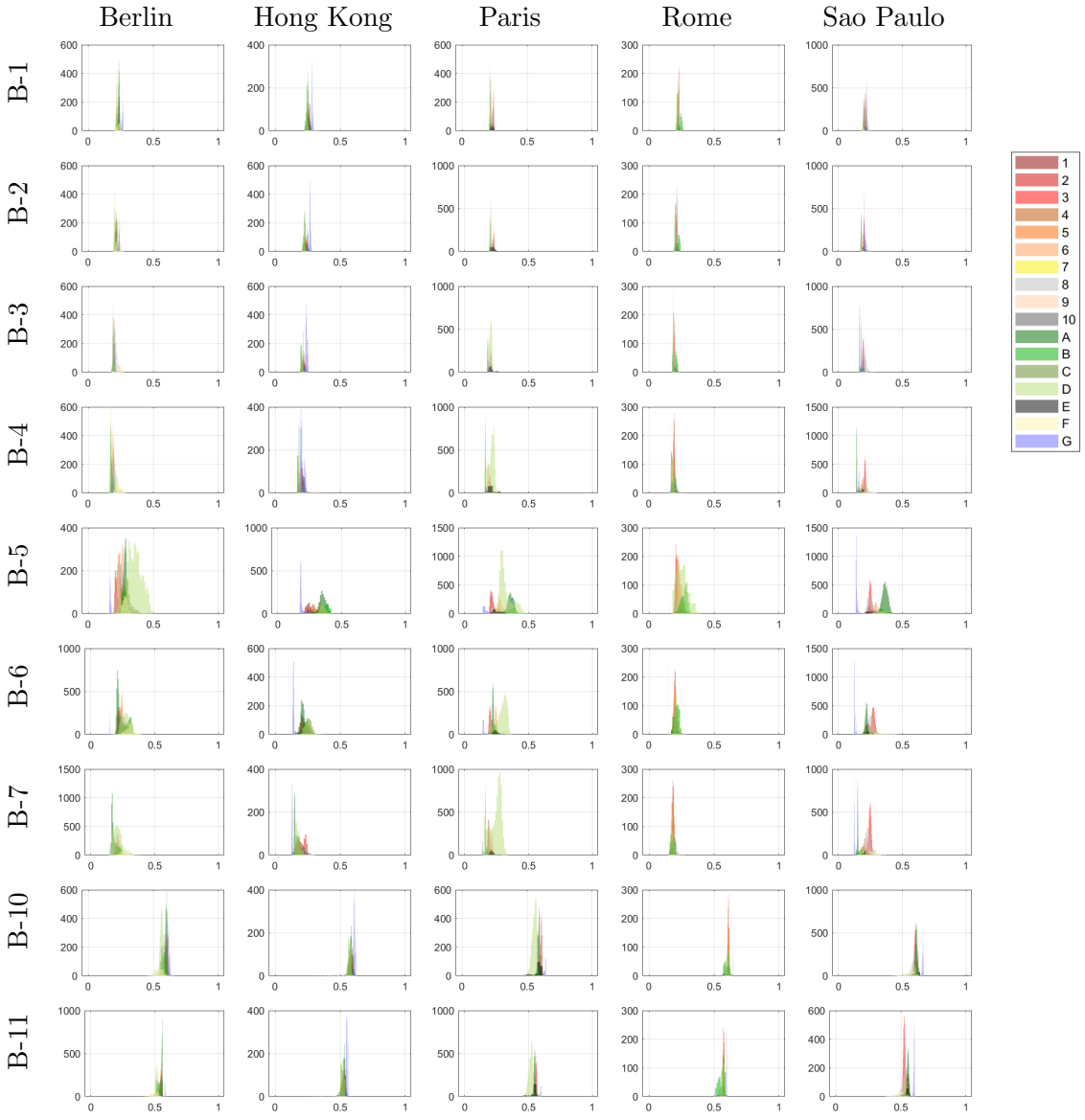

Figure 5.16: **Class Distributions for L8-2:** Each histogram plot illustrates the class distributions for each LCZ class. Across the columns the distribution for each city is shown. Across the rows, in the distribution for each band in the Landsat-8, view 2 images. The x-axis of each histogram shows the spectral value of each pixel, the images have been normalized to have values between 0 and 1. The y-axis represents the number of counts across a small range of spectral values using MATLAB's `histogram` function. The counts are not normalized across all plots.

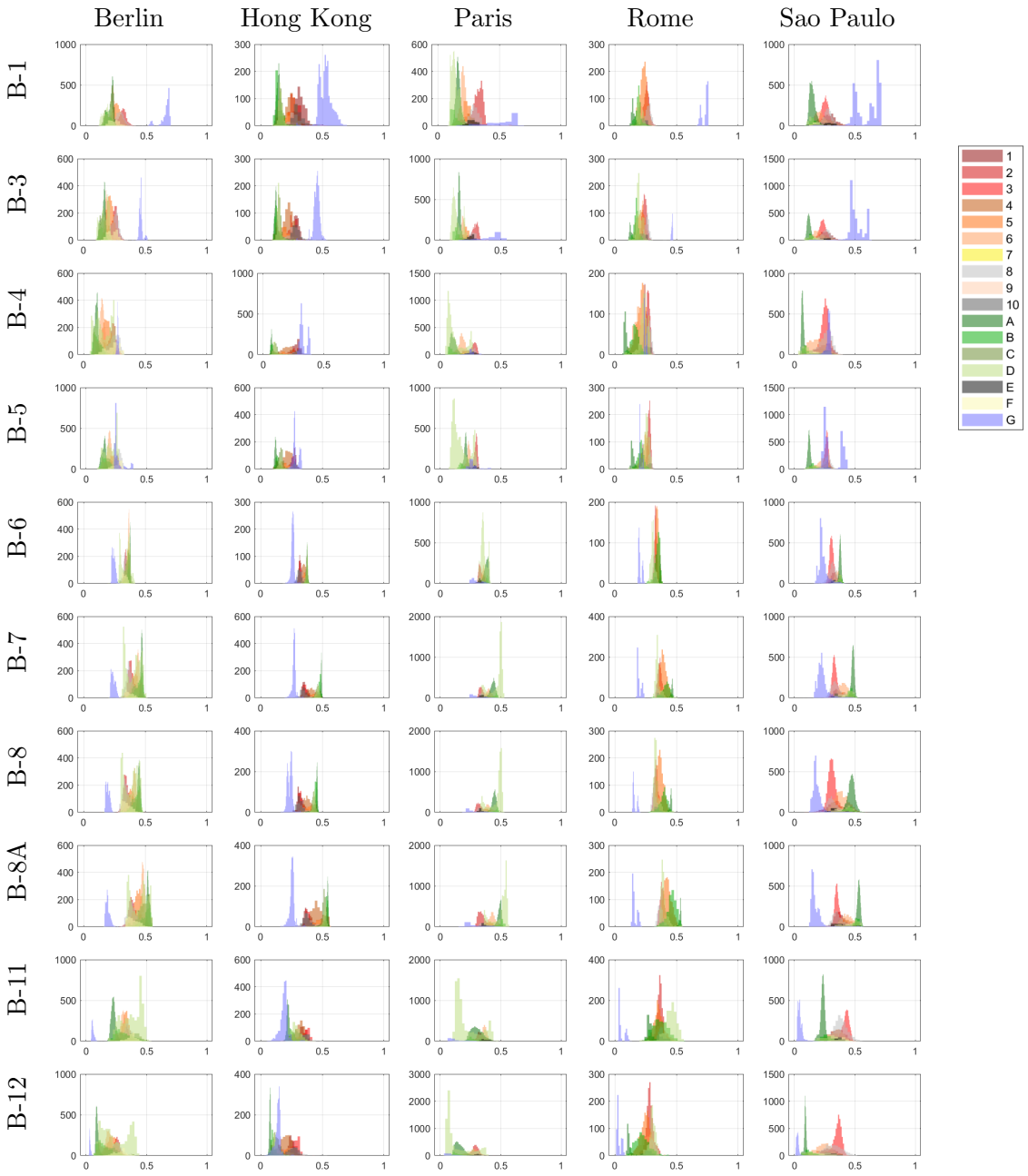

Figure 5.17: **Class Distributions for S2:** Each histogram plot illustrates the class distributions for each LCZ class. Across the columns the distribution for each city is shown. Across the rows, in the distribution for each band in the Sentinel-2 images. The x-axis of each histogram shows the spectral value of each pixel, the images have been normalized to have values between 0 and 1. The y-axis represents the number of counts across a small range of spectral values using MATLAB's `histogram` function. The counts are not normalized across all plots.

Chapter 6

Conclusion and Future Work

Although SSMA and SEMA are useful for fusing multiple view data for land use classification, they are only applicable when pairwise dissimilarities are provided by an expert user or are derived from pairs of known class labels. In this thesis, we introduced a new method, SSNE, that we conduct two sets of experiments on: classification of known views and classification of unknown views.

In Experiment 1, we seek to classify a known city where 50% of the data is used for training and 50% of the data is used for testing. We showed that by modifying the SSMA/SEMA objective functions to incorporate a normalization term, the multiple view data can be projected into a latent space even when there are no pairwise dissimilarities available. We showed that in land-use classification scenarios where pairwise dissimilarities are available, the resulting latent space embedding (SSNE) can be used to yield similar classification performance to the embeddings found by SSMA/SEMA. More importantly, however, is that in scenarios where the multiple views are spatially registered but no pairwise dissimilarities are available, SSNE can exploit the spatial alignment to yield similar classification results to situations where pairwise dissimilarities are available. Further, we demonstrated that manifold alignment is robust to the use of different classifiers.

In Experiment 2, we attempt to classify an unknown city when the classifier is trained on several other cities. We show that this is a hard problem that cannot necessarily be solved with only the application of manifold alignment. However, we do assume that the images for each city are spectrally aligned. Using a better alignment system or other image preprocessing techniques such as atmospheric calibration may yield a solution to this problem.

There are many possibilities for extension of this work; we discuss several here. First, with our proposed algorithm SSNE, we apply a normalization factor to SSMA/SEMA. We test the robustness of manifold alignment as preprocessing technique for three classifiers: an SVM, an LDA classifier, and a random forest classifier. However, we do not discuss the results of optimizing these classifiers or utilize more sophisticated classifiers such as Artificial Neural Networks or Deep Learning techniques. Deep learning could be applied for classification or used to develop an autoencoder for preforming dimensionality reduction; the latter would be the easier to apply to the currently training set due to both the number of ground truth samples alliable and that is could be directly compared to the use of SSNE.

Second, the Data Fusion contest provides several Open Street Maps as an auxiliary data set; this map data could be utilized as additional views for each city. This would demonstrate the

ability to combine image and non-image datasets using manifold alignment. In this application, the map data from Open Street Maps still has a spatial component that could be exploited.

Third, there is further investigation required regarding the selection of the weight parameters for SSNE. Overall, there are slight differences as we varied the parameters over a few values. Further, it would be worthwhile to explore how the optimal parameters for SSNE when applied to the contest data set could be generalized to other datasets.

Fourth, for the case of classify unknown views, it would be worthwhile to apply our experiment to the testing cities provided by the Data Fusion contest. This is a harder problem as we limit our dataset to the image pixels corresponding to the provided ground truth. We discuss in the results of Experiment 2 that we assume that the each view for each city is spectrally aligned, however this is a poor assumption; in further experiments it would perhaps be worthwhile to treat each city as an "independent view" that must be aligned. The application of other preprocessing techniques, such as atmospheric correction, to the image data in addition to manifold alignment would be of interest.

Bibliography

[1] 2017 IEEE GRSS Data Fusion Contest. `http://www.grss-ieee.org/community/technical-committees/data-fusion/data-fusion-contest`.

[2] D. K. Agrafiotis. Stochastic proximity embedding. *Journal of Computational Chemistry*, 24(10):1215–1221, 2003.

[3] F. Amato, J. Havel, A. Gad, and A. El-Zeiny. Remotely sensed soil data analysis using artificial neural networks: A case study of El-Fayoum depression, Egypt. *ISPRS International Journal of Geo-Information*, 4:677–696, 06 2015.

[4] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE Trans. Geoscience and Remote Sensing*, 43(3):441–454, March 2005.

[5] B. Bechtel, P. J. Alexander, J. Bhner, J. Ching, O. Conrad, J. Feddema, G. Mills, L. See, and I. Stewart. Mapping local climate zones for a worldwide database of the form and function of cities. *ISPRS International Journal of Geo-Information*, 4(1):199–219, 2015.

[6] M. Belgiu and L. Drgu. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[8] J. Benedetto, W. Czaja, J. Dobrosotskaya, T. Doster, K. Duke, and D. Gillis. Semi-supervised learning of heterogeneous data in remote sensing imagery. *Proc. SPIE*, 8401:840104–840104–12, 2012.

[9] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, June 1998.

[10] N. D. Cahill, W. Czaja, and D. W. Messinger. Schroedinger eigenmaps with nondiagonal potentials for spatial-spectral clustering of hyperspectral imagery. *Proc. SPIE*, 9088:908804–908804–13, 2014.

[11] N. D. Cahill and P. G. Immel. Semi-supervised normalized embeddings for land-use classification from multiple view data. *Proc. SPIE*, 10644, 2018.

[12] J. Chen, H. Fang, and Y. Saad. Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection. *J. Mach. Learn. Res.*, 10:1989–2012, December 2009.

[13] M. Chi, R. Feng, and L. Bruzzone. Classification of hyperspectral remote-sensing data with primal SVM for small-sized training dataset problem. *Advances in Space Research*, 41(11):1793 – 1799, 2008.

[14] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[15] A. Criminisi, J. Shotton, and E. Konukoglu. *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*, volume 7, pages 81–227. NOW Publishers, Foundations and Trends® in Computer Graphics and Vision: No 2-3, pp 81-227, number = 2-3, edition, January 2012.

[16] W. Czaja and M. Ehler. Schroedinger eigenmaps for the analysis of biomedical data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1274–1280, 2013.

[17] Q. Du. Modified Fisher's linear discriminant analysis for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(4):503–507, Oct 2007.

[18] M. Fauvel, J. Chanussot, and J. Benediktsson. Kernel principal component analysis for the classification of hyperspectral remote sensing data of urban areas. *EURASIP Journal on Advances in Signal Processing*, 2009(783194):1–14, 2009.

[19] J. Geletic and M. Lehnert. GIS-based delineation of local climate zones: the case of medium-sized central European cities. *Moravian Geographical Reports*, 24(3), 2016.

[20] D. B. Gillis and J. H. Bowles. Hyperspectral image segmentation using spatial-spectral graphs. volume 8390, pages 83901Q–83901Q–11, 2012.

[21] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294 – 300, 2006. Pattern Recognition in Remote Sensing (PRRS 2004).

[22] A. Halevy. *Extensions of Laplacian Eigenmaps for Manifold Learning*. PhD thesis, University of Maryland, College Park, 2011.

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

[24] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins. Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(1):49–57, Feb 2010.

[25] J. Harsanyi and C. Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):779–785, July 1994.

[26] J. E. Johnson. *Schroedinger Eigenmaps for Manifold Alignment of Multimodal Hyperspectral Images*. PhD thesis, Rochester Institute of Technology, 2016.

[27] J. E. Johnson, C. M. Bachmann, and N. D. Cahill. Manifold alignment with Schroedinger eigenmaps. *Proc. SPIE*, 9840:98401K–98401K–11, 2016.

[28] D.H. Kim and L.H. Finkel. Hyperspectral image processing using locally linear embedding. pages 316–319, March 2003.

[29] A. D. Kulkarni and B. Lowe. Random forest algorithm for land cover classification. *International Journal on Recent and Innovative Trends in Computing and Communication (IJRITCC)*, 4(3):58–63, 2016.

[30] D. Lunga and O. Ersoy. Spherical stochastic neighbor embedding of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):857–871, Feb 2013.

[31] U. Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[32] N. A. Mahmon, N. Ya'acob, and A. L. Yusof. Differences of image classification techniques for land use and land cover classification. In *2015 IEEE 11th International Colloquium on Signal Processing Its Applications (CSPA)*, pages 90–94, March 2015.

[33] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, Aug 2004.

[34] G. Mountrakis, J. Im, and C. Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247 – 259, 2011.

[35] G. Mountrakis, J. Im, and C. Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259, 2011.

[36] NASA. remote sensing and lasers. `https://www.nasa.gov/centers/langley/news/factsheets/RemoteSensing.html`, March 1998. FS-1998-03-35-LaRC.

[37] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[38] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[39] OpenStreetMap contributors. Planet dump retrieved from `https://planet.osm.org` . `https://www.openstreetmap.org`, 2017.

[40] M. Pal. Random forests for land cover classification. *Geosci. Remote Sens. Symp.*, 3516:3510 – 3512 vol.6, 08 2003.

[41] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[42] Y. Pei, F. Huang, F. Shi, and H. Zha. Unsupervised image matching based on manifold alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1658–1664, Aug 2012.

[43] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, et al. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, 113:S110–S122, 2009.

[44] G. M. Senseman, C. F. Bagley, and S. A. Tweddale. Accuracy assessment of the discrete classification of remotely-sensed digital data for landcover mapping. In *USACERL Technical Report EN-95/04*, pages 1–27, April 1995.

[45] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737, Jun 1997.

[46] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[47] I. D. Stewart and T. R. Oke. Local climate zones for urban temperature studies. *Bulletin of the American Meteorological Society*, 93:1879–1900, 2012.

[48] T. Janos, F. Tunde. Geoinformatics. `https://www.tankonyvtar.hu/en/tartalom/tamop425/0032_terinformatika/ch04s04.html`, 2008. Section 4.3 Hyperspectral.

[49] D. Tuia and G. Camps-Valls. Kernel manifold alignment for domain adaptation. *PloS one*, 11(2):e0148655, 2016.

[50] 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*, 52(12):7708–7720, Dec 2014.

[51] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[52] T. Wang, C. Tan, L. Chen, and Y. Tsai. Applying artificial neural networks and remote sensing to estimate chlorophyll-a concentration in water body. In *2008 Second International Symposium on Intelligent Information Technology Application*, volume 1, pages 540–544, Dec 2008.

[53] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 313–319 vol.1, Oct 2003.

[54] T. Zhang, J. Yang, D. Zhao, and X. Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7):1547–1553, 2007.

[55] Z. Zhang and M. I. Jordan. Multiway spectral clustering: A margin-based perspective. *Statistical Science*, 23(3):383–403, 2008.

[56] G. Zhu and D. G. Blumberg. Classification using ASTER data and SVM algorithms; the case study of Beer Sheva, Israel. *Remote Sensing of Environment*, 80(2):233 – 240, 2002.