Claremont Colleges
Scholarship @ Claremont

HMC Senior Theses | HMC Student Scholarship

2016

Pattern Recognition in High-Dimensional Data

Matthew Dannenberg
Harvey Mudd College

Recommended Citation
Dannenberg, Matthew, "Pattern Recognition in High-Dimensional Data" (2016). *HMC Senior Theses*. 76.
https://scholarship.claremont.edu/hmc_theses/76

This Open Access Senior Thesis is brought to you for free and open access by the HMC Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in HMC Senior Theses by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.

Pattern Recognition in High-Dimensional Data

Matthew Dannenberg

Weiqing Gu, Advisor

Satyan Devadoss, Reader

Department of Mathematics

May, 2016

Copyright © 2016 Matthew Dannenberg.

The author grants Harvey Mudd College and the Claremont Colleges Library the nonexclusive right to make this work available for noncommercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of the author. To disseminate otherwise or to republish requires written permission from the author.

Abstract

Vast amounts of data are produced all the time. Yet this data does not easily equate to useful information: extracting information from large amounts of high dimensional data is nontrivial. People are simply drowning in data. A recent and growing source of high-dimensional data is hyperspectral imaging. Hyperspectral images allow for massive amounts of spectral information to be contained in a single image. In this thesis, a robust supervised machine learning algorithm is developed to efficiently perform binary object classification on hyperspectral image data by making use of the geometry of Grassmann manifolds. This algorithm can consistently distinguish between a large range of even very similar materials, returning very accurate classification results with very little training data. When distinguishing between dissimilar locations like crop fields and forests, this algorithm consistently classifies more than 95 percent of points correctly. On more similar materials, more than 80 percent of points are classified correctly. This algorithm will allow for very accurate information to be extracted from these large and complicated hyperspectral images.

Contents

List of Figures

Acknowledgments

I would like to thank my parents for their support during the process of writing this thesis.

I would like to thank Sarah Lichtman, for her support and for her continuous stream of useful MATLAB advice, without which this thesis's algorithm could never have been tested.

Over and above all else, I would like to thank my advisor, Professor Weiqing Gu, without whose guidance this thesis would not have been possible.

Chapter 1

Introduction

From traffic footage to internet connections to satellite scans and more, the modern world generates massive quantities of data all of the time. Just the storage of data on such a scale is an extremely difficult endeavor, let alone the processing. Yet eventually this data must be processed in order to extract useful information, or it serves no good to anyone. However, many large datasets also depend on a huge number of variables, or are composed of a large number of pixels. In such sets of data, a single point may be represented as an element of a very high-dimensional vector space. This thesis will attempt to develop new techniques to efficiently process high-dimensional data.

In particular, this thesis focuses on object classification on hyperspectral images. Hyperspectral images promise to encode massive amounts of information about a spatial region into a very convenient representation. These images merge spectral, and geometric information into a coherent picture which is so far relatively unexplored by mathematicians. Moreover, there is a high level of demand for efficient processing algorithms for hyperspectral images. These images have applications in spectroscopy, food processing, geology, counterterrorist operations, and environmental analysis. However, simple processing algorithms are not always sufficient to make good use of the information contained within a hyperspectral image. This thesis has produced an algorithm for object classification on static hyperspectral images, with the hope that future work may expand upon it to develop a real time algorithm for object classification on dynamic hyperspectral images, a largely unexplored topic without effective algorithms.

Chapter 2

Relevant Background

2.1 The Grassmann Manifold

The Grassmann Manifold, or Grassmannian, $Gr(k, n)$ is a compact smooth manifold consisting of all k-dimensional subspaces of $\mathbb{R}^n$. Its properties make it a natural setting for efficient subspace tracking and dimensional reduction algorithms. In particular, optimization on a Grassmann manifold will allow for problems requiring the choice of an optimal subspace to be solved in a manner independent of the representation of that subspace. In this section, a representation of this manifold and some of the geometric properties which will be useful in the development and description of numerical methods will be developed.

2.1.1 The Quotient Representation

While the Grassmannian is not itself a Lie group, it can be represented as a quotient of Lie groups, particularly orthogonal groups. This gives Grassmannians a natural structure as homogenous spaces.

Definition 2.1. *The orthogonal group $O(n)$ is the group of all real orthogonal matrices A under matrix multiplication. Written formally this is*

$$O(n) = \left\{A \in GL_n(\mathbb{R}) : AA^T = I\right\}.$$

It can also be thought of as the set of all ordered orthonormal bases of $\mathbb{R}^n$, where each orthogonal matrix is constructed simply by appending the basis vectors into a matrix. Most significantly, $O(n)$ is a compact Lie group, which is to say that it is a compact manifold with a group structure

where the group operation is a differentiable map from $O(n)\times O(n)$ to $O(n)$. Knowing that $O(n)$ is a smooth manifold, we are first motivated to search for the tangent space to $O(n)$, as any quotient of orthogonal groups will inherit some subspace of the tangent space.

Proposition 2.1. *For $Y \in O(n)$, the tangent space to Y is given by all matrices $A \in GL(n, \mathbb{R})$ such that $Y^T A$ is skew-symmetric.*

Proof. This can be seen immediately by taking a differentiable path $\alpha : (-\epsilon, \epsilon) \to O(n)$ with $\alpha(0) = Y$ and $\alpha'(0) = A$, noting that $Y^T Y = I$, and taking a derivative at 0. □

As the dimension of a Lie group is equal to the dimension of its tangent space at the origin, this further implies that $O(n)$ is $n(n-1)/2$ dimensional. Knowing the tangent space of $O(n)$ is sufficient to allow for the creation of a natural notion of length based upon the inner product on the tangent space, which can again be inherited by the quotient group. For two tangent vectors A and B to $O(n)$ at Y, a natural inner product is

$$\langle A, B \rangle = \mathrm{tr} A^T B.$$

For $A = B$, this induces the Frobenius norm $\| \cdot \|_F$ and a length on $O(n)$.

Definition 2.2. *If $\gamma : [a, b] \to O(n)$ is a differentiable map, then the length of γ is*

$$\int_a^b \|\gamma'(x)\|_F \, dx.$$

At this point, we may define the Grassmann manifold as a quotient of orthogonal groups.

Definition 2.3. *The Grassmann manifold $Gr(k, n)$ is the quotient manifold given by*

$$Gr(k, n) \cong O(n)/(O(k) \times O(n-k)).$$

In particular, defining the Grassmann manifold in this manner implies a convenient manner of representing elements of it as equivalence classes of matrices. In particular, an element may be represented as an equivalence class of "tall" $n \times p$ matrices with orthonormal columns, invariant under right-multiplication by any element of $O(k)$. This representation of the Grassmann manifold will later in this thesis be useful for achieving maps between arbitrary matrices and elements of a Grassmann manifold. To more concretely grasp a point on $Gr(k, n)$, consider an example.

Example 2.1. *The two matrices*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \textit{and} \qquad \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{\sqrt{2}} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

are different representatives of the same equivalence class in $Gr(3,5)$.

With some effort, this representation allows for a simple expression for the tangent space to the Grassmann manifold, noting that the Frobenius norm on the tangent space is inherited from the orthogonal group.

Proposition 2.2. *Edelman et al.* (1998)
The tangent space to $Gr(k,n) \cong O(n)/(O(k)\times O(n-k))$ *at* $Y \in Gr(k,n)$ *consists of all matrices of the form*

$$Y \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} C & 0 \\ 0 & D \end{pmatrix}$$

for A *an* $(n-p)\times p$ *matrix,* $C \in O(p)$*, and* $D \in O(n-p)$.

It is not immediately clear that this new definition of the Grassmann manifold is consistent with the first given definition, that the Grassmann manifold consists of all k-dimensional linear subspaces of $\mathbb{R}^n$. To see this, recall that the orthogonal group may be represented as all ordered orthonormal bases of $\mathbb{R}^n$. Then $O(n)/O(n-k)$, the so-called **Stiefel manifold**, may be represented as all ordered orthogonal k-frames on $\mathbb{R}^n$. Any k-frame specifies a particular subspace of $\mathbb{R}^n$, though a single subspace may be represented by multiple elements of the Stiefel manifold. This is addressed in $(O(n)/O(n-k))/O(k)$, as the quotient by $O(k)$ identifies all distinct k-frames which specify the same subspace of $\mathbb{R}^n$.

2.1.2 Distance on the Grassmann Manifold

In the previous section, a Riemannian metric on Grassmannians was introduced using the structure inherited from the orthogonal group. However, there are many distinct notions of distance on Grassmann manifolds, and many optimization problems do not indicate clearly which should be used.

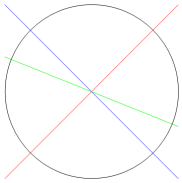

Figure 2.1 Each point in $Gr(1,2)$ represents a line. Identifying each line with the two points on the unit circle it intersects, a very natural notion of distance may be defined as the minimum angle between two pairs of these intersection points.

As such, other notions of distance may prove more natural or more efficient in certain applications. In addition, calculating the distance between two points on a Grassmann manifold using the equation in Definition 2.2 is cumbersome and computationally intensive. This section will introduce several more geometrically motivated notions of distance.

To motivate a metric on a Grassmann manifold, one must obtain a geometric notion of distance between two subspaces. Clearly neither the minimum or maximum distance between points on the subspaces are sufficient, as all subspaces intersect and distinct subspaces will always have points arbitrarily far apart. As a simple example observe the manifold $Gr(1,2)$ consisting of lines through the origin in $\mathbb{R}^2$, shown in Figure 2.1.

This suggests that distance between subspaces may be defined using the angles between them. Of course, in higher dimensions the angle between two subspaces is not a well-defined notion. To rigorize this notion, principal angles are introduced.

Definition 2.4. *Chepushtanova* (2015)
Let U and V be two elements of $Gr(k,n)$, represented as orthonormal $n \times k$ matrices. The principal angles $\theta_1, \ldots, \theta_k$ are recursively defined in increasing order by

$$\begin{aligned} \theta_i = \underset{u_i \in span(U),\ v_i \in span(V),\ \|u_i\|=\|v_i\|=1}{minimize} \quad & \arccos u_i^T v_i \\ subject\ to \quad & u_i^T u_j = v_i^T v_j = 0,\ j = 1, \ldots, i-1. \end{aligned} \tag{2.1}$$

Let $\theta \in \mathbb{R}^k$ denote the vector of principal angles. A common statistical quantity, the canonical correlation of the two subspaces, is simply the vector of the cosines of principal angles. While the optimization-based definition of principal angles is mathematically clear, a more computationally efficient calculation or these quantities is given by the following algorithm.

Principal Angle Computation Chepushtanova (2015)

1. Let $U, V \in Gr(k, n)$ be represented by orthonormal matrices.
2. Define $(Y, \Sigma, Z) = \text{svd}(U^T V)$.
3. Then $\theta = \arccos(\text{diag}(\Sigma))$.

While many distinct metrics and pseudometrics may be defined based on principal angles, three have been chosen for testing in this paper.

Definition 2.5. *Let $U, V \in Gr(k, n)$.*
The geodesic metric is defined by $d_g(U, V) = \|\theta\|$.
The chordal metric is defined by $d_c(U, V) = \|\sin\theta\|$.
The ℓ-smallest pseudometric is defined by $d_\ell(U, V) = \sqrt{\sum_{i=1}^{\ell} \theta_i^2}$.

2.1.3 Geodesics and Parallel Transport

On a Euclidean space, the idea of moving in a straight line is a fairly simple notion - maintain a constant velocity. Intuitively, extending this notion to a manifold should suggest that the tangent vector along a "straight" path should remain constant. Yet this is not possible, as the tangent space of a manifold varies from point to point, making it more complicated to maintain a consistent notions of parallelism. To recapture this concept of a straight line, one may introduce the parallel transport of a vector tangent to the manifold. As Grassmann manifolds are Riemannian, given a Riemannian metric on $Gr(k, n)$ there exists a unique symmetric Riemannian connection, immediately defining a covariant derivative. Then the parallel transport of a tangent vector along a path is defined as a vector field restricted to the path such that the covariant derivative is identically 0. A path is said to be a geodesic if its tangent vectors are parallel everywhere along it.

In this section, the Riemannian metric induced from the orthogonal group is used to define geodesics and parallel transport on a Grassmann manifold; other metrics will result in slightly different definitions of these concepts. On the Grassmann manifold $Gr(k, n)$, there exist methods running in $O(nk^2)$ steps by which both geodesic curves and parallel transports of tangent vectors along geodesics may be determined. Quickly determining these is essential to many numerical methods on the Grassmann manifold.

Proposition 2.3. *Edelman et al.* (1998)
The Grassmann manifold geodesic $Y(t)$ with $Y(0) = Y \in Gr(k, n)$ and $Y'(0) = A$ is given by

$$Y(t) = \begin{pmatrix} YV & U \end{pmatrix} \begin{pmatrix} \cos \Sigma t \\ \sin \Sigma t \end{pmatrix} V^T \tag{2.2}$$

where $U\Sigma V^T$ is the compact singular value decomposition of A. The distance as specified in Definition 2.2 *along a geodesic between two points is the geodesic distance.*

Proposition 2.4. *Edelman et al.* (1998)
Let A and B be tangent vectors to the Grassmann manifold at Y. The parallel transport of B along the geodesic starting at Y with initial direction $Y'(0) = A$ is

$$\tau B(t) = \left(\begin{pmatrix} YV & U \end{pmatrix} \begin{pmatrix} -\sin \Sigma t \\ \cos \Sigma t \end{pmatrix} U^T + (I - UU^T) \right) B \tag{2.3}$$

where $U\Sigma V^T$ is the compact singular value decomposition of A.

These results are particularly powerful in that after the singular value decomposition has been calculated, all remaining operations tend to amount to matrix multiplication, which is both easily parallelizable and fairly simply made very efficient.

2.2 Hyperspectral Imaging

A hyperspectral image intuitively functions much like human eyes do. Within human eyes, specialized cells capture incident light in three separate bands - red, green, and blue. This raw information is transferred via an optical nerve to the brain, where it is efficiently processed and converted into a single colored image. Similarly, in a hyperspectral image, one takes a desired frequency range and divides it into a large number of narrow frequency bands. A complex system of optics is placed in a hyperspectral camera in order to be able to properly distinguish between incident photons in each separate band. At every pixel of the image, intensity data for each band is collected. While of course this data can then be assembled into a single image to mirror the human brain process, leaving the bands separate yields a far more interesting set of data. With many hyperspectral images containing more than one-hundred frequency bands, such images essentially contain hundreds of full frequency spectra, one at every pixel of the

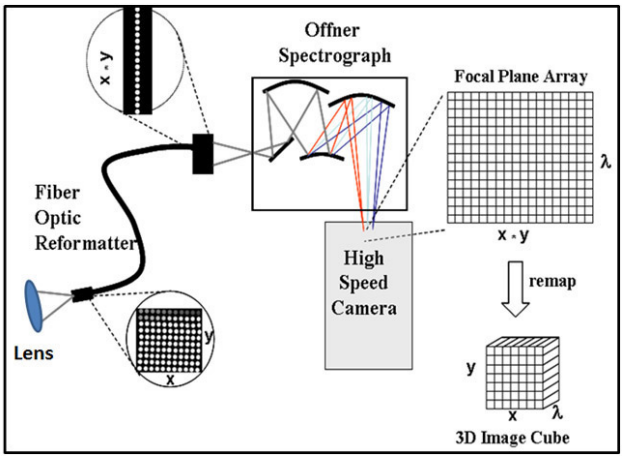

Figure 2.2 One possible construction of a hyperspectral camera. While the precise optical arrangement varies, virtually all hyperspectral cameras use very similar types of designs to this one. (a)

image. A clever analysis of such an image may thus take into account both spectroscopic and spatial information. Hyperspectral videos, or dynamic hyperspectral images, also allow for temporal information to be analyzed.

2.2.1 Hyperspectral Camera Construction

The actual construction of a hyperspectral camera is a complicated affair, generally involving the inclusion of complicated and expensive optical arrays as in Figure 2.2 so as to be able to properly separate out the different wavelength bands required with sufficient accuracy and loss of signal strength. However, some progress has been made in developing more inexpensive hyperspectral cameras, such as in Habel et al. (2012). While the technology for hyperspectral images has existed for some time, devices capable of capturing dynamic hyperspectral images in real time are a very new development. This is both due to the difficulty of capturing sufficient statistics of frequency data in a short time window for a frame of a video and due to the difficulty of transferring and creating such large amounts of data as fast as it is being created. A promising architecture for the acquisition of hyperspectral video is demonstrated in Wang et al. (2015). However, modern hyperspectral video cameras are prohibitively expensive and thus not accessible to most civilians. This paper focuses on the analysis of stan-

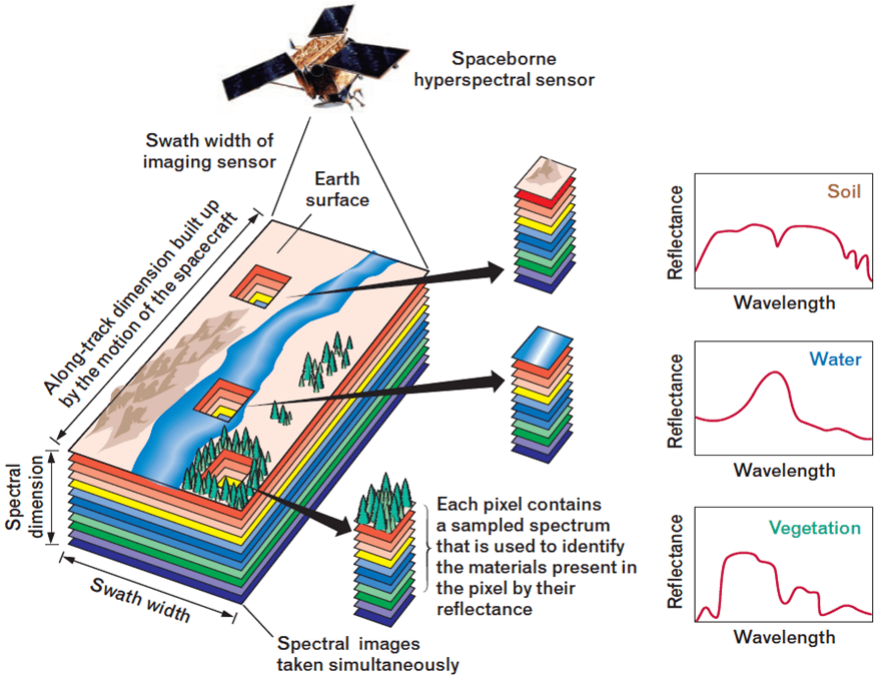

Figure 2.3 Satellite-mounted hyperspectral cameras are capable of searching huge amounts of terrain for desirable materials. Shaw and Burke (2003)

dard hyperspectral imaging due to the lack of freely available hyperspectral video.

2.2.2 Applications of Hyperspectral Imaging

Due to the sheer amount of data contained in both dynamic and static hyperspectral images, they are useful for a large number of tasks. The two which are of the most interest to this thesis are remote sensing for object detection and subsequent classification as well as dynamic object tracking. The applicability of hyperspectral imaging to these tasks is clear - one has all of the information necessary to detect, classify, and track objects quite easily with the combination of frequency and spatial data which hyperspectral cameras provide.

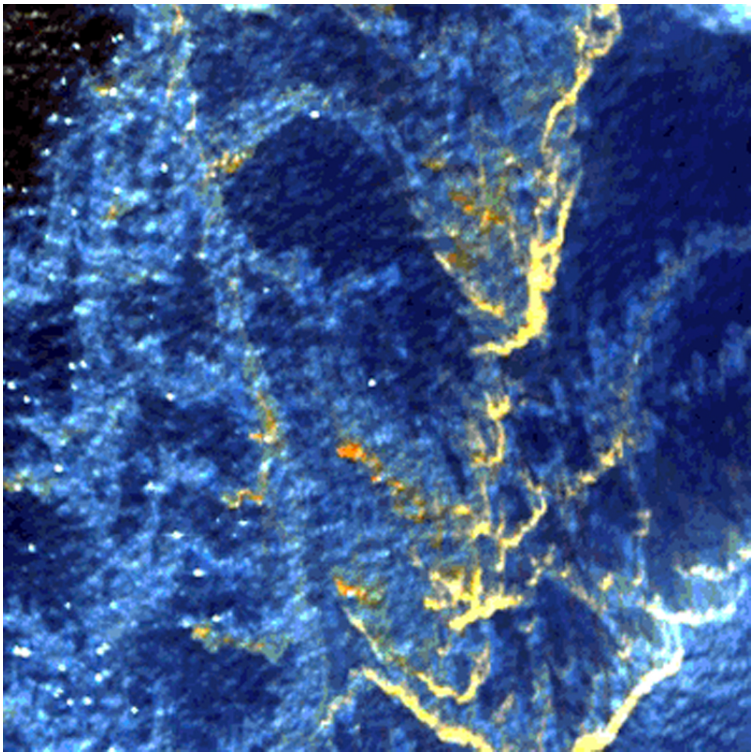

Figure 2.4 Hyperspectral object classification can be used to keep track of dangerous toxins, like oil spills in large bodies of water. (b)

Remote Sensing

A broad overview of the uses for hyperspectral imaging in remote sensing may be found in Shaw and Burke (2003). For such an application, consider a hyperspectral camera mounted on a satellite or aircraft. As the craft progresses, the hyperspectral camera may capture images of wide swaths of terrain passing below as shown in Figure 2.3. However, the data need not be time dependent provided the vehicle is moving sufficiently quickly, although multiple images can be taken at different times. Naively, the data corresponding to a point on the ground provides detailed knowledge of the precise materials present at that spot, potentially giving detailed information relevant for geological, botanical, or national security purposes such as in Figure 2.4. However, among other confounding effects, atmospheric noise and scattering effects can have tremendous impacts on the data, a problem analyzed in Griffin and Burke (2003). The strong relationship between hyperspectral device behavior and atmospheric effects is explored and a predictive model forecasting hyperspectral camera system performance developed in Kerekes and Baum (2003). Moreover, successful automatic target detection algorithms have been developed in Manolakis et al. (2003). However, the techniques there developed are strongly influenced by the specific hardware details of particular satellite-mounted hyperspectral devices and do not rely on Grassmann manifold techniques. Hence, it is strongly suspected that more efficient and more general methods may be developed so as to allow for more streamlined and more effective classification. This task is the primary focus of this paper.

Dynamic Object Tracking

Unlike remote sensing, dynamic object tracking is a purely time-dependent operation. In such a task, imagine trying to keep track of a truck known to be loaded with dangerous material. An autonomous tracking algorithm could allow a device like a drone to follow the truck, potentially saving lives. Significant progress has been made on the problem of object tracking with standard videos, such as with algorithms like He et al. (2014). However, the sheer amount of data present in dynamic hyperspectral images renders many such algorithms unable to cope. There is also an expected increase in reward, though. While object tracking in standard video is powerful, hyperspectral video offers many new channels over which tracking can occur. While many gases do not emit visible light in standard conditions,

they are likely more receptive to some lower frequency bands, allowing for them to be tracked as well. Moreover, objects which may initially appear identical on a standard video can be distinguished in a hyperspectral video. Imagine that when tracking a truck, the truck passes under a bridge. Then, a second truck not carrying any material but otherwise identical leaves the bridge with the original staying behind. Some wavelengths of light may be particularly well suited to passing through the truck's exterior, allowing for a hyperspectral image to distinguish between an empty and non-empty truck. Some progress has been made with the tracking of gases using tools like persistent homology as in Chepushtanova (2015) or very simplistic dimensional reduction methods like principal component analysis as in Gerhart et al. (2013).

Chapter 3

Literature Review

While the analysis of hyperspectral images with Grassmannian techniques is still largely unexplored, many related algorithms do exist. In this section, many of these algorithms are detailed. Several of these methods are used in the object classification algorithm presented in this paper. Other methods are included in this section in order to create a database of useful Grassmannian methods and dimensional reduction techniques for use in further investigations.

3.1 Function Optimization on the Grassmann Manifold

There are many techniques for dimension reduction using Grassmann manifolds. One straightforward approach is to find the optimal reduced-dimension subspace via an optimization problem on a Grassmann manifold. Some highly efficient algorithms to perform such optimization already exist, a few of which are detailed here.

3.1.1 Conjugate Gradient Descent

Conjugate gradient descent methods are a class of iterative methods designed to find the minimum of a function. In a conjugate gradient algorithm, a step down is iteratively taken along a direction specified by a sum of the negative gradient at a point and the direction of the previous step as illustrated by the second conjugate gradient step in Figure 3.1. In effect, an inner product is locally defined for the optimization problem at hand and

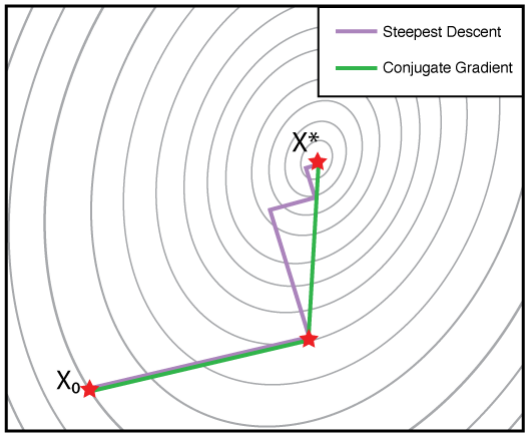

Figure 3.1 Conjugate gradient methods can be seen to converge far more rapidly than standard gradient descent methods. Cockett

a Gram-Schmidt orthogonalization procedure is iteratively performed and subtracted off. These algorithms are of particular use as they converge superlinearly and each step is relatively simple to compute. Here we describe the algorithm for implementing Polak-Ribiére conjugate gradient descent on a Grassmann manifold, which can be implemented in $O(nk^2)$ time on $Gr(k, n)$.

Polak-Ribiére Conjugate Gradient to Minimize $F(Y)$ on a Grassmann Manifold. Edelman et al. (1998)

1. Choose initial $Y_0 \in Gr(k, n)$. Let F_{Y_0} be the matrix of partial derivatives of F with respect to each element of Y_0. Define $G_0 = (I - Y_0 Y_0^t)F_{Y_0}$ and let $H_0 = -G_0$.

2. Iterate all following steps over j from 0 until the process is terminated.

3. Minimize $F(Y_j(t))$ over all t where $Y_j(t)$ is the geodesic starting at Y_j with initial direction H_j. Let t_j be the t at which the minimum occurs and let $Y_{j+1} = Y_j(t_j)$.

4. Define $G_{j+1} = (I - Y_{j+1} Y_{j+1}^t)F_{Y_{j+1}}$.

5. Parallel transport both G_j and H_j along $Y_j(t)$ to the point Y_{j+1}, calling these τG_j and τH_j.

6. Compute the new direction of travel

$$H_{j+1} = -G_{j+1} + \gamma_j \tau H_j \quad \text{where} \quad \gamma_j = \frac{\text{tr}(G_{j+1} - \tau G_j)^t G_{j+1}}{\|G_j\|_F}.$$

7. Conjugate gradient algorithms must be occasionally reset, so let $H_{j+1} = -G_{j+1}$ if $j + 1 \equiv 0 \pmod{p}(n - p)$.

3.1.2 Newton's Method

Newton's method is another iterative algorithm which can find the minimum of a function. In a single Newton's method step, a step is taken in the direction of the negative gradient by an amount which depends on the inverse of the function's Hessian. This method tends to converge quadratically, but each step is not as simple to compute as in a conjugate gradient method. In particular, while conjugate gradient steps could be computed purely in terms of the value of a function and its gradient, Newton's method also requires knowledge of the Hessian. In practice, this can cause each iteration to be more difficult to compute relative to a conjugate gradient step. In Edelman et al. (1998) there is a standard implementation of Newton's method on a Grassmann manifold which can also be implemented in $O(nk^2)$ time on $Gr(k, n)$.

Of course, an iterative numerical method requires an initial guess of a correct solution and may behave sensitively with respect to the choice of that guess. In particular, it is important to know how inaccurate the initial guess can be without unexpected behavior, which may effect how far can a tracked object move in a single frame of a dynamic hyperspectral image without being lost by the tracking algorithm, for example. When using Newton's method to estimate eigenspaces of operators by using the Grassmann manifold, a novel improvement to Newton's method developed in Absil et al. (2003) is of use. It tends to enlarge bases of attraction about minima on the manifold without a significant performance cost. This leads to a far more stable implementation.

3.1.3 Stochastic Gradient Descent

Stochastic gradient descent is a more specialized algorithm than the previous two. It seeks to efficiently find a minimum of a function which is written as a sum of many component differentiable functions. In particular, if many component functions are complicated or take a very similar form to one another, then computing the gradient of the total function by summing the gradient of every component function becomes a particular inefficient task. Instead, component functions can be sampled and the gradient of the total function estimated from the gradients of those chosen components. After the direction and step size have been determined, this method behaves precisely like a standard gradient descent method. A robust and versatile version of this algorithm has been brought to a Grassmann manifold in He and Zhang (2014). This paper develops an adaptive stochastic gradient descent algorithm, where the step size is chosen adaptively so as to optimize the convergence rate. In general, stochastic gradient descent can only promise a linear rate of convergence, but each calculated step is made significantly simpler than in other methods.

3.2 Linear Dimension Reduction Techniques

Many high-dimensional data sets actually contain only a smaller number of dimensions of actual information hidden among the many other dimensions. In particular, it is often the goal to take a high-dimensional data set and obtain a compact representation with very few dimensions. Linear dimension reduction techniques begin with data in $\mathbb{R}^n$ and seek to find a k-dimensional hyperplane which characterizes the data as well as possible without a significant loss of valuable information. Many methods have been developed for dimension reductions. These methods are often less stable due to the assumption of linearity and avoid the use of manifolds. They are very simple to implement and very computationally efficient.

3.2.1 Principal Component Analysis

Principal component analysis, or PCA, is a linear dimension reduction technique. It seeks to understand in which directions the data most significantly varies, as seen visually in Figure 3.2. It produces an ordered list of vectors of decreasing significance. In a standard dimension reduction procedure, the first k vectors in this list may be chosen. Here is a simple method of

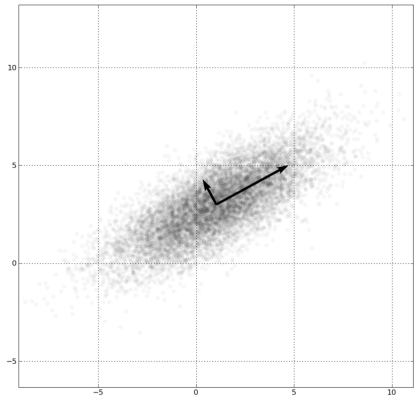

Figure 3.2 Principal component analysis examines the directions of most extreme variation in a data set. (c)

performing a principal component analysis.

Principal Component Analysis

1. Let X be the matrix of data.
2. Compute the singular value decomposition $X = U\Sigma V^t$.
3. The principal component decomposition of X is then given by $P = U\Sigma$, where each column of P is a principal component of X.
4. The ith singular value provides a weighting for the ith principal component.

Principal component analyses are exceedingly simple to compute, giving them a strong advantage over most other dimensionality reduction techniques. However, when attempting to classify high-dimensional data, the subspace chosen by a principal component analysis may not be useful. Consider a facial classification problem where the task is to identify whether an individual in an image is French or German. While dimensionality reduction can be useful in such cases, a principal component analysis is likely be dominated by other factors such as biological gender which more heavily influence appearance.

There are a number of useful extensions of principal component analysis as well. While a standard PCA is influenced heavily by any outliers in data, a robust PCA is designed to behave in a more stable manner. Kernel PCA methods allow for nonlinear dimensionality reduction to be performed with little more computation than a standard PCA. These methods use a kernel to bring non-linear data to a linear space to be processed. However, such methods do not tend to consider the underlying manifold structure which the data may possess.

Many of the currently existing algorithms for dimensionality reduction and hyperspectral image processing use variations of principal component analysis methods. These methods are expected to produce inferior results to the more geometric Grassmann manifold methods. It is in part the goal of this thesis to replace the use of PCA with the use of Grassmannian methods.

3.2.2 Support Vector Machines

Support vector machines, or SVMs, are supervised machine learning tools useful for dimensional reductions on binary classification tasks. A standard support vector machine takes a training set of data divided into two classes. The support vector machine will then determine the linear subspace separating the two classes such that the orthogonal distance between the closest two elements of the two classes is maximized. The dimension can then be reduced by only considering the orthogonal complement of the selected subspace. Support vector machines can also be generalized using kernel methods so as to be able to use nonlinear separating curves.

3.2.3 Sparse Support Vector Machines

Support vector machines can also be made more efficient through the use of sparsity. High-dimensional data is often sparse, which is to say that any data depends only on a small number of indices relative to the total number of dimensions. Moreover, forms of sparsity may be induced using clever choices of norm, particularly in this case an ℓ_1 norm. In such cases, sparse support vector machines, or SSVMs, are of use. Inducing sparsity causes a support vector machine to output more polarized results, strongly favoring a few dimensions while mostly neglecting all others. For very high-dimensional systems, this allows the majority of dimensions to be ignored, dramatically speeding up computation. The existence and methodology of sparse support vector machines for dimension reduction is given in Bi et al.

(2003).

Formally, a sparse support vector machine seeks to define two parallel hyperplanes

$$\{x : w^T x + b = -1\} \qquad \text{and} \qquad \{x : w^T x + b = 1\}, \tag{3.1}$$

such that the margin between them given by $2/\|w\|_{\ell_1}$ is maximized. If the data is separable, these two hyperplanes are to be chosen so that they separate the two classes of data. If not, an error factor is introduced. This leads to the optimization problem (with e the all 1s vector, D the diagonal matrix of class labelings for each training point, X the matrix of data point coordinates, and C a positive parameter characterizing the tradeoff between separability and margin size),

$$\begin{aligned} \underset{w,b,\zeta}{\text{minimize}} \quad & \|w\|_1 + Ce^T\zeta \\ \text{subject to} \quad & D(Xw + be) \geq e - \zeta, \quad \zeta \geq 0. \end{aligned} \tag{3.2}$$

Ultimately, this optimization problem may be converted into a linear program as expressed in Chepushtanova (2015) equation (2.8). Such sparse support vector machines have been applied to the classification of hyperspectral images in Chepushtanova (2015) and Chepushtanova and Kirby (2015). The algorithm presented in this paper is a modification of the method used in these papers. Implementation of these methods in this thesis's analyses have commonly reduced 12000-dimensional spaces to at most 3 significant dimensions.

3.3 Nonlinear Dimension Reduction Techniques and Manifold Embedding Methods

While linear dimension reduction techniques have their advantages, such as fast computation time, they are incredibly inefficient when significant nonlinearity is introduced to a problem. Suppose, as an example, the task of tracking a pendulum in a high-resolution video. Each frame can be viewed as a single point in an extraordinarily high-dimensional space. However, in actuality the pendulum possesses very few degrees of freedom, tracing a nonlinear path through the video along a simple manifold. Nonlinear dimensional reduction techniques can serve to characterize this simple manifold rather than the vastly larger space, allowing for much simpler processing and simpler visualization. This section will introduce a series

of nonlinear dimension techniques. In addition, these techniques may also serve to embed abstract manifolds into a Euclidean space where they may be more easily analyzed.

Many of these techniques are, in a Euclidean setting, dependent on the Manifold Hypothesis, an assumption that a high-dimensional data set does not truly depend on the entire space it is embedded in. Instead, this hypothesis posits that the data is actually sampled from an underlying lower-dimensional smooth manifold. This reduces many data analysis questions to determining an approximate form of the underlying sample manifold, from which a natural geometry on the data arises. In real applications, this hypothesis is often valid and is a good initial assumption.

3.3.1 Multidimensional Scaling

Given some collection of high-dimensional data, multidimensional scaling is a method of constructing a configuration of points in a lower-dimensional Euclidean space based purely on the distances between the points. It attempts to create an isometric embedding in the minimum possible dimensional space. In this paper, multidimensional scaling is primarily used as an embedding method to bring points on an abstract manifold into a Euclidean space.

Multidimensional Scaling Kruskal and Wish (1978)

1. Take as input an specified order for a set of p datapoints and D a $p \times p$ matrix of distances between those points.
2. Then, let $M = PAP$ for $P = I - \frac{1}{p}ee^T$, $A_{ij} = -\frac{1}{2}D_{ij}^2$, and e the all 1s vector.
3. Write $M = X^T X$.
4. Columns of X represent points of data embedded in a Euclidean space.

While this method is extremely versatile and simple to implement, constructing the distance matrix can be extremely computationally intensive for points on an abstract manifold. When used as a method to embed a Grassmann manifold into Euclidean space, determining this distance matrix requires $O(p^2)$ distance computations, each of which requires an SVD to be computed.

3.3.2 Locally Linear Embedding

Locally linear embedding, or LLE, is a sophisticated nonlinear dimensional reduction technique. For each datapoint, the nearest neighbors are selected and points are thus connected by edges, forming a distance-weighted graph which approximates the manifold. The method then takes advantage of the local linearity of the manifold in order to represent each point by a linear combination of its neighbors, with some small error term. In each case, we obtain a local understanding of the graph and thus the intrinsic manifold describing the data. This manifold is then mapped to as low dimensional a linear space as possible while preserving the observed local properties. The method is described further in Saul and Roweis (2000) and Roweis and Saul (2000). An analysis of this algorithm is also provided in Belkin and Niyogi (2003). This method runs in $O(dn^2)$ where d is the number of original dimensions and n the number of original data points. This quadratic growth in the number of datapoints can make locally linear embedding less feasible on extremely large datasets.

3.3.3 Laplacian Eigenmaps

Much like locally linear embedding, Laplacian eigenmaps as introduced in Belkin and Niyogi (2003) seeks to convert data into a weighted graph in order to understand the structure of an underlying lower-dimensional manifold. However, the Laplacian eigenmaps algorithm is motivated by the optimal embedding problem on a manifold, potentially obtaining better results than locally linear embedding. In a manifold optimal embedding problem, we seek a map to find a map f which has unit $\mathcal{L}^2$ norm from a manifold to a low dimensional Euclidean space such that the integral over a standard measure

$$\int_{\mathcal{M}} \|\nabla f(x)\|^2$$

is minimized. Finding such an f on a manifold can be treated as an eigenvalue problem for the Laplace-Beltrami operator. In building a nearest neighbor graph out of data sampled on a manifold, the Laplace-Beltrami operator may be approximated by the graph Laplacian and a similar minimization process thus carried out.

Laplacian Eigenmaps Belkin and Niyogi (2003)

1. Construct an adjacency graph. This can be done by choosing a number of nearest neighbors for each point or by connecting all datapoints lying within a certain distance of each other.

2. Choose a positive real parameter t. Provide a heat-kernel weighting on the adjacency graph between datapoints x_i and x_j of

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

   if x_i and x_j have an edge between them and 0 otherwise.

3. Define the diagonal weight matrix D by

$$D_{ii} = \sum_k W_{ki}$$

   and $D_{ij} = 0$ for $i \neq j$. Define the graph Laplacian L by

$$L = D - W.$$

4. Compute the eigenvalues and eigenvectors for the generalized eigenvector problem

$$Lf = \lambda Df,$$

   Note that if there are no isolated points in the adjacency graph, one can simply invert D and obtain a standard eigenvector problem.

5. Choose a positive integer m as the number of dimensions for a low-dimensional embedding. Choose the eigenvectors f_k for $1 \leq k \leq m$ corresponding to the m smallest strictly positive eigenvalues. Define an embedding into $\mathbb{R}^m$ by

$$x_i \to (f_1(i), f_2(i), \ldots, f_m(i)).$$

Laplacian eigenmaps very well captures local geometric information of the data in an embedding. However, it takes the number of dimensions which are output as a parameter specified by the user. This denies the approach the ability to adaptively choose an ideal low-dimensional space in which to place output. More significantly, there is as of yet no known way in which to choose an optimum value for the parameter t, though

experimental evidence in Belkin and Niyogi (2003) indicates that such an optimum value likely exists and that the performance of the algorithm can depend on the value of t. However, if the graph is sufficiently well-connected, simply choosing $t = \infty$ (ensuring weights of 1 for any connected edges) does capture a reasonable amount of information about the data's structure.

3.4 Grassmannian Methods for Video Data Analysis

A number of efficient algorithms using a Grassmann manifold for tasks dealing with high-dimensional data have been developed. While this thesis's algorithm is not designed for dynamic hyperspectral images, this section serves as a starting point from which Grassmannian methods for such an analysis can be built. This section will serve to describe how such algorithms represent and process data.

3.4.1 Persistent Homology for Hyperspectral Videos

This algorithm found in Chepushtanova (2015) makes use of persistent homology in order to understanding changes over time in a hyperspectral video. Persistent homology is a novel technique from topological data analysis which allows for significant topological data to be extracted as well as the relevant length scales on which such patterns occur. Imagine as an example two sets of planar data, one of which is randomly distributed on $[-1, 1] \times [-1, 1]$ and the other which lies on the unit circle. The second data set has a topological feature, namely the hole in the data which is presumably an important feature of the way in which it was generated. One tool in simplicial homology is the Vietoris-Rips complex, built by varying a parameter $\epsilon > 0$ is varied over some range of values. For each ϵ a simplicial complex is generated, where some set of $k + 1$ points is made into a k-simplex if the pairwise distances between them is less than ϵ. Examining the simplicial homology of the generated complex as a function of ϵ gives access to a tremendous amount of topological data like the presence of holes while also providing geometric information like relevant length scales for a set of data.

In order to apply this technique to a hyperspectral video, the video must first be decomposed into manageable frames which can then be compared in order to gain temporal information. Analyzing a series of frames in Euclidean space is too computationally intense. Instead, take an $x \times y$

frame with z frequency bands and reform it into an $xy \times z$ matrix. By taking the singular value decomposition of this matrix and choosing an orthogonal matrix as a representative, an entire frame of the hyperspectral video may be represented as a point on a Grassmann manifold $Gr(z, xy)$. After bringing an entire video to the Grassmann manifold, the Vietoris-Rips complex is built and simplicial homologies calculated.

In Chepushtanova (2015), this technique is used in order to detect the flow of a gas in a hyperspectral video. However, the value of this algorithm is somewhat limited. While it is particularly good at tracking the presence and the topological behavior of changes, it is not good at distinguishing between multiple distinct simultaneous changes, such as if a gas were being released with a moving background behind. Moreover, this technique is not easily capable of showing where a change is happening, only noting its presence and its behavior.

3.4.2 GRASTA

GRASTA as presented in He et al. (2011) is a novel approach to subspace-tracking from incomplete information. It has demonstrated usefulness in robust matrix completion and real-time separation of background from foreground in standard video data. GRASTA utilizes a dynamic subspace-tracking algorithm. Note that best capturing an object moving in video over time requires the use of a dynamically updating collection of pixels, which is to say a different choice of subspace. In order to properly capture behavior while staying efficient, an algorithm must both perform dimension reductions to low-dimensional subspaces while also maintaining the ability to alter the subspace as time progresses should a new one become more representative of the data.

GRASTA relies upon repeatedly updating four quantities. First, given an estimate of the relevant low-dimensional subspace which a problem has been reduced to, $\hat{U}_t$, as well as the current observations, GRASTA updates an estimated sparse vector s, weighting vector w, and dual vector y. This is done using an ADMM algorithm, detailed shortly. These three vectors are relevant in order to produce a new estimated subspace $\hat{U}_{t+1}$. This is carried out using an Augmented Lagrangian and a simple gradient descent method on the Grassmann manifold to move between subspace estimates, with the standard assumption that the optimal subspace does not change significantly between individual time-steps. The size of the gradient steps are chosen adaptively so as to allow for both fast convergence when a

subspace is static and fast adaptation when the relevant subspace changes from frame to frame. It is through alternation of these two primary steps that GRASTA obtains a fast convergence rate. The algorithm can be found more explicitly described in He et al. (2011), yet here it suffices to provide a rough description and discuss t-GRASTA, a more general version of this algorithm, in more detail.

GRASTA is an exceedingly efficient algorithm, able to process standard video data in real time at 57 frames per second. Given an initially n-dimensional subspace which is to be brought to d dimensions at any given time, GRASTA requires $O(nd^2)$ steps to run, asymptotically similar to most algorithms relying upon dimensional reduction on a Grassmann manifold. It's conceivable that a modification of this algorithm could be made to run in close to real time on a hyperspectral video.

3.4.3 t-GRASTA

Presented in He et al. (2014), t-GRASTA, or transformed GRASTA, is a generalization of GRASTA. It is particularly suited for image alignment in the presence of large amounts of noise and corruption and dealing with robust subspace-tracking. One of the most significant improvements of t-GRASTA over GRASTA is in its ability to handle a changing background, such as in the case of a camera jitter or camera attached to a moving object. However, dealing with such a situation requires more than linear subspace-tracking, as the geometric transforms on a video from frame to frame may be nonlinear, leading to a general manifold-tracking problem. This is remedied in t-GRASTA by approximating this general manifold locally by a union of subspaces, allowing Grassmannian methods to continue to play a part in a usual subspace-tracking problem.

Much like GRASTA, t-GRASTA functions by alternating between two computations, one of which is an alternating directions method of multipliers, or ADMM, method. This method takes as input an $n \times d$ matrix U, a normalized transformed image $I \circ \tau$ represented in $\mathbb{R}^n$ where τ is the current best estimate for the true geometric transform on the image and the Jacobian matrix J of the image. In addition, the algorithm has four parameters, a penalty constant ρ, a tolerance ϵ^{tol}, and maximum number of iterations K. It returns a weighting vector $w \in \mathbb{R}^d$, a collection of sparse outliers $e \in \mathbb{R}^n$, the parameters for a locally linear approximation to τ called $\Delta\tau'$, and a dual vector $\lambda \in \mathbb{R}^n$ for optimization purposes.

ADMM Solver for a Locally Linear Problem He et al. (2014)

1. Initialize w, e, $\Delta\tau$, and λ by $w = e = \Delta\tau = \lambda = 0$ and initialize $\mu = 1$.

2. Set $P = \left(U^t U\right)^{-1} U^t$, $F = \left(J^t J\right)^{-1} J^t$, and

$$h(w, e, \Delta\tau) = Uw + e - I \circ \tau - J\Delta\tau.$$

3. Iterate through the following steps either K times or at least one time ending as soon as

$$\|h(w, e, \Delta\tau)\|_{\ell^2} \leq \epsilon^{tol}.$$

4. In this step, we proceed with the understanding that as soon as the value of a variable is updated, its new value is used immediately afterward in all calculations. Update the value of $\Delta\tau$ by

$$\Delta\tau_{new} = F\left(Uw + e - I \circ \tau + \frac{1}{\mu}\lambda\right)$$

and the value of w by

$$w_{new} = P\left(I \circ \tau + J\Delta\tau - e - \frac{1}{\mu}\lambda\right).$$

Update the value of e by making use of an element-wise soft thresholding operator $S_{\frac{1}{\mu}}$ which in effect acts as a slight barrier to values significantly above $\frac{1}{\mu}$ by

$$e_{new} = S_{\frac{1}{\mu}}\left(I \circ \tau + J\Delta\tau - Uw - \frac{1}{\mu}\lambda\right).$$

Then update the value of λ by

$$\lambda_{new} = \lambda + \mu h\left(w, e, \Delta\tau\right),$$

and μ by

$$\mu_{new} = \rho\mu.$$

5. Here end the loop if $\|h(w, e, \Delta\tau)\|_{\ell^2} \leq \epsilon^{tol}$ or if the loop has been run through K times.

6. Output the most recent values of $w, e, \Delta\tau$, and λ.

With the ADMM method established, the full algorithm for the online mode of t-GRASTA can be given. This algorithm takes as input a series of L $n \times d^\ell$ orthonormal matrices U^ℓ, chosen as representatives of particular subspaces $S^\ell \in Gr(d^\ell, n)$ for integer ℓ from 1 to n. It also requires a sequence of N unaligned images, which can be chosen from a video, I_i and the parameters of the initial transformation as best as can be approximated τ_i^0 for integer i from 1 to N. It returns a series of iteratively approximated subspaces U_i^ℓ and the corresponding transformation parameters τ_i^L obtained after the processing of the ith image.

t-GRASTA Fully Online Mode He et al. (2014)

1. Nest two loops, with the outer loop iterating through i from 1 to N and the inner loop iterating through ℓ from 1 to L.

2. Produce an updated value of the Jacobian matrix J of the ith image
$$J_i^\ell = \left.\frac{\partial(I_i \circ \zeta)}{\partial\zeta}\right|_{\zeta=\tau_i^\ell}$$
and produce an updated version of the normalized transformed image $I \circ \tau$ by
$$I_i \circ \tau_i^\ell = \frac{\text{vec}\left(I_i \circ \tau_i^\ell\right)}{\|\text{vec}\left(I_i \circ \tau_i^\ell\right)\|_{\ell^2}}.$$

3. Produce an estimate for the weight vector w_i^ℓ, the sparse outliers e_i^ℓ, the locally linear transformation approximation $\Delta\tau_i^\ell$, and the dual vector λ_i^ℓ using the ADMM algorithm with $I_i \circ \tau_i^\ell$, J_i^ℓ, and the current estimate of the subspace U_i^ℓ.

4. Set an augmented Lagrangian
$$\mathcal{L}(U, w, e, \Delta\tau, \lambda) = \|e\|_{\ell^1} + \lambda^t h(w, e, \Delta\tau) + \frac{\mu}{2}\|h(w, e, \Delta\tau)\|_{\ell^2}^2.$$

Compute the gradient of this Lagrangian by first letting

$$\Gamma = \left(I - U_t^\ell (U_t^\ell)^t\right)\left(\lambda_i^\ell + \mu h\left(w_i^\ell, e_i^\ell, \Delta\tau_i^\ell\right)\right)$$

and then setting

$$\Delta\mathcal{L} = \Gamma(w_i^\ell)^t.$$

5. Set the step-size η_i^ℓ. This can be chosen a number of ways, but t-GRASTA is not as sensitive to its choice as GRASTA was.

6. Update the estimated subspace

$$U_{i+1}^\ell = U_i^\ell + \left[\left(\cos\left(\eta_i^\ell \|\Gamma\| \|w_i^\ell\|\right) - 1\right) U_t \frac{w_i^\ell}{\|w_i^\ell\|} - \sin\left(\eta_i^\ell \|\Gamma\| \|w_i^\ell\|\right) \frac{\Gamma}{\|\Gamma\|}\right] \frac{(w_i^\ell)^t}{\|w_i^\ell\|},$$

and the updated transformation parameters

$$\tau_i^{\ell+1} = \tau_i^e ll + \Delta\tau_i^\ell.$$

7. Upon the completion of both loops, output all U_i^{L+1} and τ_i^{L+1}.

For our purposes, giving the algorithm suffices, a full discussion of the robustness and generality of the algorithm can be found in He et al. (2014) where it was presented.

Chapter 4

Object Classification Algorithm

In order to classify objects in a hyperspectral image, an algorithm based upon the classification on embedded Grassmann manifolds technique in Chepushtanova (2015) has been developed. Empirical evidence suggests that this algorithm is able to extraordinarily accurately distinguish between different objects in a hyperspectral image. Here this algorithm and implementation will be described in detail, as well as a range of parameter variations and potential improvements which may be pursued in future work on this topic. This algorithm has also been implemented in MATLAB, with the code for that implementation made freely available.

Binary Supervised Classification of a Hyperspectral Image

1. Begin with a region of a hyperspectral image with n bands and a collection of pixels with known classification, as well as the two classes to be distinguished.
2. Divide the region into a grid of smaller blocks and iterate over blocks.
3. From the training data, choose 9 at a time and assemble them into a point on $Gr(9, n)$, creating a number of training points on the manifold.
4. For all pixels in the block, assemble the 9 surrounding pixels into a point on $Gr(9, n)$.

5. Determine the matrix of pairwise distances between all points on $Gr(9, n)$ using a specified metric.
6. Embed the points on $Gr(9, n)$ into a Euclidean space using multidimensional scaling.
7. Using the embedded training points, train an SSVM and select relevant dimensions.
8. Train a standard SVM on the remaining low-dimensional space and classify all embedded points representing pixels.
9. Return a 2-dimensional grid containing the classification result for each pixel in the image.

Most notably, given n bands and a region containing p pixels, this algorithm runs in $O(np)$, though the constant involved is extremely large and generally depends on the square of the number of pixels in each block.

4.0.1 Initializing the Algorithm

This is a supervised algorithm, so ultimately a set of preclassified training data must be obtained. On images where ground truth is not available, these locations must be identified by hand. Before beginning the algorithm, a human must create a list of types of material to be classified and assign a collection of locations on the image to each class. These pixels need not be in the specified region of the hyperspectral image. Several attempts to use clustering methods to obtain training data and thus obtain an unsupervised algorithm were inconclusive, as such methods are extremely computationally intensive and must be run before any hyperspectral band selection or dimension reduction processes occur.

4.0.2 Determining Block Size

Following the determination of training data, the program may be initialized. One of the most significant input parameters is the size of the blocks into which the hyperspectral image is divided. In running, the vast majority of this algorithm's time is generally occupied in computing the matrix of pairwise distances between points on the Grassmann manifold, which when p points are on the manifold requires $O(p^2)$ operations. As such,

shrinking the size of each block will decrease the number of distance calculations which need to be carried out. However, the number of times all other steps in the algorithm will need to be carried out will increase, so selecting the optimal block size requires these costs to be balanced. One further consideration when determining block size comes from the random selection of a subset of training data. This allows for a small possibility of an ineffective sample, which may cause significant errors in the classification data of a single block. This effect may be mitigated by selecting a sufficient quantity of training data and using fewer blocks. Generally, selecting square blocks with a width between 10 and 15 pixels sufficiently balances these concerns.

4.0.3 Bringing Training Data to the Grassmann Manifold

In order to bring pixels on a hyperspectral image to points on a Grassmann manifold, the tall orthogonal matrix representation is utilized. Suppose $u_1, \ldots, u_k$ are a collection of pixels (themselves represented as column vectors) known to have the same classification. Then, by appending them together obtains a matrix M. Two distinct methods of mapping such a matrix to an orthogonal tall matrix in a meaningful way were considered. In the current implementation, a compact singular value decomposition is found so that $M = U\Sigma V^T$ for U and V orthogonal and Σ diagonal. As U has the same dimensions as M, each such matrix M is associated with U. This map is differentiable and numerical evidence suggests that it well captures the properties of M relevant to a Grassmannian analysis.

However, singular value decompositions are in general computationally intensive, so a simpler method has also been developed. However, it is still unknown if it exhibits the same degree of stability as is possessed by the singular value decomposition method. In this method, compute the QR decomposition of M so that $M = QR$ for Q orthogonal and R upper triangular. Then, selecting the first few columns of Q so as to obtain a matrix with the same dimensions as M implies a mapping from M to a Grassmann manifold. As QR decompositions are significantly simpler to calculate than SVDs, future analyses will seek to observe if this mapping captures information as effectively.

Whichever process is used, the current implementation requires that 9 training points be selected for every point on the Grassmann manifold. This technique is then repeated until a sufficient quantity of points on the manifold is obtained. Sampling with replacement is acceptable, as the

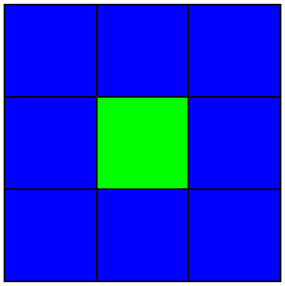

Figure 4.1 Each pixel, marked in green, is aggregated into a matrix by combining it with the 8 pixels, marked in blue, surrounding it. This matrix is then brought to the Grassmann manifold to represent the center pixel.

difference of even a single pixel will change the representative point on $Gr(9, n)$.

4.0.4 Bringing Pixels to the Grassmann Manifold

The same techniques which allow for training data to be brought to a Grassmann manifold can also allow for each pixel to be represented on the manifold. However, when analyzing training data, one could always able to select a number of pixels with the same classification, as theirs was already known. In the case of an arbitrary pixel, it is likely not possible to select other pixels with the same classification, making the aggregation process more complicated. Instead, one can work to choose with pixels which are relatively likely to have the same classification. If the hyperspectral camera is assumed to have a resolution such that a single pixel is far smaller than any significant object within the image, then a simple solution is to choose the 9 pixels closest to the initial pixel, as demonstrated in Figure 4.1. It is for this reason that the algorithm operates on $Gr(9, n)$ rather than any other Grassmann manifold.

Much of the touted power of hyperspectral images comes from the ability to merge spatial information with spectroscopic information. While the nearest-neighbor solution described above is adequate for most cases, it is possible that more sophisticated spatial configurations can also be used to develop innovative classification techniques. Future work will focus on this possibility. Moreover, when possible, configurations including more pixels generally produce more accurate classification results, as Chepushtanova (2015) demonstrates that it becomes more likely that two training classes on

$Gr(k, n)$ will be separable as k increases.

4.0.5 Distance Matrix Computation

At this point in the algorithm, both training points and pixels within a block lie on a Grassmann manifold. In order to apply an embedding method, a distance matrix must be computed. While any of the notions of distance given in Definition 2.5 (or any other coherent metrics or pseudometrics on a Grassmann manifold) may be used, the current implementation uses the geodesic distance. While results from Chepushtanova (2015) suggest that the 1-smallest pseudometric returns the most efficient classification, experiments with altered metrics did very little to effect accuracy. Accordingly, since the geodesic metric allows for more consistency with earlier notions of parallel transport, its use was continued in this task.

This portion of the algorithm requires, by far, the most computation time. Significant future work will focus on determining a new approach which will avoid the use of a distance matrix, since there is no non-time intensive manner in which one can be found on a manifold.

4.0.6 Obtaining an Embedding

Having determined a distance matrix for all points on $Gr(9, n)$, the following step is to embed all data in a Euclidean space so that very simple and fast dimension reduction techniques may be applied. To this effect, multidimensional scaling is used. If the chordal metric is used to calculate the distance matrix, then this embedding is an isometry as noted in Chepushtanova (2015). For other metrics, there may be a small amount of error introduced. Future analyses may focus on obtaining embeddings by using LLE, Laplacian Eigenmaps, or Isomap.

4.0.7 Restricting to Significant Dimensions

Having obtained an embedding, all points are now represented on an extremely high-dimensional Euclidean space. The vast majority of these dimensions are virtually irrelevant to the analysis, so a sparse support vector machine is trained on the embedded training data to restrict attention purely to the most significant dimensions to classification. This training process yields a vector of weights w for the SSVM. Dimensions are ordered by the absolute value of their corresponding weight. In this order, dimensions are kept until the absolute value of the weighting vector component drops by

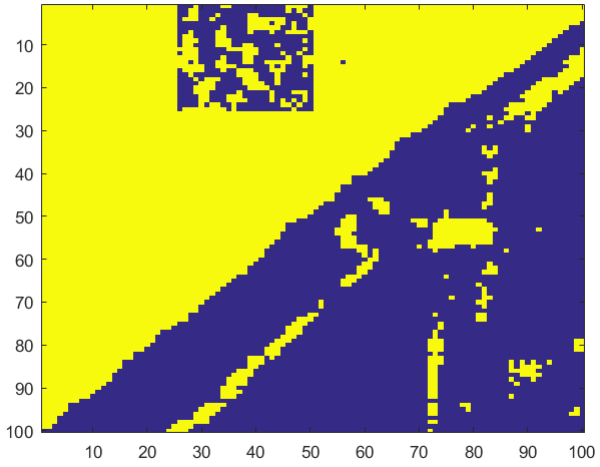

Figure 4.2 This image is the output of the algorithm after analyzing a large portion of an image of Washington D.C., particularly a segment showing some coastline. It is distinguishing between water (yellow) and grass (blue). Inland errors are mostly due to other, non-grass and non-water objects appearing in the image. In the water, a block error may be seen where classification results are effectively random.

a factor of 10 from one entry to the next. All subsequent dimensions are neglected.

4.0.8 Classifying All Pixels

Following this procedure, the pixel data is now simply enough represented and in a sufficiently low-dimensional space that it may be directly classified. A new standard support vector machine is trained on the low-dimensional training data. All pixels in the block are then classified by this support vector machine.

4.0.9 Obtaining the Output

This analysis is conducted separately within each block. Each result is combined into a large 2-dimensional grid which is output as an image, like that of Figure 4.2, which contains 16 blocks. Particularly significant in this image is the very accurate classification of the beach as well as the block

error. Several attempts to automatically detect such block errors have not yet been successful, future work may yield additional insight into this issue.

Chapter 5

Results

After being implemented, the previously described algorithm was tested on several hyperspectral images, with the aim to gauge its capabilities and accuracy. Analysis was primarily focused on a set of two hyperspectral images. On the former, no ground truth was available, whereas the latter had available ground truth. Moreover, the geometry of objects in the latter was far simpler than in the former, causing it to be an easier test for the algorithm. Accordingly, the former serves as a useful feasibility analysis, while more statistical accuracy results can be obtained through an analysis of the latter. The results from each image's analysis are presented in this chapter.

5.1 Feasibility Analysis

In this section, the results from the analysis of a HYDICE (Hyperspectral Digital Imagery Collection Experiment) image of the Washington D.C. National Mall are given. This image contains 191 wavelength bands with wavelengths from 400 micrometers to 2500 micrometers and was taken by a hyperspectral camera mounted on a low-flying plane. A detailed discussion of the capabilities and technical specifications of HYDICE may be found in Mitchell (1995). No ground truth was available, and the image contained a large number of distinct features. The classes considered most relevant for this work were grass, forest, road, building, and water. While each class contained a fair amount of variation, such as recognizably different building materials or differences between saltwater and freshwater, they were considered to be sufficiently homogenous to be characterized together.

A visualization of this dataset may be seen in Figure 5.1. Several results from the analysis of this image will be presented as a series of pairwise classifications between classes of objects.

5.1.1 Water and Grass

Results distinguishing water and grass were among the easiest to obtain. These object classes are very simple to distinguish by eye on a hyperspectral image, so training data is more accurate and more abundant. In addition, grass and water features tend to be larger, so significant portions of the image can be rendered without major concerns over the presence of other classes. As such, large images with recognizable features from the overall HYDICE image like Figures 5.2 and 5.3 may be generated. Moreover, the algorithm was extraordinarily effective at distinguishing between these classes - most SSVM dimension reductions resulted with only one significant dimension. The class separation in a block where two dimensions were relevant may be seen in Figure 5.4. A complete classification result on a large region like these typically requires less than 3 minutes on a standard laptop.

5.1.2 Forest and Grass

Distinguishing between a forested region and a grassy region is one of the most difficult tasks for such a classification algorithm, as the two types of objects look very similar over the majority of the spectrum. There are only a few noticeable differences, as may be seen in Figures 5.5 and 5.6. Despite this, the algorithm was still able to easily distinguish between the two, though with noticeably more errors. In addition, about 8 dimensions were consistently left after the dimension reduction, making visualization more difficult and indicating a more difficult classification task. Such a result may be seen in Figure 5.7. The algorithm's success at this task is particularly useful, as this distinction is extremely important for uses in measuring deforestation and similar environmental issues.

5.1.3 Buildings and Grass

Determining whether or not a pixel is a building or grass is made more difficult by the diverse construction of the buildings in the area. Many buildings have wildly different spectroscopic profiles, so to guarantee accuracy more pre-classified training data was required and it needed to be

Figure 5.1 This image is a visualization of the HYDICE Washington D.C. hyperspectral image. It was formed by choosing 3 particularly relevant wavelength bands and combining them into an RGB image. As such, the colors are not accurate. In the top of the image, the most significant object classes identified are grass, forest, water, and roads. In the bottom of the image, the most significant object classes are grass, buildings, and roads.

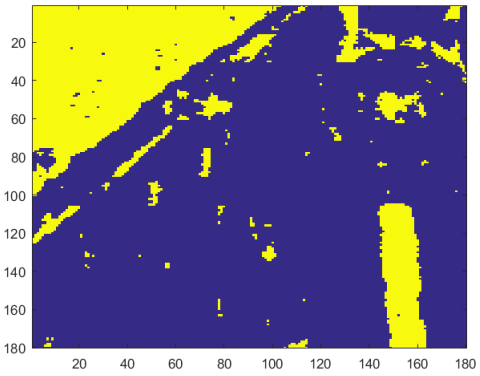

Figure 5.2 This image presents a classification result on the top left portion of the HYDICE image. Water is depicted in yellow and grass in blue. One can see that the algorithm accurately classifies the top left water and the reflecting pool water in the bottom right. The block width was set to 15 pixels, and 13 training points were assembled on the Grassmann manifold for each class.

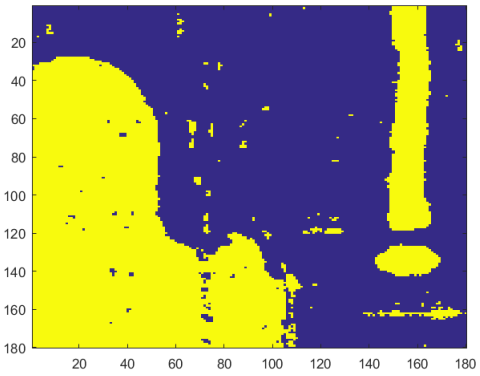

Figure 5.3 This image presents a classification result on the upper left portion of the HYDICE image. Water is depicted in yellow and grass in blue. Again, the algorithm accurately determines the locations of bodies of water. The block width was set to 15 pixels, and 13 training points were assembled on the Grassmann manifold for each class.

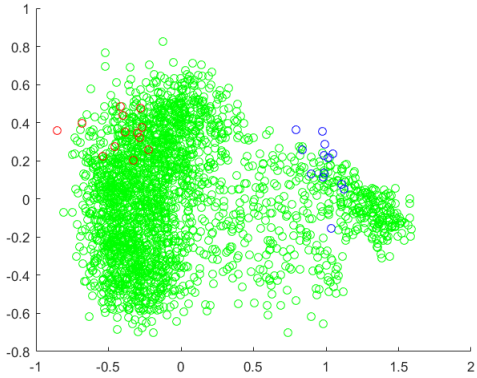

Figure 5.4 This image displays the low-dimensional representation of the data from a single block after the SSVM, when only two dimensions remained. Red points are water training data, blue points are grass training data, and green points are all pixels. There is a clear separation between the two classes. The block width in this data was 25 pixels, and 13 training points were assembled on the Grassmann manifold for each class.

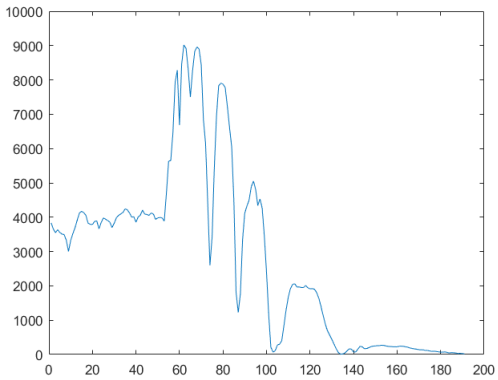

Figure 5.5 This image depicts a fairly typical spectrum for a grass pixel on the HYDICE dataset.

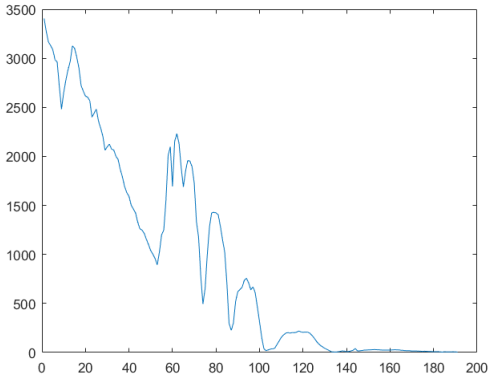

Figure 5.6 This image depicts a fairly typical spectrum for a forest pixel on the HYDICE dataset.

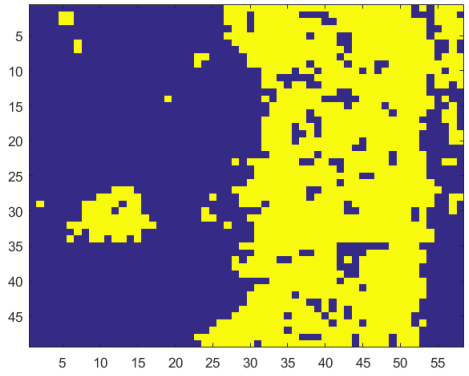

Figure 5.7 This image presents a classification result in the upper portion of the image immediately to the left of the reflecting pool (long rectangular body of water) in a region around the white dot seen in Figure 5.1. Grass is depicted in yellow and forest is depicted in blue. There are more frequent errors in this classification, but it still clearly distinguishes between a grassy region and a forested one. This entire image is a single block, and 20 training points were assembled on the Grassmann manifold for each class.

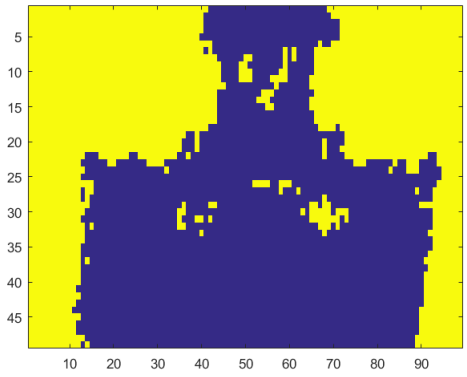

Figure 5.8 This image presents a classification result in the center of the bottom of the HYDICE image, focusing on the Capitol building. Grass is depicted in yellow and the building in blue. This classification was extremely accurate, with almost all errors coming from attempts to classify roads. This entire image is a single large block and 20 training points were assembled on the Grassmann manifold for each class.

sampled from nearly all buildings. After carrying out this process, the algorithm performed admirably in the classification task. When tasked with distinguishing between grass and buildings in the vicinity of the Capitol building, Figure 5.8 was obtained, clearly distinguishing the two. The algorithm encounters difficulty when roads appear in the image, but otherwise classifies very effectively. The number of dimensions remaining after the SSVM was consistently between 2 and 4. Figure 5.9 demonstrates the distribution of data in a block where 3 dimensions remained. This figure demonstrates a large amount of clustering behavior in the pixel data, a good sign if an supervised algorithm is to be designed at a future date.

5.2 Statistical Analysis

In this section, the results from the analysis of an AVIRIS (Airborne Visible / Infrared Imaging Spectrometer) dataset taken of crop fields in a large region taken by Baumgardner et al. (2015) are given. This image contains 220 wavelength bands with wavelengths from 400 micrometers to 2500 micrometers and also uses a hyperspectral camera mounted on a low-flying plane. In this case, while the large image is of a similar size or larger as

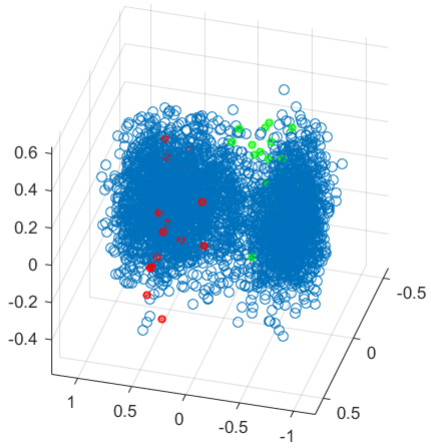

Figure 5.9 This image displays the low-dimensional representation of the data from a single block of the HYDICE image after the SSVM, when only three dimensions remained. Blue points are pixel datapoints, red points are the building training dataset, and green points are the grass training dataset. Clustering behavior in the pixel data is clearly apparent. In this case there are 20 training points per class.

	Grass	Hay	Soybeans	Woods
Grass		99.0%	86.3%	100%
Hay	83.1%		65.5%	77.8%
Soybeans	83.5%	89.2%		99.0%
Woods	100%	100%	100%	

Table 5.1 Each entry in this table is the percentage of correct entries in a binary classification of the classes corresponding to the row and column, when the region being considered was actually entirely composed of the row entry's material. For each task, a region containing about 400 pixels was chosen, and 30 training points were generated on the Grassmann manifold for each class. In general, the algorithm is able to correctly classify a large majority of each field.

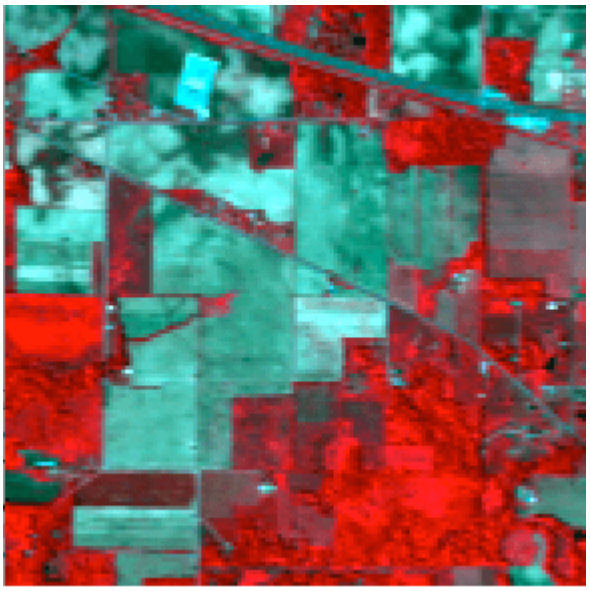

Figure 5.10 This image is a visualization of the AVIRIS dataset. It was formed by choosing 3 particularly relevant wavelength bands and combining them into an RGB image. As such, the colors are not accurate. In this image, fields may be clearly distinguished from their surroundings, but not all different crops may be clearly distinguished from each other.

Figure 5.11 This image displays the ground truth of the AVIRIS dataset, labeling all classified regions. While the entire image is not classified, sufficient quantities are to allow for significant statistical results to be obtained.

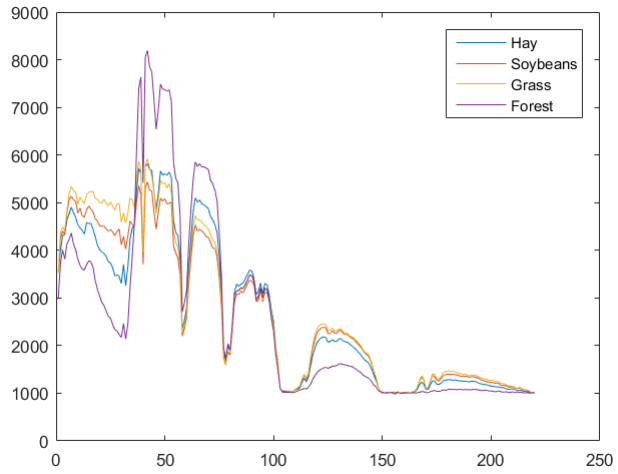

Figure 5.12 This image displays an average spectrum from pixels chosen from each of the four classes primarily being compared in this section. These spectra are all much more similar than spectra in the HYDICE image were.

compared to the HYDICE dataset, the section as seen in Figure 5.10 that is being analyzed is very small, just 145×145 pixels. In this region, the researchers who collected the data clearly denoted the ground truth in Figure 5.11. This image is far more easily divided into distinct components with different materials than the HYDICE image was. In order to obtain error statistics, four materials were chosen. These are pastured or mowed grass, windrowed hay, min soybeans, and woods. A typical spectrum for each may be found in Figure 5.12. These materials all have spectra with very similar characteristics, making this more of a challenging classification task. A subset of a field of each material was used for training data, constituting about 125 pixels for each class. To obtain an accuracy rate, a non-training subset of one of the fields corresponding to each class was distinguished with the other classes, and the number of incorrect pixels counted. These results are given in Table 5.1.

While the algorithm in general performs very well, there are a number of surprising results from this table. Most notable among them is the asymmetry. In general, it is very easy for this method to ascertain that an object is not composed of hay, but very difficult to accurately proclaim that an object is made of hay. Similarly, classifying forests is a very simple task, as may be expected based on how distinct the spectra appears in Figure 5.12. Repeated trials obtain results consistent with those in the table.

Chapter 6

Conclusions and Future Work

Hyperspectral imaging is still a relatively new topic, and there is still great room for exploration and new results in the field. The extraordinary utility of hyperspectral imaging promises that this subject will have great value of some time. In this thesis, an algorithm for object classification in hyperspectral images was developed and its capabilities tested. This supervised algorithm assembles data into points on a Grassmann manifold, which is then embedded into a high-dimensional Euclidean space. Subsequent dimensional reductions allow for efficient classification. This method is versatile and powerful, and is able to process large portions of an image relatively quickly. This result vindicates the use of Grassmann manifolds in the analysis of hyperspectral images. In general, results demonstrate very low error rates on more clearly distinct materials, such as in the HYDICE image distinguishing water from grass, when the final low-dimensional space being considered is only one or two dimensional. In the few cases when the algorithm's error rate is higher, it is correlated with dimensional reductions failing to reduce the problem to a sufficiently low number of dimensions.

While the algorithm performs admirably, there are still many potential improvements which can be made to benefit both its accuracy and its computational efficiency. Future work may focus on clustering and classification directly on the Grassmann manifold, rather than embedding into an ambient Euclidean space. Such an improvement may render the result more intuitive and may remove the need to decompose the image into blocks. In addition, rigorous analyses of the quantity and quality of training data required must be carried out, as it remains unknown how sensitive the algorithm is to variations in training data. In addition, it is hoped that future

work will be able to efficiently convert this algorithm into an unsupervised method. The existing method has already served to demonstrate that data is clustered on the Grassmann manifold a priori, so it remains feasible that a technique like k-means clustering could serve to eliminate the need for humans to preclassify data. Efforts to extend the existing algorithm to multiclass classification show promise using similar methods as those in Huang et al. (2013). Future work may more rigorously test the efficacy of such generalizations.

Each of the previous proposals are relatively incremental improvements to the existing algorithm. Yet there are also fundamentally distinct ways in which a Grassmann manifold may be used to tackle this task. For example, current work on this thesis has large relied upon non-statistical techniques. However, there is a wealth of algorithms relying upon statistical computations on Grassmann manifolds such as in Turaga et al. (2011) or using particle filtering methods on affine Grassmann manifolds as in Shirazi et al. (2014). A comprehensive literature search has revealed few extensions of such statistical methods to the analysis of hyperspectral images, so such an approach remains largely unexplored and may be of great value. In addition, there is great opportunity for more physics-motivated analyses. Future work may take advantage of the atmospheric analyses from Shaw and Burke (2003) and Griffin and Burke (2003) to better understand how to mitigate these confounding effects. Further, reasonable assumptions suggest an error in every band at every pixel due to the inherent Poisson statistics involved in detection of photons in that wavelength band. Future methods may wish to be robust under errors of this type.

In addition to presenting this new algorithm, it is hoped that this thesis will serve as a stepping stone for future research, both in its presented algorithm and in the contained survey of relevant machine learning and manifold optimization methods. A particularly tantalizing future goal is the classification of objects in dynamic hyperspectral images in real time. Much future work will focus on this difficult-to-achieve task. It is only within the last several years that the technology required to capture dynamic hyperspectral images has existed, and they are still largely inaccessible to the general public. As such, the development of such a real-time method would be a significant achievement and could open up an entirely new area of data-enabled science and applications. If the proposed algorithm of this paper is to be modified for this task, each of the goals described in the second paragraph must be implemented. It is particularly important that optimization occurs directly on the Grassmann manifold as that enables

this time-intensive operation to only be carried out once every time step instead of multiple times. This is vital for a real-time algorithm.

The new technology of hyperspectral imaging holds great promise for machine learning algorithms. Very few previously existing technologies contain such a vast amount of information about the world around us in such a condensed and clear manner. It is the author's hope that an effective and efficient technique to analyze hyperspectral imaging will lead to great new insights across disciplines in spectrometry, computer vision, and remote sensing among many other uses.

Bibliography

a. Capturing 3d hyperspectral image cubes for dynamic events | spie homepage: Spie. http://spie.org/newsroom/technical-articles/5495-capturing-3d-hyperspectral-image-cubes-for-dynamic-events. (Visited on 11/11/2015).

b. Free data samples - spectir - advanced hyperspectral & geospatial solutions. http://www.spectir.com/free-data-samples/. (Visited on 11/5/2015).

c. Principal component analysis - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Principal_component_analysis. (Visited on 11/21/2015).

Absil, P-A, Robert Mahony, and Rodolphe Sepulchre. 2004. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica* 80(2):199–220.

Absil, P-A, Rodolphe Sepulchre, Paul Van Dooren, and Robert Mahony. 2003. A newton algorithm for invariant subspace computation with large basins of attraction. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, 2352–2357. IEEE.

Balzano, Laura, Benjamin Recht, and Robert Nowak. 2010. High-dimensional matched subspace detection when data are missing. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, 1638–1642. IEEE.

Balzano, Laura, and Stephen J Wright. 2013. On grouse and incremental svd. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, 1–4. IEEE.

Baumgardner, Marion F., Larry L. Biehl, and David A. Landgrebe. 2015. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site

3. doi:doi:/10.4231/R7RX991C. URL https://purr.purdue.edu/publications/1947/1.

Belkin, Mikhail, and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15(6):1373–1396.

Bi, Jinbo, Kristin Bennett, Mark Embrechts, Curt Breneman, and Minghu Song. 2003. Dimensionality reduction via sparse support vector machines. *The Journal of Machine Learning Research* 3:1229–1243.

Boumal, N., B. Mishra, P.-A. Absil, and R. Sepulchre. 2014. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research* 15:1455–1459. URL http://www.manopt.org.

Chepushtanova, Sofya. 2015. *Algorithms for Feature Selection and Pattern Recognition on Grassmann Manifolds*. Ph.D. thesis, Colorado State University. Libraries.

Chepushtanova, Sofya, and Michael Kirby. 2015. Classification of hyperspectral imagery on embedded grassmannians. *arXiv preprint arXiv:150200946* .

Cockett, Rowan. . conjugategradient.png (543ÃŮ441). http://www.row1.ca/s/presentations/2013/MSc2PhD/images/conjugateGradient.png. (Visited on 12/1/2015).

Edelman, Alan, Tomás A Arias, and Steven T Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications* 20(2):303–353.

Gerhart, Torin, Justin Sunu, Lauren Lieu, Ekaterina Merkurjev, Jen-Mei Chang, Jérôme Gilles, and Andrea L Bertozzi. 2013. Detection and tracking of gas plumes in lwir hyperspectral video sequence data. In *SPIE Defense, Security, and Sensing*, 87,430J–87,430J. International Society for Optics and Photonics.

Griffin, Michael K, and H-h K Burke. 2003. Compensation of hyperspectral data for atmospheric effects. *Lincoln Laboratory Journal* 14(1):29–54.

Habel, Ralf, Michael Kudenov, and Michael Wimmer. 2012. Practical spectral photography. In *Computer Graphics Forum*, vol. 31, 449–458. Wiley Online Library.

Hamm, Jihun, and Daniel D Lee. 2009. Extended grassmann kernels for subspace-based learning. In *Advances in Neural Information Processing Systems*, 601–608.

Harandi, Mehrtash T, Mathieu Salzmann, and Richard Hartley. 2014. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *Computer Vision–ECCV 2014*, 17–32. Springer.

He, Jun, Laura Balzano, and John Lui. 2011. Online robust subspace tracking from partial information. *arXiv preprint arXiv:11093827* .

He, Jun, Laura Balzano, and Arthur Szlam. 2012. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 1568–1575. IEEE.

He, Jun, Dejiao Zhang, Laura Balzano, and Tao Tao. 2014. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing* 32(10):800–813.

He, Jun, and Yue Zhang. 2014. Adaptive stochastic gradient descent on the grassmannian for robust low-rank subspace recovery and clustering. *arXiv preprint arXiv:14124044* .

Huang, Lingkang, Hao Helen Zhang, Zhao-Bang Zeng, and Pierre R Bushel. 2013. Improved sparse multi-class svm and its application for gene selection in cancer classification. *Cancer informatics* 12:143.

Karrasch, Daniel. 2014. An introduction to grassmann manifolds and their matrix representation. .

Kerekes, John P, and Jerrold E Baum. 2003. Hyperspectral imaging system modeling. *Lincoln Laboratory Journal* 14(1):117–130.

Kruskal, Joseph B, and Myron Wish. 1978. *Multidimensional scaling*, vol. 11. Sage.

Manolakis, Dimitris, David Marden, and Gary A Shaw. 2003. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal* 14(1):79–116.

Mitchell, Peter A. 1995. Hyperspectral digital imagery collection experiment (hydice). In *Satellite Remote Sensing II*, 70–95. International Society for Optics and Photonics.

Murzina, Marina VA, and J Paul Farrell. 2005. Dynamic hyperspectral imaging. In *Nondestructive Evaulation for Health Monitoring and Diagnostics*, 135–144. International Society for Optics and Photonics.

Peng, Yigang, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. 2012. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(11):2233–2246.

Roweis, Sam T, and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.

Santamar, Mauricio Villegas. 2011. *Contributions to high-dimensional pattern recognition*. Ph.D. thesis, Universitat de Barcelona.

Saul, Lawrence K, and Sam T Roweis. 2000. An introduction to locally linear embedding. *unpublished Available at: http://www cs toronto edu/~ roweis/lle/publications html* .

Shaw, Gary A, and Hsiao-hua K Burke. 2003. Spectral imaging for remote sensing. *Lincoln Laboratory Journal* 14(1):3–28.

Shirazi, Sareh, Mehrtash T Harandi, Brian C Lovell, and Conrad Sanderson. 2014. Object tracking via non-euclidean geometry: A grassmann approach. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, 901–908. IEEE.

Shirazi, Sareh, Mehrtash T Harandi, Conrad Sanderson, Azadeh Alavi, and Brian C Lovell. 2012. Clustering on grassmann manifolds via kernel embedding with application to action analysis. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 781–784. IEEE.

Skauli, Torbjørn, and Joyce Farrell. 2013. A collection of hyperspectral images for imaging systems research. In *IS&T/SPIE Electronic Imaging*, 86,600C–86,600C. International Society for Optics and Photonics.

Turaga, Pavan, Ashok Veeraraghavan, Anurag Srivastava, and Rama Chellappa. 2011. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(11):2273–2286.

Wang, Lizhi, Zhiwei Xiong, Dahua Gao, Guangming Shi, Wenjun Zeng, and Feng Wu. 2015. High-speed hyperspectral video acquisition with a

dual-camera architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4942–4950.

Wong, Yung-Chow. 1967. Differential geometry of grassmann manifolds. *Proceedings of the National Academy of Sciences of the United States of America* 57(3):589.