DISSERTATION

GRASSMANN, FLAG, AND SCHUBERT VARIETIES IN APPLICATIONS.

Submitted by

Timothy P. Marrinan

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2017

Doctoral Committee:

Advisor: Michael Kirby
Co-Advisor: Chris Peterson

Mahmood R. Azimi-Sadjadi
Dan Bates
Bruce Draper

ABSTRACT

GRASSMANN, FLAG, AND SCHUBERT VARIETIES IN APPLICATIONS.

This dissertation develops mathematical tools for signal processing and pattern recognition tasks where data with the same *identity* is assumed to vary linearly. We build on the growing canon of techniques for analyzing and optimizing over data on Grassmann manifolds. Specifically we expand on a recently developed method referred to as the flag mean that finds an average representation for a collection data that consists of linear subspaces of possibly different dimensions. When prior knowledge exists about relationships between these data, we show that a point analogous to the flag mean can be found as an element of a Schubert variety to incorporates this theoretical information. This domain restriction relates closely to a recent result regarding point-to-set functions. This restricted average along with a property of the flag mean that prioritizes weak but common information, leads to practical applications of the flag mean such as chemical plume detection in long-wave infrared hyperspectral videos, and a modification of the well-known diffusion map for adaptively visualizing data relationships in 2-dimensions.

ACKNOWLEDGEMENTS

I have encountered many supporters since the onset of my mathematical expedition. I have been loved and encouraged throughout my studies, and I'm certain I could not have persisted without this support. If you loved me, thank you. My success is yours as well.

I am deeply grateful to my advisors Michael Kirby and Chris Peterson for sharing their brilliance, guidance, and patience. They mean the world to me and I admire them greatly. I want to thank Bruce Draper, J. Ross Beveridge, and Louis Scharf for the stimulating discussions that we have shared and for the collaborations that will hopefully remain fruitful. Thank you for your wisdom, time, and encouragement. I want to thank all of my colleagues at Colorado State for their companionship and engagement, especially the members of the Pattern Analysis Lab like Tegan Emerson. I want to thank Paul and Marti Marrinan for the love and stability they provided throughout every stage of my development, and I want to thank my dog Reggie for his patience and unwavering affection. You're a good dog Reggie.

Lastly, I would like to thank my grandfather, Jim Mueller. He was one of the brightest lights I have known, and I wouldn't be the person I am today without his influence.

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGEMENTS iii

Chapter 1. INTRODUCTION 1

1.1. Pattern Recognition and Signal Processing 1

1.2. Overview 3

Chapter 2. BACKGROUND 5

2.1. Linear Subspace Models for Data Analysis 5

2.2. Grassmann Manifolds 7

2.3. Flag Manifolds and Schubert Varieties 18

2.4. Grassmannian Averages 22

Chapter 3. SCHUBERT VARIETY CONSTRAINED OPTIMIZATION 28

3.1. Introduction 28

3.2. General Problem 29

3.3. Subspace Constrained Averaging 36

3.4. Tangent Space Decomposition Using Affine Patches 42

3.5. Summary 50

Chapter 4. HYPERSPECTRAL CHEMICAL PLUME DETECTION 53

4.1. Introduction 53

4.2. Flag-based Chemical Plume Detection 56

4.3. Data Set Description 57

4.4. Quantitative Results on Synthetic Data 58

4.5. Qualitative Results 61
4.6. Summary 63
Chapter 5. ADAPTIVE VISUAL SORT AND SUMMARY 65
5.1. Introduction 65
5.2. Adaptive Visual Sort and Summary 68
5.3. Qualitative Results 77
5.4. Summary 79
Chapter 6. CONCLUSION 81
6.1. Contributions 81
6.2. Future Work 82
BIBLIOGRAPHY 85

CHAPTER 1

INTRODUCTION

1.1. Pattern Recognition and Signal Processing

The focus of this dissertation is in developing tools and algorithms for pattern recognition and signal processing tasks that exploit underlying geometric structure in high-dimensional data sets. Pattern recognition encompasses a broad class of applications that involve classifying inputs into identity categories based on features found within the data. The classification is often accomplished via machine learning algorithms in both supervised and unsupervised contexts. Signal processing describes the abstract process of extracting or transferring usable information from digital signals of all types. The scope of this dissertation is in building theory and applications through these two lenses where data samples from within a class demonstrate structured variations; specifically variation that can be represented at least locally by a linear subspace. Such tasks arise in a variety of problems including activity modeling and recognition [70], shape analysis [52], appearance recognition [49], action classification [37], face recognition [31], noisy image classification [45], chemical detection in hyperspectral images [46], and general manifold clustering [5, 29] to name a few. In each of the aforementioned applications data can be represented by linear subspaces that span some form of variation intrinsic to the class or identity of a sample. In this context, the natural place to perform data analysis is the Grassmann manifold, i.e., the parametrization of k-dimensional subspaces of an n-dimensional space, because elements of this space are invariant under changes of basis and related metrics reflect this invariance.

Finding a map between data and a structured manifold allows us to analyze the data using the distances between manifold representatives as a measure of similarity. A simplistic

analogy is that when comparing travel routes on the Earth, it behooves us to measure distance around the globe rather than using straight-line distances which may pass through the mantle or core. An alternative to finding manifold representatives for data points is to treat data as elements of a high-dimensional vector space. In this context, researchers looks to uncover structure within the data cloud itself that might indicate relationships between samples. Both perspectives build bridges to areas of mathematics and statistics from which algorithm developers can draw tools and inspiration. In this dissertation, we stick to the former perspective and thus the job of finding an appropriate mapping between data and a manifold is a crucial step that relates directly to the success of the representation and resulting applications.

Once a map between data and the appropriate manifold has been established, many of these problems benefit from computing summary information for the manifold representatives. Such tools are used to identify what is common between samples corresponding to a single class, but distinct from samples outside of that class. Along this vein numerous averages have been developed for data on Grassmann manifolds. One average of particular interest is the flag mean, which associates a nested sequence of subspaces with a collection of points such that the elements of the flag are the best averages of each fixed dimension. The flag mean is valuable tool because it behaves like a median, in that it best represents the dominant process in a collection of data, but it retains the computational advantages of a mean. It exists as an fast method for summarizing data clouds on Grassmann manifolds, and is the result of an algebraic solution to a geometrically motivated cost function.

Another relevant tool recently developed for data analysis on Grassmann manifolds is a notion of point-to-set distances that agrees with the natural point-to-point Grassmannian

distances in the presence of appropriate restrictions. This technique was discovered concurrently and proved independently by Schwickerath [60] and by Ye and Lim [72]. This development opens the door for advances in optimization and analysis when the domain of a problem is restricted to a subset of points on a Grassmannian.

In this dissertation we attempt to generalize the problem solved by the flag mean to include domain constraints. We look for an element of a Schubert variety, or a subset of points on a Grassmannian that obey a sequence of intersection constraints, that averages the data. In this context the data is not constrained to live on the variety, that requirement is only applied to the resulting average representation. The flag mean and it's subsequent generalization allow us to connect mathematical advances with algorithms for solving practical problems in pattern recognition and signal processing, as was our original goal.

1.2. Overview

In Chapter 2, we provide some general background. As much as possible we strive for this dissertation to be self-contained. Therefore we begin by defining the objects of interest, Grassmann, flag, and Schubert varieties, and proceed by explaining their context in data analysis. We give an explicit derivation for the flag mean, and distinguish it from other subspace averages. We discuss point-to-set distances and explain their relationship to the task at hand.

In Chapter 3, we develop the machinery for Schubert variety constrained optimization. We describe and prove the scenarios for which we can find an optimal solution. We also introduce an iterative algorithm that achieves a locally optimal solution in a more general context. We demonstrate examples of subspace constrained averaging and Schubert variety

constrained averaging, and evaluate the solutions using a decomposition of the tangent space of a point represented by an affine patch of the Grassmannian.

Based on the theory discussed in Chapter 2 and Chapter 3, we create an algorithm for chemical plume detection in hyperspectral videos in Chapter 4. We discuss the difficulty of extracting usable signals from long-wave infrared hyperspectral data, and we introduced some of the common existing algorithms for detecting material signatures in such data. We then contrast our algorithm with the baseline algorithms on real data that allows for practical, qualitative comparisons, and on synthetic data which allows for quantitative results.

Chapter 5 moves to an application of the flag mean in the domain of visual sorting of nanoparticle images for forensic analysis. Here we discuss a host of algorithms for dimensionality reduction, and modify the existing diffusion map to generate a robust visualization algorithm that is cued on prior knowledge about the data at hand. This updated algorithm allows users to create clusters of related images and display a global spatialization based on other images relationships to these clusters.

Finally, we review our contributions in Chapter 6. We discuss open questions and data sets where these techniques might yield interesting and useful results.

CHAPTER 2

BACKGROUND

2.1. Linear Subspace Models for Data Analysis

High-dimensional data has become ubiquitous in pattern recognition, signal processing, and hosts of other application domains, but the often used moniker "big data" ambiguously refers to many distinct representations. It can be applied to streaming data such as Twitter feeds where the number of features per sample is small, but the number of samples is massive and dynamic. It can be used to describe genetic data where observations are only made a few times, but the number of observed pathways is huge. It can refer to monstrously large databases where look ups are costly, and big data can describe applications where both the number of samples and the size of each sample are big, as in analyzing content in YouTube videos. Each aspect of largeness comes with its own set of challenges.

The most common and simplistic setting for big data analysis is a high-dimensional Euclidean space. In this context, each sample is represented by a vector whose entries correspond to quantitative features of the sample. For example, a black and white digital image is a matrix whose entries correspond to the light intensity measured at that pixel. If this image is raster scanned, i.e., if the rows or columns are concatenated to form a vector, then the resulting vector can be treated as a point in a vector space whose dimension is equal to the number of pixels in the image. The benefits of treating an image (or any datum) as a point in a vector space are that many tools exist for statistics, optimization, and analysis in this simple setting.

However, in many applications this representation is insufficient. Suppose that you have two black and white images where one is the photo negative of the other. The scenes

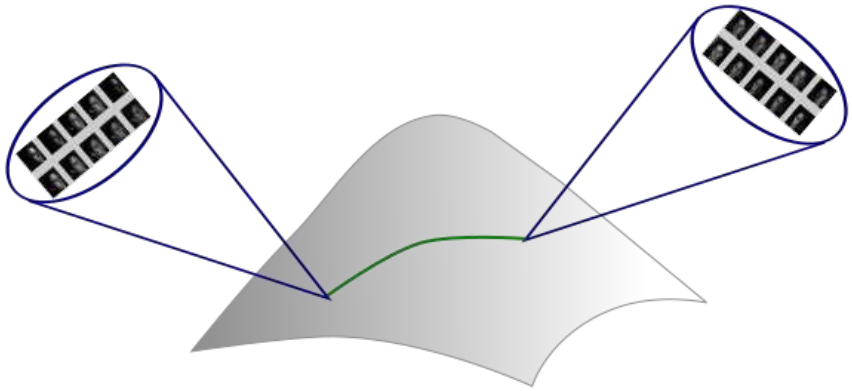


FIGURE 2.1. Illustration of illumination spaces as points on a Grassmann manifold. The basis for each subspace is a set of images of a single person with varying illumination. The pose of the subject is distinct for each subspace, because that change to the image is not linear. Therefore, the *identity* of each sample is tied to the subject and the pose. The green curve represents the minimal geodesic, or the shortest path between the two points that respects the geometry of the manifold.

visualized are the same, but the colors are inverted. Are these distinct images two examples of the same thing? Do they share an identity? The average of these two images as points in a Euclidean space would be a vector with all entries equal which represents no discernible scene. If the identity of these samples is the same, shouldn't the average share that identity? Issues like this one motivate the need for data representations that encode more of the identity of a sample. Thus we build a theoretical bridge to manifolds where the geometry makes sense of some of the structure known to be present in the data. In particular, in this dissertation the notion of a sample will correspond to a linear subspace of a high-dimensional Euclidean space. This representation allows us to include variation in our model, making it robust to addition and scalar multiplication. There are new issues that arise because of this representation, however. Not all mathematical tools that exist in Euclidean space have been generalized to work in this setting. It will be our task then to discuss the existing techniques and develop novel solutions in areas of need. Using subspaces to represent data invokes a natural connection with the Grassmann manifold as a setting for data analysis.

2.2. Grassmann Manifolds

Definition 2.2.1. *The* ***Grassmann manifold*** $\mathrm{Gr}(k, \mathcal{V})$ *is a manifold whose points parametrize the subspaces of dimension* k *inside the vector space* $\mathcal{V}$.

In this dissertation, we will assume that $\mathcal{V}$ is an n-dimensional real vector space. We denote by $\mathrm{Gr}(k, n)$ the Grassmann manifold of k-dimensional subspaces of $\mathbb{R}^n$, $GL(k)$ denotes the general linear group of invertible $k \times k$ matrices and $O(k)$ denotes the orthogonal group of $k \times k$ orthogonal matrices. Let $\mathbb{R}^{n\times k}$ denote the vector space of $n \times k$ matrices with real entries and let $(\mathbb{R}^{n\times k})^\circ$ denote the open submanifold of full rank $n \times k$ matrices. For each $Y \in (\mathbb{R}^{n\times k})^\circ$, let $[Y]$ denote the column space of Y. There is a surjective map $\phi : (\mathbb{R}^{n\times k})^\circ \to \mathrm{Gr}(k, n)$ given by $\phi(Y) = [Y]$ (with $[Y]$ identified with its corresponding point on $\mathrm{Gr}(k, n)$). It is clear that $\phi(X) = \phi(Y)$ if and only if there exists an $A \in GL(k)$ such that $XA = Y$. Thus a point q on $\mathrm{Gr}(k, n)$ corresponds to a k-dimensional subspace V_q of $\mathbb{R}^n$ and can be represented by any element of a $GL(k)$ orbit of a full rank $n \times k$ matrix, Y, whose column space $[Y]$ is equal to V_q.

If Y is a representative with orthonormal columns and if $B \in O(k)$ then YB will be another representative with orthonormal columns. Identifying $n \times k$ orthonormal bases with the same span gives an interpretation of the Grassmannian as a quotient space. That is,

$$\mathrm{Gr}(k, n) \simeq \mathrm{O}(n) \Big/ \mathrm{O}(k) \times \mathrm{O}(n-k)\,. \tag{1}$$

As a quotient space, a point on the Grassmann manifold can be written as the set

$$[Q] \simeq \left\{ Q \begin{pmatrix} Q_k & 0 \\ 0 & Q_{n-k} \end{pmatrix} \;\middle|\; Q \in \mathrm{O}(n),\ Q_k \in \mathrm{O}(k),\ Q_{n-k} \in \mathrm{O}(n-k) \right\}, \tag{2}$$

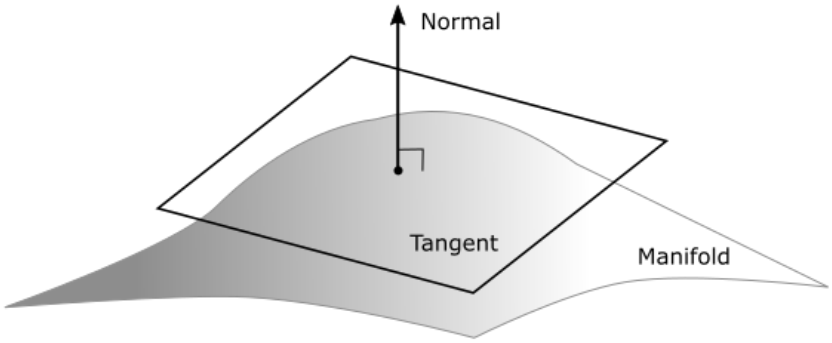


FIGURE 2.2. Illustration of the tangent and normal spaces of a Grassmannian.

which is equivalent to the set of $n \times k$ matrices whose columnspace is equal to the span of the first k columns of Q. We will abuse notation slightly and use $[Q]$ to represent either the span of the first k columns of an $n \times k$ matrix or the span of the first k columns of an $n \times n$ matrix. The quotient space interpretation makes it easy to see that the dimension of the Grassmannian is $k(n-k)$.

2.2.1. TANGENT AND NORMAL SPACE. An n-dimensional manifold is a space that looks locally like $\mathbb{R}^n$. The Grassmannian falls into a category of manifolds known as differentiable manifolds because we can compute derivatives to curves on the manifold. The derivative to a curve $\gamma : [0,1] \to \mathrm{Gr}(k,n)$, at the point $[X]$, is a vector $\gamma'(t)$ lying in the vector space $\mathrm{T}_X\mathrm{Gr}(k,n)$ known as the tangent space of $\mathrm{Gr}(k,n)$ at $[X]$.

Intuitively, the tangent space at a point can be visualized as the plane tangent to the manifold as shown in Figure 2.2. The Grassmannian, $\mathrm{Gr}(k,n)$, has dimension $k(n-k)$ and therefore the tangent space centered at any point does as well. The normal space is the orthogonal complement of the tangent space, and thus has dimension $n - k(n-k)$ in this instance. The normal space at a point $[X] \in \mathrm{Gr}(k,n)$ is the set of matrices of the form

$$Y = XA \tag{3}$$

where A is a $k \times k$ anti-symmetric matrix. Therefore the tangent space of the Grassmannian at $[X]$, $\mathrm{T}_X\mathrm{Gr}(k,n)$, is the set of all $n \times k$ vectors Z such that

$$X^T Z = 0 \quad \text{or} \quad Z = X^\perp B. \tag{4}$$

The matrix $B \in \mathbb{R}^{(n-k)\times k}$ corresponds to the directions free of rotations mixing the basis given by the columns of X.

There is an inner product $\langle \cdot, \cdot \rangle_{[X]}$ uniquely defined on each tangent space of the manifold, which allows us to measure the length of a curve. Let $\gamma : [0,1] \to \mathrm{Gr}(k,n)$ be a differentiable curve, and define the length of γ to be

$$\mathrm{L}(\gamma) \doteq \int_0^1 \|\gamma'(t)\|_{\gamma(t)} dt \tag{5}$$

where $\| \cdot \|_X$ is the norm induced by the inner product in the tangent space $\mathrm{T}_X\mathrm{Gr}(k,n)$.

2.2.2. Geodesics. Geodesics on manifolds generalize the concept of straight lines in $\mathbb{R}^n$. Geometrically, a straight line in $\mathbb{R}^n$ can be thought of as the image of a curve $\gamma(t) : [0,1] \to \mathbb{R}^n$ with zero acceleration for all $t \in [0,1]$. A geodesic on $\mathrm{Gr}(k,n)$ observes the same property, which is to say, the acceleration vector is normal to the manifold at every point along a geodesic. Additionally, a geodesic between two points represents the shortest curve connecting the two points in the sense of the length function defined in Equation 5. The distance between $[X], [Y] \in \mathrm{Gr}(k,n)$ is thus a function of the length of the minimal geodesic between the two points. A geodesic $\gamma(t) : [0,1] \to \mathrm{Gr}(k,n)$ emanating from $\gamma(0) = [Q]$ with

$Q \in O(n)$ can be written as

$$\gamma(t) = Q\mathrm{e}^{t\mathbf{B}} \quad \text{with} \quad \mathbf{B} = \begin{pmatrix} 0 & -B^T \\ B & 0 \end{pmatrix} \tag{6}$$

for any $B \in \mathbb{R}^{(n-k)\times(n-k)}$, and the derivative of $\gamma(t)$ is $\gamma'(t) = \gamma(t)\mathbf{B}$ [13]. In light of this definition and the quotient representation of the Grassmannian indicated in Equation 2, the Grassmannian geodesic emanating from the point $[Q] \in \mathrm{Gr}(k,n)$ is $[\gamma(t)]$. This is however not a very useful formula for computations. In order to compute geodesics, or to move on the Grassmannian in the direction of tangent vectors, we need maps between the manifold and the tangent space of the manifold at a point.

DEFINITION 2.2.2. *For every $Z \in \mathrm{T}_X\mathrm{Gr}(k,n)$ there exists a unique geodesic $\gamma(t) : [0,1] \to \mathrm{Gr}(k,n)$ that depends on $[X]$ and Z such that $\gamma(0) = [X]$ and $\gamma'(0) = Z$. The mapping $\mathrm{Exp}_X : \mathrm{T}_X\mathrm{Gr}(k,n) \to \mathrm{Gr}(k,n)$ defined by*

$$\mathrm{Exp}_X(Z) \doteq \gamma(1) \tag{7}$$

is called the ***exponential map of Z at $[X]$.***

The exponential map computes the retraction of a tangent vector onto the Grassmannian, and can be calculated as

$$Exp_X(Z) = XV\cos(\Sigma)V^T + U\sin(\Sigma)V^T \tag{8}$$

where $U\Sigma V^T$ is the thin singular value decomposition of the tangent vector Z. Pseudocode for computing the exponential map can be found in Algorithm 1, and leads directly to the computation of a geodesic as follows.

Algorithm 1 The exponential map of Z at $[X]$

function $\text{EXP}_X([X], Z)$
  $U\Sigma V^T \leftarrow$ thin SVD(Z)
return $XV\cos(\Sigma)V^T + U\sin(\Sigma)V^T$

THEOREM 2.2.1 (Computing Geodesics on the Grassmann manifold). *Let* $\gamma : [0, 1] \to \text{Gr}(k, n)$ *be a curve with initial point* $\gamma(0) = [X] \in \text{Gr}(k, n)$ *and tangent vector* $\gamma'(0) = Z \in \text{T}_X\text{Gr}(k, n)$. *A point on the geodesic emanating from* $[X]$ *in the direction of* Z *can be computed as*

$$\gamma(t) = XV\cos(t\Sigma)\,Q^T + U\sin(t\Sigma)\,Q^T \tag{9}$$

where $U\Sigma V^T$ *is the thin singular value decomposition of the tangent vector* Z *and* Q *is any element of* $O(k)$.

The proof of Theorem 2.2.1 can be found in §2.5.1 of [13]. It relies on the definition of the Grassmannian geodesic as a quotient of the orthogonal group quotient described in Equation 6 and a clever parametrization of the singular value decompositions of the matrix B. It should be clear that the computational formula for a geodesic on the Grassmannian is very closely related to the exponential map. In fact, the only difference is evaluation at the point t along that path, as well as the post-multiplication by the matrix Q. According to Edelman *et al.*, leaving off the matrix Q would give another representative of the same equivalence class, however, the tangent vectors to $\gamma(t)$ would have to be modified similarly [13]. Pseudocode for constructing a geodesic in the direction of a tangent vector can be found in Algorithm 2.

Algorithm 2 The geodesic emanating from $[X]$ in the direction Z.

function $\text{GEODESIC}([X], Z, t)$
  $U\Sigma V^T \leftarrow$ thin SVD(Z)
return $XV\cos(t\Sigma)\,V^T + U\sin(t\Sigma)\,V^T$

DEFINITION 2.2.3. *The inverse of the exponential map,* $\mathrm{Log}_X : \mathrm{Gr}(k,n) \to \mathrm{T}_X\mathrm{Gr}(k,n)$, *is referred to as the* ***logarithmic map****. It takes an element of the manifold to a point in the tangent space at* $[X]$*, and is computed as*

$$\mathrm{Log}_X([Y]) = U\Sigma V^T \tag{10}$$

where $U\Theta V^T$ *is the thin singular value decomposition of the matrix* $(I - XX^T)Y(X^TY)^{-1}$ *and* $\Sigma = \tan^{-1}(\Theta)$.

$\mathrm{Log}_X([Y])$ maps $[Y]$ into the tangent space of the manifold at $[X]$, $\mathrm{T}_X\mathrm{Gr}(k,n)$. The logarithmic map is only defined within a convex ball of $[X]$, the size of which is determined by the dimension of $\mathrm{Gr}(k,n)$. More details pertaining to the size of a convex ball on the Grassmannian can be found in [5]. There is no obvious closed form equation for the logarithmic map, but the algorithm described in Definition 2.2.3 can be verified as follows. Given $[X], [Y] \in \mathrm{Gr}(k,n)$, we seek to find $Z \in \mathrm{T}_X\mathrm{Gr}(k,n)$ such that $\mathrm{Exp}_X(Z) = Y$. Let $U\Sigma V^T$ be the thin singular value decomposition of Z so that U is an $n \times k$ slice of an orthonormal matrix, Σ is a $k \times k$ diagonal matrix, and V is a $k \times k$ orthonormal matrix. Then

$$Y = XV\cos(\Sigma)V^T + U\sin(\Sigma)V^T \tag{11}$$

as per the definition of the exponential map. We need to solve for $U, \Sigma,$ and V in order to reconstruct Z. Since Z is a tangent vector to $[X]$, $X^TZ = 0$. Thus we know that

$$X^TY = X^T(XV\cos(\Sigma)V^T + U\sin(\Sigma)V^T) \tag{12}$$

$$= V\cos(\Sigma)V^T \tag{13}$$

and that

$$(I - XX^T)Y = Y - XX^TY \tag{14}$$

$$= XV\cos(\Sigma)V^T + U\sin(\Sigma)V^T - XV\cos(\Sigma)V^T \tag{15}$$

$$= U\sin(\Sigma)V^T. \tag{16}$$

Leveraging these equalities and rearranging terms, we see that

$$(I - XX^T)Y(X^TY)^{-1} = U\sin(\Sigma)V^T\left(V\cos(\Sigma)V^T\right)^{-1} \tag{17}$$

$$= U\sin(\Sigma)\cos^{-1}(\Sigma)V^T \text{ (because } V \text{ is orthogonal)} \tag{18}$$

$$= U\tan(\Sigma)V^T. \tag{19}$$

So that finally if we compute $U\Theta V^T$ as the SVD of $(I - XX^T)Y(X^TY)^{-1}$, we can construct $Z = U\Sigma V^T$ with $\Sigma = \tan^{-1}(\Theta)$ as desired. Pseudocode for computing the logarithmic map in a computationally efficient way can be found in Algorithm 3.

Algorithm 3 The logarithmic map of $[Y]$ at $[X]$

function $\text{LOG}_X([X], [Y])$
  $U\Sigma V^T \leftarrow$ thin SVD $\left(Y(X^TY)^{-1} - X\right)$
  $\Theta \leftarrow \tan^{-1}(\Sigma)$
return $U\Theta V^T$

The last relevant map on the Grassmann manifold that we will discuss is parallel translation. The main idea is that we can easily move tangent vectors through paths on the manifold using our equation for geodesics. However, because the direction of the vector is constant while it travels and the manifold is not flat, the vector will likely not be tangent to the manifold in it's new location. This issue is corrected by removing the normal component from the vector after each infinitesimally small step on the manifold.

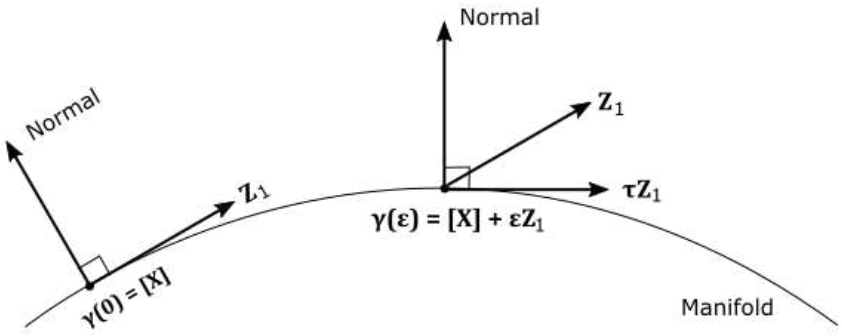


FIGURE 2.3. Illustration of parallel translation of a tangent vector Z along a curve $\gamma(t)$ on a Grassmann manifold. τZ represents the translated version of Z that is tangent to the point $\gamma(\epsilon) = [X] + \epsilon Z$.

THEOREM 2.2.2 (Parallel translation on the Grassmann manifold). *Let Z_1 and Z_2 be tangent vectors to a point $[X] \in \mathrm{Gr}(k, n)$. The parallel translation of Z_1, denoted τZ_1, along the geodesic in the direction of Z_2 is*

$$\tau Z_1(t) = \left(-XV \sin(t\Sigma)U^T + U \cos(t\Sigma)U^T + \left(I - UU^T\right)\right) Z_1 \tag{20}$$

where $U\Sigma V^T$ is the thin singular value decomposition of Z_2.

The proof of Theorem 2.2.2 can be found in §8.1.1 of [2]. A cartoon version of this process can be seen in Figure 2.3. In order to illustrate parallel transport in two dimensions, we set $Z_2 = Z_1$. That is two say, the tangent vector Z_1 will be moved along the geodesic $\gamma(t)$ in the direction it is already pointing, Z_1. The basepoint for Z_1 is the point $\gamma(0) = [X]$. After traveling an infinitesimally small distance ϵ along the geodesic, the difference between the vectors Z_1 and τZ_1 located at $\gamma(\epsilon) = [X] + \epsilon Z_1$is the removal of the normal component of Z_1. The pseudocode for computing the parallel translation of a tangent vector can be found in Algorithm 4, and simply reiterates Theorem 2.2.2. There is, however, a maximum distance that a tangent vector can be transported via this method because it relies on the

exponential and logarithmic maps. This distance is the maximum radius of a convex ball on a particular Grassmann manifold, and is a function of the dimension of the manifold. An explicit computation of this distance can be found in [5].

Algorithm 4 τZ_1, the parallel translation of Z_1 along the geodesic emanating from $[X]$ in the direction Z_2.

function ParTrans$([X], Z_1, Z_2, t)$
  Ensure $X^T X = I$
  $U\Sigma V^T \leftarrow$ thin SVD (Z_2)
return $\left(-XV\sin(t\Sigma)U^T + U\cos(t\Sigma)U^T + \left(I - UU^T\right)\right) Z_1$

2.2.3. Metrics and Similarity Measures. Since we use matrices to represent points on a Grassmannian, metrics need to be independent of our choice of coordinates and therefore orthogonally invariant.

Definition 2.2.4. *Let* $d : \mathrm{Gr}(k,n) \times \mathrm{Gr}(k,n) \to \mathbb{R}$ *be a metric. The metric,* d*, is said to be* ***orthogonally invariant*** *if for every* $[X], [Y] \in \mathrm{Gr}(k,n)$ *and every* $A \in O(n)$*,* $d([X],[Y]) = d([AX],[AY])$.

It has long been known that the principal angles between linear subspaces are orthogonally invariant, because they depend only on the relative position of the subspaces. Thus numerous distance metrics on Grassmannians have been developed as functions of principal angles [7].

Definition 2.2.5. *Let* $[X]$ *and* $[Y]$ *be subspaces of* $\mathbb{R}^n$ *with* $q = \min\{\dim([X]), \dim([Y])\}$. *The* ***principal angles*** $\theta_k \in [0, \pi/2]$ *between* $[X]$ *and* $[Y]$ *are defined for* $k = 1, 2, \ldots, q$ *by*

$$\cos\theta_k = \max_{v\in[X]} \max_{u\in[Y]} u^T v = u_k^T v_k, \tag{21}$$

subject to the constraints $\|u\| = \|v\| = 1$, *and* $u_j^T u_k = v_j^T v_k = 0$ *for* $j = 1, 2, \ldots k-1$. *The vectors* $\{u_1, u_2, \ldots u_q\}$ *and* $\{v_1, v_2, \ldots v_q\}$ *are the* ***principal vectors*** *of the pair of spaces.*

If Q_X and Q_Y are orthonormal bases for $[X]$ and $[Y]$, the principal angles and vectors between $[X]$ and $[Y]$ can be calculated by finding the thin singular value decomposition of $Q_X^T Q_Y$ [7]. Write the decomposition as $Q_X^T Q_Y = U\Sigma V^T$. The principal vectors of $[X]$ are the columns of the matrix $Q_X U$, the principal vectors of $[Y]$ are the columns of $Q_Y V$, and the principal angles between the spaces are the inverse cosines of the singular values, i.e. $\cos\theta_k = \sigma_k$. Note that if $A, B \in O(k)$, then the matrix $Q_X^T Q_Y$ has the singular values as $(Q_X A)^T (Q_Y B)$. Two metrics arising as functions of principal angles are relevant in this dissertation, the geodesic distance based on arc length and the projection Frobenius norm. A method for computing the principal angles between two subspaces of arbitrary subspace dimension can be found in Algorithm 5.

Algorithm 5 Principal angles separating $[X]$ and $[Y]$

function $\boldsymbol{\Theta}([X],[Y])$
  Ensure $X^T X = I$, $Y^T Y = I$
  $r \leftarrow \min\{\dim([X]), \dim([Y])\}$
  $U\Sigma V^T \leftarrow$ thin SVD $\left(X^T Y\right)$, such that $[\sigma_1, \ldots, \sigma_r]^T = diag(\Sigma)$
  for $i = 1, \ldots, r$ **do**
    if $\sigma_i < 1 \times 10^{-8}$ **then**
      $\theta_i \leftarrow \cos^{-1}(\sigma_i)$
    else
      $\theta_i \leftarrow \sqrt{2(1-\sigma_i)}$
return $[\theta_1, \ldots, \theta_r]^T$

Definition 2.2.6. *If* $[X], [Y] \in Gr(k, n)$, *then the* ***geodesic distance based on arc length*** *between the two is defined as*

$$d([X],[Y]) \doteq \|\boldsymbol{\Theta}\|_2, \tag{22}$$

where $\mathbf{\Theta}$ *is the* k*-dimensional vector of principal angles between* $[X]$ *and* $[Y]$*.*

This is the canonical metric on the Grassmann manifold in the sense that it is equivalent to the Euclidean metric in the tangent space of a single point on the Grassmannian. Pseudocode for computing the geodesic distance based on arc length can be found in Algorithm 6.

Algorithm 6 Geodesic distance based on arc length between $[X]$ and $[Y]$

function $d([X],[Y])$
  $\Theta \leftarrow \mathbf{\Theta}([X],[Y])$
return $\|\Theta\|_2$

DEFINITION 2.2.7. *Let* $[X],[Y] \in Gr(k,n)$*. The* ***projection Frobenius norm*** *is*

$$d_{pF}([X],[Y]) \doteq 2^{-\frac{1}{2}}\|XX^T - YY^T\|_F. \tag{23}$$

It is an elementary exercise to show that d_{pF} is an orthogonally invariant metric on $\mathrm{Gr}(k,n)$. The projection Frobenius norm arises from the identification of points in $\mathrm{Gr}(k,n)$ with $n \times n$ projection matrices of rank k. This distance can also be computed as the ℓ_2-norm of the vector of the sines of the principal angles between $[X]$ and $[Y]$. That is, $d_{pF}([X],[Y]) = \|\sin\mathbf{\Theta}\|_2$ where $\mathbf{\Theta}$ is the k-dimensional vector of principal angles between $[X]$ and $[Y]$[13]. It can be be shown that for $[X] \neq [Y]$, $d([X],[Y]) > d_{pF}([X],[Y])$, and that these metrics are asymptotically equivalent. Thus for points close together, $d([X],[Y]) \approx d_{pF}([X],[Y])$. Additionally, since both metrics are based on principal angles, distances on $\mathrm{Gr}(k,n)$ are bounded. A method for computing the Projection Frobenius norm between two subspaces can be found in Algorithm 7.

PROPOSITION 2.2.3. *For all* $[X],[Y] \in Gr(k,n)$*,*

$$d([X],[Y]) \leq (\pi/2)\sqrt{k} \quad \textit{and} \quad d_{pF}([X],[Y]) \leq \sqrt{k}. \tag{24}$$

Algorithm 7 Projection Frobenius norm between $[X]$ and $[Y]$

function $d_{pF}([X],[Y])$
  $\Theta \leftarrow \mathbf{\Theta}([X],[Y])$
return $\|\sin(\Theta)\|_2$

PROOF. Let $[X],[Y] \in \mathrm{Gr}(k,n)$. Then $k = \min\{\dim([X]),\dim([Y])\}$, and $\theta_i \in [0,\pi/2]$ for $i = 1,2,\ldots k$. Thus we have

$$d([X],[Y]) = \|\mathbf{\Theta}\|_2 = \sqrt{\sum_{i=1}^{k}\theta_i^2} \leq \sqrt{\sum_{i=1}^{k}(\pi/2)^2} = \sqrt{k(\pi/2)^2} = (\pi/2)\sqrt{k}, \quad \text{and} \tag{25}$$

$$d_{pF}([X],[Y]) = \|\sin\mathbf{\Theta}\|_2 = \sqrt{\sum_{i=1}^{k}\sin^2(\theta_i)} \leq \sqrt{\sum_{i=1}^{k}1} = \sqrt{k} \quad \text{as desired.} \tag{26}$$

□

2.3. Flag Manifolds and Schubert Varieties

The Grassmann manifold will be the main setting for data analysis in this dissertation. However some applications require us to generalize or restrict our view to objects beyond a set of fixed-dimensional subspaces. One overt advantage to representing data as points on a Grassmannian is that natural linear variation is included in each representative. Sometimes though, different dimensions of a subspace are more important than others. This weighting suggests a connection to flag manifolds.

On the other hand, restricting analysis to a subset of points on a Grassmannian may also provide a more accurate picture of an application if not all subspaces are within the domain of the problem. There are many possible restrictions, with one of the simplest being a collection of subspaces that overlap with a distinguished subspace or sequence of subspaces in some prescribed dimensions. This type of constraint characterizes a Schubert variety, and will see use in the applications that follow.

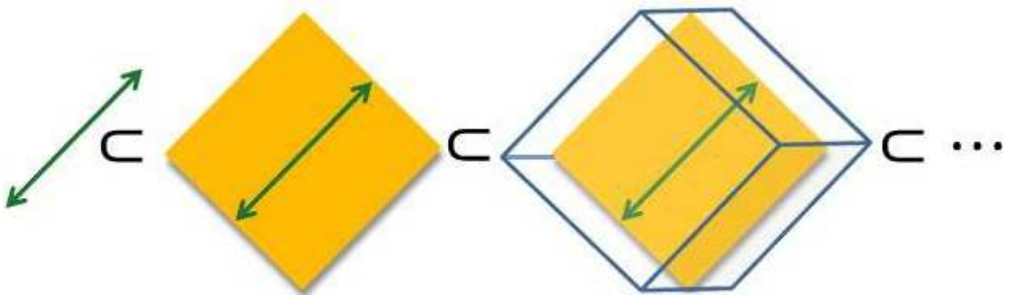


FIGURE 2.4. Illustration of a nested sequence of subspaces that would correspond to a point on the flag manifold $\mathrm{FL}(n; [1, 2, 3, \ldots])$.

2.3.1. FLAG MANIFOLDS. We begin by describing flags, because they are prerequisite for understanding Schubert varieties.

DEFINITION 2.3.1. *Let* $\boldsymbol{q} = \{q_0, q_1, q_2, \ldots, q_M, q_{M+1}\}$ *be a strictly increasing set of integers such that* $0 = q_0 < q_1 < q_2 < \ldots < q_M < q_{M+1} = n$. *A* ***flag***, $\boldsymbol{F}$, *in* $\mathbb{R}^n$ *of type* $\boldsymbol{q}$ *is a nested sequence of subspaces*

$$\boldsymbol{F} \doteq \{0\} \subset [S_1] \subset [S_2] \subset \cdots \subset [S_M] \subset [S_{M+1}] = \mathbb{R}^n \tag{27}$$

such that $dim([S_i]) = q_i$. *The* ***flag manifold***, $FL(n; \boldsymbol{q})$, *is the manifold whose points correspond to all flags of type* $\boldsymbol{q}$.

If the signature of the flag, $\boldsymbol{q}$, includes all of the natural numbers between 0 and n, the resulting flags are referred to as full flags. Figure 2.4 shows an illustration of the nested structure present in the low-dimensional elements of a full flag. If the set of integers that defines the flag includes only one integer other than zero and the ambient dimension, $\boldsymbol{q} = \{0, q_1, n\}$, then the flag manifold $\mathrm{FL}(n; \boldsymbol{q})$ is equivalent to the Grassmann manifold $\mathrm{Gr}(q_1, n)$. Alternatively, the flag manifold can be thought of as a submanifold of the product of the Grassmann manifolds $\mathrm{Gr}(q_i - q_{i-1}, n)$ for $i = 1 \ldots M + 1$.

The idea that the flag manifold is a generalization of the Grassmann manifold and the structure of the elements in the set described in Equation 2 hint at a natural quotient space interpretation for a flag manifold. Let $\boldsymbol{q} = \{0 = q_0, q_1, q_2, \ldots, q_M, q_{M+1} = n\}$. Then we have

$$\mathrm{FL}(n; \boldsymbol{q}) \simeq \mathrm{O}(n) \Big/ \mathrm{O}(q_1) \times \mathrm{O}(q_2 - q_1) \times \cdots \times \mathrm{O}(n - q_M) \,, \tag{28}$$

suggesting that $\dim(\mathrm{FL}(n; \boldsymbol{q})) = \sum_{i=1}^{M} q_i(q_{i+1} - q_i)$. In particular, a full flag has dimension $n(n-1)/2$. We will not deal with distances between points on flag manifolds, but it is simple to see how Grassmannian distances can be used as the basis for metrics in that setting. The sum of the Grassmann distances between elements of the same size, will generate a metric between points on a flag manifold. For more explanation of flag manifolds and their geometry, refer to [48].

2.3.2. Schubert Varieties. Schubert varieties carve out a subset of points on a Grassmann manifold that all intersect with a distinguished sequence of subspaces. The most common setting for discussing Schubert varieties is enumerative geometry, or intersection theory. Early questions in the field were things like, "How many lines intersect four given lines in $\mathbb{R}^3$?" A thorough treatment of Schubert varieties from the enumerative geometry perspective can be found in [18]. Schubert varieties have garnered much less interest as a setting for data analysis, however some advances have been made recently [60, 72]. There are several equivalent definitions, but we find the following to be the most intuitive with respect to our description of the Grassmann manifold.

Definition 2.3.2. *Fix a Grassmannian* $\mathrm{Gr}(k, n)$ *and a flag* $\boldsymbol{F} = \{0\} \subset [W_1] \subset [W_2] \subset \cdots \subset [W_M] \subset \mathbb{R}^n$ *with signature* $\boldsymbol{q} = \{0 = q_0, q_1, q_2, \ldots, q_M, q_{M+1} = n\}$. *Given an increasing sequence of integers* $\boldsymbol{\alpha} = \{0 = \alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_M \leq \alpha_{M+1} = k\}$ *where* $\alpha_i \leq \min\{q_i, k\}$, *the*

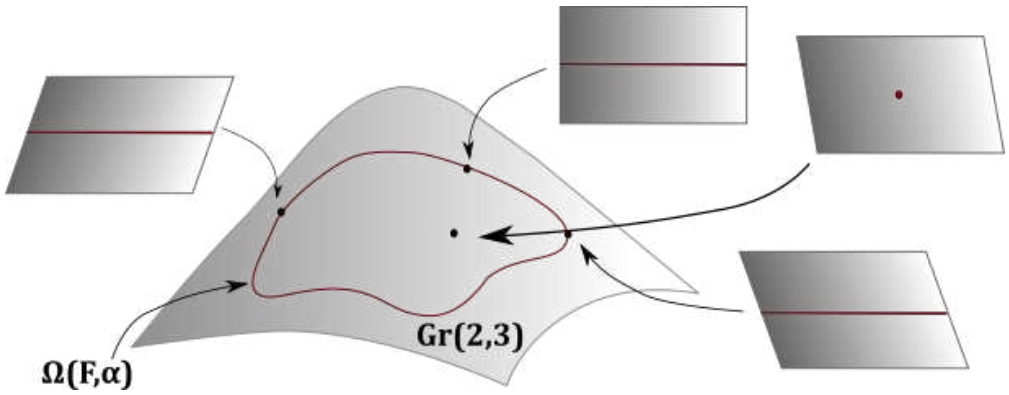


FIGURE 2.5. The grey surface represents the Grassmannian of 2-planes in $\mathbb{R}^3$, $\mathrm{Gr}(2,3)$. The closed, red curve represents the Schubert variety of planes that fully contain the span of the second canonical basis vector $[e_2]$. That is $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ for $\boldsymbol{F} = [e_2]$ and $\boldsymbol{\alpha} = 1$. Points not on the closed curve intersect $[e_2]$ as points, as demonstrated by the plane in the upper right.

associated ***Schubert variety****,* $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$*, is defined as*

$$\Omega(\boldsymbol{F}, \boldsymbol{\alpha}) \doteq \{[X] \in \mathrm{Gr}(k,n) \mid \dim([X] \cap [W_i]) \geq \alpha_i\}\,. \tag{29}$$

The dimension of this Schubert variety is equal to $\sum_{i=1}^{M}(q_i - \alpha_i)$. An example illustration of a simple Schubert variety can be seen in Figure 2.5. As with flag manifolds, we are not presently interested in computing distances between points on a Schubert variety. Since the varieties carve out a subset of points on a Grassmannian, we can use Grassmann distances if we want to measure similarity without respecting the geometry of the Schubert variety. However, we care about the distance between a given point on a Grassmann manifold and the Schubert variety itself.

In the case where the flag that defines the Schubert variety contains a single subspace, i.e. $\boldsymbol{F} = \{0\} \subset W \subset \mathbb{R}^n$, the distance between a point and the variety has been described by Schwickerath [60].

PROPOSITION 2.3.1. *Let* $\boldsymbol{F} = \{0\} \subset W \subset \mathbb{R}^n$ *and* $\boldsymbol{\alpha} = \{0, \alpha, k\}$. *Given the Schubert variety* $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ *and a point* $[X] \in Gr(k, n)$, *the principal angles between* $[X]$ *and any point* $[Y] \in \Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ *are bounded below by*

$$[\mathbf{0}_{k-\alpha}, \boldsymbol{\Theta}_{1,\ldots,\alpha}([W], [X])] \leq \boldsymbol{\Theta}_{1,\ldots,k}([Y], [X]), \tag{30}$$

where Θ *is the vector of principal angles separating* $[X]$ *and* $[Y]$ *as defined in Algorithms 5. In other words, each principal angle on the left-hand side is less than or equal to the corresponding principal angle on the right-hand side.*

The proof of Proposition 2.3.1 can be found in [60], however the point that realizes this lower bound can be constructed as follows. Given that W, X are orthonormal bases for $[W], [X]$ respectively, let $U\Sigma V^T = W^T X$ be a singular value decomposition. The left principal vectors are then written as Wu_i for $i = 1, \ldots, k$ where u_i is the ith column of U. Let $[Z_\alpha] \doteq [Wu_1 | Wu_2 | \cdots | Wu_\alpha]$, and let $[Z_{k-\alpha}]$ be any set of $k - \alpha$ orthonormal vectors from the intersection of $[X]$ and the orthogonal complement of $[Z_\alpha]$. The lower bound on the principal angles is then achieved by any point $[Z] \in \Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ where $[Z] = [Z_\alpha | Z_{k-\alpha}]$. It should be clear that there is often not a unique point that achieves this lower bound. It may be less obvious that $[Z_\alpha]$, the portion of $[Z]$ that intersects $[W]$, may also not be unique. Colloquially, this is because the Schubert variety is not a convex set on the Grassmann manifold and thus projection is not unique. Further details can be found in [60, 72].

2.4. Grassmannian Averages

For the applications in this dissertation we are concerned with grouping subspaces based on similarity. In some contexts a nearest neighbors comparison can give a complete picture

of the interconnections between points, but when the data is noisy it is possible that relationships are established because of noise rather than signal. Examples of such a scenario can be found in [45]. For this reason, subspace averages are often used to identify common information in related subsets of the data. As in Euclidean space there are numerous ways to average points on Grassmannians; each with benefits and drawbacks. This dissertation will focus primarily on the flag mean, however for context we now explain the common methods for computing subspace averages and attempt to build intuition about their properties.

2.4.1. The Karcher mean. The Karcher mean, $\boldsymbol{\mu}_K$, is the intrinsic or canonical mean on the Grassmann manifold. It is the point that minimizes the mean squared error using the canonical metric,

$$\boldsymbol{\mu}_K = \underset{[\mu]\in \mathrm{Gr}(k,n)}{\arg\min} \sum_{i=1}^{P} d([X_i],[\mu])^2. \tag{31}$$

The Karcher mean is most commonly found by using an iterative algorithm like Newton's method or first-order gradient descent [1, 5]. These algorithms exploit the matrix Exp and Log maps to move the data to and from the tangent space of a single point at each step. A unique optimal solution is guaranteed for data that lives within a convex ball on the Grassmann manifold, but in practice not all data sets satisfy this criterion [5, 30, 47]. Using the geodesic distance based on arc length, Proposition 2.2.3 shows that the maximum distance between two points on $\mathrm{Gr}(k,n)$ is $(\pi/2)\sqrt{k}$. As illustrated by Begelfor and Werman the convexity radius is $\pi/4$ [5]. This means that if the point cloud being averaged has a radius greater than $\pi/4$ the logarithmic map is no longer bijective, and the Karcher mean is no longer unique. Pseudocode for a steepest descent method for computing the Karcher mean can be seen in Algorithm 8. This method is adapted from [5].

Algorithm 8 The Karcher mean of $\{[X_1], [X_2], \ldots, [X_P]\}$ given error tolerance ϵ

function $\boldsymbol{\mu}_K(\{[X_1], [X_2], \ldots, [X_P]\}, \epsilon)$
  $\mu_1 \leftarrow X_1$
  while $d([\mu_i], [\mu_{i+1}]) > \epsilon$ **do**
    $Z \leftarrow \frac{1}{P} \sum_{j=1}^{P} \mathrm{Log}_{\mu_i}(X_j)$
    $\mu_{i+1} \leftarrow \mathrm{Exp}_{\mu_i}(Z)$
return μ_{i+1}

2.4.2. THE L_2-MEDIAN. The L_2-median, $\boldsymbol{\mu}_{L_2}$, is one of many ways of generalizing the median for 1-dimensional data into higher dimensions. It is referred to by many names [11, 23, 63] including the spatial median, the geometric median, the mediancentre, and confusingly the L_1-median. By any name, the L_2-median is the point that minimizes the sum of the distances to the sample points, rather than the sum of the squares of the distances. For subspace data it solves

$$\boldsymbol{\mu}_{L_2} = \underset{[\mu] \in \mathrm{Gr}(k,n)}{\arg\min} \sum_{i=1}^{P} d([X_i], [\mu]), \tag{32}$$

where again $d([X_i], [\mu])$ is the geodesic distance based on arc length. As a direct generalization of the median for 1-dimensional data, the L_2-median is robust to outliers [11]. That is to say, if the data being averaged comes from multiple underlying processes, $\boldsymbol{\mu}_{L_2}$ will better represent the dominant process rather than the entire set of data. This is in contrast to the behavior of the Karcher mean, which represents the center of mass.

Methods for finding $\boldsymbol{\mu}_{L_2}$ also take advantage of the matrix exponential and logarithmic maps, and thus fall prey to the same uniqueness condition as the Karcher mean. One such method comes from Fletcher *et al.*, and adapts the Weiszfeld algorithm to Riemannian manifolds [17]. Pseudocode for this method can be found in Algorithm 9.

One distinction between the Karcher mean and the L_2-median is that the latter is robust to outliers. A common method for measuring the robustness from the statistics literature is

Algorithm 9 The L_2-median of $\{[X_1], [X_2], \ldots, [X_P]\}$ given error tolerance ϵ

function $\boldsymbol{\mu}_{L_2}(\{[X_1], [X_2], \ldots, [X_P]\}, \epsilon)$
  $\mu_1 \leftarrow X_1$
  while $d([\mu_i], [\mu_{i+1}]) > \epsilon$ **do**
    $Z \leftarrow \sum_{j=1}^{P} \frac{\mathrm{Log}_{\mu_i}([X_j])}{d([\mu_i],[X_j])} * \left(\sum_{i=1}^{P} \frac{1}{d([\mu_i],[X_j])}\right)^{-1}$
    $\mu_{i+1} \leftarrow \mathrm{Exp}_{\mu_i}(Z)$
  return μ_{i+1}

the finite sample breakdown point. Without giving an explicit definition, the finite sample breakdown point is the fraction of samples that can be corrupted to infinity before the average is corrupted to infinity as well. The Grassmann manifold is compact, and all metrics on it are bounded, so this point cannot be computed directly. However, when the cost function of the L_2-median is used on an unbounded manifold the breakdown point is 0.5, meaning that half the data can be corrupted arbitrarily before the L_2-median is corrupted [17]. On the other hand, when the cost function of the Karcher mean is evaluated on an unbounded manifold, the breakdown point is 0. If any data point is pulled infinitely away, the mean will be pulled with it. For more information on robust statistics, refer to Peter Huber's book [26].

2.4.3. The flag mean. In many applications, it can be natural and advantageous to represent aspects of data through subspaces lying in a fixed ambient space that are of differing dimensions. In such applications, a set of subspaces live naturally on a collection of Grassmann manifolds rather than on a single Grassmann manifold. Suppose that $[X] \in \mathrm{Gr}(k_1, n)$ and $[Y] \in \mathrm{Gr}(k_2, n)$ for $k_1 < k_2$. As illustrated in Bjork and Golub's foundational paper [7], there will be p_1 principal angles between $[X]$ and $[Y]$ and we can redefine $d_{pF}([X], [Y])$ as the ℓ_2-norm of the vector of the sines of the k_1 principal angles between $[X]$ and $[Y]$. Note that d_{pF} is no longer a metric due to the possibility of $d_{pF}([X], [Y]) = 0$ while $[X] \neq [Y]$ (for instance, if $[X]$ is a proper subspace of $[Y]$).

The flag mean, denoted $\boldsymbol{\mu}_{pF}$, is a nested sequence of subspaces that is central to a subspace point cloud in the sense that the kth subspace within the flag is the best k-dimensional representation of the data with respect to a cost function based on the projection Frobenius norm. Let $\{[X_i]\}_{i=1}^P$ be a finite collection of subspaces of $\mathbb{R}^n$ such that $X_i^T X_i = I$. Let $\widetilde{Q} = \{q_1, \ldots, q_P\}$ be a collection of positive integers, and suppose that $\dim([X_i]) = q_i$ for $i = 1 \ldots P$. We can consider $\{[X_i]\}_{i=1}^P$ to be a point cloud in the disjoint union of a set of Grassmannians, $\coprod_{\widetilde{Q}} \mathrm{Gr}(q_i, n)$.

For these subspaces we wish to find the 1-dimensional subspace $[u^{(1)}] \in \mathrm{Gr}(1, n)$ that minimizes the sum of the squares of projection Frobenius norms between itself and $[X_i]$ for $i = 1 \ldots P$. The projection Frobenius norm loses its distinction as a metric when it is used to compare points that do not live on the same manifold. However, it still measures the similarity between the objects. Metrics exist in the scenario where points live on just two Grassmann manifolds, but even in this case the similarity measures are more useful than the actual metrics. This topic is discussed at length in [60, 72]. Thus we aim to solve

$$\begin{aligned} &\underset{[u^{(1)}] \in \mathrm{Gr}(1,n)}{\arg\min} \quad \sum_{i=1}^{P} d_{pF}([u^{(1)}], [X_i])^2 \\ &\text{subject to} \quad u^{(1)T} u^{(1)} = 1. \end{aligned} \tag{33}$$

After finding the optimal $[u^{(1)}]$, the problem is extended to find a sequence of 1-dimensional subspaces that optimize Equation 33 with additional constraints. By solving

$$\begin{aligned} &\underset{[u^{(j)}] \in \mathrm{Gr}(1,n)}{\arg\min} \quad \sum_{i=1}^{P} d_{pF}([u^{(j)}], [X_i])^2 \\ &\text{subject to} \quad u^{(j)T} u^{(j)} = 1 \\ &\qquad\qquad\quad u^{(j)T} u^{(k)} = 0 \qquad \text{for } k < j, \end{aligned} \tag{34}$$

it is possible to find r ordered 1-dimensional subspaces, $\{[u^{(1)}], [u^{(2)}], \ldots, [u^{(r)}]\}$, where r is the dimension of the span of $\cup_{i=1}^{P}[X_i]$. These subspaces are then central to the collection of points $\{[X_i]\}_{i=1}^{P}$. From this sequence of mutually orthogonal vectors, the flag mean is defined explicitly as

$$\begin{aligned} \boldsymbol{\mu}_{pF} = \quad & \operatorname{span}\{u^{(1)}\} \subset \operatorname{span}\{u^{(1)}, u^{(2)}\} \subset \\ & \ldots \subset \operatorname{span}\{u^{(1)}, \ldots, u^{(r)}\}. \end{aligned} \tag{35}$$

While the subspaces $\{[u^{(1)}], [u^{(2)}], \ldots, [u^{(r)}]\}$ are derived iteratively, they can actually be computed analytically. It has been shown that $\{[u^{(1)}], [u^{(2)}], \ldots, [u^{(r)}]\}$ can be computed as the left singular vectors of the matrix $\mathbf{X} = [X_1|X_2|\ldots|X_P]$, where X_i is an orthonormal basis for $[X_i]$ [12]. The pseudocode for computing the flag mean can be found in Algorithm 10. More recently, Santamaria *et al.* have shown that for a given set of data, there is an optimal subspace dimensions that can be chosen from $\boldsymbol{\mu}_{pF}$ to minimize the mean squared error of a slightly modified cost function [57].

Algorithm 10 The flag mean of $\{[X_1], \ldots [X_P]\}$

function $\boldsymbol{\mu}_{pF}(\{[X_1], \ldots [X_P]\})$
  Ensure $X_i^T X_i = I$ for $i = 1, \ldots, P$
  $\mathbf{X} \leftarrow [X_1|X_2|\ldots|X_P]$
  $r \leftarrow \dim\left(\operatorname{span}\left\{\cup_{i=1}^{P}[X_i]\right\}\right)$
  $U\Sigma V^T \leftarrow$ thin SVD($\mathbf{X}$), such that $U = [u^{(1)}|u^{(2)}|\ldots|u^{(r)}]$
return $\left\{\left[u^{(1)}\right], \left[u^{(1)}|u^{(2)}\right], \ldots, \left[u^{(1)}|\ldots|u^{(r)}\right]\right\}$

CHAPTER 3

SCHUBERT VARIETY CONSTRAINED OPTIMIZATION

3.1. Introduction

Many of the applications discussed in this dissertation represent data as points on a Grassmann manifold. Finding each element of the flag mean for these points can be thought of as an unconstrained optimization problem, because the only constraints observed are those which keep our solutions on the appropriate manifold. In Euclidean space, the simplest constraints to impose on optimization problems are usually linear, that is, they require the solution to live within some linear subspace of the ambient space. Points in Euclidean space are 0-dimensional objects, thus overlap with a linear subspace and containment in a linear subspace are equivalent. On Grassmann manifolds however the two relations can be unique, and Schubert varieties are used to describe those sets of points. Thus we would like to impose a constraint on our Grassmannian optimization that requires the solution to be an element of a Schubert variety as a generalization of the linear constraints used in Euclidean space.

There are ready-made applications of this type of constrained optimization in geometric analysis and signal processing. In geometric multi-resolution analysis as described by Allard *et al.*, the goal is to approximate a non-linear low-dimensional manifold structure of a point cloud in high-dimensional Euclidean space by fitting piecewise affine spaces of appropriate dimension to the data [4]. The problem of identifying intrinsic dimension of data and approximating its structure has been recognized as important in numerous applications such as the analysis of sounds, images, gene arrays, and EEG signals [4]. Incorporating Schubert variety constraints into the solution of Allard *et al.* would allow researchers to restrict

approximations of data to agree with linear subspaces that represent physical or domain specific knowledge of problems that might not be well captured in observed data.

In related signal processing applications, Hagege and Francos, and Yavo *et al.* try to discover low-dimensional linear embeddings of images under geometric deformations [22, 71]. The former authors investigate the noise-free case and the latter team tackles the problem in the presence of contaminated observations. In both instances, the authors look to approximate a true, underlying, linear space from observed data. However, it is possible that a theoretical linear space exists that should contain the observations. The prior work by Hagege and Francos suggests that such an oracle subspace does, in fact, exist [21]. In this case, it would make sense to incorporate information from both the observed samples and the oracle subspace into the low-dimensional embedding. This is an ideal example of an application for Schubert variety constrained averaging as we will describe it.

3.2. General Problem

Given a flag $\boldsymbol{F} = \{0\} \subset [W_1] \subset \cdots \subset [W_M] \subset \mathbb{R}^n$ such that $\dim([W_i]) = q_i$ as prescribed by the signature $\boldsymbol{q} = \{0 = q_0, q_1, \ldots, q_M, q_{M+1} = n\}$, let $\boldsymbol{\alpha} = \{0 = \alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_M \leq \alpha_{M+1} = k\}$ be a sequence of integers and $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ be the associated Schubert variety. The generic goal of Schubert variety constrained optimization is then to find a point on a Grassmann manifold $[X] \in \mathrm{Gr}(k, n)$ that minimizes

$$\begin{aligned} &\underset{[X]\in\mathrm{Gr}(k,n)}{\arg\min} && f([X]) \\ &\text{subject to} && \dim\left([X] \cap [W_j]\right) \geq \alpha_j, \end{aligned} \tag{36}$$

for $j = 1 \ldots M$ and some differentiable function $f : \mathrm{Gr}(k, n) \to \mathbb{R}$.

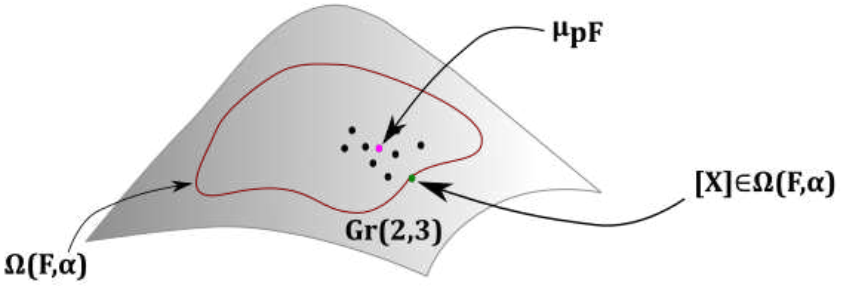


FIGURE 3.1. A schematic representation of Schubert variety constrained averaging on $\mathrm{Gr}(2,3)$. The closed curve represents a Schubert variety. The black dots represent the data to be averaged, $[Q_1], [Q_2], \ldots, [Q_P] \in \mathrm{Gr}(2,3)$. The fuchsia dot, $\boldsymbol{\mu}_{pF}$, marks the flag mean of the data, and the green dot, $[X]$, indicates the point on the Schubert variety that minimizes the cost function.

This problem is likely too broad to solve in one shot, therefore we restrict our attention to a sub-problem. First, we will be minimizing a class of functions that can be written as the sum of the squared distances between a collection of observed data on a Grassmann manifold, $[Q_1], [Q_2], \ldots, [Q_P] \in \mathrm{Gr}(k,n)$, and the point of interest. That is, $f([X]) \doteq \sum_{i=1}^{P} d([X],[Q_i])^2$ for some distance function d. Minimizing the f will result in an average of the observed data restricted to the Schubert variety. Second, we will choose a flag that consists of just a single subspace, $\boldsymbol{F} = \{0\} \subset W \subset \mathbb{R}^n$, with $\dim(W) = j$. It will be easy to generalize our approach for more general flags following the same method. For our initial investigation we will further restrict the distance function to be the projection Frobenius norm, $d_{pF}([X],[Q_i]) = 2^{-\frac{1}{2}}\|XX^T - Q_iQ_i^T\|_F$ so the optimization problem from Equation 36 is modified to become

$$\begin{aligned} &\underset{[X]\in\mathrm{Gr}(k,n)}{\arg\min} && \frac{1}{P}\sum_{i=1}^{P} d_{pF}([X],[Q_i])^2 \\ &\text{subject to} && \dim\left([X]\cap[W]\right) \geq \alpha. \end{aligned} \tag{37}$$

Figure 3.1 depicts a schematic of Schubert variety constrained averaging on $\mathrm{Gr}(2,3)$. The

fuchsia dot, $\boldsymbol{\mu}_{pF}$, indicates the unconstrained average, or the 2-dimensional element of the flag mean, while green point, $[X]$, is the sought after minimizer that lives on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$.

3.2.1. Gradient descent on Grassmannians. In Algorithm 8 we presented a simple steepest descent method for solving the Karcher mean optimization problem using the tools developed in Section 2.2. In this limited case, the gradient was determined by averaging tangent space representations of the data, and stepping along the geodesic in the direction of that average. For other real functions of points on the Grassmannian, the gradient can be computed as follows.

Definition 3.2.1. *Let $f : \mathrm{Gr}(k, n) \to \mathbb{R}$ be a differentiable function and define $f_X \in \mathbb{R}^{n \times k}$ to be the matrix of partial derivatives of f with respect to the elements of the matrix X, i.e.,*

$$(f_X)_{ij} = \frac{\partial f}{\partial X_{ij}}. \tag{38}$$

*The **gradient of the real-valued function** f **at** $[X]$ is then defined to be the tangent vector ∇f such that*

$$tr(f_X^T Z) = d(\nabla f, Z) \doteq tr\left((\nabla f)^T Z\right) \tag{39}$$

for all tangent vectors $Z \in \mathrm{T}_X\mathrm{Gr}(k, n)$.

The second definitional equality in Equation 39 comes from the fact that the geodesic distance based on arc length is equivalent to the Euclidean distance in the tangent space. Solving for ∇f such that $X^T \nabla f = 0$ we get the formula, $\nabla f = (I - XX^T) f_X$, that works

for any real-valued, differentiable function. The formula is presented as pseudocode in Algorithm 11. This generic gradient expression will allow us to apply the Riemannian conjugate gradient method to our optimization problem, which boasts superlinear convergence rates.

Algorithm 11 The gradient of f at $[X]$

function $\nabla f([X])$
  Ensure $X^T X = I$
  $(f_X)_{ij} \leftarrow \frac{\partial f}{\partial X_{ij}}$
return $f_X - XX^T f_X$

3.2.2. Conjugate gradient. We follow the method for conjugate gradient modified for the Grassmann manifold that was detailed by Edelman, Arias, and Smith [13]. Colloquially, this method minimizes a sequence of line searches along geodesics on the manifold where search directions are determined using only gradient information (which in turn is used to approximate information about the Hessian). At the $(k-1)$st step of the algorithm, we step from $[X_{k-1}]$ to $[X_k]$ by minimizing the objective function $f : \mathrm{Gr}(k,n) \to \mathbb{R}$ along the geodesic $\gamma_{k-1}(t)$ emanating from $[X_{k-1}]$ in the tangent direction Z_{k-1}. The next search direction is then chosen to be a weighted combination of the gradient at the new location and the translated version of the old search direction. If Z_{k-1} is the search direction for the geodesic originating at $[X_{k-1}]$, then the updated search direction will be

$$Z_k = \nabla f([X_k]) + \alpha_k \tau Z_{k-1} \tag{40}$$

for some choice of α_k, where τZ_{k-1} is the parallel translated version of Z_{k-1} as described in Theorem 2.2.2. For our implementation of conjugate gradient we use the Polak-Ribière formula,

$$\alpha_k = \frac{\langle Z_k, Z_k - \tau Z_{k-1} \rangle_{[X_k]}}{\langle Z_{k-1}, Z_{k-1} \rangle_{[X_{k-1}]}}, \tag{41}$$

which approximates the second derivative using the difference of gradients. Pseudocode for conjugate gradient on the Grassmann manifold can be seen in Algorithm 12.

Algorithm 12 Conjugate gradient on the Grassmann manifold

function ConjGrad($f, [X_0]$)
  Ensure $X_0^T X_0 = I$
  $Y_0 \leftarrow \nabla f([X_0])$
  $Z_0 \leftarrow -Y_0$
  $\gamma_0(t) \leftarrow \text{Geodesic}([X_0], Z_0, t)$
  $t_0 \leftarrow \arg\min f(\gamma_0(t))$ ▷ Minimize using any line search algorithm
  while $t_k \neq 0$ **do**
    $[X_{k+1}] \leftarrow \gamma_k(t_k)$
    $Y_{k+1} \leftarrow \nabla f([X_{k+1}])$
    $\tau Z_k \leftarrow \text{ParTrans}([X_k], Z_k, Z_k, t_k)$
    $\tau Y_k \leftarrow \text{ParTrans}([X_k], Y_k, Z_k, t_k)$
    $\alpha_k \leftarrow {}^{\langle Y_{k+1}, Y_{k+1} - \tau Y_k \rangle_{[X_{k+1}]}}/_{\langle Y_k, Y_k \rangle_{[X_k]}}$ ▷ Polak-Ribière formula
    $Z_{k+1} \leftarrow -Y_{k+1} + \alpha_k \tau Z_k$
    $\gamma_{k+1}(t) \leftarrow \text{Geodesic}([X_{k+1}], Z_{k+1}, t)$
    $t_{k+1} \leftarrow \arg\min f(\gamma_{k+1}(t))$
    $k \leftarrow k + 1$
  return $\gamma_k(t_k)$

3.2.3. The Schubert variety constraint as a penalty function. The descent method characterized in Algorithm 12 requires that we be able to compute a gradient at each iterate in order to step forward. In general circumstances this is not a restrictive requirement as we have straightforward methods for computing tangent vectors on the Grassmann manifold. However, restricting the domain of the optimization to a Schubert variety significantly complicates this process, because Schubert varieties are singular. Since more than one tangent space can computed at these singular points, the direction of descent cannot be easily chosen. To the best of our knowledge, there is no closed form method for computing tangent vectors that point strictly within a Schubert variety. Since we have no general form for the tangent vectors at every point on the Schubert variety, we look to add the intersection constraints into the optimization as penalty terms.

For a thorough treatment of penalty methods, please consult [8]. The general idea is that constraints in the optimization problem are moved into the objective function via a penalty function which increases the cost of a solution when the constraint is violated. The amount that violating a constraint affects the cost is controlled via a penalty parameter. The penalty method employed here starts with an infeasible solution (a point that is not on the Schubert variety), and iteratively increases the weight of the penalty parameter until the constraint is met. At each iteration the penalty parameter is fixed and the problem is minimized with respect to the points on the Grassmannian. This process creates a sequence of iterates with non-decreasing costs. Chong and Zak show that under convexity constraints for the feasible region, the upper bound of these costs is the minimum of the constrained problem [8]. Pseudocode for the penalty method can be found in Algorithm 13.

Algorithm 13 Penalty method for constrained optimization

function PENALTYMETHOD($f, P, [X_0], \epsilon$)
  Select $\eta > 1$ ▷ Set growth parameter
  Select λ_0 ▷ Set initial value of penalty parameter
  Ensure $[X_0] \notin \Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ ▷ Choose infeasible starting point
  $[X_1] \leftarrow \arg\min f([X]) + \lambda_0 P([X])$
  while $d([X_{k-1}], [X_k]) > \epsilon$ **do**
    $\lambda_k \leftarrow \eta \lambda_{k-1}$
    $[X_k] \leftarrow \arg\min f([X]) + \lambda_{k-1} P([X])$
  return $[X_k]$

The optimization problem in Equation 37 can be rewritten using this standard penalty method to include the Schubert variety constraint in the objective function. To do so, we first parametrize all points on $\Omega(\boldsymbol{W}, \boldsymbol{\alpha})$ using a fixed basis $\hat{W} \in O(n)$ and a coefficient matrix $A \in \mathbb{R}^{n \times k}$. Choose

$$[\hat{W}] \doteq [w_1|w_2|\cdots|w_j|w_1^\perp|w_2^\perp|\cdots|w_{n-j}^\perp] = [W|W^\perp] \tag{42}$$

where the columns of $W^{\perp}$ complete W to a basis for $\mathbb{R}^n$ and $\hat{W}$ is orthonormal. In a slight abuse of notation, let $O(n \times k)$ indicate the set of matrices with n rows and k columns whose columns are mutually orthogonal and unit length. Define

$$\mathcal{A} \doteq \left\{ A \in \mathbb{R}^{n \times k} \;\middle|\; A = \left[\begin{array}{c|c} B & \\ \hline C & \mathrm{D} \end{array}\right],\ B \in O(j \times \alpha),\ C = \mathbf{0},\ D \in O(n \times (k-\alpha)) \right\} \tag{43}$$

to be the set of all matrices that can be written as the span of orthonormal columns whose lower left block is all zeros. In the block structure defined, B is a $j \times \alpha$ slice of an orthonormal matrix, C is an $(n-j) \times \alpha$ matrix of zeros, and D is an $n \times (k-\alpha)$ slice of an orthonormal matrix such that it's columns are orthogonal to the first α columns of A. Thus for all $[X] \in \Omega(\boldsymbol{W}, \boldsymbol{\alpha})$, there exists a $A \in \mathcal{A}$ such that $[X] = [\hat{W}A]$.

Now by replacing $[X]$ in Equation 37 with $[\hat{W}A]$ we define $f : \mathrm{Gr}(k,n) \times \mathbb{R}_{>0} \to \mathbb{R}$ as a function of $[A]$ and the penalty, λ. That is,

$$f([A], \lambda) = \frac{1}{M} \sum_{i=1}^{M} d_{pF}([\hat{W}A], [Q_i])^2 + \lambda \left\| \begin{bmatrix} 0_L & I_L \end{bmatrix} A \begin{bmatrix} I_R \\ 0_R \end{bmatrix} \right\|_F^2 \tag{44}$$

where I_L and I_R are both identity matrices such that $I_L \in \mathbb{R}^{(n-j)\times(n-j)}$, and $I_R \in \mathbb{R}^{(k-\alpha)\times(k-\alpha)}$. The second term in Equation 44 is equivalent to the the penalty times the squared Frobenius norm of the lower left block of $[A]$, that is, $\lambda \|C\|_F^2$ and is zero if and only if $C = \mathbf{0}$. Thus when the penalty term is zero, $[X] = [\hat{W}A]$ is a point on the Schubert variety $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$.

The derivative of the augmented objective function is taken with respect to A instead of $[X]$, however the method for computing that gradient remains straightforward. For tips on taking derivatives of matrix functions, refer to [39, 53]. The result of this differentiation is,

$$f_A([A],\lambda) = \left(I - \frac{2}{M}\sum_{i=1}^{M}\hat{W}^T Q_i Q_i^T \hat{W}\right) A + 2\lambda \begin{bmatrix} 0 & 0 \\ 0 & I_L \end{bmatrix} A \begin{bmatrix} I_R & 0 \\ 0 & 0 \end{bmatrix} \tag{45}$$

and then $\nabla f([A]) = f_A - AA^T f_A$ as prescribed by Algorithm 11. The augmented objective function in Equation 44 and its gradient can be used with a combination of the penalty method described by Algorithm 13 and the conjugate gradient method for the Grassmann manifold presented in Algorithm 12 to perform Schubert variety constrained optimization on the Grassmann manifold. A practical example will follow in Section 3.4.

3.3. Subspace Constrained Averaging

Neither the conjugate gradient nor the penalty method are specific to the cost function that we have chosen. Thus, so long as we can compute a derivative with respect to the coefficient matrix, $[A]$, we will be able to implement this method for Schubert variety constrained optimization. However, for this particular cost function, we are able to find an algebraic solution in certain cases. We present Lemma 3.3.1 and subsequently Theorem 3.3.2 as a technique for finding the subspace constrained average of a set of points on a Grassmann manifold. We begin by finding the corner of the distinguished subspace $[W]$ that best averages the data.

Lemma 3.3.1. ***Subspace constrained average*** *Let $\{[Q_i]\}_{i=1}^{P} \in \mathrm{Gr}(k,n)$ be a finite collection of linear subspaces with respective orthonormal bases Q_i for $i = 1 \ldots P$. Given an additional subspace $[W] \in \mathrm{Gr}(j,n)$ with orthonormal basis W and an integer $\alpha \leq \min\{j,k\}$, the α-dimensional subspace, $[X]$, that minimizes*

$$\begin{aligned} &\underset{[X]\in Gr(\alpha,n)}{\arg\min} \quad f([X]) = \frac{1}{P}\sum_{i=1}^{P} d_{pF}([X],[Q_i])^2 \\ &\text{subject to} \quad [X] \subseteq [W], \end{aligned} \tag{46}$$

is the span of the matrix WV, where $V = [v_1|v_2|\dots|v_\alpha]$ are the α dominant eigenvectors of the symmetric, $j \times j$ matrix Y, defined as

$$Y = W^T \left(\sum_{i=1}^{M} Q_i Q_i^T \right) W. \tag{47}$$

Proof. We initially wish to solve the problem for $\alpha = 1$. That is,

$$\begin{aligned} [x] = &\underset{[x]\in \mathrm{Gr}(1,n)}{\arg\min} \quad \frac{1}{P}\sum_{i=1}^{P} d_{\mathrm{pF}}([x],[Q_i])^2 \\ &\text{subject to} \quad [x] \subseteq [W]. \end{aligned} \tag{48}$$

Since $[x]$ is a 1-dimensional subspace of $[W]$, we can write $x = Wv$ for some $v \in \mathbb{R}^j$, and v can be chosen such that $v^T v = 1$. Thus Equation 48 can be rewritten as

$$\begin{aligned} [v] = &\underset{v\in\mathbb{R}^j}{\arg\min} \quad \frac{1}{P}\sum_{i=1}^{P} d_{\mathrm{pF}}([Wv],[Q_i])^2 \\ &\text{subject to} \quad v^T v = 1. \end{aligned} \tag{49}$$

To solve this problem for v, we can rewrite the cost function using the singular value decomposition of $(Wv)^T Q_i$. That is, $(Wv)^T Q_i = U\Sigma V^T$ where $U = \pm 1$, $\Sigma = \cos\theta$, and V is a unit

vector in R^k. Applying this decomposition we have,

$$
\begin{aligned}
[v] &= \arg\min_{v\in\mathbb{R}^j} \quad \frac{1}{P}\sum_{i=1}^{P} d_{\mathrm{pF}}([Wv],[Q_i])^2 \\
&= \arg\min_{v\in\mathbb{R}^j} \quad \frac{1}{P}\sum_{i=1}^{P} \frac{1}{\sqrt{2}}\|\sin\theta_i\|_2^2 \\
&= \arg\min_{v\in\mathbb{R}^j} \quad \frac{1}{\sqrt{2}}\sum_{i=1}^{P} \sin^2\theta_i \\
&= \arg\max_{v\in\mathbb{R}^j} \quad \frac{1}{\sqrt{2}}\sum_{i=1}^{P} \cos^2\theta_i \\
&= \arg\max_{v\in\mathbb{R}^j} \quad \frac{1}{\sqrt{2}}\sum_{i=1}^{P} (Wv)^T Q_i Q_i^T (Wv) \\
&= \arg\max_{v\in\mathbb{R}^j} \quad \frac{1}{\sqrt{2}} v^T W^T \left(\sum_{i=1}^{P} Q_i Q_i^T\right) Wv.
\end{aligned} \tag{50}
$$

Let $Y = W^T\left(\sum_{i=1}^{P} Q_i Q_i^T\right) W$. Then the entire problem can be written via the method of Lagrange multipliers as a function to be maximized,

$$
f(v,\lambda) = \frac{1}{\sqrt{2}} v^T Y v - \lambda(v^T v - 1). \tag{51}
$$

The first order necessary conditions for optimality are satisfied by points that simultaneously solve the partial derivatives of f with respect to v and λ. That is, the points that solve

$$
\frac{\partial f}{\partial v} = \frac{2}{\sqrt{2}} Y v - 2\lambda v \tag{52}
$$

$$
\frac{\partial f}{\partial \lambda} = v^T v - 1. \tag{53}
$$

Clearly the (v,λ)-pair that maximizes Equation 51 is the eigenvector, v, of Y associated with the largest eigenvalue, λ.

This result is extended for values of $\alpha > 1$ by solving the base case, $[x_1]$, as described, and then including the additional constraint $x_i^T x_j = 0$ for all $i < j$. Since Y is real and symmetric it is possible to find eigenvectors that form and orthogonal basis for the columnspace of Y. Thus the point $[X] \in \mathrm{Gr}(\alpha, n)$ that minimizes Equation 48 is the span of the matrix WV, where $V = [v_1|v_2|\dots|v_\alpha]$ are the α dominant, mutually orthogonal eigenvectors of Y. $\square$

Algorithm 14 Subspace constrained average

function $\boldsymbol{\mu}_{SCA}([W], \alpha, \{[Q_i]\}_{i=1}^M)$
  Ensure $W^T W = I$
  Ensure $\alpha \leq \min\{j, k\}$
  $Y \leftarrow W^T \left(\sum_{i=1}^M Q_i Q_i^T\right) W$
  $\Lambda V \leftarrow YV$ ▷ Eigenvector decomposition of Y
  return $[Wv_1|\cdots|Wv_{\alpha_1}]$ ▷ The α dominant eigenvectors from V

Lemma 3.3.1 provides the machinery to find the subspace of the distinguished space $[W]$ that is closest to the collection of points $\{[Q_i]\}_{i=1}^P$. Pseudocode for computing $\boldsymbol{\mu}_{SCA}$, the subspace constrained average, can be found in Algorithm 14. If we can extend the basis for $\boldsymbol{\mu}_{SCA}$ to span a k-dimensional space, these α-dimensions will satisfy the overlap requirement for the Schubert variety $\Omega([W], \alpha)$. Theorem 3.3.2 demonstrates how to choose those $k - \alpha$ dimensions to be the best remaining dimensions in the orthogonal complement, which leads to a point on the Schubert variety that contains the best α-dimensional subspace of $[W]$.

THEOREM 3.3.2. ***Subspace constrained flag of averages*** *Let* $\{[Q_i]\}_{i=1}^P \in \mathrm{Gr}(k, n)$ *be a collection of linear subspaces. Given a flag* $\boldsymbol{F} = \{0\} \subset [W_1] \subset [W_2] \subset \cdots \subset [W_M] \subset \mathbb{R}^n$ *with* $\dim([W_i]) = q_i$ *and a sequence of integers* $\boldsymbol{\alpha} = \{0 = \alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_M \leq \alpha_{M+1} = k\}$ *where* $\alpha_i \leq \min\{m_i, k\}$*, define a Schubert variety*

$$\Omega(\boldsymbol{F}, \boldsymbol{\alpha}) \doteq \{[X] \in \mathrm{Gr}(k, n) \mid \dim([X] \cap [W_i]) \geq \alpha_i\}. \tag{54}$$

There exists an element $[X^*] \in \Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ *such that* $[X^*] = [X^{(1)}] \oplus [X^{(2)}] \oplus \cdots \oplus [X^{(M+1)}]$ *where* $[X^{(i)}] \subseteq [W_i] \Big\backslash \left\{[X^{(1)}] \oplus [X^{(2)}] \oplus \cdots \oplus [X^{(i-1)}]\right\}$ *is a subspace of dimension* $\alpha_i - \alpha_{i-1}$ *that minimizes the function* $f_{(i)} : \Omega([W_i] \Big\backslash \left\{[X^{(1)}] \oplus [X^{(2)}] \oplus \cdots \oplus [X^{(i-1)}]\right\}, \alpha_i - \alpha_{i-1}) \subset \mathrm{Gr}(\alpha_i - \alpha_{i-1}, n) \to \mathbb{R}$ *defined as*

$$f_{(i)}([X]) = \frac{1}{P} \sum_{i=1}^{P} d_{pF}([X], [Q_i])^2. \tag{55}$$

The proof is constructive and invokes Lemma 3.3.1 multiple times. For clarity, the purpose of Theorem 3.3.2 is to find a point on the Schubert variety, $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$, that averages the data under the restriction that $[X^{(1)}]$ is the best α_1-dimensional portion of $[W_1]$, $[X^{(1)}] \oplus [X^{(2)}]$ is the best α_2-dimensional subspace of $[W_2]$ such that it contains the best subspace of $[W_1]$, and so on. In some cases this $[X^*]$ will be the point on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha}) \subseteq \mathrm{Gr}(k, n)$ that minimizes $f([X]) = \frac{1}{P} \sum_{i=1}^{P} d_{\mathrm{pF}}([X], [Q_i])^2$, but not in all instances. These cases will be elucidated following the proof.

Proof. Using Lemma 3.3.1, find $[X^{(1)}] \subseteq [W_1]$ as the solution to

$$\begin{aligned} \left[X^{(1)}\right] = \underset{[X] \in \mathrm{Gr}(\alpha_1, n)}{\arg\min} \quad & \frac{1}{P} \sum_{i=1}^{P} d_{\mathrm{pF}}([X], [Q_i])^2 \\ \text{subject to} \quad & [X] \subseteq [W_1]. \end{aligned} \tag{56}$$

Define $[\tilde{W}_2] \doteq [W_2] \Big\backslash [X^{(1)}] = \left[\left(I - X^{(1)} X^{(1)T}\right) W_2\right]$ which can be computed by projecting $[W_2]$ into the orthogonal complement of $[X^{(1)}]$, and finding an orthonormal basis for the

projection. $[X^{(2)}] \subseteq [\tilde{W}_2]$ is then found as the solution to

$$\begin{aligned} \left[X^{(2)}\right] = \underset{[X]\in \mathrm{Gr}(\alpha_2-\alpha_1,n)}{\arg\min} \quad & \frac{1}{P}\sum_{i=1}^{P} d_{\mathrm{pF}}([X],[Q_i])^2 \\ \text{subject to} \quad & [X] \subseteq [\tilde{W}_2] \end{aligned} \tag{57}$$

which ensures that $\dim\left([X^{(1)}] \oplus [X^{(2)}] \cup [W_2]\right) \geq \alpha_2$ as desired. Iterate this procedure to construct $[X^{(1)}], [X^{(2)}], \ldots, [X^{(M)}]$ as described. Let $[\tilde{W}_{(M+1)}] \doteq \mathbb{R}^n \Big\backslash \left\{[X^{(1)}] \oplus [X^{(2)}] \oplus \cdots \oplus [X^{(M)}]\right\}$, and find $[X^{(M+1)}]$ as the solution to

$$\begin{aligned} [x] = \underset{[x]\in \mathrm{Gr}(k-\alpha_M,n)}{\arg\min} \quad & \frac{1}{P}\sum_{i=1}^{P} d_{\mathrm{pF}}([x],[Q_i])^2 \\ \text{subject to} \quad & [x] \subseteq [\tilde{W}_{(M+1)}]. \end{aligned} \tag{58}$$

If we define $[X^*] = [X^{(1)}] \oplus [X^{(2)}] \oplus \cdots \oplus [X^{(M+1)}]$, then $[X^*] \in \Omega(\boldsymbol{F}, \boldsymbol{\alpha}) \subseteq \mathrm{Gr}(k,n)$ with the desired properties. □

3.3.1. When do the two problems agree? The problem solved by Theorem 3.3.2 is not the same as the one presented in Equation 37, but they do have overlaps. For this reason, we refer to the result of Theorem 3.3.2 as a subspace constrained flag of averages, rather than a Schubert variety constrained average. The result of Theorem 3.3.2 is a point $[X^*] \in \mathrm{Gr}(k,n)$ that is also an element of the Schubert variety $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$. $[X^*]$ has the property that it contains the best α_1-dimensional subspace of $[W_1]$, $[X^{(1)}]$, and the best α_2-dimensional subspace of $[W_2]$ with the caveat that it must also contain $[X^{(1)}]$, and so on.

Returning to the problem in Equation 37, The flag of interest is $\boldsymbol{F} = \{0\} \subset [W] \subset \mathbb{R}^n$ with $\dim([W]) = j$. $[X^*]$ is then the solution to Equation 37 when the distinguished subspace $[W]$ is fully contained within $[X^*]$ or when $[X^*]$ is a subspace of $[W]$. In other words, if $j \leq k$ and $\boldsymbol{\alpha} = 0 < j < n$ or if $k \geq j$ and $\boldsymbol{\alpha} = 0 < k < n$, then $[X^*]$ is the optimal average

Algorithm 15 Subspace constrained flag of averages

function $\boldsymbol{\mu}_{SFCA}(\boldsymbol{F}, \boldsymbol{\alpha}, \{[Q_i]\}_{i=1}^{M})$
  Ensure $\boldsymbol{F} = \{0\} \subset [W_1] \subset \cdots \subset [W_p] \subset \mathbb{R}^n$
  Ensure $\boldsymbol{\alpha} = \{0 = \alpha_0, \alpha_1, \ldots, \alpha_p, \alpha_{p+1} = k\}$
  $[X^{(1)}] \leftarrow \boldsymbol{\mu}_{SCA}([W_1], \alpha_1, \{[Q_i]\}_{i=1}^{M})$
  for $i = 1 \ldots M$ **do**
    $[\tilde{W}_{i+1}] \leftarrow \left(I - \left[X^{(i)}|X^{(i-1)}|\cdots|X^{(1)}\right]\left[X^{(i)}|X^{(i-1)}|\cdots|X^{(1)}\right]^T\right) W_{i+1}$
    $[X^{(i+1)}] \leftarrow \boldsymbol{\mu}_{SCA}([W_1], \alpha_{i+1} - \alpha_i, \{[Q_i]\}_{i=1}^{M})$
  return $\left[X^{(1)}\middle|X^{(2)}\middle|\cdots\middle|X^{(M+1)}\right]$

of $\{[Q_i]\}_{i=1}^{P} \in Gr(k, n)$ restricted to $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$. However, the second case is equivalent to reducing the dimension of the ambient space and finding the unconstrained average so the result should not be surprising. Pseudocode implementing the subspace constrained flag of averages can be found in Algorithm 15.

3.4. Tangent Space Decomposition Using Affine Patches

In limited cases, we can decompose the tangent space for a point on a Schubert variety into the dimensions that point within the variety and those that do not. These toys examples can be useful in gaining intuition into more general cases. Let us examine an example where Theorem 3.3.2 provides the optimal solution to Equation 37. That is, the subspace constrained average is equivalent to the Schubert variety constrained average.

Let

$$[W] = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{59}$$

and define a Schubert variety, $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$, on $\mathrm{Gr}(2, 4)$ such that $\boldsymbol{F} = \{0\} \subset [W] \subset \mathbb{R}^4$ and $\boldsymbol{\alpha} = \{0 < 1 < 2\}$. Points on this Schubert variety can be represented without loss of

generality as the columnspace of matrices in the form

$$[Y] = \begin{bmatrix} 1 & 0 \\ 0 & y_1 \\ 0 & y_2 \\ 0 & y_3 \end{bmatrix} \tag{60}$$

for $y_1, y_2, y_3 \in \mathbb{R}$ such that $y_1^2 + y_2^2 + y_3^2 = 1$, however this is not the most useful representation for the example at hand. There is an affine patch of the Grassmannian where all points can be written as

$$[X] = \begin{bmatrix} 1 & 0 \\ x_1 & x_2 \\ 0 & 1 \\ x_3 & x_4 \end{bmatrix} \tag{61}$$

for $x_1, x_2, x_3, x_4 \in \mathbb{R}$, which includes all points on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ in the form of Equation 60 where $y_2 \neq 0$. From this representation for an affine patch a spanning set of tangent vectors should be relatively obvious. Remember from Equation 4 that the tangent vectors to $[X]$ on $\mathrm{Gr}(k, n)$ are the matrices Z of size $n \times k$ such that $X^T Z = 0$. Additionally, since the dimension of $\mathrm{Gr}(2, 4)$ is $k(n-k) = 4$, we are looking for four linearly independent vectors to span the tangent space. One basis for the tangent space is then

$$\frac{\partial X}{\partial x_1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \frac{\partial X}{\partial x_2} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \frac{\partial X}{\partial x_3} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad \frac{\partial X}{\partial x_4} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{62}$$

These vectors do not necessarily point within the Schubert variety. In fact, the dimension of $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ is $\sum_{i=1}^{M} q_i - \alpha_i = (0-0) + (1-1) + (4-2) = 2$, so the tangent space to the Schubert variety at a point is 2-dimensional as well. Our choice of affine patch representation anticipated this dimension, so a basis for this space is relatively clear. Vectors $\frac{\partial X}{\partial x_2}$ and $\frac{\partial X}{\partial x_4}$ are both tangent to almost all points on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$, and we can see that for such points the span of $[X] + c_2 \frac{\partial X}{\partial x_2} + c_4 \frac{\partial X}{\partial x_4}$ ($c_2, c_4 \in \mathbb{R}$ and sufficiently small) will also contain $[W]$ as required.

3.4.1. Computational examples. We already have a proof of Theorem 3.3.2, but using the tangent space decomposition we can provide empirical evidence that the theory is correct. Let $[W]$ and $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ be defined as above. The set of data to be averaged will be 4 points on $\mathrm{Gr}(2, 4)$ that are generated as perturbation of a random point with variance $\sigma = 0.5$. In this case, the random point is

$$[Q_0] = \begin{bmatrix} -0.6665 & -0.0415 \\ -0.6927 & -0.2602 \\ 0.1042 & 0.2700 \\ 0.2549 & -0.9261 \end{bmatrix} \tag{63}$$

and the data to be averaged are

$$[Q_1] = \begin{bmatrix} -0.2471 & -0.6550 \\ 0.9269 & -0.3733 \\ -0.1701 & -0.3295 \\ -0.2254 & -0.5683 \end{bmatrix}, \quad [Q_2] = \begin{bmatrix} -0.4100 & 0.6882 \\ -0.8080 & -0.0021 \\ -0.1075 & -0.4538 \\ 0.4093 & 0.5661 \end{bmatrix}, \tag{64}$$

$$[Q_3] = \begin{bmatrix} -0.7921 & -0.4687 \\ -0.3160 & -0.2333 \\ 0.0061 & -0.0122 \\ 0.5222 & -0.8519 \end{bmatrix}, \quad [Q_4] = \begin{bmatrix} -0.5008 & -0.0127 \\ -0.8652 & -0.0225 \\ 0.0021 & -0.0493 \\ -0.0257 & 0.9985 \end{bmatrix} \tag{65}$$

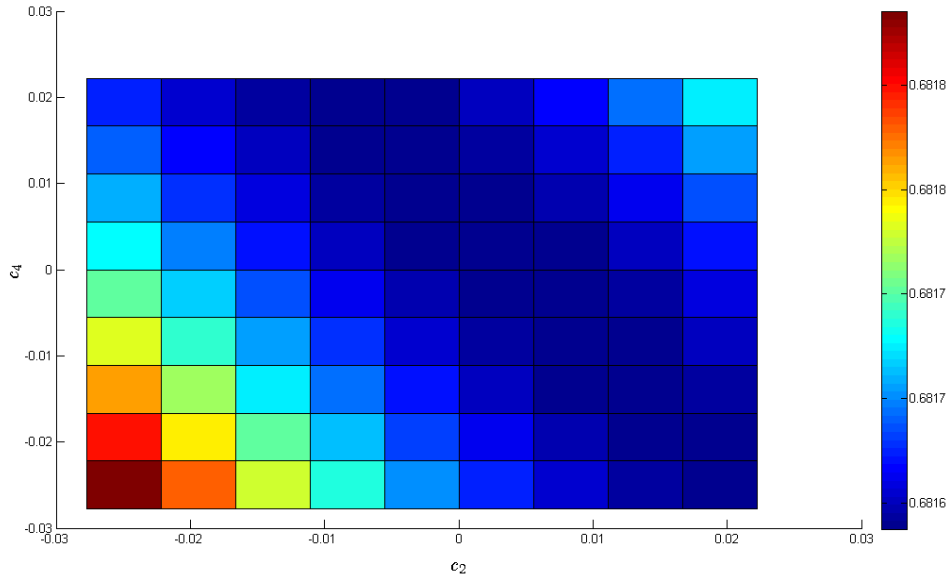
The subspace $[W]$ is not contained in any of them and the minimal principal angle separating it from any of these four points is 0.4018 radians between $[W]$ and $[Q_3]$. The unconstrained average of these four points, that is 2-dimensional element from the flag mean, is

$$[\boldsymbol{\mu}_{pF}] = \begin{bmatrix} 0.6866 & 0.0284 \\ 0.6499 & -0.4716 \\ -0.0026 & -0.0458 \\ 0.3259 & 0.8802 \end{bmatrix} \tag{66}$$

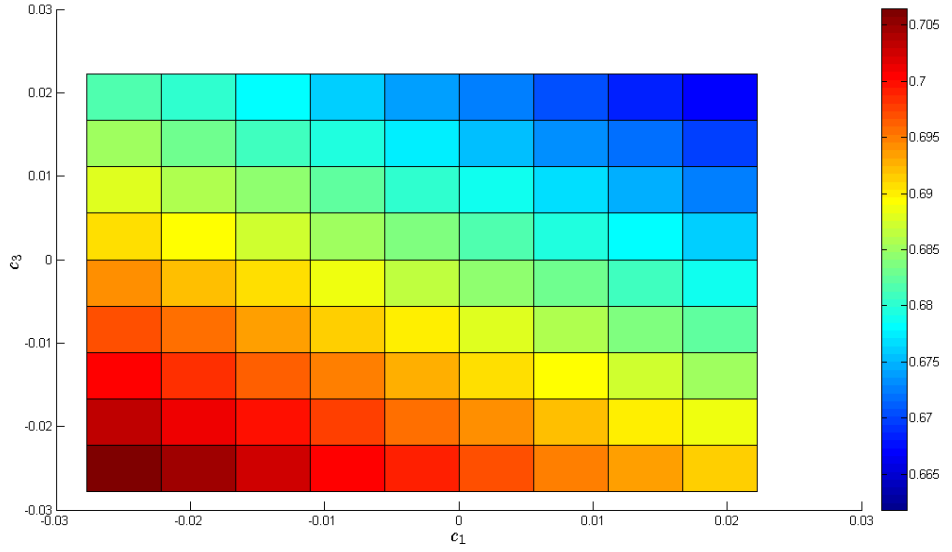
and has a cost of $\frac{1}{4}\sum_{i=1}^{4} d_{\mathrm{pF}}([\boldsymbol{\mu}_{pF}], [Q_i])^2 = 0.3371$. Clearly this does not observe the Schubert variety constraint, in fact it is 0.8132 radians away from the subspace of interest, but it gives us a baseline against which to compare our constrained averages. Using the method described in Algorithm 15, the subspace constrained flag of averages is

$$[\boldsymbol{\mu}_{SFCA}] = \begin{bmatrix} 1.0000 & 0 \\ 0 & -0.5198 \\ 0 & -0.0456 \\ 0 & 0.8530 \end{bmatrix} \tag{67}$$

and has a cost of $\frac{1}{4}\sum_{i=1}^{4} d_{\mathrm{pF}}([\boldsymbol{\mu}_{SFCA}], [Q_i])^2 = 0.6816$. Using Algorithm 12 c Algorithm 13 identifies the same minimizer as the Schubert variety constrained average.



(A) Cost for points of the form $[\boldsymbol{\mu}_{SFCA}] + c_2\frac{\partial X}{\partial x_2} + c_4\frac{\partial X}{\partial x_4}$. The minimum is at $c_2 = c_4 = 0$, indicating that $[\boldsymbol{\mu}_{SFCA}]$ is the local minimum.



(B) Cost for points of the form $[\boldsymbol{\mu}_{SFCA}] + c_1\frac{\partial X}{\partial x_1} + c_3\frac{\partial X}{\partial x_3}$. The minimum is at $c_1 = c_3 = 0.225$, thus we can decrease the cost if we leave the Schubert variety.

FIGURE 3.2. Cost surfaces for points near $[\boldsymbol{\mu}_{SFCA}]$, the subspace constrained average of $\{[Q_i]\}_{i=1}^4$.

Points of the form $[\boldsymbol{\mu}_{SFCA}] + c_2\frac{\partial X}{\partial x_2} + c_4\frac{\partial X}{\partial x_4}$ where $c_2, c_4 \in \mathbb{R}$ and are small enough stay within the tangent space of the Schubert variety. Thus we can see that $[\boldsymbol{\mu}_{SFCA}]$ is at least

a local minimum in the feasible region. Figure 3.2a shows how the cost of the optimization problem changes if we nudge $[\boldsymbol{\mu}_{SFCA}]$ in a direction that stays on the Schubert variety. The minimum over this region is for $c_2 = c_4 = 0$, indicating that we are at a local solution.

However, if we move in directions that are tangent to $[\boldsymbol{\mu}_{SFCA}]$ on $\mathrm{Gr}(2,4)$ but do not stay within the Schubert variety $[\boldsymbol{\mu}_{SFCA}]+c_1\frac{\partial X}{\partial x_1}+c_3\frac{\partial X}{\partial x_3}$, then we can observe a decrease in the cost of our objective function. Figure 3.2b shows the cost associated with nudging the average in these directions. The direction $0.0225\frac{\partial X}{\partial x_1} + 0.0225\frac{\partial X}{\partial x_3}$ appears to provide the steepest decrease with an associated cost of 0.665. Unfortunately the point that achieves this cost reduction is out of the feasible region and no longer contains $[W]$. Thus Figure 3.2 suggests that we are simultaneously solving the sequence of functions described by Equation 55 and the cost function in Equation 37, so the subspace constrained flag of averages is equivalent to the Schubert variety constrained average in this instance.

It is not always the case that the same point will satisfy both Equation 55 and Equation 37. With a small modification of the previous example we can demonstrate a case when the solutions are distinct. Let

$$[W] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{68}$$

and again define a Schubert variety, $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$, on $\mathrm{Gr}(2,4)$ such that $\boldsymbol{F} = \{0\} \subset [W] \subset \mathbb{R}^4$ and $\boldsymbol{\alpha} = \{0 < 1 < 2\}$. Points on this Schubert variety are slightly trickier to represent

generically. However if we start with a matrix of the form

$$[Y] = \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \\ 0 & y_5 \\ 0 & y_6 \end{bmatrix} \tag{69}$$

and require that

$$y_1^2 + y_2^2 = 1 \tag{70}$$

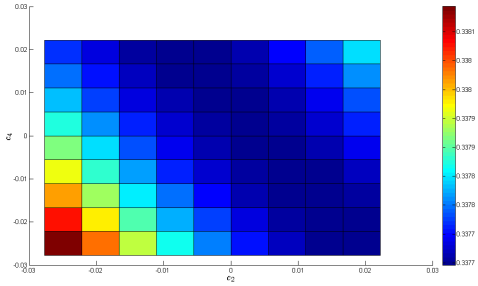
$$y_2^2 + y_4^2 + y_5^2 + y_6^2 = 1 \tag{71}$$

$$y_1 y_2 + y_3 y_4 = 0 \tag{72}$$

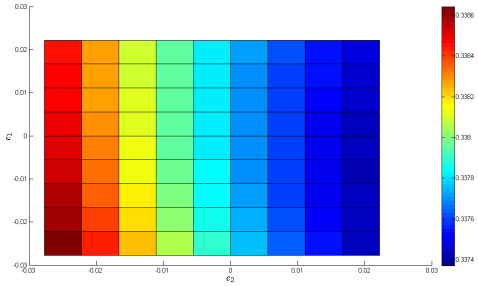
then we will be able to describe any point on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ by the columnspace of $[Y]$. Even though this generic representation is more involved than the one for the 1-dimensional $[W]$, we can use the same affine patch from Equation 61 to describe a large subset of this Schubert variety as well. Additionally, the tangent vectors from Equation 62 span the tangent space here as well. The main difference between these two examples is the dimension of the Schubert variety has increased to $\sum_{i=1}^{M} q_i - \alpha_i = (0-0) + (2-1) + (4-2) = 3$ with $\frac{\partial X}{\partial x_1}$, $\frac{\partial X}{\partial x_2}$, and $\frac{\partial X}{\partial x_4}$ all pointing within the Schubert variety.

Computing the subspace constrained flag of averages for the data yields the point

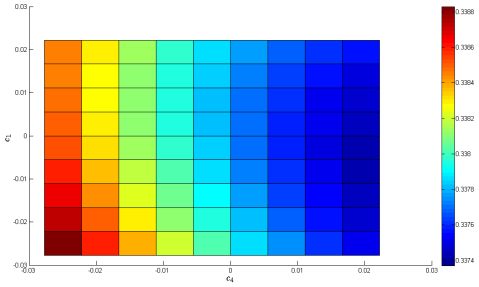
$$[\boldsymbol{\mu}_{SFCA}] = \begin{bmatrix} 0.6553 & 0.2570 \\ 0.7554 & -0.2230 \\ 0 & -0.0432 \\ 0 & 0.9393 \end{bmatrix} \tag{73}$$



(A) Cost for points of the form $[\boldsymbol{\mu}_{SFCA}] + c_2\frac{\partial X}{\partial x_2} + c_4\frac{\partial X}{\partial x_4}$. The minimum is at $c_2 = c_4 = 0$.



(B) Cost for points of the form $[\boldsymbol{\mu}_{SFCA}] + c_1\frac{\partial X}{\partial x_1} + c_2\frac{\partial X}{\partial x_2}$. The minimum is at $c_1 = 0.022$, $c_2 = -0.017$.



(C) Cost for points of the form $[\boldsymbol{\mu}_{SFCA}] + c_1\frac{\partial X}{\partial x_1} + c_4\frac{\partial X}{\partial x_4}$. The minimum is at $c_1 = 0.022$, $c_4 = -0.011$.

FIGURE 3.3. Cost surfaces for points near $[\boldsymbol{\mu}_{SFCA}]$, the subspace constrained average of $\{[Q_i]\}_{i=1}^4$, that remain on the Schubert variety. Figure 3.3b and Figure 3.3c do not have minimums at the origin, indicating that $[\boldsymbol{\mu}_{SFCA}]$ is not a local minimum of Equation 37.

with an associated cost of 0.3377. The three obvious 2-dimensional cost surfaces that we can create from points in the form $[\boldsymbol{\mu}_{SFCA}] + c_1\frac{\partial X}{\partial x_1} + c_2\frac{\partial X}{\partial x_2} + c_4\frac{\partial X}{\partial x_4}$ indicate that while this is the optimal solution for subspace constrained averaging, it is not a local minimum for the Schubert variety constrained cost function from Equation 37. Figure 3.3 shows these three cost surfaces.

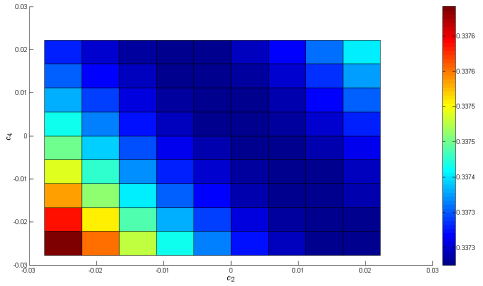
If we follow the path of steepest descent we eventually arrive at a locally optimal solution for the Schubert variety constrained average,

$$[\boldsymbol{\mu}_{Schub}] = \begin{bmatrix} -0.6336 & 0.2646 \\ -0.7737 & -0.2167 \\ 0 & -0.0431 \\ 0 & 0.9387 \end{bmatrix}. \tag{74}$$

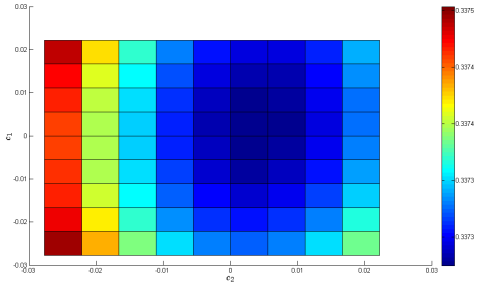
This point lives on $\Omega(\boldsymbol{F}, \boldsymbol{\alpha})$ and has an associated cost of 0.3371. Additionally, the cost surfaces shown in Figure 3.4 indicate that nudging this point within the variety will increase the cost. This optimal solution can also be found using the conjugate gradient with penalty method when initiated with a nearby starting point. These two points are very close together, the geodesic distance based on arc length between them is 0.0268, but this is an extremely simply example. It is not always the case that the distinct solutions are that close together.

3.5. Summary

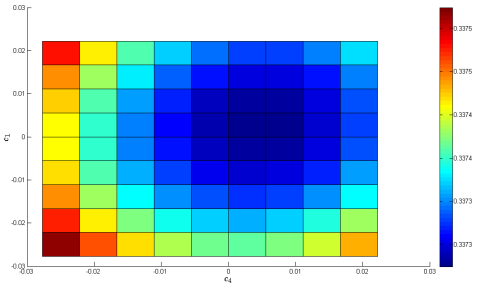
In Chapter 3 we introduced the machinery for performing Schubert variety constrained optimization in some cases. Specifically, when we rewrite the Schubert variety constraint using a coefficient matrix, we can convert the constraint into a second term in the objective function. This allows us to leverage the penalty method in conjunction with conjugate gradient to perform Schubert variety constrained optimization for any objective function that we can differentiate with respect to the coefficient matrix. In Subsection 3.2.3 we were formulating the coefficient matrix A to suit the cost function described by Equation 37, which has only one subspace $[W]$ in it's flag. The formulation is much more general than



(A) Cost for points of the form $[\boldsymbol{\mu}_{Schub}] + c_2\frac{\partial X}{\partial x_2} + c_4\frac{\partial X}{\partial x_4}$. The minimum is at $c_2 = c_4 = 0$.



(B) Cost for points of the form $[\boldsymbol{\mu}_{Schub}] + c_1\frac{\partial X}{\partial x_1} + c_2\frac{\partial X}{\partial x_2}$. The minimum is at $c_1 = c_2 = 0$.



(C) Cost for points of the form $[\boldsymbol{\mu}_{Schub}] + c_1\frac{\partial X}{\partial x_1} + c_4\frac{\partial X}{\partial x_4}$. The minimum is at $c_1 = c_4 = 0$.

FIGURE 3.4. Cost surfaces for points near $[\boldsymbol{\mu}_{Schub}]$, the Schubert variety constrained average of $\{[Q_i]\}_{i=1}^4$, that remain on the Schubert variety. Figure 3.3a, Figure 3.3b and Figure 3.3c all have minimums at the origin, indicating that $[\boldsymbol{\mu}_{Schub}]$ is a local minimum of Equation 37.

that, however. We could design a block coefficient matrix in the style of Equation 43 to suit any flag. First, we would modify $\hat{W}$ so that the span of the first q_i columns is a basis for subspace $[W_i]$. Then we would create blocks within A of the appropriate dimension to mix columns for each subspace in the flag. That is, the first block in the upper right of A would be a slice of an orthonormal matrix of size $j_1 \times \alpha_1$, with a block of zeros of size $(n - j_1) \times \alpha_1$ below it. To the right of these columns would be a block of size $j_2 \times (\alpha_2 - \alpha_1)$ that is a slice of an orthonormal matrix, and zeros filling out the rows beneath. This pattern continues until all of the proper subspaces in the flag have been accounted for and the remaining columns

complete the matrix to a basis for $\mathbb{R}^n$. Of course the penalty function would have to modified slightly as well to ensure that zeros ended up in the proper places in the coefficient matrix once the optimization was complete.

Although this process can be extended for Schubert varieties of all types, the efficacy of the method leaves much to be desired. In experimental trials, the initial guess of the solution needs to be close to the actual solution for the method to converge to the correct optimum. It appears that the penalty method, at least in our implementation, creates very steep walls around iterates as they near the feasible region. This means that we converge to stationary points based on our augmented cost function, but they are not actually local solutions to the initial problem. Additionally, because the Schubert variety constraint is not convex it is possible that there are many local solutions to Equation 37 that are not globally optimal.

The other main contribution of this chapter was to introduce the notion of a subspace constrained flag of averages as described by Theorem 3.3.2. The idea is that a different problem can be solved to find the portion of each subspace in a flag that is closest to a collection of data. The flag and/or point constructed from spanning these portions is sometimes equivalent to the solution to the Schubert variety constrained average. Section 3.4 provided two concrete examples of this process; one where the solutions agreed and one where they did not. In these simple examples we were able to decompose the tangent space at a point on the Grassmannian into the directions that stay within the Schubert variety and those that do not. This allowed us to identify when solutions to the subspace constrained averaging problem do not solve the Schubert variety constrained averaging problem.

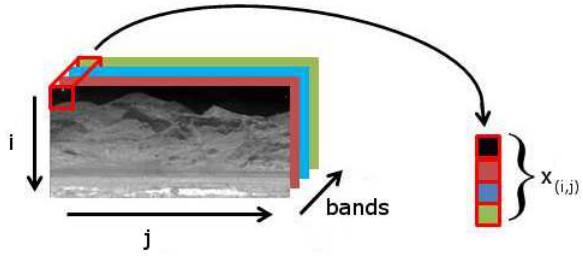
CHAPTER 4

HYPERSPECTRAL CHEMICAL PLUME DETECTION

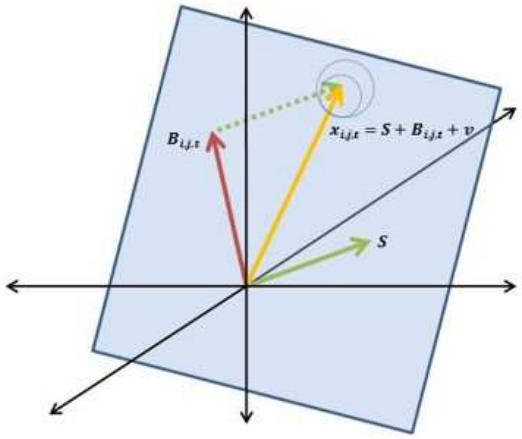
4.1. Introduction

The problem of detecting chemical plumes in long-wave infrared hyperspectral images has grown increasingly popular in the last decade. Military and civilian applications are abundant as can be inferred from the variety of disciplines that research the topic [40, 64, 50, 20, 69]. Most algorithms for chemical plume detection, such as matched filters [58, 59], normalized matched filters [44], and subspace detectors [32], use information from a single hyperspectral image to perform their detections. The accuracy of these methods are improved by supervised estimation of plume-free background statistics. For images generated by ground-level sensors, the problem of estimating the radiance of the background clutter is exacerbated by the unequal distances between objects in the scene and the sensor itself.

One unsupervised method for estimating the plume-free background of an image is to keep the sensor in a static location and image the scene at a time known to be free of chemical agents. This alleviates the difficulty of compensating for differences in path-length between objects and the sensor, and ensures that the background estimate will not be contaminated with the chemical being detected. Using a static sensor to repeatedly image the same scene is referred to as a 'persistent stare' application and can be useful for monitoring emissions from industrial facilities or as an early warning of dangerous chemicals near cities or sensitive government buildings [69, 20, 33, 16]. What this method does not account for are changes in lighting and temperature due to the change in time between the images being used for detection. These physical changes affect the at-sensor measurements, and must be taken into account by any technique using this type of background estimate.



(A) Each pixel in a long-wave infrared hyperspectral image corresponds to a vector whose elements are the response of that pixel to different wavelengths of infrared light.



(B) The linear model for the observed spectrum of pixel $x_{i,j,t}$, most simply consists of the sum of the gas spectrum (S), the background of that pixel ($B_{i,j,t}$) and Gaussian noise (ν).

FIGURE 4.1. Pixels of hyperspectral images correspond to vectors that can be decomposed into distinct endmembers.

In this chapter, we propose a method for detecting chemical plumes that utilizes temporal information to estimate the plume-free background of a hyperspectral image. With this information, we substitute an element of the flag mean for each hyperspectral pixel in an image that contains dimensions which span the background clutter, the changes in illumination and temperature, and the signature of the chemical agent of interest. By measuring the similarity between the laboratory signature for a chemical and the representative flags, we determine a scalar statistic that predicts the presence of the chemical at each pixel in an image. This method of *including* background information in the model for each pixel, rather than projecting away from it, provides a sensitive detector that appears to provide improved detection accuracy for weak or optically thin chemical plumes. [1]

[1]The material in this chapter was largely published in [46], and was a collaboration with J. Ross Beveridge, Bruce Draper, Michael Kirby, and Chris Peterson.

There are numerous thorough and well written introductions to hyperspectral image processing available for reference [40, 42]. Thus we review only the details relevant for the technique at hand. The most widely used spectral model is the linear mixing model [3]. This model assumes that the observed spectrum of a pixel consists of a linear combination of distinct spectral endmembers. In other words, pixel (i, j) is represented by a vector $x_{i,j} \in \mathbb{R}^b$ where b is the number of spectral bands captured by the imaging device. Then we can write $x_{i,j} = S + B_{i,j} + \nu$ where S is the spectrum of the gas, B is the spectrum of the background at that pixel location, and ν is white noise. Alternatively, either S or $B_{i,j}$ can be replaced by matrices whose columns correspond to endmembers. In this case, the matrices would be multiplied by a unit length vector that provides a convex combination of these endmembers. Figure 4.1 shows an illustration depicting the correspondence between a pixel in the scene and a linear combination of spectral endmembers.

Algorithm 16 Adaptive cosine (coherence) estimator detection statistic

function $\eta_{\text{ACE}}(x, s, \hat{\Gamma}, \hat{\mu})$
  Ensure $\hat{\Gamma}$ is invertible. ▷ Use diagonal loading if necessary
  $\tilde{x} \leftarrow x - \hat{\mu}$
  $\tilde{s} \leftarrow s - \hat{\mu}$
  return $\frac{\left(\tilde{s}^T \hat{\Gamma}^{-1} \tilde{x}\right)^2}{\left(\tilde{s}^T \hat{\Gamma}^{-1} \tilde{s}\right)\left(\tilde{x}^T \hat{\Gamma}^{-1} \tilde{x}\right)}$

Two of the detection algorithms in the literature that assume a linear mixing model are the adaptive cosine (or coherence) estimator (ACE) [32, 59] and the matched filter (MF) [43]. The ACE algorithm has become particularly popular in practice because of it broad applicability, simple implementation, and speed [41]. Assume that $s, x, \hat{\mu} \in \mathbb{R}^b$ such that s is the spectrum of a target signature, x is the spectrum of a test pixel, $\hat{\mu}$ is the sample mean of the pixels, and $\hat{\Gamma} \in \mathbb{R}^{d \times d}$ is the sample covariance of the mean subtracted pixels. The ACE detection statistic is then the cosine of the angle between whitened versions of target

signature and the test pixel. The pseudocode for computing the ACE detection statistic is presented in Algorithm 16, which takes a value of 1 if the whitened target spectrum and the whitened pixel are collinear, and a value of 0 if they are orthogonal.

Alternatively, the commonly used implementation of the matched filter assumes that the target and background classes have the same covariance matrix and that the observed spectra have uncorrelated components [43]. This assumption is can be satisfied, in part, in a scenario where there is no structured interference in the background pixels. Pseudocode for the computation of the matched filter can be found in Algorithm 17.

Algorithm 17 Matched filter detection statistic

function $\eta_{\mathrm{MF}}(x, s, \hat{\Gamma}, \hat{\mu})$
  Ensure $\hat{\Gamma}$ is invertible. ▷ Use diagonal loading if necessary
  $\tilde{x} \leftarrow x - \hat{\mu}$
  $\tilde{s} \leftarrow s - \hat{\mu}$
  return $\frac{\tilde{s}^T \hat{\Gamma}^{-1} \tilde{x}}{\tilde{s}^T \hat{\Gamma}^{-1} \tilde{s}}$

Thresholding these statistics allows us to compute a binary detection mask on a hyperspectral image. However, to determine the possible effectiveness of this thresholding we report empirical receiver-operator-characteristic (ROC) curves and area under the curve (AUC) scores. Of particular interest is the front end of the ROC curves, where the false positive rate is low.

4.2. Flag-based Chemical Plume Detection

Given a hyperspectral movie with pixelwise correspondence between frames, the flag-based detection algorithm preprocesses the data by creating a subspace from the span of a hyperspectral pixel at a time known to be free from gas, and the associated pixel from the frame under test, that is $[X_{i,j,t}] = \mathrm{span}\{x_{i,j,0}, x_{i,j,t}\}$. If a pixel contains the target spectrum in one frame and not the other the subspace will be 2-dimensional. Thus $[X_{i,j,t}] \in \mathrm{Gr}(2, b)$ or

$\mathrm{Gr}(1, b)$, where b is the number of spectral bands in the movie. Three horizontally adjacent subspaces (pixels) are then averaged using the flag mean from Algorithm 10, pushing mutual information to the front and creating a new representative

$$[\overline{X}_{i,j,t}] = \text{the 3-dimensional element of } \boldsymbol{\mu}_{pF}\left([X_{i,j-1,t}], [X_{i,j,t}], [X_{i,j+1,t}]\right) \tag{75}$$

for the pixel. If all of the subspaces being averaged are 2-dimensional, the largest element of the flag mean could potentially be a 6-dimensional subspace. If the three subspaces being averaged came from pixels with homogeneous background, the 1-dimensional element of the flag will represent the background spectrum, thus in practice this dimension is discarded. However, if two of the three subspaces being averaged contain the target spectrum, the median-like property of the flag mean will push this into the second or third element of the flag mean depending on the magnitude of change in ambient conditions. The detection statistic for the flag-based detection technique can be computed using the pseudocode in Algorithm 18.

Algorithm 18 Flag-based detection statistic

function $\eta_{\mathrm{F}}([X_{i,j-1,t}], [X_{i,j,t}], [X_{i,j+1,t}], s)$
  $[\overline{X}_{i,j,t}] \leftarrow \boldsymbol{\mu}_{pF}\left([X_{i,j-1,t}], [X_{i,j,t}], [X_{i,j+1,t}]\right)$ ▷ 3-dimensional element of $\boldsymbol{\mu}_{pF}$
  $[s] \leftarrow \mathrm{span}\{s\}$ ▷ A 1-dimensional subspace
  $\Theta_1 \leftarrow \boldsymbol{\Theta}([\overline{X}_{i,j,t}], [s])$ ▷ Smallest angle via Algorithm 5
  return $1 - \frac{2\Theta_1}{\pi}$

4.3. Data Set Description

The long-wave infrared data for which we compute detections is a 4-dimensional array (hyperspectral movie) from the Fabry - Pérot Interferometer Sensor Data Set of size 256 rows $\times$ 256 columns $\times$ 20 bands $\times$ 561 frames created by the Naval Research lab [35]. The spectrometer used to collect this data is an imaging spectroradiometer that operates

efficiently in the 811 micron range. An explosive burst was used to launch and disperse the simulant Triethyl Phosphate (TEP) near frame 111 of the movie. The hyperspectral images from this movie will be used to demonstrate the effectiveness of the detection algorithms on plume-free images into which target signatures have been synthetically added, and on images after the release of the simulant to use in demonstrate practical scenarios.

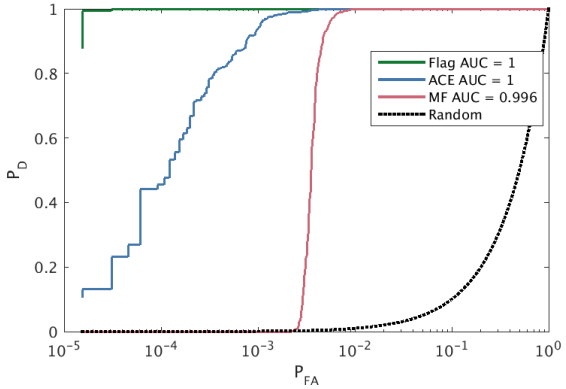
4.4. Quantitative Results on Synthetic Data

The statistic computed by the flag-based detection algorithm reduces to a monotonic function of the principal angle separating the target signature from a test subspace, however, the process used to generate that representative subspace encodes a stronger signal if a neighbor of that pixels also contains the target. Hence this algorithm improves detections for signals with very small signal-to-interference-plus-noise ratios (SINR). This property will first be demonstrated on hyperspectral images where target signatures have been synthetically added to a plume-free scene at different SINRs. This artificial data will provide ground truth by which to benchmark the three algorithms quantitatively. The SINR is computed as the ratio of the signal power to the total power of the background plus white noise. Specifically, artificial plume pixels were created as $x_{i,j} = S\alpha + B_{i,j} + \nu$ where S is the signature for TEP, α is a constant that is scaled to achieve the desired SINR, $B_{i,j}$ is the background spectrum at pixel (i, j), and ν is white noise with variance $\sigma^2 = 0.2$ and covariance Γ as described below. Thus

$$\text{SINR} = \frac{\frac{1}{b}\sum_{k=1}^{b}(S(k)\alpha)^2}{\frac{1}{b}\sum_{k=1}^{b}(B_{i,j}(k))^2 + \sigma^2} \tag{76}$$

is the computed ratio and is typically reported in decibels as $\text{SINR}_{\text{dB}} = 10\log_{10}(\text{SINR})$.

(A) Randomly generated synthetic binary detection mask.



(B) ROC curves for TEP added to the hyperspectral image with $\text{SINR}_{\text{dB}} = 5$.

FIGURE 4.2. Detection accuracy for TEP inserted into a plume-free hyperspectral image.

Each wavelength used in the observations of the Fabry - Pérot data was selected to maximize detection sensitivity of TEP, however, the adjacent bands are highly correlated. Thus, we follow the lead of Sakla *et al.* [56] and utilize a first-order Markov-based model, defined as $\nu \sim N_b[0, \Gamma]$ to generate the additive white noise. The covariance matrix Γ is defined as $\Gamma = \sigma^2 \boldsymbol{R}$ where $\boldsymbol{R}$ is the Toeplitz correlation matrix defined according to the first-order Markov model [27],

$$\boldsymbol{R} = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^3 & \rho^{b-1} \\ \rho & 1 & \rho & \rho^2 & \cdots & \rho^{b-2} \\ \rho^2 & \rho & 1 & \rho & \cdots & \rho^{b-3} \\ \vdots & & & & \ddots & \vdots \\ \rho^{b-2} & \cdots & \rho^2 & \rho & 1 & \rho \\ \rho^{b-1} & \cdots & \rho^3 & \rho^2 & \rho & 1 \end{bmatrix} \tag{77}$$

and σ^2 is a fixed variance. In the experiments the variance was fixed at $\sigma^2 = 0.2$. The method for estimating ρ also follows that of Sakla *et al.* [56].

Using the above method, and scaling α to generate the appropriate SINR_{dB}, TEP was added to a randomly generated rectangle accounting for 1% of the pixels in a plume-free

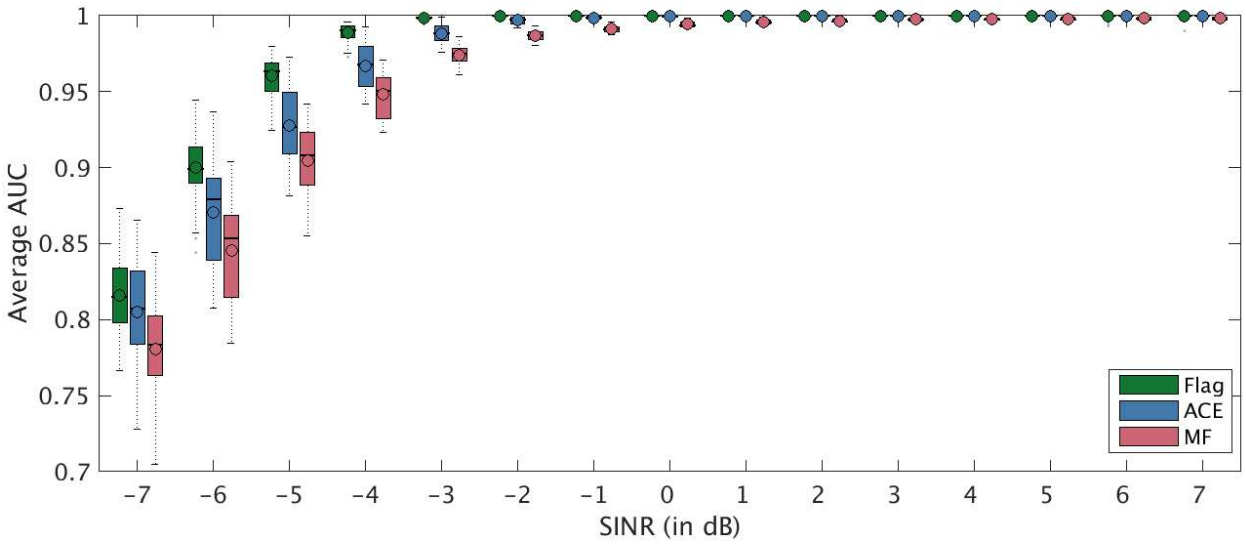


FIGURE 4.3. Box plots representing the AUC scores for detections of synthetically added TEP. Rectangular plumes were randomly generated 100 times for each reported SINR level.

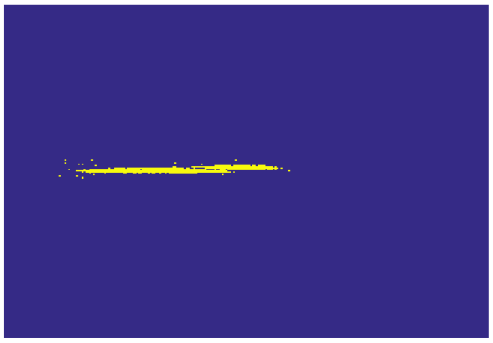
hyperspectral image. An example of one such binary detection mask is shown in Figure 4.2a. White noise with the same variance and covariance was added to all pixels of the image not included in the plume. To compute detections as fairly as possible, the plume mask was used by the MF and ACE algorithms so that calculated background statistics would be as accurate as possible. The flag-based algorithm used a plume-free version of the same image with different white noise to create the initial 2-dimensional subspaces to be averaged. The experiment was repeated 100 times at each SINR_{dB} level to generate the box plot of AUC scores in Figure 4.3. As can be seen, the biggest advantage of the flag-based detection algorithm over the ACE and MF methods comes at very low SINR_{dB} levels. However, even when the SINR_{dB} level is positive and all methods do well, the flag-based detection has a probability of detection for a low probability of false alarm as can be seen in Figure 4.2b.

4.5. Qualitative Results

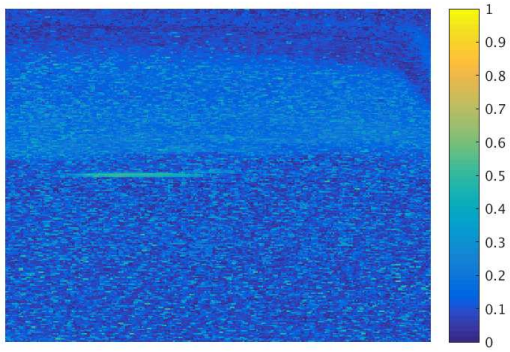
Other researchers have used temporal information to segment hyperspectral videos as a way to detect and track chemical plumes [19, 65]. These methods do not predict whether or not a chemical agent is present, but rather cluster related hyperspectral pixels so that contiguous regions can all be identified as containing a chemical or not with another method. In an attempt to provide some measure of the success of our detections on real data, we have employed a less sophisticated supervised clustering technique to find the pixels associated with the gas plume in frame 150 of the Fabry - Pérot data, i.e. after the simulant was released. This clustering will then be used as an approximate plume mask for computing ROC curves on our various detections in that frame.

To generate the approximate plume mask, first, the temporal singular value decomposition of each pixel in the hyperspectral movie was compute. That is to say a basis was found for the space spanned by all of the spectra of a single pixel through time. The background of each pixel was estimated as the span of the first three dimensions of its basis. Each pixel was then projected into the orthogonal complement of this basis to remove most of the background information. The resulting background-removed hyperspectral pixels were clustering using k-means. The cluster membership along with an visualization of the spectral mean of the image was manually inspected to determine which clusters contained the plume, and those pixels were used to create the approximate plume mask. This process was heavily supervised, and while replicable, was not automated. An example of the plume mask generated for frame 150 can be seen in Figure 4.4a.

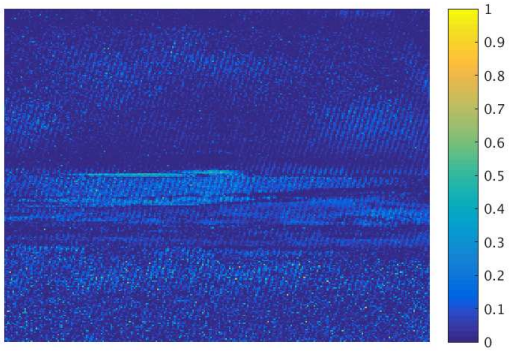
Detections were performed on images using the three algorithms. For the flag-based algorithm, the first scene of the hyperspectral movie which is free of simulant, was used as the plume-free frame to build the initial 2-dimensional subspaces for each pixel. For the MF



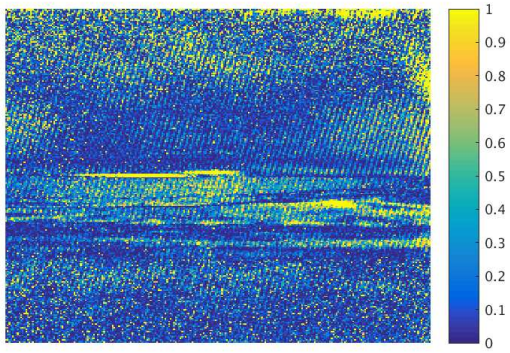
(A) Approximate plume mask computed by clustering background-removed hyperspectral pixels.



(B) Detection map for flag-based algorithm.



(C) Detection map for ACE algorithm.



(D) Detection map for matched filter algorithm

FIGURE 4.4. Detection scores associated with frame 150 of the Fabry - Pérot data. Higher scores indicate a greater likelihood of TEP being present in a given pixel.

and ACE algorithms, the approximate plume mask was used to achieve a better estimate of the background statistics. Figure 4.4 shows the detections maps for each algorithm on frame 150 of the data, and Figure 4.5 shows the associated ROC curves computed from the approximate plume mask.

In Figure 4.4d, the scores from matched filter algorithm, we see the highest detection statistics. However this does not necessarily translate to the best detection performance.

Many of the high scoring detections are false alarms in the upper right sky pixels. In Figure 4.4c we see the detections of the ACE algorithm. These are mostly low scores, but the plume is distinct along the horizon. Contrast is more important than high scores achieving a good level of detection, which is generally good for the ACE detections. There are a handful of speckle noise errors in the foreground of the image that generate a poor probability of detection for small probabilities of false alarm however. This agrees with the information in the ROC curves that suggests that the ACE and MF algorithms do not perform quite as well in the left-hand side of the ROC curves in Figure 4.5. Overall the AUC for the flag-based algorithm is higher than the two standard algorithms, and it performs better on the left side of the ROC curve. This can be identified in the detection images by the slightly better contrast between the plume and the speckle noise detections in the background.

From the AUC scores computed to be between 0.78 and 0.86, the results on synthetic data in Figure 4.3 would suggest that this image has a SINR_{dB} around -7 however this cannot be computed exactly. The somewhat contiguous, higher detection statistics in the sky in Figure 4.4b may be attributable to the change in temperature between the first frame of the movie and the 150th frame as there is approximately a 6 second lag between frames during the capturing process and the video was capture in the morning.

4.6. Summary

In this chapter we presented a novel flag manifold based method for detecting chemical plumes in long-wave infrared hyperspectral movies. The technique leverages knowledge of the radiance of the background scene, taken from a frame of a hyperspectral movie at a time known to be free of chemical agents, to improve the detection of chemical signatures in other frames of the movie.

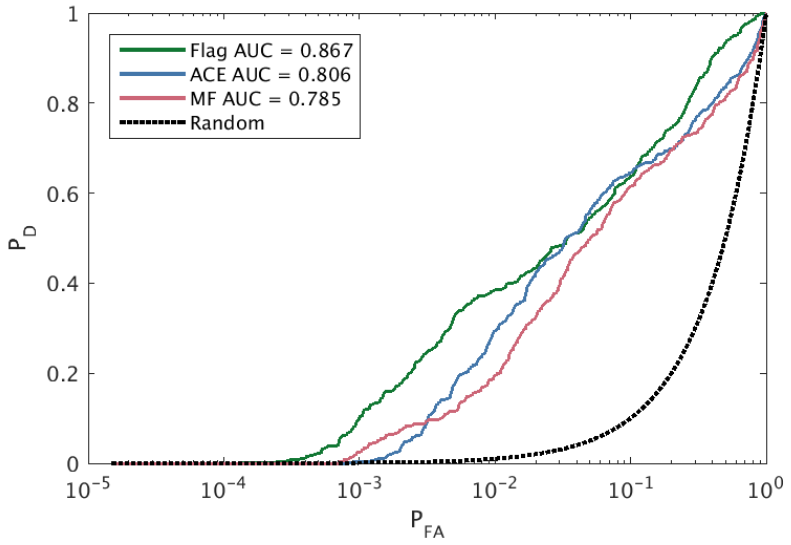


FIGURE 4.5. ROC curves generated from detections on frame 150 of the data using approximate plume as truth.

The technique used to create the flags pushes information about the background clutter, ambient conditions, and potential chemical agents into the leading elements of the flags. The result of exploiting this temporal information by way of the flag structure is a novel algorithm for detecting gas plumes that appears to be sensitive to the presence of weak plumes.

Quantitative results on synthetic data show that the flag-based algorithm consistently performs better on data when the SINR_{dB} is low, and beats the ACE and MF algorithms in probability of detection for low probabilities of false alarm even when the SINR_{dB} is high. Qualitative experiments suggest that these results hold true on real data, when images of the scene are available at a time known to be free of the target signature.

CHAPTER 5

ADAPTIVE VISUAL SORT AND SUMMARY

5.1. Introduction

Forensic analysis of nanoparticles is often conducted through the collection and identification of electron microscopy images to determine the origin of suspected nuclear material. Each image is carefully studied by experts for classification of materials based on texture, shape, and size. Manually inspecting large image datasets takes enormous amounts of time. However, automatic classification of large image datasets is a challenging problem due to the complexity involved in choosing image features, the lack of training data available for effective machine learning methods, and the lack of availability of user interfaces to parse through images. Therefore, a significant need exists for automated and semi-automated methods to help analysts perform accurate image classification in large image datasets.

The overarching goal of this chapter is to create a 2-dimensional visualization of a collection of data that reflects semantic, or describable, relationships between the data [2]. On the surface, there are many algorithms that already address this issue. See, for example, nonlinear embeddings like Isomap [68], Locally Linear Embedding [54], Laplacian Eigenmap [6], Diffusion Map [9], etc. While these methods retain a great deal of relationship information in their low dimensional embeddings, it may not be related to the primary objective of the data analysis. There are many scenarios in which each method excels or fails. Instead, we present INStINCt, our Intelligent Signature Canvas, as a method for quickly organizing image data

[2]The work in this chapter was done in collaboration with Elizabeth Jurrus, Nathan Hodas, Nathan Baker, and Mark D. Hoover, and was supported by the Pacific Northwest National Laboratory National Security Directorate PhD Internship Program. The novel contributions to this dissertation are mainly contained in Section 5.2.3. The full text of the publication that resulted from this work can be found in [29].

in a web-based canvas framework. Images are partitioned using small sets of example images, chosen by users, and presented in an optimal layout based on features derived from convolutional neural networks [61]. The optimal layout chosen by the system is the result of applying the diffusion map embedding to a graph whose edges are weighted by the angle between individual images and the flag means of subsets of images chosen by the user as anchors for the visualization.

To demonstrate the value of such a technique, assume that a collection of data consists of images to be classified by a domain scientist. An oncologist might look at a massive collection of blood cell images to determine by morphological appearance if circulating tumor cells (CTCs) are present. It is a non-trivial task to identify and count the CTCs even with advanced machine learning techniques because the expected ratio of CTCs to healthy blood cells is incredibly small and many healthy cells may appear malformed if they have been imaged poorly. Pathologists can manually judge whether cells that have been automatically classified as CTCs are in fact cancerous but this requires significant time on the part of the physician or technician. Our task in this analogy would be to then sort the collection of images based on their relationship to some small collection of cells that the human operator has manually identified to be CTCs. If our initial sorting of the images contained errors, we would like the human operator to be able to affect the sorting globally by interacting with a small number of images directly (rather than with parameters in the model).

The idea of allowing users to interact with mathematical models to improve clustering and classification has a long history of research in the semi-supervised clustering domain. Methods such as iPCA [28], SCREEN [67], and many others allow user guidance to affect their lower dimensional embeddings. However, these methods require that users manipulate parameters directly. In iPCA, for example, users adjust the weights associated with the

various basis dimensions to adjust the view of the data. While this type of manipulation will respect the restrictions of the mathematical model, it may frustrate users who are experts in other domains but not in data reduction techniques. Instead, we would like human operators to be able to manipulate the visualization directly, and to have the intent of that interaction translated into parameter updates.

A pipeline called Visual to Parametric Interaction (V2PI) where users interact purely in the visual domain, but affect changes to the model parameters that update the underlying mathematical model and hence the visualization on a global scale was introduced by Leman et al. [36]. The pipeline that Leman proposes appears to build on work of Endert et al. [14] that argues against direct manipulation of model parameters, saying "The drawback of [direct parametric] interaction is that users are expected to be experts in the underlying model that generates the visualization".

The results of the INStINCt framework are demonstrated qualitatively using particle images from the Capstone Depleted Uranium (DU) Aerosol Characterization and Risk Assessment Program [51], which a team led by Pacific Northwest National Laboratory conducted under the auspices of the U.S. Army Public Health Command. This dataset consists of a large set of backscattered electron (BSE) images from scanning electron microscopy (SEM) of aerosol samples collected during perforation of an Abrams tank and a Bradly vehicle with DU muntions. The details of the aerosol collection in the high-energy environment of the Capstone study have been described by Holmes *et al.* [25]. The motivation for examining the images of the particles is to determine the particle morphology, especially in the nano-size range. This morphology provides insight into the relationship between the chemical formation, the solubility, and the dissolution rates [51] present during their formation. As reported by Krupka *et al.* [34], ultrafine aerosols of aluminum and iron from the vehicle armor were

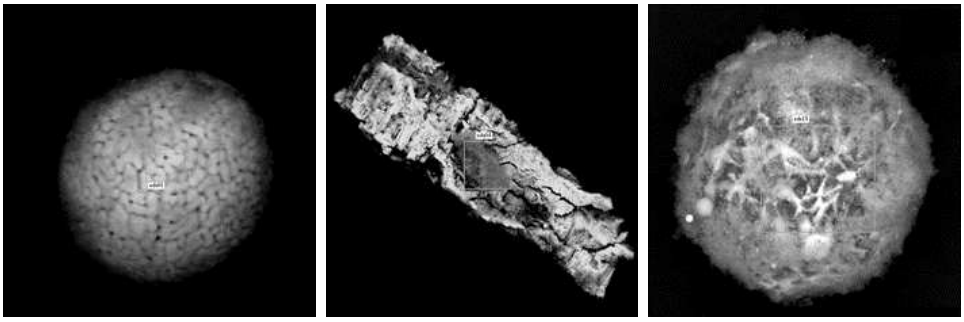


FIGURE 5.1. Example nanoparticle images from the CDC Capstone DU Aerosol Study.

observed in the analysis of the Capstone particle images. The mechanism of the formation for such particles in the nano-size range from high temperature processes involves vaporization-condensation mechanisms. Guilemette and Parkhurst [38] further reported that little to no evidence was obtained that demonstrated the existence of DU nanoparticles. Examples of BSE images of a selection of these particles can be seen in Figure 5.1.

5.2. Adaptive Visual Sort and Summary

The INStINCt framework was developed to enable efficient organization of SEM images through the integration of a web-based interface, state-of-the-art image feature vectors, and a new clustering technique. The user interacts with the INStINCt interface through a light weight web client application that provides a virtual canvas for organizing images and enabling different users to collaborate on the same datasets. The canvas displays images represented by a 4096-dimensional feature vector computed from OverFeat, a convolutional neural network [61]. Since we cannot easily visualize these high dimensional feature vectors, we attempt to map these feature vectors into a 2-dimensional space while preserving their class relations. The method we implemented to perform dimensionality reduction is an adaptive spatialization method. First, the user selected images are clustered using DBSCAN [15] and "anchors" for each group are computed. The positions of the remaining images are

updated in the 2-dimensional spatialization using a coordinates determined by the similarity of the images to these clusters. Similarity is measured by the angle between a feature vector and the low-dimensional subspace of created as an anchor. The relationships between unlabeled images and small, user-selected clusters of images are visualized via diffusion map [9] in the canvas. This representation is therefore distinct from the related L-Isomap embedding where the algorithm attempts to preserve distances between landmarks, and fits the remaining data into the embedding by triangulating distances from the landmarks [10]. One of the advantages of our method for representing the clusters is that in the inevitable case that the visualization does not represent all the of the semantic information recognized by the domain expert, the visualization can be globally updated by moving a small number of samples closer to or further from the clusters.

5.2.1. User Interface. Central to the INStINCt implementation is the web-based software for displaying images and recording human operator interactions. We leverage the activeCanvas web application from Hodas *et al.* [24] to display the computed visualizations. The activeCanvas allows image files to be uploaded to a server, from which they can be organized into a 2D layout by means of JSON files that contain most relevantly, file names, (x, y)-coordinates, and a Boolean flag that indicates whether or not an image was moved on the canvas. The activeCanvas includes the ability to zoom and pan across the canvas as well as an image magnifier that allows users to see an enlarged version of the currently selected image. It also includes the ability to group select multiple images to quickly arrange images on screen as the user desires according to the task at hand.

Via a representational state transfer (RESTful) interface, the user sends the JSON file, containing new (x, y)-coordinates and the updated "moved" Boolean flag for each image, to the server. The server analyzes only the images the user moved, and computes a new location

for all images on the screen according the distance-to-anchors diffusion map embedding described in Section 5.2.3. The updated (x, y)-coordinates are placed in the JSON, and that information is pushed back to the web-interface. Images on the screen smoothly translate to the new positions calculated by the server. The full translation takes about 1.5 seconds, providing the user with a visual trace of the changes. The user then adjusts additional images according to their tasks, based on the new positions, to provide additional input to the prediction algorithm. The cycle repeats until the user is satisfied with all the positions of the images on the screen, and then the user may export the JSON to another application for further analysis such as building a classifier. Additionally, users can upload supplementary images as part of the main workflow, adding additional data to the layout if needed.

5.2.2. Feature Selection. Feature selection for computer vision is a challenging problem. There are many methods for finding relevant features in images and creating feature vectors to represent each image. Some of the most relevant work in this area focuses on keypoint detection and aggregation of similar keypoints to identify similarities across images [66]. Instead of trying to find the best keypoint detector for our data, we chose to represent each image with a 4096-dimensional feature vector that is computed as the output of Sermanet *et al.*'s convolutional neural network called OverFeat [61]. OverFeat was exhaustively trained to identify object boundaries on a massive corpus of real-life images from the ImageNet Large Scale Visual Recognition Challenge [55]. It turns out that this training also acts as a very good general feature descriptor for image data [62]. We found that when we performed a qualitative evaluation of the OverFeat features against standard image descriptors, the OverFeat features provided better separation of different particle images.

5.2.3. Dimensionality Reduction. As a result of the mapping from the original pixel space of the images to the feature space via the convolutional neural network, the SEM data it may be possible for pathologists to discern the existence of the relevant clusters in $\mathbb{R}^{4096}$. However visualization in 4096 dimensions is not feasible, so we attempt to preserve class relations in a reduced dimension visualization. Most low dimensional mappings look to preserve pairwise relationships between all data points either locally or globally under the assumption is that little is known about the data itself. Conversely, our application domain is centered around a human operator who has working knowledge of the data, and some relevant relationships. The operator's goal is to explore other hypothetical relationships in the data or to quickly organize the images with respect to the known, relevant information rather than to all possible information.

With this task in mind, we look to display points based on their relationship to user selected clusters of data. Starting from a naive 2-dimensional layout of the image data, a user specifies a few small collections of related images by moving them into separate, spatially contained regions of the visualization. The initial layout presented to the user could be as simple as a tessellation of the images or randomly generated (x, y)-coordinates. We would like to illustrate the improvements that our adaptive method for visual sort and summary provides over existing techniques, so we generate our initial layout using the well-known nonlinear embedding technique diffusion map [9].

Diffusion map constructs a Gaussian kernel from the distances between data points; a technique that works equally well in Euclidean space or on a manifold equipped with a metric. The initial description of diffusion map involved point clouds in Euclidean space, and thus used Euclidean distance between points as a measure of similarity. Here data has been mapped to points on a Grassmann manifold so distances for this kernel will are

measured using the minimum principal angle between two subspaces. As described by Björck and Golub [7], principal angles can be computed for subspaces of different dimensions, and specific to our application it will work for measuring the angle between one feature vector and a subspace generated from the span of a collection of feature vectors.

Algorithm 19 t-step diffusion map [9]

function DIFFUSIONMAP(Y, ϵ, t)
  Ensure Y is an $N \times N$ similarity matrix.
  Ensure $\epsilon > 0$ ▷ ϵ controls the width of the Gaussian.
  Ensure $t > 0$ ▷ t-step diffusion process.
  $W_{i,j} \leftarrow \exp(\frac{-Y_{i,j}}{\epsilon})$ ▷ Compute the Gaussian kernel.
  $d_i \leftarrow \sum_{j=1}^{N} W_{i,j}$ ▷ Compute degree
  $L_{i,j} \leftarrow \frac{W_{i,j}}{d_i}$ ▷ Compute normalized graph Laplacian
  $L_{i,j} \leftarrow \frac{L_{i,j}}{(d_i d_j)}$ ▷ Approximate Laplace-Beltrami operator
  $D_{i,j} \leftarrow \sum_{j=1}^{N} L_{i,j}$ ▷ Compute sampling density
  $M \leftarrow D^{-1}L$ ▷ Re-normalize
  $A \leftarrow M^t$ ▷ Compute t-step diffusion probabilties
  $\Lambda V \leftarrow AV$ ▷ Eigendecomposition of M
  Ensure $V = [v_1|v_2|\ldots|v_N]$ are ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$
  for i = 1 ...N **do**
    $y_i \leftarrow [v_2(i)|v_3(i)]^T$ ▷ $y_i \in \mathbb{R}^2$
  return $\{y_i\}_{i=1}^{N}$ ▷ 2-dimensional diffusion map coordinates

This Gaussian kernel is used to compute a normalized graph Laplacian, and each row of the Laplacian is rescaled based on the sampling density so as to find the 1-step transition probabilities. The eigenvectors corresponding to the largest magnitude eigenvalues of this matrix then represent the coordinates of the low dimensional embedding, with the exception that the first eigenvector is constant and is thus dismissed. Figure 5.2 shows the initial layout of the CDC images using a standard 1-step diffusion map.

The dimensionality reduction performed in the INStINCt framework is cued by user-created clusters. Thus our description here will follow this process. As an example, suppose a user manually creates three small clusters of related images by moving them into distinct,

spatially constrained areas of the activeCanvas as is shown by the circled images in Figure 5.3. The interaction with the activeCanvas is recorded using a JSON file that indicates the new (x, y)-coordinates of the moved images as well as the Boolean flag showing that they have been touched. This file can be exported from the activeCanvas if further analysis and computations are desired.

Because of a intentional limitation in the architecture, we are not allowed knowledge of the previous locations of the images. Thus we cannot determine the user's intentions strictly based on an image's updated coordinates. For example, we would not know that two images have been moved closer together from the information in the JSON file. We would only know that both images were moved and their current coordinates. Instead, we interpret the human operators actions by clustering the images that have been moved based on their 2-dimensional coordinates. For this task we employ the Density-based spatial clustering of applications with noise (DBSCAN) algorithm of Ester *et al.* [15]. The DBSCAN method looks for areas of high density in the data to create clusters. DBSCAN was chosen because it does not require knowledge about the number of clusters ahead of time, it can find clusters of arbitrary shape, and it allows for some points to remain unclustered if they are not located in dense regions. The final point is advantageous because sometimes the user wishes to inject class information by pushing dissimilar images further apart, rather than by grouping similar images. Thus the INStINCt framework does not require that an image movement away from a cluster necessarily forces that image into another cluster. One additional note, the DBSCAN method assumes that clusters are of relatively equal size, so skewed group sizes generated by a user may yield less desirable results. Pseudocode implementing the DBSCAN method can be found in Algorithm 20.

Algorithm 20 DBSCAN Algorithm [15]

function DBSCAN($\mathcal{Y}, \{y_j\}_{j\in\mathcal{Y}}, \epsilon$, MinPts)
  Ensure $\mathcal{Y} \subseteq \mathcal{X}$ ▷ Index set of images that were moved by user
  Ensure $y_j \in \mathbb{R}^2$ for all $j \in \mathcal{Y}$ ▷ 2-dimensional coordinates of moved images
  Ensure $\epsilon > 0$ ▷ Upper bound on neighborhood diameter
  Ensure MinPts≥ 0 ▷ Lower bound cluster size
  $i \leftarrow 0$
  $C_i \leftarrow \{\}$ ▷ Create an empty index set for noise points
  for $j \in \mathcal{Y}$ **do**
    if y_j has been visited **then**
      Continue to next point
    Mark y_j as visited
    NbrPts $\leftarrow \{k \in \mathcal{Y} | d(y_j, y_k) \leq \epsilon\}$ ▷ Indices of points near y_j
    if |NbrPts| $<$ MinPts **then**
      Mark y_j as noise
    else
      $i \leftarrow$ Next Cluster
      $C_i \leftarrow \{C_i \cup j\}$ ▷ Add the index of y_j to cluster i
      for $p \in$ NbrPts **do**
        if y_p has not been visited **then**
          Mark y_p as visited
          NbrPts$^{(p)} \leftarrow \{k \in \mathcal{Y} | d(y_p, y_k) \leq \epsilon\}$ ▷ Indices of points near y_p
          if |NbrPts$^{(p)}$| $\geq$ MinPts **then**
            NbrPts$\leftarrow \{\text{NbrPts} \cup \text{NbrPts}^{(p)}\}$ ▷ Combine the indices of points
        if p is not in any cluster **then**
          $C_i \leftarrow \{C_i \cup p\}$
  $K \leftarrow \max(\{i\})$
  return $\{C_i\}_{i=0}^{K}, K$ ▷ Sets of cluster indices and total number of clusters

The DBSCAN clustering step determines which images a user thinks are similar. The inferred knowledge from the clusters is then used to create a subspace "anchor" as a representative for each similarity group. An anchor consists of a subspace average of the feature vectors for the images in that cluster. We impose the constraint that the space spanned by each anchor must not overlap with the anchors for any of the other clusters. Additionally, moved images that were not assigned to a cluster are represented in the anchor of the nearest cluster with a reduced weight proportional to their distance from that group.

For instance, let $\{x_i\}_{i\in\mathcal{X}}$ be a collection of data with $x_i \in \mathbb{R}^{4096}$ for all i, and let $\{y_i\}_{i\in\mathcal{X}}$ be the coordinates of these feature vectors in an initial, arbitrary 2-dimensional spatialization. Assume that a human operator moved some images into two dense clusters, and moved a few other images into a non-clustered region away from those sets. The index set of all moved images is $\mathcal{Y}$ and their coordinates are updated for the vectors in the set $\{y_i\}_{i\in\mathcal{Y}}$. Let $\mathcal{C}_1 \subseteq \mathcal{Y}$ and $\mathcal{C}_2 \subseteq \mathcal{Y}$ be index sets for the elements of cluster 1 and 2 respectively so that $\{x_i\}_{i\in\mathcal{C}_1}$ and $\{x_i\}_{i\in\mathcal{C}_2}$ are the collections feature vectors for each cluster. Let $\mathcal{C}_0 \subseteq \mathcal{Y}$ be an index set that denotes the images that were moved, but not placed in a cluster by DBSCAN, so that $\{x_i\}_{i\in\mathcal{C}_0}$ is the collection of feature vectors associated with these images, and we have $\mathcal{Y} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_0$. Finally, let $\mathcal{C}_0^{(1)} \subseteq \mathcal{C}_0$ be a subset of those indices corresponding to the points that are closer in 2-dimensional Euclidean distance to cluster 1 than cluster 2 and vice versa for $\mathcal{C}_0^{(2)}$. The set of feature vectors associated with these points would then be $\{x_i\}_{i\in\mathcal{C}_0^{(1)}}$ and $\{x_i\}_{i\in\mathcal{C}_0^{(2)}}$ respectively.

The anchor U_1 for cluster 1 is computed as follows. Associate a weight of $\lambda_i = 1$ with each element of $\{x_i\}_{i\in\mathcal{C}_1}$ and a weight of

$$\lambda_i = \frac{1}{\min\{\|y_i - y_j\|_2 \text{ for } j \in \mathcal{C}_1\}} \tag{78}$$

with each element of $\{x_i\}_{i\in\mathcal{C}_0^{(1)}}$. The anchor U_1 is then computed as the weighted flag mean of this input data using the method described by Draper *et al.* [12]. The anchor U_2 is computed for cluster 2 in an analogous fashion. Algorithm 21 contains pseudocode for computing the anchor of one cluster.

Using these subspace anchors, the principal angle between each feature vector in the data set and each of these anchors are computed. Let N be the number of points in the data set and let K be the number of clusters that the human operator has formed. Suppose Q is a

Algorithm 21 Compute an anchor for an INStINCt cluster

function InstinctAnchor($\mathcal{X}, \mathcal{C}_k, \mathcal{C}_0^{(k)}, \{x_i\}_{i\in\mathcal{X}}, \{y_i\}_{i\in\mathcal{X}}$)
  Ensure $x_i \in \mathbb{R}^{4096}$ for all $i \in \mathcal{X}$. ▷ Feature vectors of full dataset
  Ensure $y_i \in \mathbb{R}^2$ for all $i \in \mathcal{X}$ ▷ 2-dimensional coordinates of all images
  for $i \in \mathcal{C}_k$ **do**
    $\lambda_i \leftarrow 1$
    $[x_i] \leftarrow$ Orthonormal basis for span$\{x_i\}$
  for $i \in \mathcal{C}_0^{(k)}$ **do**
    $\lambda_i \leftarrow \frac{1}{\min\{\|y_i - y_j\|_2 \text{ for } j\in\mathcal{C}_k\}}$
    $[x_i] \leftarrow$ Orthonormal basis for span$\{x_i\}$
  $\mathcal{D} \leftarrow \{[\lambda_i x_i]\}_{i\in\mathcal{C}_k} \cup \{[\lambda_i x_i]\}_{i\in\mathcal{C}_0^{(k)}}$ ▷ Set of weighted subspaces to be averaged
  $\left\{\left[u^{(1)}\right], \left[u^{(1)}|u^{(2)}\right], \ldots, \left[u^{(1)}|\ldots|u^{(r)}\right]\right\} \leftarrow \boldsymbol{\mu}_{pF}(\mathcal{D})$ ▷ Weighted flag mean of $\mathcal{D}$
  return $[u_1|u_2|u_3]$ ▷ 3-dimensional element of $\boldsymbol{\mu}_{pF}$

square, symmetric matrix of dimension $(N + K) \times (N + K)$. Populate the elements in the lower left $K \times N$ block of Q, and symmetrically the upper right $N \times K$ block of Q, with the principal angles between the feature vectors and the anchors. The values in the upper left $N \times N$ submatrix are set to a small constant, α_1, and the values lower right $K \times K$ submatrix are set to a relatively large constant α_2. From this symmetric matrix, we compute the 1-step diffusion map to get the updated coordinates of the visualization, $\{\tilde{y}_i\}_{i\in\mathcal{X}}$. The pseudocode for a full INStINCt update is shown in Algorithm 22.

The result of setting the pairwise distances of the images, i.e. the upper left block of Q, to a small constant is that images want to collapse to a single point in the diffusion map embedding. Only their relative distances to the anchors pull them apart. Similarly, the large constant value set for the lower right block of Q attempts to push the distinct clusters away from each other. As we will show in the following Section (and in Figure 5.4), this provides something like a gradient between the different types of particles in our dataset.

Algorithm 22 INStINCt Visualization Update

function InstinctUpdate($\mathcal{X}, \mathcal{Y}, \{x_i\}_{i\in\mathcal{X}}, \{y_i\}_{i\in\mathcal{X}}, \alpha_1, \alpha_2$)
  Ensure $\mathcal{Y} \subseteq \mathcal{X}$ ▷ Index set of images that were moved by user
  Ensure $x_i \in \mathbb{R}^{4096}$ for all $i \in \mathcal{X}$. ▷ Feature vectors of full dataset
  Ensure $y_i \in \mathbb{R}^2$ for all $i \in \mathcal{X}$ ▷ 2-dimensional coordinates of all images
  Ensure $\alpha_1 << \alpha_2$
  $\{\mathcal{C}_i\}_{i=0}^{K}, K \leftarrow \text{DBSCAN}(\mathcal{Y}, \{y_j\}_{j\in\mathcal{Y}}, \frac{\sqrt{2}}{3}, 3)$ ▷ Do DBSCAN on moved images
  for $k = 1 \dots K$ **do**
    $\mathcal{C}_0^{(k)} \leftarrow \{i | y_i \text{ is closest to cluster } \mathcal{C}_k\}$
    $[U_k] \leftarrow \text{InstinctAnchor}(\mathcal{X}, \mathcal{C}_k, \mathcal{C}_0^{(k)}, \{x_i\}_{i\in\mathcal{X}}, \mathcal{X}, \{y_i\}_{i\in\mathcal{X}})$
  $N \leftarrow |\mathcal{X}|$
  $A \leftarrow N \times N$ matrix with all values equal to α_1
  Diagonal of $A \leftarrow 0$
  $B \leftarrow K \times K$ matrix with all values equal to α_2
  Diagonal of $B \leftarrow 0$
  $C \leftarrow N \times K$ matrix of zeros
  for $i = 1 \dots N$ **do**
    for $k = 1 \dots K$ **do**
      $C(i,k) \leftarrow d([x_i], [U_k])$ ▷ Geodesic distance via Algorithm 6
  $Q \leftarrow \begin{bmatrix} A & C^T \\ C & B \end{bmatrix}$ ▷ Construct similarity matrix
  $\{\tilde{y}_i\}_{i=1}^{N+K} \leftarrow \text{DiffusionMap}(Q, 1, 1)$ ▷ 1-step diffusion map
  return $\{\tilde{y}_i\}_{i=1}^{N}$ ▷ Updated 2-dimensional coordinates

5.3. Qualitative Results

In our application domain we assume that the human operator has some knowledge about the data, and they are looking to explore other hypothetical relationships or to organize the data relative to this known information rather than to all the information available. Thus, we are only looking to display points based on their relationship to some user selected clusters of data. Figure 5.2 shows an initial layout of the CDC data using a standard 1-step diffusion map. From this initial layout, the user manually creates three small clusters of related images by moving them in the INStINCt framework, as is shown by the circled images in Figure 5.3. The purpose of the clustering step is to determine which images the user thinks are related. Once that information is gathered, we create an "anchor" to represent each cluster. The anchor consists of an average of the feature vector representations of the images in that

FIGURE 5.2. Initial diffusion map layout of the CDC data.

FIGURE 5.3. CDC data with three user created clusters of images.

cluster. The remaining images are arranged according to their distances to those anchors. The results of the associated update is shown in Figure 5.4.

FIGURE 5.4. CDC data with three clusters after the distance-to-anchor diffusion map update.

These initial results using clustering indicate there is a strong correlation between shape and texture and particle type. However, this data is missing labels and requires an approach that combines clustering on feature data with input from the user to perform semi-automated classification on particle types.

5.4. SUMMARY

This chapter presents a new method for user analysis of particle image data produced by SEM. The work presented in this chapter represents an exploration into an adaptive data visualization technique through the representation of high dimensional spaces. The results appear to reflect semantic relationships in a way that is hopefully valuable to human operators. Users indicate relationships via the positions chosen on the client canvas, which can be indicative of shape, size, texture. The novel contributions to this dissertation are mostly contained in Section 5.2.3 and include the method for dimensionality reduction that

preserves distances between data points and user selected clusters, the practical use of the flag mean as a stand-in for a cluster of data points, and the use of principal angles between subspaces as a measure of similarity for a diffusion map embedding.

There are a number of directions for extending this research in future work. First, we have not compared our representation to any previous methods for visualization. This is because many of the algorithms for visualization are not publicly available, but more specifically none of the previous methods would work out of the box in the context described, as they have largely been implemented for data living in a Euclidean space. Generalizing these methods for subspace data would allow for a fair comparison of methods. We would also like to incorporate multiple feature vectors for each particle image. For the case of the data set at hand, each particle has been imaged exactly once and therefore corresponds to a single feature vector. However, the method implemented in INStINCt is flexible enough to handle the span of multiple feature vectors as a representative for a single particle. This situation could arise if the particles were additionally analyzed as high-resolution secondary electron images. We also would like to take into account the scale of each particle to incorporate size to our analysis. Finally, we would like to extend this proof-of-concept to include a database that would enable more flexibility for saving user input, such as image groupings, labels, and possible explicit relationships.

CHAPTER 6

CONCLUSION

6.1. Contributions

This dissertation focused on developing theory and algorithms for pattern recognition and signal processing on Grassmann manifolds. Specifically we made the following contributions:

- We introduced a parametrization for points on a Schubert variety that allows us to reformulate a Schubert variety domain restriction as a penalty term.
- The parametrization allowed us to implement a penalty method algorithm based around a conjugate gradient method for Grassmann manifolds to descend onto the solution of the Schubert variety constrained averaging problem through a sequence of infeasible points that converge to a locally optimal solution.
- We introduced subspace constrained averaging as an alternative method for finding domain-restricted averages of Grassmannian data, and proved that the optimal solution to this problem can be found algebraically.
- We demonstrated the similarity and difference between the Schubert variety constrained average and the subspace constrained average using a novel decomposition of affine patches of the tangent space at a point.
- We described a novel algorithm for detecting targets of interest in long-wave infrared hyperspectral images, and provided evidence that it outperforms competing detection algorithms when the signal-to-noise-plus-interference ratio of the data is very low.
- We identified a method for generating useful synthetic data for evaluating hyperspectral target detection methods.

- We modified the popular diffusion map embedding to create an updatable visualization for inferring relationships between image data.
- We provided simple and thorough algorithms for implementing all of the techniques described in this dissertation.

6.2. Future Work

There are a number of directions that this work can be advanced in the future. With respect to Chapter 3, we would first like to extend the theory relating to the Schubert variety constrained averaging problem. Even when the subspace constrained average is not the optimum of Schubert variety constrained problem, there is evidence that the optimum is nearby. What does the solution look like in a general case? Can we describe how close the initial guess must be for the penalty method and conjugate gradient algorithm to descend on an actual local optimum? Can we use the tangent space decomposition of affine patches to create a more robust descent algorithm for generic examples? Along with these theoretical questions, we would also like to implement this constrained averaging in the applications described in Section 3.1.

For the hyperspectral image analysis done in Chapter 4, the algorithm described for identifying targets relies on multiple images of the same scene and pixelwise correspondence across the images. This requirement is sometimes easily satisfied, but in other circumstances nearly impossible to meet. An algorithm was developed for this dissertation as a snapshot method for doing target detection with the flag mean that only requires a single image of the scene. The algorithm attempts to better approximate the observed spectrum in the image by reconstructing the the library spectrum for the target from a small number of in-scene endmembers using sparse canonical correlation analysis. The algorithm was not

presented in this dissertation because some real issues of practicality remained unresolved. The algorithms tries to approximate a the laboratory *radiance* spectrum using the real-life *absorption* spectrum in an image. It isn't clear that this approximation should exist as a sparse linear combination of spectra from the scene if the absorption spectrum is saturated. Despite initially promising results, the final prognostication was that the space spanned by these endmembers was typically too large to be useful. It would often include the target spectrum, but would introduce a large number of false positives from other spectra that were close to the identified subspace. Improving this snapshot method or finding a new solution to the single image problem would prove very valuable in practical detection scenarios.

The work in Chapter 5 is largely self-contained, and the code for the active canvas that displays the images at the coordinates computed by the distance-to-anchors diffusion map is not freely available. Thus further work in that direction would require the recreation of some existing material. For those involved with the INStINCt project however, the framework developed leaves ample room for advancement. The distance-to-anchors diffusion map embedding was computing using geodesic distances and flag means, although the data examined is only represented by 1-dimensional subspaces. Fattening those representations with additional images or different measurements of the same particles would significantly improve the utility of the adaptive visualization because there would be more features for the averages to identify similarity within. This functionality is already coded, and just requires appropriate data to exploit it.

The theory and applications developed in this dissertation broaden the toolset for performing pattern recognition and signal processing on Grassmann manifolds. We view this work as a starting point for further exploration into Schubert varieties as an alternative means of including variability in a model rather than identifying invariant features. The

techniques described in this dissertation are applicable to a wide range of big data problems given that the flag and Grassmannian representations encode large volumes of data as a single point. This method of including variability tends to produce more robust algorithms, and we are thus better able to leverage the geometric relationships contained in the original data.

The Matlab code implementing the experiments and algorithms described in this dissertation (with the exception of some portions of Chapter 5) will be made publicly available at www.tmarrinan.com for transparency and reproducibility.

BIBLIOGRAPHY

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80:199–220, 2004. 23

[2] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 14

[3] John B Adams, Milton O Smith, and Alan R Gillespie. Imaging spectroscopy: Interpretation based on spectral mixture analysis. *Remote geochemical analysis: Elemental and mineralogical composition*, 7:145–166, 1993. 55

[4] William K Allard, Guangliang Chen, and Mauro Maggioni. Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis*, 32(3):435–462, 2012. 28

[5] E. Begelfor and M. Werman. Affine invariance revisited. In *CVPR*, volume 2, pages 2087 – 2094, 2006. 1, 12, 15, 23

[6] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001. 65

[7] A. Björck and G.H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973. 15, 16, 25, 72

[8] E. Chong and S. Zak. *An introduction to optimization*, volume 76. John Wiley & Sons, 2013. 34

[9] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 65, 69, 71, 72

[10] Vin De Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, pages 721–728, 2003. 69

[11] Yadolah Dodge and Valentin Rousson. Multivariate L1 mean. *Metrika*, 49(2):127–134, 1999. 24

[12] B. Draper, M. Kirby, J. Marks, T. Marrinan, and C. Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014. 27, 75

[13] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis and Applications*, 20(2):303–353, 1998. 10, 11, 17, 32

[14] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, and Chris North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011. 67

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 68, 73, 74

[16] Dennis F Flanigan. Hazardous cloud imaging: a new way of using passive infrared. *Applied optics*, 36(27):7027–7036, 1997. 53

[17] P Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1 Suppl):S143, 2009. 24, 25

[18] William Fulton and Piotr Pragacz. *Schubert varieties and degeneracy loci*. Springer, 1998. 20

[19] Torin Gerhart, Justin Sunu, Lauren Lieu, Ekaterina Merkurjev, Jen-Mei Chang, Jérôme Gilles, and Andrea L Bertozzi. Detection and tracking of gas plumes in lwir hyperspectral video sequence data. In *SPIE Defense, Security, and Sensing*, pages 87430J–87430J. International Society for Optics and Photonics, 2013. 61

[20] Michel Grutter, Roberto Basaldud, Edgar Flores, and Roland Harig. Optical remote sensing for characterizing the spatial distribution of stack emissions. In *Advanced Environmental Monitoring*, pages 107–118. Springer, 2008. 53

[21] Rami Hagege and Joseph M Francos. Parametric estimation of affine transformations: An exact linear solution. *Journal of Mathematical Imaging and Vision*, 37(1):1–16, 2010. 29

[22] Rami R Hagege and Joseph M Francos. Universal manifold embedding for geometrically deformed functions. *IEEE Transactions on Information Theory*, 62(6):3676–3684, 2016. 29

[23] JBS Haldane. Note on the median of a multivariate distribution. *Biometrika*, 35(3-4):414–417, 1948. 24

[24] Nathan Oken Hodas and Alex Endert. Adding semantic information into data models by learning domain expertise from user interaction. *arXiv preprint arXiv:1604.02935*, 2016. 69

[25] Thomas D Holmes, Raymond A Guilmette, Yung Sung Cheng, Mary Ann Parkhurst, and Mark D Hoover. Aerosol sampling system for collection of capstone depleted uranium particles in a high-energy environment. *Health physics*, 96(3):221–237, 2009. 67

[26] Peter J Huber. *Robust statistics*. Springer, 2011. 25

[27] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989. 59

[28] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009. 66

[29] Elizabeth Jurrus, Nathan Hodas, Nathan Baker, Tim Marrinan, and Mark D Hoover. Adaptive visual sort and summary of micrographic images of nanoparticles for forensic analysis. In *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*, pages 1–6. IEEE, 2016. 1, 65

[30] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977. 23

[31] Michael Kirby and Lawrence Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1):103–108, 1990. 1

[32] Shawn Kraut, Louis L Scharf, and L Todd McWhorter. Adaptive subspace detectors. *Signal Processing, IEEE Transactions on*, 49(1):1–16, 2001. 53, 55

[33] Robert Kroutil, Paul Lewis, Mark Thomas, Timothy Curry, David Miller, Roger Combs, and Alan Cummings. Emergency response chemical detection using passive infrared spectroscopy. *SPIE Newsroom*, 2006. 53

[34] Kenneth M Krupka, Mary Ann Parkhurst, Kenneth Gold, Bruce W Arey, Evan D Jenson, and Raymond A Guilmette. Physicochemical characterization of capstone depleted uranium aerosols iii: morphologic and chemical oxide analyses. *Health physics*, 96(3):276–291, 2009. 67

[35] Naval Research Laboratory. Fabry-pérot interferometer sensor data set. In *Algorithms for Threat Detection Data Repository (Not publicly available)*. Colorado State University, 2008. 57

[36] Scotland C Leman, Leanna House, Dipayan Maiti, Alex Endert, Chris North, et al. Visual to parametric interaction (v2pi). *PloS one*, 8(3):e50474, 2013. 67

[37] Y.M. Lui, J.R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR*, pages 833–839, 2010. 1

[38] Glenn I Lykken and Berislav Momcilovic. Comment on the capstone depleted uranium (du) aerosol characterization and risk assessment study. *Health physics*, 98(1):77, 2010. 68

[39] J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics: Texts and References Section. Wiley, 1999. 35

[40] D Manolakis, S Golowich, and R DiPietro. Long-wave infrared hyperspectral remote sensing of chemical clouds: A focus on signal processing approaches. *Signal Processing Magazine, IEEE*, 31(4):120–141, 2014. 53, 55

[41] D Manolakis, M Pieper, E Truslow, T Cooley, M Brueggeman, and S Lipson. The remarkable success of adaptive cosine estimator in hyperspectral target detection. In *SPIE Defense, Security, and Sensing*, pages 874302–874302. International Society for Optics and Photonics, 2013. 55

[42] Dimitris Manolakis, David Marden, and Gary A Shaw. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1):79–116, 2003. 55

[43] Dimitris Manolakis and Gary Shaw. Detection algorithms for hyperspectral imaging applications. *Signal Processing Magazine, IEEE*, 19(1):29–43, 2002. 55, 56

[44] Dimitris Manolakis, Christina Siracusa, and Gary Shaw. Hyperspectral subpixel target detection using the linear mixing model. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(7):1392–1409, 2001. 53

[45] T. Marrinan, J.R. Beveridge, B. Draper, M. Kirby, and C. Peterson. Flag manifolds for the characterization of geometric structure in large data sets. In *Numerical Mathematics and Advanced Applications-ENUMATH 2013*, pages 457–465. Springer, 2015. 1, 23

[46] T. Marrinan, J.R. Beveridge, B. Draper, M. Kirby, and C. Peterson. Flag-based detection of weak gas signatures in long-wave infrared hyperspectral image sequences. In *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2016. 1, 54

[47] Tim Marrinan, Bruce Draper, J Ross Beveridge, Michael Kirby, and Chris Peterson. Finding the subspace mean or median to fit your need. In *CVPR*, pages 1082–1089. IEEE, 2014. 23

[48] D. Monk. The geometry of flag manifolds. *Proceedings of the London Mathematical Society*, 3(2):253–286, 1959. 20

[49] Hiroshi Murase and Shree K Nayar. Visual learning and recognition of 3D objects from appearance. *IJCV*, 14(1):5–24, 1995. 1

[50] Erin M O'Donnell, David W Messinger, Carl Salvaggio, and John R Schott. Identification and detection of gaseous effluents from hyperspectral imagery using invariant algorithms. In *Defense and Security*, pages 573–582. International Society for Optics and Photonics, 2004. 53

[51] M.A. Parkhurst, E.G. Daxon, G.M. Lodde, F. Szrom, R.A. Guilmette, L.E. Roszell, G.A. Falo, and C.B. McKee. Depleted uranium aerosol doses and risks: summary of us assessments. *Columbus, Ohio: Battelle Pres*, 2005. 67

[52] V. Patrangenaru and K.V. Mardia. Affine shape analysis and image analysis. *Proc. 22nd Leeds Ann. Statistics Research Workshop*, July 2003. 1

[53] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008. 35

[54] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 65

[55] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014. 70

[56] Adel A Sakla, Wesam A Sakla, and Mohammad S Alam. Hyperspectral target detection via discrete wavelet-based spectral fringe-adjusted joint transform correlation. *Applied optics*, 50(28):5545–5554, 2011. 59

[57] I. Santamaria, L.L. Scharf, C. Peterson, M. Kirby, and J. Francos. An order fitting rule for optimal subspace averaging. In *Statistical Signal Processing Workshop (SSP), 2016 IEEE*, pages 1–4. IEEE, 2016. 27

[58] Louis L Scharf. *Statistical signal processing*, volume 98. Addison-Wesley Reading, MA, 1991. 53

[59] Louis L Scharf and L Tood McWhorter. Adaptive matched subspace detectors and adaptive coherence estimators. In *Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on*, pages 1114–1117. IEEE, 1996. 53, 55

[60] Anthony N. Schwickerath. *Linear models, signal detection, and the Grassmann manifold*. PhD thesis, Colorado State University, 2014 Fall. 3, 20, 21, 22, 26

[61] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 66, 68, 70

[62] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 70

[63] Christopher G Small. A survey of multidimensional medians. *International Statistical Review/Revue Internationale de Statistique*, pages 263–277, 1990. 24

[64] Govert W Somsen and Tom Visser. Liquid chromatography/infrared spectroscopy. *Encyclopedia of Analytical Chemistry*, 2000. 53

[65] Justin Sunu, Jen-Mei Chang, and Andrea L Bertozzi. Simultaneous spectral analysis of multiple video sequence data for lwir gas plumes. In *SPIE Conference on Defense, Security, and Sensing*, 2014. 61

[66] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 70

[67] Wei Tang, Hui Xiong, Shi Zhong, and Jie Wu. Enhancing semi-supervised clustering: a feature projection perspective. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 707–716. ACM, 2007. 66

[68] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 65

[69] Pierre Tremblay, Simon Savary, Matthias Rolland, André Villemaire, Martin Chamberland, Vincent Farley, Louis Brault, Jean Giroux, Jean-Luc Allard, Éric Dupuis, et al. Standoff gas identification and quantification from turbulent stack plumes with

an imaging fourier-transform spectrometer. In *SPIE Defense, Security, and Sensing*, pages 76730H–76730H. International Society for Optics and Photonics, 2010. 53

[70] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011. 1

[71] Ziv Yavo, Joseph M Francos, Ignacio Santamaria, and Louis L Scharf. Estimating the mean manifold of a deformable object from noisy observations. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, pages 1–5. IEEE, 2016. 29

[72] K. Ye and L.-H. Lim. Schubert varieties and distances between subspaces of different dimensions. *Journal on Matrix Analysis and Applications*, 37(3):1176–1197, 2016. 3, 20, 22, 26