DISSERTATION

MEAN VARIANTS ON MATRIX MANIFOLDS

Submitted by

Justin D Marks

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2012

ABSTRACT

MEAN VARIANTS ON MATRIX MANIFOLDS

The geometrically elegant Stiefel and Grassmann manifolds have become organizational tools for data applications, such as illumination spaces for faces in digital photography. Modern data analysis involves increasingly large-scale data sets, both in terms of number of samples and number of features per sample. In circumstances such as when large-scale data has been mapped to a Stiefel or Grassmann manifold, the computation of mean representatives for clusters of points on these manifolds is a valuable tool.

We derive three algorithms for determining mean representatives for a cluster of points on the Stiefel manifold and the Grassmann manifold. Two algorithms, the normal mean and the projection mean, follow the theme of the Karcher mean, relying upon inversely related maps that operate between the manifold and the tangent bundle. These maps are informed by the geometric definition of the tangent bundle and the normal bundle. From the cluster of points, each algorithm exploits these maps in a predictor/corrector loop until converging, with prescribed tolerance, to a fixed point. The fixed point acts as the normal mean representative, or projection mean representative, respectively, of the cluster. This method shares its principal structural characteristics with the Karcher mean, but utilizes a distinct pair of inversely related maps. The third algorithm, called the flag mean, operates in a context comparable to a generalized Grassmannian. It produces a mean subspace of arbitrary dimension. We provide applications and discuss generalizations of these means to other manifolds.

ACKNOWLEDGEMENTS

I would like to thank Chris Peterson and Michael Kirby for their constant support and guidance during my graduate mathematics education. I also thank the Pattern Analysis Laboratory group for establishing warm community and fruitful collaboration, which have combined to nurture my research. I recognize the love and prayers of my parents, Jean and Jeff Marks, who have encouraged me in my scholastic pursuits from childhood. I praise Jesus for gifting me with a mind to comprehend mathematics, and for releasing moments of insight and inspiration.

TABLE OF CONTENTS

1. INTRODUCTION

In the introduction to this thesis, we provide background information about nonlinear manifolds. In particular, we highlight the geometry of matrix manifolds. We explore literature describing the geometry foundational to Stiefel and Grassmann manifolds, literature focused on the computation of manifold means, and literature relating to nonlinear manifold analysis. The geometric setting of matrix manifolds will be referenced for the construction of each of our mean representative algorithms.

1.1 Matrix Manifolds

This section briefly summarizes the geometric structure of matrix manifolds, the mathematical setting for this thesis. Specifically, we provide an explicit description of the tangent and normal spaces at each point of the Stiefel and Grassmann manifolds. These manifolds are frequently used to perform analysis on subspaces or pools of data points.

First, let us define a manifold. A n-dimensional *topological manifold* is a topological space which looks locally like n-dimensional Euclidean space. A *differentiable manifold* allows for defining tangent vectors and tangent spaces. *Riemannian manifolds* are differentiable manifolds that have a metric on the tangent space that allows measurements of distances, angles, etc. The Stiefel and Grassmann manifolds are examples of Riemannian manifolds, and we will describe the particular structure of their tangent spaces below.

1.1.1 Stiefel Manifold

The Stiefel manifold $St(n, p)$ is defined as the collection of all ordered p-tuples of orthonormal vectors in $\mathbb{R}^n$. Thus, points on the Stiefel manifold are in one-to-one correspondence with

$n \times p$ orthonormal matrices. For example, $St(3,2)$ includes the three distinct points $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, and $\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$.

We measure the distance between two points, X and Y, on the Stiefel manifold as $d_S^2(X,Y) = trace(X-Y)^T(X-Y)$. This is simply the standard Euclidean 2-norm naturally applied to Stiefel manifold matrices embedded in $\mathbb{R}^{n \times p}$.

Let P be a point on the Stiefel manifold $N = St(n,p)$ and let Y be the $n \times p$ orthonormal matrix corresponding to P. The tangent space to N at P is the vector space of all $n \times p$ matrices Δ such that $Y^T\Delta$ is skew symmetric, i.e. $Y^T\Delta + \Delta^T Y = 0$. The normal space to N at P is the vector space of matrices $\{YS\}$ where S is any $p \times p$ symmetric matrix. We find the geometric framework for this description in [11].

1.1.2 Grassmann Manifold

The Grassmann manifold $Gr(n,p)$ is defined as the collection of all p-dimensional subspaces of $\mathbb{R}^n$. For example, $Gr(3,2)$ is the set of all planes through the origin in $\mathbb{R}^3$. For computations, we utilize orthonormal matrix representatives for points on the Grassmannian. In other words, an $n \times p$ matrix Y, with $Y^TY = I$, is a representative for the p-dimensional subspace spanned by the columns of Y, denoted $[Y]$. Consider that $[Y] \in Gr(3,2)$ is the xy-plane for choices of orthonormal basis matrix Y such as $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$, or any other orthonormal matrix that has column space equal to the xy-plane.

Given two p-dimensional subspaces $[X]$ and $[Y]$ of $\mathbb{R}^n$, there are p principal angles $0 \leq \theta_1 \leq \theta_2 \leq ... \leq \theta_p \leq \frac{\pi}{2}$ between $[X]$ and $[Y]$. The principal angles may be computed as the inverse cosine of the singular values of X^TY, where X and Y are orthonormal matrix representatives for $[X]$ and $[Y]$, respectively. We measure the distance between two points $[X]$ and $[Y]$ on the Grassmann manifold $Gr(n,p)$ as $d_G([X],[Y]) = \sqrt{\sum_{i=1}^{p} \theta_i^2}$. This is known as the arc length metric and imposes a particular, commonly used geometry on the Grassmann manifold.

Let V be a fixed p-dimensional subspace of $\mathbb{R}^n$ and let Y be an $n \times p$ orthonormal matrix whose column space is equal to V. Let $V = [Y]$ be the corresponding point on the Grassmann manifold $M = Gr(n,p)$. The tangent space to M at $[Y]$ can be identified with the vector space of all $n \times p$ matrices Δ such that $Y^T \Delta = 0$. The normal space to M at $[Y]$ can be identified with the vector space of matrices $\{YS\}$ where S is any $p \times p$ matrix. This characterization does not depend on the choice of representative Y for $[Y]$. The geometric framework for this description is presented in [11].

1.1.3 The Concept of Mean Representatives

Suppose we have a cluster of points on the Stiefel or Grassmann manifold. We desire to compute a mean representative for the cluster. While a mean within a vector space is easy to compute, performing the task on a matrix manifold is not obvious. The Karcher Mean is one algorithm to calculate a matrix manifold mean, which is described in 1.1.4. On the other hand, we constructed a suite of new manifold mean methods. We have devoted a chapter of this thesis to each of these algorithms. They are titled the normal mean, the projection mean, and the flag mean. After presenting our three new algorithms, we will offer a chapter containing applications and a computational complexity analysis. As a point of reference and comparison, we describe the Karcher mean in this introduction.

1.1.4 Karcher Mean

The Karcher mean [6] enables computation of a mean representative for a cluster of points on the Grassmann manifold. The algorithm exploits **Exp** and **Log** maps in a predictor/corrector loop until convergence to a fixed point. This fixed point minimizes the sum of the squared distances to the given points subject to the constraint of lying on the manifold. The key technique is to reduce the manifold problem to a vector space mean computation by utilizing tangent spaces to the manifold.

The Karcher mean $[\mu_{KM}] \in Gr(n,p)$ of C points $\{[X_i]\}_{i=1}^{C} \in Gr(n,p)$, $X_i^T X_i = I_p$, is the solution of

$$\underset{[Y]\in Gr(n,p)}{\operatorname{argmin}} \sum_{i=1}^{C} d_G([Y],[X_i])^2$$

Algorithm 1, borrowed from [6], is used to compute $[\mu_{KM}]$ and shares features with our algorithms. The *Gexp* and *Glog* maps, defined in [6], are a family of maps parametrized by the Grassmann manifold. At each point of the Grassmann manifold and for suitably defined neighborhoods, the *Gexp* and *Glog* maps are mutual inverses.

Algorithm 1 Calculate $[\mu_{KM}]$ on $Gr(n,p)$

```
Require: $[X_i] \in Gr(n,p), 1 \le i \le C$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  $\mu \Leftarrow X_1$
  repeat
    $\delta \Leftarrow \frac{1}{C}\sum_{i=1}^{C} Glog(\mu, X_i)$
    $\mu \Leftarrow Gexp(\mu, \delta)$
  until $||\delta||_F < \epsilon$
  $\mu_{KM} \Leftarrow \mu$
```

2. A NORMAL/TANGENT BUNDLE ALGORITHM FOR AVERAGING POINT CLOUDS ON GRASSMANN AND STIEFEL MANIFOLDS

In this paper, we derive iterative algorithms for determining representatives for a cluster of points on the Grassmann manifold and on the Stiefel manifold. Each of the algorithms depend upon a pair of maps parametrized by the underlying manifold. Based on a characterization of the tangent and normal space at each point, the pair of maps allow one to move from the manifold to a given tangent space and vise-versa. Iterating in a manner similar to the Karcher mean algorithm, we observe rapid convergence to a fixed point which acts as a mean for the cluster.

2.1 Introduction

In computational data analysis it is common to appeal to established geometric theory to develop algorithms and process data [29, 7, 26, 17, 10, 19, 14, 25, 1]. As a particular example, the Grassmann manifold and Stiefel manifold are matrix manifolds which have found use as a setting in which to classify and make comparisons between large data sets [18, 9, 28, 11, 12, 24]. Given a cluster of points on either of these manifolds, we describe procedures to find a single point, on the manifold, which represents the cluster. The algorithms for the Stiefel manifold, $St(n, p)$, arise from its natural embedding in $\mathbb{R}^{n\times p}$ as the set of all full rank orthonormal $n \times p$ matrices. The algorithms for the Grassmann manifold, $Gr(n, p)$, arise from its realization as a quotient space of $St(n, p)$. These models of the manifolds lead to an explicit description of the normal and tangent spaces at each point. These explicit descriptions lead to maps which are used to build iterative algorithms for each manifold. For each algorithm, the input is a cluster of points and the output is a single point which arises as a fixed point of the iteration. The fixed point can be taken as a mean representative of the cluster.

2.2 Mathematical Background

This section summarizes the mathematical setting that we consider. In particular, it provides an explicit description of the tangent and normal spaces at each point of the Grassmann and Stiefel manifolds.

2.2.1 Stiefel Manifold

The Stiefel manifold $St(n, p)$ is defined as the collection of all ordered p-tuples of orthonormal vectors in $\mathbb{R}^n$. Thus, points on the Stiefel manifold are in one-to-one correspondence with $n \times p$ orthonormal matrices. We measure the distance between two points, X and Y, on the Stiefel manifold as $d_{\mathcal{S}}^2(X, Y) = trace(X - Y)^T(X - Y)$.

Let Y be an $n \times p$ orthonormal matrix corresponding to a point on $St(n, p)$. The tangent space to $St(n, p)$ at Y is the vector space of all $n \times p$ matrices Δ such that $Y^T\Delta$ is skew symmetric, i.e. $Y^T\Delta + \Delta^T Y = 0$. The normal space to $St(n, p)$ at Y is the vector space of matrices $\{YS\}$ where S is any $p \times p$ symmetric matrix. The geometric framework for this description is presented in [11].

2.2.2 Grassmann Manifold

Let $\mathbb{R}$ denote the field of real numbers. The Grassmannian $Gr(n, p) = Gr_{\mathbb{R}}(n, p)$ is defined as the collection of all p-dimensional subspaces of $\mathbb{R}^n$. We can give $Gr(n, p)$ a geometric structure by expressing it as a *homogeneous space* for a matrix group. For instance, since points on $Gr(n, p)$ correspond to p-dimensional subspaces of $\mathbb{R}^n$, we can represent a point via an $n \times p$ orthonormal matrix. In other words, an $n \times p$ matrix Y, with $Y^T Y = I$, is a representative for the p-dimensional subspace spanned by the columns of Y, denoted $[Y]$. However, if Q is an orthonormal $p \times p$ matrix then Y and YQ represent the same point on $Gr(n, p)$ since they have the same column space. Thus there is a surjective map from $St(n, p)$ to $Gr(n, p)$ by mapping an orthonormal $n \times p$ matrix Y to its column space $[Y]$ and the fiber above $[Y]$ is $\{YQ \mid Q \in O(p)\}$.

Thinking of Y as the first p columns of an orthonormal $n \times n$ matrix, we can obtain the identification $Gr(n,p) = O(n)/(O(n-p) \times O(p))$ (where $O(n)$ denotes the orthogonal group of real, orthonormal, $n \times n$ matrices). More specifically, if we begin with a point in $O(n)$ with block structure $[Y|W]$, where W is an orthonormal $n \times (n-p)$ matrix, we can right multiply by any matrix of the form

$$\left[\begin{array}{c|c} O(p) & 0 \\ \hline 0 & O(n-p) \end{array}\right]$$

and generate a matrix in $O(n)$ for which the first p columns span the same subspace as Y.

Given two points $[X], [Y]$ on $Gr(n,p)$ there is a p-tuple of *principal angles*, $0 \leq \theta_1 \leq \theta_2 \leq ... \leq \theta_p \leq \frac{\pi}{2}$, between the corresponding subspaces V_X, V_Y. The first principal angle, θ_1, is defined as the minimum angle that occurs between vectors in V_X and vectors in V_Y. The first principal vectors $u_1 \in V_X, w_1 \in V_Y$ are a pair of unit length vectors achieving this minimum angle. The k^{th} principal angle is defined iteratively as the smallest angle that occurs between vectors in V_X and vectors in V_Y subject to the constraint of these vectors being orthogonal to all previously obtained principal vectors. In other words,

$$\theta_1 = \min\{\arccos\left(\frac{< u, w >}{|u| \cdot |w|}\right) | u \in V_X, w \in V_Y\}.$$

The first principal vectors $u_1 \in V_X, w_1 \in V_Y$ satisfy

$$\arccos(< u_1, w_1 >) = \theta_1, \|u_1\| = \|w_1\| = 1.$$

The k^{th} principal angle is defined by

$$\theta_k = \min\{\arccos\left(\frac{< u, w >}{|u| \cdot |w|}\right) | u \in V_X, w \in V_Y; u \perp u_j, w \perp w_j \text{ for } j < k\}.$$

The k^{th} principal vectors $u_k \in V_X, w_k \in V_Y$ satisfy

$$\arccos(< u_k, w_k >) = \theta_k, \|u_k\| = \|w_k\| = 1; u_k \perp u_j, w_k \perp w_j \text{ for } j < k.$$

The p principal angles between the subspaces V_X and V_Y are necessarily independent of the choice of basis for V_X and V_Y. It is a remarkable fact that the principal angles can be computed as the inverse cosine of the singular values of X^TY where X and Y are any orthonormal representatives for $[X]$ and $[Y]$. The independence of the singular values of X^TY from the choice of orthonormal matrix representatives of $[X]$ and $[Y]$ is a consequence of the following fact: If A, B are orthonormal $p \times p$ matrices and if $X' = XA, Y' = YB$ then the singular values of X^TY are equal to the singular values of X'^TY'. Details on the connection between the singular values of the matrix product and principal angles between corresponding subspaces can be found in [8]. For this paper, we define the distance between two points $[X]$ and $[Y]$ on the Grassmann manifold $Gr(n,p)$ as $d_{\mathcal{G}}([X],[Y]) = \sqrt{\sum_{i=1}^{p} \theta_i^2}$. Since the metric is defined in terms of the principal angles between V_X and V_Y and since principal angles are invariant under the action of the orthogonal group, the metric itself is orthogonally invariant.

Let Y be an $n \times p$ orthonormal matrix whose column space corresponds to a point $[Y]$ on $Gr(n,p)$. The tangent space to $Gr(n,p)$ at $[Y]$ is the vector space of all $n \times p$ matrices Δ such that $Y^T\Delta = 0$. The normal space to $Gr(n,p)$ at $[Y]$ is the vector space of matrices $\{YS\}$ where S is any $p \times p$ matrix. This characterization does not depend on the choice of representative Y for $[Y]$. Again, the geometric framework for this description is presented in [11].

2.2.3 Closest Orthonormal Matrix

Let T be a full rank $n \times p$ matrix with thin SVD given by $T = U\Sigma V^T$. As observed by [20], the closest orthonormal matrix to T, with respect to the *Frobenius* norm, is given by UV^T.

In other words

$$B = \underset{X^TX=I}{\arg\min} ||X - T||_F = UV^T.$$

To see this we rewrite the optimization problem as

$$B = \underset{X^TX=I}{\arg\max}\, tr(X^TT).$$

Through the Lagrange function, we obtain the solution

$$B = \frac{1}{2}T\Lambda^{-1} \tag{2.2.1}$$

where Λ is a full symmetric matrix of Lagrange multipliers.

Substitution of (2.2.1) into $B^TB = I$ yields $\Lambda^2 = \frac{1}{4}(T^TT)$, so $\Lambda = \frac{1}{2}(T^TT)^{\frac{1}{2}}$. Using $T = U\Sigma V^T$ and $U^TU = I_p$, we get

$$\Lambda = \frac{1}{2}(V\Sigma^2V^T)^{\frac{1}{2}} = \frac{1}{2}V\Sigma V^T.$$

Finally, from (2.2.1),

$$B = \frac{1}{2}T\Lambda^{-1} = \frac{1}{2}U\Sigma V^T(2V\Sigma^{-1}V^T) = UV^T.$$

To ensure this unique stationary point is a maximum for B, we compute the Hessian with respect to X, which is $-2\Lambda = -V\Sigma V^T$. As the eigenvalues of the Hessian are the negatives of the singular values of T, the Hessian is negative definite. Now, the second order sufficient condition states that we have a maximum for B, and thus a minimizer for the original objective function.

2.3 Manifold Normal Maps

For both the Grassmann and the Stiefel manifolds, this section describes explicit maps from "the tangent space to the manifold at P" to "a manifold neighborhood of P". One can view each parametrized family of maps as a continuously differentiable map from the "tangent bundle to the Grassmann (resp. Stiefel) manifold" to the "Grassmann (resp. Stiefel) manifold" whose restriction to each tangent space behaves well at the origin. Such maps have been referred to as *retraction* maps [22, 2, 4] and serve a useful role in building algorithms on manifolds. With suitably restricted domains, each of the maps in this section has an inverse which can also be described in explicit terms. We use each map, together with its inverse, to construct an iterative algorithm (in the spirit of the Karcher mean algorithm) that takes a cluster of points on a manifold as input and produces a single representative point on the manifold as output.

2.3.1 Grassmann Manifold Normal map

Let $[Y]$ be a point in $\mathcal{G} = Gr(n,p)$ (with Y an $n \times p$ orthonormal matrix). Let $T_{[Y]}$ be the tangent space to $\mathcal{G}$ at $[Y]$ and let $T_Y^* = T_{[Y]} + Y$. Thus if $T \in T_Y^*$ then we can write $T = Y + \Delta$ with $Y^T\Delta = 0$. In the following Proposition, we verify that T is full rank.

Proposition 2.3.1. *If $\Delta \in T_{[Y]}$ then $Y + \Delta$ has rank p.*

Proof. Consider the thin singular value decomposition (SVD), $Y + \Delta = U\Sigma V^T$, where U is an $n\times p$ orthonormal matrix, Σ is a $p\times p$ diagonal matrix, and V is a $p\times p$ orthogonal matrix. Note that $Y^T\Delta = 0$ and that $Y^TY = I_p$. Multiply each side by Y^T to obtain $I_p = Y^TU\Sigma V^T$. Since each matrix has rank at most p and since their product has rank p, we conclude that each matrix in the product has rank exactly p. In particular, $U\Sigma V^T$ has rank p. $\square$

Let T be an $n \times p$ matrix. Recall from 2.2.3 that if the thin SVD is $T = U\Sigma V^T$, then the closest orthonormal matrix to T is UV^T. With this in mind, define $R_{\mathcal{G},Y} : T_{[Y]} \to \mathcal{G}$ by

$$R_{\mathcal{G},Y}(\Delta) = [UV^T]$$

where $Y + \Delta = U\Sigma V^T$. Note that the map depends on Y instead of $[Y]$.

The image of this map is a neighborhood $Nb_{[Y]}$ of $[Y]$ on $\mathcal{G}$. Now we identify the inverse map $R^{-1}_{\mathcal{G},Y} : Nb_{[Y]} \to T_{[Y]}$. Let $[X]$ be a point in $Nb_{[Y]}$ and let $T^*_Y = T_{[Y]} + Y$. Since $R_{\mathcal{G},Y}$ maps points on T^*_Y to the nearest point on $\mathcal{G}$, the inverse map should be a normal vector to $\mathcal{G}$ at the point $[X]$. Thus we seek a normal vector XS, S of size $p \times p$, such that $XS = Y + \Delta$ with $\Delta \in T_{[Y]}$ (an intuitive picture is shown in Figure 2.1). This requires that $Y^T(XS - Y) = 0$, which implies $XS = X(Y^TX)^{-1}$. We note that Y^TX is nonsingular unless $[X]$ intersects the null space of $[Y]$ non-trivially. It follows that $R^{-1}_{\mathcal{G},Y}([X]) = X(Y^TX)^{-1} - Y$. Note that if A is a $p \times p$ orthogonal matrix, then $R^{-1}_{\mathcal{G},Y}([XA]) = XA(Y^TXA)^{-1} - Y = X(Y^TX)^{-1} - Y = R^{-1}_{\mathcal{G},Y}([X])$. In other words, the map is well defined on $\mathcal{G}$.

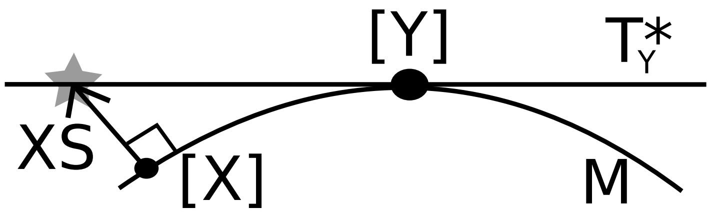

Fig. 2.1: Our derivation centers on the point of intersection of the shifted tangent space $T_{[Y]} + Y$ and the normal space to the Grassmann manifold at $[X]$.

In the following Proposition, we verify algebraically that $(R^{-1}_{\mathcal{G},Y} \circ R_{\mathcal{G},Y})$ is the identity map.

Proposition 2.3.2. *If $T \in T_{[Y]}$, then $(R^{-1}_{\mathcal{G},Y} \circ R_{\mathcal{G},Y})(T) = T$.*

Proof. Let $T \in T_{[Y]}$. Consider the SVD decomposition $T + Y = U\Sigma V^T$. Since T is in $T_{[Y]}$, we have $Y^TT = 0$ so $Y^T(T+Y) = I_p$ and $Y^TU = V\Sigma^{-1}$. Therefore

$$\begin{aligned}(R_{\mathcal{G},Y}^{-1} \circ R_{\mathcal{G},Y})(T) &= R_{\mathcal{G},Y}^{-1}([UV^T]) \\ &= UV^T(Y^TUV^T)^{-1} - Y \\ &= U(Y^TU)^{-1} - Y \\ &= U(V\Sigma^{-1})^{-1} - Y \\ &= T + Y - Y \\ &= T.\end{aligned}$$

$\square$

We will use the pair of maps $R_{\mathcal{G},Y}(T) = [UV^T]$ (with $T + Y = U\Sigma V^T$) and $R_{\mathcal{G},Y}^{-1}([X]) = X(Y^TX)^{-1} - Y$ in an iterative algorithm to produce the *Grassmann manifold normal mean* of a set of points on $\mathcal{G}$.

2.3.2 Stiefel Manifold Normal map

Parallel to the Grassmannian case, we let T_Y be the tangent space to $\mathcal{S} = St(n,p)$ at Y. Define $R_{\mathcal{S},Y} : T_Y \to \mathcal{S}$ by

$$R_{\mathcal{S},Y}(\Delta) = UV^T$$

where $Y + \Delta = U\Sigma V^T$ is the thin SVD. The image of $R_{\mathcal{S},Y}$ is a neighborhood Nb_Y of Y on $St(n,p)$.

We next formulate the inverse map $R_{\mathcal{S},Y}^{-1} : Nb_Y \to T_Y$. Let X be a point on Nb_Y. We define the map so that $R_{\mathcal{S},Y}^{-1}(X)$ is of the form $Y + \Delta$ for some $\Delta \in T_Y$ and is in the normal space to $\mathcal{S}$ at X. To achieve this, we require a normal vector XS, where S is a $p \times p$ symmetric matrix, such that $XS = Y + \Delta$. Skew symmetry generates the equation

$$Y^T(XS - Y) + (XS - Y)^TY = 0.$$

Noting that $Y^TY = I_p$ we have

$$MS + S^TM^T = 2I_p \tag{2.3.1}$$

where $M = Y^TX$. Pairing (2.3.1) with the symmetry equation $S - S^T = 0$ yields a system of equations that can be solved for $\hat{S}$. We then write

$$R^{-1}_{\mathcal{S},Y}(X) = X\hat{S} - Y.$$

The computation of $\hat{S}$ is as follows. Let $a^{(i)}$ and $a_{(i)}$ denote the i^{th} column and i^{th} row of A, respectively. Suppose $1 \leq i \leq j \leq p$. From (2.3.1) we obtain $m_{(i)}s^{(j)} + {s^{(i)}}^T {m_{(j)}}^T = 2\delta_{ij}$, or

$$m_{(i)}s^{(j)} + m_{(j)}s^{(i)} = 2\delta_{ij} \tag{2.3.2}$$

where δ_{ij} is the Kronecker delta. From symmetry, for all $1 \leq i < j \leq p$, $s_{ij} - s_{ji} = 0$, and due to redundancy of the linear equations of (2.3.2), we have

$$\begin{aligned} m_{(i)}s^{(j)} + m_{(j)}s^{(i)} = 0, \quad 1 \leq i < j \leq p \\ m_{(i)}s^{(i)} = 1, \quad 1 \leq i \leq p. \end{aligned} \tag{2.3.3}$$

We combine these p^2 equations into a sparse linear system $A\mathbf{s} = b$. Note that $\mathbf{s} = vec(S^T)$, b is of size $p^2 \times 1$ and contains 0's and 1's, and A is of size $p^2 \times p^2$ and contains elements of M, 0's, 1's, and -1's. We solve for $\mathbf{s}$ and reshape $\mathbf{s}$ to arrive at $\hat{S}$, of size $p \times p$.

We will use the pair of maps $R_{\mathcal{S},Y}(T) = UV^T$ (with $T + Y = U\Sigma V^T$) and $R^{-1}_{\mathcal{S},Y}(X) = X\hat{S} - Y$ (with $\hat{S}$ as described above) in an iterative algorithm to produce the *Stiefel manifold normal mean* of a set of points on $\mathcal{S}$.

2.4 Normal Mean Algorithms

Using each pair of inversely-related maps, we describe an iterative algorithm to compute the *normal mean* for a collection of points on either the Grassmann manifold or Stiefel manifold. Figure 2.2 depicts the underlying idea of a manifold mean. The center white point is the origin in the tangent space and identifies the point on the manifold corresponding to this space. Essentially, $[Y]$ (resp. Y) is the normal mean of a finite set of points $\mathcal{P} \subset Gr(n,p)$ (resp. $\mathcal{P} \subset St(n,p)$) if their images in $T_{[Y]}$ (resp. T_Y) sum to be the origin. To find such a point, we begin with a guess, $[\mu]$ (resp. μ), for the mean and then use $R^{-1}_{\mathcal{G},\mu}$ (resp. $R^{-1}_{\mathcal{S},\mu}$) to map the elements of $\mathcal{P}$ to the tangent space $T_{[\mu]}$ (resp. T_μ). We then average them within the tangent space to get a vector δ. Mapping δ to the manifold using $R_{\mathcal{G},\mu}$ (resp. $R_{\mathcal{S},\mu}$) yields an updated mean approximation $[\mu]$ (resp. μ). We repeat until $||\delta||_F < \epsilon$ for some prescribed ϵ, which implies that $[\mu]$ (resp. μ) is "centered".

2.4.1 Algorithm for Grassmann Manifold

The fundamental maps that are used in the algorithm are $R_{\mathcal{G},Y}(T) = [UV^T]$ (with $T + Y = U\Sigma V^T$) and $R^{-1}_{\mathcal{G},Y}([X]) = X(Y^TX)^{-1} - Y$. Recall that while $R^{-1}_{\mathcal{G},Y}([X])$ is defined in terms of the choice of representative X for $[X]$, its value does not depend on the choice of representative. However, our maps are also defined in terms of a representative Y for $[Y]$. In order to establish an algorithm on $Gr(n,p)$, we need properties that are independent of the choice of representative. The next Proposition provides one of the steps in this direction.

Proposition 2.4.1. *If A is a $p \times p$ orthogonal matrix and if X, Y are $n \times p$ orthonormal matrices then $R^{-1}_{\mathcal{G},YA}([X]) = (R^{-1}_{\mathcal{G},Y}([X]))A$.*

Proof. Since $R^{-1}_{\mathcal{G},Y}([X]) = X(Y^TX)^{-1} - Y$, we have

$$\begin{aligned}
R^{-1}_{\mathcal{G},YA}([X]) &= X((YA)^TX)^{-1} - YA \\
&= X(A^TY^TX)^{-1} - YA \\
&= X(Y^TX)^{-1}(A^T)^{-1} - YA \\
&= X(Y^TX)^{-1}A - YA \\
&= (X(Y^TX)^{-1} - Y)A. \\
&= (R^{-1}_{\mathcal{G},Y}([X]))A.
\end{aligned}$$

□

Corollary 2.4.2. *Let $\mathcal{P}$ be a finite set of C points in $Gr(n,p)$. If A is a $p \times p$ orthogonal matrix and if Y is an $n \times p$ orthonormal matrix then*

$$\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y}([X_i]) = 0 \iff \frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},YA}([X_i]) = 0.$$

The goal of the iterative algorithm is to find an element $[Y] \in Gr(n,p)$ such that $\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y}([X_i]) = 0$. While the previous proposition and corollary show that the final answer is independent of the choice of representative Y for $[Y]$, the next proposition shows how to construct a sequence of points on $Gr(n,p)$ such that each point in the sequence is also independent of the choice of representative.

Proposition 2.4.3. *Let $\mathcal{P}$ be a finite set of C points in $Gr(n,p)$. If A is a $p \times p$ orthogonal matrix and if Y is an $n \times p$ orthonormal matrix then*

$$R_{\mathcal{G},Y}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y}([X_i])) = R_{\mathcal{G},YA}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},YA}([X_i])).$$

Proof. We have

$$\begin{aligned} R_{\mathcal{G},Y}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y}([X_i])) &= R_{\mathcal{G},Y}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} (X_i(Y^TX_i)^{-1} - Y)) \\ &= R_{\mathcal{G},Y}((\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} X_i(Y^TX_i)^{-1}) - Y) \\ &= [UV^T] \end{aligned}$$

where $U\Sigma V^T$ is the thin SVD of $(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} X_i(Y^TX_i)^{-1}) - Y + Y$. In a similar manner

$$\begin{aligned} R_{\mathcal{G},YA}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},YA}([X_i])) &= R_{\mathcal{G},YA}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y}([X_i])A) \\ &= R_{\mathcal{G},YA}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} (X_i(Y^TX_i)^{-1} - Y)A) \\ &= R_{\mathcal{G},YA}((\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} X_i(Y^TX_i)^{-1})A - YA) \\ &= [UV^TA] \end{aligned}$$

since $U\Sigma V^TA$ is the thin SVD of $(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} X_i(Y^TX_i)^{-1})A - YA + YA$. We conclude by remarking that $[UV^T] = [UV^TA]$. □

Given a finite collection of points $\mathcal{P} \subset Gr(n,p)$ and an initial point $[Y_0] \in Gr(n,p)$, we can consider the sequence of points $[Y_0], [Y_1], [Y_2], \ldots$ on $Gr(n,p)$ where

$$[Y_{t+1}] = R_{\mathcal{G},Y_t}(\frac{1}{C}\sum_{[X_i]\in\mathcal{P}} R^{-1}_{\mathcal{G},Y_t}([X_i])).$$

By Proposition 2.4.3, this is a well defined sequence of points on $Gr(n,p)$. More precisely, the sequence is independent, at each step, of the choice of representative Y_t for $[Y_t]$.

Algorithm 2, found below, uses the properties of the maps $R_{\mathcal{G},Y}$ and $R^{-1}_{\mathcal{G},Y}$ to build an iterative algorithm that is very much in the spirit of the Karcher mean algorithm from [6].

In the Karcher mean algorithm, the corresponding maps are **Exp** and **Log** maps derived from the model of $Gr(n,p)$ and $St(n,p)$ as quotients of $SO(n)$. The geometric intuition that drives the algorithm is captured in Figure 2.2.

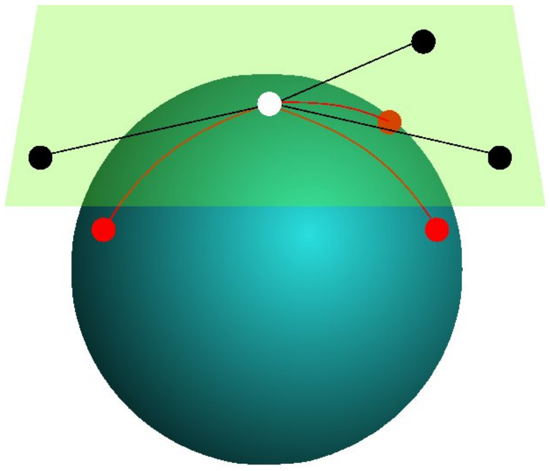

Fig. 2.2: Three points on the matrix manifold and their associated tangent vectors. Upon convergence of the algorithm the center white point is both the origin of the tangent space and the average value of the three tangent vectors.

Algorithm 2 Calculate Normal Mean, $[\mu_N]$, on $Gr(n,p)$

```
Require: $[X_i] \in Gr(n,p), 1 \le i \le C$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  $[\mu] \Leftarrow [X_1]$
  repeat
    $\delta \Leftarrow \frac{1}{C} \sum_{i=1}^{C} R_{\mathcal{G},\mu}^{-1}([X_i])$
    $[\mu] \Leftarrow R_{\mathcal{G},\mu}(\delta)$
  until $||\delta||_F < \epsilon$
  $[\mu_N] \Leftarrow [\mu]$
```

2.4.2 *Principal angles, orthogonally invariant functions, and F-Normal means*

Given two p-dimensional subspaces $[X]$ and $[Y]$ of $\mathbb{R}^n$, there are p principal angles $0 \le \theta_1 \le \theta_2 \le ... \le \theta_p \le \frac{\pi}{2}$ between $[X]$ and $[Y]$. The principal angles may be computed as the inverse cosine of the singular values of $X^T Y$ [8].

Definition 2.4.4. Consider the p-tuple of principal angles between $[X]$ and $[Y]$, $\Theta([X],[Y]) = (\theta_1, \ldots, \theta_p)$. We define $\Theta([X_1],[Y_1]) > \Theta([X_2],[Y_2])$ if the vector $\Theta([X_1],[Y_1]) - \Theta([X_2],[Y_2])$ is non-negative with at least one positive entry.

Let $F : Gr(n,p) \times Gr(n,p) \to \mathbb{R}$ be a bivariate function on the Grassmann manifold. Let $O(n)$ denote the orthogonal group. If $F([X],[Y]) = F([AX],[AY])$ for any $[X],[Y] \in Gr(n,p)$ and any $A \in O(n)$, then F is said to be *orthogonally invariant*. If F is orthogonally invariant then the value of $F([X],[Y])$ can be written in terms of the principal angles between $[X]$ and $[Y]$.

Definition 2.4.5. We say that a bivariate function F on $Gr(n,p)$ respects natural distance functions if

- F is orthogonally invariant
- $F([X],[Y]) \geq 0$ with equality if and only if $[X] = [Y]$.
- $\Theta([X_1],[Y_1]) > \Theta([X_2],[Y_2]) \implies F([X_1],[Y_1]) \geq F([X_2],[Y_2])$.

The orthogonal invariance of F allows us to think of F as a function on the set $S = \{(x_1, \ldots, x_k) | 0 \leq x_1 \leq \cdots \leq x_k \leq \pi/2\}$.

Given a continuous, bivariate function F on $Gr(n,p)$ that respects natural distance functions, there are two useful modifications that can be made to Algorithm 2 by weighting the averaging stage of the algorithm by the function F. These are found below.

Perhaps the main advantage of the F-Normal Mean is the ability to dampen (or exaggerate) the effect of outliers. For instance, by having the function F saturate quickly for relatively small values of $\Theta([X_1],[X_2])$, a single outlier will have little effect on the output of Algorithm 3. An advantage of the Directional F-Normal Mean is to fundamentally change

Algorithm 3 Calculate F-Normal Mean, $[\mu_N]$, on $Gr(n,p)$

```
**Require:** $[X_i] \in Gr(n,p), 1 \leq i \leq C$; $\epsilon > 0$
**Ensure:** $X_i^T X_i = I$
  $[\mu] \Leftarrow [X_1]$
  **repeat**
    $\delta \Leftarrow \frac{1}{C}\sum_{i=1}^{C} F([u],[X_i]) R_{\mathcal{G},\mu}^{-1}([X_i])$
    $[\mu] \Leftarrow R_{\mathcal{G},\mu}(\delta)$
  **until** $||\delta||_F < \epsilon$
  $[\mu_N] \Leftarrow [\mu]$
```

Algorithm 4 Calculate Directional F-Normal Mean, $[\mu_N]$, on $Gr(n,p)$

```
**Require:** $[X_i] \in Gr(n,p), 1 \leq i \leq C$; $\epsilon > 0$
**Ensure:** $X_i^T X_i = I$
  $[\mu] \Leftarrow [X_1]$
  **repeat**
    $\delta \Leftarrow \frac{1}{C}\sum_{i=1}^{C} F([u],[X_i]) \frac{R_{\mathcal{G},\mu}^{-1}([X_i])}{\|R_{\mathcal{G},\mu}^{-1}([X_i])\|}$
    $[\mu] \Leftarrow R_{\mathcal{G},\mu}(\delta)$
  **until** $||\delta||_F < \epsilon$
  $[\mu_N] \Leftarrow [\mu]$
```

the metric that is implicitly being used on $Gr(n,k)$. For instance, by choosing a function F that is more heavily dependent on the value of the first principal angle or by choosing F to be the Fubini-Study metric, etc.

2.4.3 Algorithm for Stiefel Manifold

The Stiefel Normal Mean is nearly identical to the Grassmann Normal Mean. The main difference is in the functions that are used in the iteration. In particular, the pair of maps we use are $R_{\mathcal{S},Y}(T) = UV^T$ (with $T + Y = U\Sigma V^T$) and $R^{-1}_{\mathcal{S},Y}(X) = X\hat{S} - Y$ (with $\hat{S}$ as described in Section 2.3.2). Contrary to the Grassmann case, we do not have to worry about whether the maps are well defined on an equivalence class. This simplifies the discussion to a presentation of the algorithm.

Algorithm 5 Calculate Normal Mean, μ_N, on $St(n,p)$

```
Require: $X_i \in St(n,p), 1 \le i \le C$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  $\mu \Leftarrow X_1$
  repeat
    $\delta \Leftarrow \frac{1}{C}\sum_{i=1}^{C} R^{-1}_{\mathcal{S},\mu}(X_i)$
    $\mu \Leftarrow R_{\mathcal{S},\mu}(\delta)$
  until $||\delta||_F < \epsilon$
  $\mu_N \Leftarrow \mu$
```

2.5 Conclusions

From a model of the Stiefel manifold as a submanifold of $\mathbb{R}^{n\times p}$ and a model of the Grassmannian as a quotient object of the Stiefel manifold, explicit matrix descriptions of the tangent and normal space to each of these manifolds can be formulated. These descriptions allow the building of pairs of maps, parametrized by the underlying manifold, whose composition is the identity on a suitably restricted neighborhood. The pair of maps are used to derive an iterative algorithm for determining a normal mean representative for a cluster of points on the Grassmann and Stiefel manifolds. This is reminiscent of the use of the **Exp** and **Log** maps on a Lie group to build the Karcher mean of a set of points. The similarities between the normal mean algorithm and the Karcher mean algorithm extend to the setting of computational complexity. However, by relying on maps different from **Exp** and **Log**, the normal mean algorithm provides alternate mean representations that are flexible and have the potential to capture different characteristics in a large data set.

3. TANGENT BUNDLE PROJECTION ALGORITHM FOR REPRESENTING POINT CLOUDS ON GRASSMANN AND STIEFEL MANIFOLDS

We derive an iterative algorithm for determining a representative for a cluster of points on the Stiefel manifold. The algorithm depends upon the tangent bundle projection operator and its inverse, which is developed according to a characterization of the tangent and normal space at each point of the manifold. From the cluster of points, the algorithm exploits these maps in a predictor/corrector loop until converging, with prescribed tolerance, to a fixed point. The fixed point acts as a *projection mean* representative of the cluster. This method shares structural characteristics with the *normal mean* and *Karcher mean*, but utilizes a distinct pair of inversely related maps. We furnish analysis towards a similar algorithm on the Grassmann manifold.

3.1 Introduction

The Grassmann manifold and Stiefel manifold are matrix manifolds which have become prominent settings in which to represent and make comparisons between large data sets collected under a variation of state. Given a cluster of points on either of these manifolds, we describe a procedure to find a single representative point. The (hypothetical) algorithm for the Grassmann manifold is based on an explicit description of the normal and tangent spaces to the manifold arising from a standard realization as a quotient space. The algorithm for the Stiefel manifold arises from its natural embedding in $\mathbb{R}^{n\times p}$ as the set of full rank orthonormal $n \times p$ matrices.

3.2 Mathematical Background

Section 2.2 provides basic definitions for Grassmann manifolds and Stiefel manifolds. In addition, the geometric setting for each of these objects, as differentiable manifolds, is described. Suggestions for further reading are provided. Therefore, we limit our discussion here.

We identify the geometry of the Grassmann manifold necessary for our current application, following the framework presented in [11]. Let $[Y]$ be a point on the Grassmann manifold $\mathcal{M} = Gr(n,p)$, where Y is a particular $n \times p$ orthonormal matrix representative for the column space of Y. Recall that the tangent space to $\mathcal{M}$ at $[Y]$ is identified with the set of all $n \times p$ matrices Δ such that $Y^T \Delta = 0$, and the normal space to $\mathcal{M}$ at $[Y]$ is identified with the set of matrices $\{YS\}$ where S is any $p \times p$ matrix. Given $[Y] \in \mathcal{M}$ and an $n \times p$ matrix X, the projection map to the tangent space at $[Y]$ is defined as $\Pi_{\mathcal{G},[Y]} : \mathbb{R}^{n \times p} \to T_{[Y]}\mathcal{M}$, $\Pi_{\mathcal{G},[Y]}(X) = (I - YY^T)X$. Note that this map is not well-defined on Grassmannian points, since, in general, every Grassmannian point is an equivalence class of matrices with different projections. We explain further in Section 3.3.1.

Similarly, we state the necessary geometric framework of the Stiefel manifold found in [11]. Let Y be a point on the Stiefel manifold $\mathcal{N} = St(n,p)$, where Y is a $n \times p$ orthonormal matrix. Recall that the tangent space to $\mathcal{N}$ at Y is identified with the set of all $n \times p$ matrices Δ such that $Y^T \Delta + \Delta^T Y = 0$, and the normal space to $\mathcal{N}$ at Y is identified with the set of matrices $\{YS\}$ where S is any $p \times p$ symmetric matrix. On the Stiefel manifold, given $Y \in \mathcal{N}$ and an $n \times p$ matrix X, the projection map to the tangent space at Y is defined as $\Pi_{\mathcal{S},Y} : \mathbb{R}^{n \times p} \to T_Y\mathcal{N}$, $\Pi_{\mathcal{S},Y}(X) = \frac{Y(Y^TX - X^TY)}{2} + (I - YY^T)X$. This is well-defined on the set of $n \times p$ orthonormal matrices, and therefore on the Stiefel manifold $St(n,p)$.

3.3 Computing the Projection Retraction

Our purpose is to build an iterative algorithm to determine a mean representative, in the vein of the normal mean. This requires both a map from the manifold to the tangent space, and the inverse map from the tangent space to the manifold.

3.3.1 Grassmann Manifold Projection Inverse

In order to secure a projection mean algorithm for the Grassmann manifold, we need to begin with a projection map that is well-defined on Grassmannian points. However, in general, given $[X] \in \mathcal{M} = Gr(n,p)$, $\Pi_{\mathcal{G},[Y]}(XA) \neq \Pi_{\mathcal{G},[Y]}(X)$ for A orthogonal. In other words, the known Grassmann manifold tangent space projection operator is not invariant to choice of orthogonal basis for $[X]$, and thus cannot be used for our Grassmann mean algorithm.

On the other hand, let us suppose, for the sake of the following analysis, that we have determined a well-defined Grassmann tangent space projection. Let $Nb_{[Y]}$ be a neighborhood of $[Y]$ on the Grassmannian $\mathcal{M}$. Then our hypothetical tangent space projection is $\tilde{\Pi}_{\mathcal{G},[Y]} : Nb_{[Y]} \to T_{[Y]}$, with

$$\tilde{\Pi}_{\mathcal{G},[Y]}([X]) = T$$

for $[X] \in Nb_{[Y]}$ and $T \in T_{[Y]}$.

We want to compute the inverse of $\tilde{\Pi}_{\mathcal{G},[Y]}$. We know that, as a projection, $\tilde{\Pi}_{\mathcal{G},[Y]}$ maps $[X] \in Nb_{[Y]}$ to a point in $T_{[Y]}\mathcal{M}$, with motion orthogonal to $T_{[Y]}\mathcal{M}$. We illustrate the projection map in Figure 3.1.

The inverse map must operate completely within the normal space to $\mathcal{M}$ at $[Y]$. Therefore, given $T \in T_{[Y]}\mathcal{M}$, we seek the point on $\mathcal{M}$ closest in Frobenius norm to $T+Y$ with motion restricted to the normal space to $\mathcal{M}$ at $[Y]$. The closest orthonormal matrix to $T+Y$ will be the representative for the closest Grassmannian point. We recall that normal space vectors at $[Y]$ have the form YS where S is a $p \times p$ matrix. This problem is illustrated in Figure 3.2.

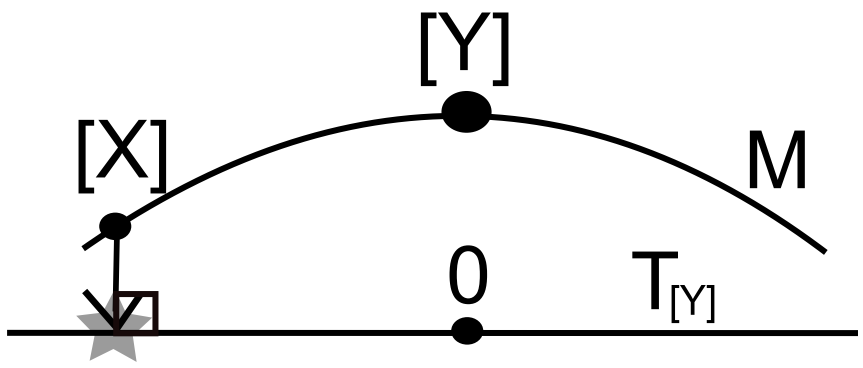

Fig. 3.1: The hypothetical tangent space projection map for the Grassmann manifold.

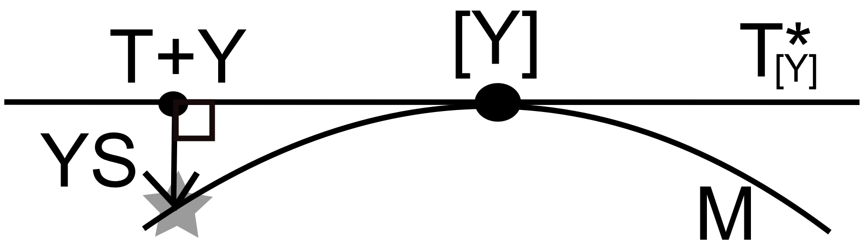

Fig. 3.2: Our derivation exploits the point of intersection of the normal space to the Grassmann manifold M and M itself.

The desired matrix S solves the following optimization problem:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|YS\|_F^2 \\ \text{subject to} \quad & (Y + T + YS)^T(Y + T + YS) = I. \end{aligned}$$

Under the observations that $Y^TT = 0$, $Y^TY = I$, and

$\|YS\|_F^2 = \text{trace}(YS)^T(YS)$, the optimization problem simplifies to

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \text{trace}(S^TS) \\ \text{subject to} \quad & S^TS + S + S^T + T^TT = 0. \end{aligned}$$

Based on the constraint, we make the substitution $S^TS = -S - S^T - T^TT$. Noting that

$\text{trace}(-S - S^T - T^TT) = -2\text{trace}(S) - \text{trace}(T^TT)$, we arrive at

$$\begin{aligned} \underset{S}{\text{maximize}} \quad & \text{trace}(S) \\ \text{subject to} \quad & S^TS + S + S^T + T^TT = 0. \end{aligned}$$

Then we define $R = S + I$ and since $R^TR = S^TS + S + S^T + I$, we write

$$\begin{aligned} \underset{R}{\text{maximize}} \quad & \text{trace}(R) \\ \text{subject to} \quad & R^TR = I - T^TT. \end{aligned}$$

We note that we must have the columns of T with 2-norm no greater than 1. Otherwise finding such an R will be infeasible over $\mathbb{R}$. Now let A be the symmetric matrix $I - T^TT$. Then $R^TR = A$ gives p^2 constraint equations, but with redundancy as the ij equation is a repeat of as the ji symmetric equation. We may write a Lagrangian with a full symmetric matrix Λ of Lagrange multipliers:

$$L(R, \Lambda) = tr(R) - tr(\Lambda(R^TR - A)).$$

We seek a maximum by determining a stationary point for $L(R, \Lambda)$. The following are the partial derivatives of L:

$$\frac{\partial L}{\partial R} = I - R(\Lambda + \Lambda^T) \tag{3.3.1}$$

$$\frac{\partial L}{\partial \Lambda} = R^TR - A.$$

Now we set both partials equal to 0. From (3.3.1) and recalling that $\Lambda = \Lambda^T$, we conclude that

$$R = \frac{1}{2}\Lambda^{-1}. \tag{3.3.2}$$

Substitution of (3.3.2) into $R^TR = A$ yields $(\Lambda^{-1})^2 = 4A$, so $\Lambda^{-1} = 2A^{\frac{1}{2}}$. Finally, from

(3.3.2),

$$R = A^{\frac{1}{2}}.$$

Translating, $S = R - I = A^{\frac{1}{2}} - I = (I - T^T T)^{\frac{1}{2}} - I$. In other words, the orthonormal matrix $Y + T + Y((I - T^T T)^{\frac{1}{2}} - I) = T + Y(I - T^T T)^{\frac{1}{2}}$ is a representative for the closest point on the Grassmann manifold to $T + Y$, with motion restricted to the normal space to the manifold at $[Y]$.

We have

$$\tilde{\Pi}_{\mathcal{G},[Y]}([X]) = T$$

as the projection map of X to the tangent space of the Grassmann manifold at $[Y]$. The inverse is

$$\Pi^{-1}_{\mathcal{G},[Y]}(T) = T + Y(I - T^T T)^{\frac{1}{2}}.$$

We verify in one direction. Let Y, $Y^T Y = I$, be a matrix representative for the point $[Y]$ on the Grassmann manifold and let $T \in T_{[Y]}$. Since $Y^T T = 0$,

$$\begin{aligned}\Pi_{\mathcal{G},[Y]}(\Pi^{-1}_{\mathcal{G},[Y]}(T)) &= (I - YY^T)(T + Y(I - T^T T)^{\frac{1}{2}}) \\ &= T + Y(I - T^T T)^{\frac{1}{2}} - Y(I - T^T T)^{\frac{1}{2}} \\ &= T.\end{aligned}$$

3.3.2 Stiefel Manifold Projection Inverse

On the Stiefel manifold, we can compute the inverse of the well-defined projection operator, $\Pi_{\mathcal{S},Y}(X) = \frac{Y(Y^T X - X^T Y)}{2} + (I - YY^T)X$. We know that $\Pi_{\mathcal{S},Y}$ maps $X \in \mathcal{N} = St(n,p)$ to a point in $T_Y\mathcal{N}$, with motion orthogonal to $T_Y\mathcal{N}$. Thus the inverse map must operate completely within the normal space to $\mathcal{N}$ at Y. Therefore, we seek the closest point on $\mathcal{N}$ to $T + Y$, $T \in T_Y\mathcal{N}$, with motion restricted to the normal space to $\mathcal{N}$ at Y. Since on the Stiefel manifold, normal space vectors at Y have the form YS where S is a $p \times p$ symmetric matrix, we seek the S that solves the following optimization problem:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \|YS\|_F^2 \\ \text{subject to} \quad & (Y+T+YS)^T(Y+T+YS) = I, S = S^T. \end{aligned}$$

Under the observations that $Y^TT + T^TY = 0$, $Y^TY = I$, and $\|YS\|_F^2 = \text{trace}((YS)^T(YS))$, the optimization problem simplifies to

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \text{trace}(S^TS) \\ \text{subject to} \quad & S^TS + S + S^T + S^TY^TT + T^TYS + T^TT = 0, S = S^T. \end{aligned}$$

We approach this problem by constructing the Lagrangian.

$$L(S, \Lambda_1, \Lambda_2) = tr(S^TS + \Lambda_1(S - S^T) + \Lambda_2(S^TS + S + S^T + S^TY^TT + T^TYS + T^TT))$$

Taking partial derivatives, we have

$$\frac{\partial tr(S^TS)}{\partial S} = 2S$$

and

$$\frac{\partial tr(\Lambda_1(S - S^T))}{\partial S} = 0,$$

where we have used the fact that Λ_1 is symmetric. Also

$$\frac{\partial tr(\Lambda_2 S^TS)}{\partial S} = S(\Lambda_2 + \Lambda_2^T),$$

$$\frac{\partial tr(\Lambda_2(S + S^T))}{\partial S} = \Lambda_2^T + \Lambda_2,$$

and

$$\frac{\partial tr(\Lambda_2(S^TY^TT + T^TYS))}{\partial S} = Y^TT\Lambda_2^T + Y^TT\Lambda_2.$$

In all,

$$\frac{\partial L(S,\Lambda_1,\Lambda_2)}{\partial S} = 2S + S(\Lambda_2+\Lambda_2^T) + \Lambda_2 + \Lambda_2^T + Y^TT\Lambda_2^T + Y^TT\Lambda_2.$$

The system of equations obtained by setting the partials to zero

$$\frac{\partial L}{\partial S} = \frac{\partial L}{\partial \Lambda_1} = \frac{\partial L}{\partial \Lambda_2} = 0$$

and noting that Λ_2 is symmetric, is

$$\begin{aligned} 0 &= S + S\Lambda_2 + \Lambda_2 + Y^TT\Lambda_2 \\ 0 &= S^T - S \\ 0 &= S^TS + S + S^T + S^TY^TT + T^TYS + T^TT. \end{aligned} \tag{3.3.3}$$

We proceed to solve for S in the first equation of (3.3.3):

$$S = -(I + Y^TT)\Lambda_2(I+\Lambda_2)^{-1}. \tag{3.3.4}$$

Let $M = Y^TT$ and $K = \Lambda_2(I+\Lambda_2)^{-1}$, and substitute (3.3.4) into the third equation of (3.3.3) to get

$$0 = K^T(I+T^TY)(I+T^TY)^TK - (I+M^TM)K - K^T(I+M^TM) + T^TT$$

where we have used the fact that $M + M^T = 0$. Since Λ_2 and $(I+\Lambda_2)^{-1}$ are symmetric, it can be shown that they are simultaneously diagonalizable, and therefore their product, K, is symmetric. Letting $A = I + M^TM$, $B = I + T^TY$, $Q = -T^TT$, and $R = I$, we have the continuous Algebraic Riccati Equation (CARE)

$$0 = -KBRB^TK + AK + KA + Q. \tag{3.3.5}$$

Note that Q is symmetric, so it has a Cholesky factorization $Q = C^TC$, which is required for a CARE. Much effort towards solving CARE's is evident in the literature [16,3]. We can solve (3.3.5) for a single K based on a Schur method in a Control Systems Toolbox from Matlab, or a new distribution of SciPy using the function call,

$$K = \textit{solve\_continuous\_are}(A, B, Q, R).$$

Given K we obtain $\Lambda_2 = (I - K)^{-1}K$, and determine S from (3.3.4). However, there is a family of solutions K to (3.3.5), not just one. We need to be able to generate the entire family, and then select the K that makes S symmetric. This is a cumbersome prospect. An alternate, simplified, problem formulation (see below) requires solving a different CARE.

3.3.3 Alternate Stiefel Projection Inverse

This is an alternate method to compute the inverse of the projection operator for the Stiefel manifold, $\Pi_{\mathcal{S},Y}(X) = \frac{Y(Y^TX - X^TY)}{2} + (I - YY^T)X$. We know that $\Pi_{\mathcal{S},Y}$ maps $X \in \mathcal{N} = St(n,p)$ to a point $T \in T_Y\mathcal{N}$, with motion orthogonal to $T_Y\mathcal{N}$. Thus the inverse map must operate completely within the normal space to $\mathcal{N}$ at Y. Therefore, we seek the closest point of $\mathcal{N}$ to $Y + T$ with motion restricted to $\mathcal{N}$ at Y. Since on the Stiefel manifold, normal space vectors at Y have the form YS where S is a $p \times p$ symmetric matrix, we seek the S that solves the following problem:

$$\begin{aligned} &\underset{S}{\text{minimize}} \quad \|YS\|_F^2 \\ &\text{subject to} \quad (Y + T + YS)^T(Y + T + YS) = I, S = S^T. \end{aligned}$$

Expanding the first constraint, $S^TS + (I + T^TY)S + S^T(I + Y^TT) + T^TT = 0$, where we have used the fact that $Y^TT + Y^TT = 0$ and $Y^TY = I$. Letting $A = I + Y^TT$, $B = I$, $Q = T^TT$, $R = -I$, and using the symmetry $S = S^T$, our constraints are expressed by the continuous algebraic Riccati equation (CARE)

$$-SBR^{-1}B^TS + A^TS + SA + Q = 0. \tag{3.3.6}$$

Note that $Q = T^TT$ satisfies the factorization requirement for a CARE. We can solve (3.3.6) for a single S in a Control Systems Toolbox from Matlab, or a new distribution of SciPy using the function call,

$$S = scipy.linalg.solve\_continuous\_are(A, B, Q, R).$$

Since this S may not be the solution we want, we need to exhaustively identify all solutions to the CARE, which can be executed in Bertini for p small [5]. We observe a finite set of 2^{p-1} real solutions S to (3.3.6). As long as X is selected in a neighborhood of Y, the matrix $Y + T + Y\hat{S}$ is seen to be equal to X when $\hat{S}$ is selected as the minimum norm S solving the CARE. We offer an example to illustrate in 3.3.4.

Summarizing,

$$\Pi_{\mathcal{S},Y}(X) = \frac{Y(Y^TX - X^TY)}{2} + (I - YY^T)X$$

is the projection map of X to the tangent space of the Stiefel manifold at Y. For $T \in T_Y\mathcal{N}$, the inverse is

$$\Pi^{-1}_{\mathcal{S},Y}(T) = Y + T + Y\hat{S}$$

where $\hat{S}$ is determined using (3.3.6) and the minimum norm criterion.

3.3.4 Stiefel Projection Inverse Example

We demonstrate by example that $\Pi_{\mathcal{S},Y}$ and $\Pi_{\mathcal{S},Y}^{-1}$ are related as inverses. Select $Y, X \in St(6,3)$ as

$$Y = \begin{bmatrix} -0.1258 & 0.3650 & 0.4525 \\ -0.6761 & 0.3656 & -0.4548 \\ -0.2394 & 0.1663 & -0.3722 \\ -0.4435 & 0.2045 & 0.6642 \\ -0.2039 & -0.2428 & 0.0007 \\ -0.4811 & -0.7776 & 0.0935 \end{bmatrix}$$

and

$$X = \begin{bmatrix} 0.0283 & 0.3836 & 0.4547 \\ -0.7164 & 0.4404 & -0.3008 \\ -0.2979 & 0.1915 & -0.3129 \\ -0.2487 & 0.2656 & 0.7396 \\ -0.2245 & -0.2145 & 0.0578 \\ -0.5338 & -0.7111 & 0.2335 \end{bmatrix}.$$

Here we have $d_{\mathcal{S}}(Y, X) = 0.3758$, which is relatively small. Then

$$T = \Pi_{\mathcal{S},Y}(X) = \begin{bmatrix} 0.1501 & 0.0275 & 0.0202 \\ -0.0621 & 0.0690 & 0.1476 \\ -0.0657 & 0.0205 & 0.0517 \\ 0.1788 & 0.0708 & 0.0975 \\ -0.0284 & 0.0258 & 0.0539 \\ -0.0721 & 0.0603 & 0.1323 \end{bmatrix}.$$

Now the solution set to the continuous algebraic Riccati equation (3.3.6) with this Y and T contains 4 unique real solutions. The members of this set took a total of 3.1 seconds to compute with Bertini [5]. The solution with smallest Frobenius norm is

$$\hat{S} = \begin{bmatrix} -0.0350 & -0.0031 & 0.0016 \\ -0.0031 & -0.0078 & -0.0141 \\ 0.0016 & -0.0141 & -0.0271 \end{bmatrix}.$$

Now we see that

$$\Pi^{-1}_{\mathcal{S},Y}(T) = Y + T + Y\hat{S} = \begin{bmatrix} 0.0283 & 0.3837 & 0.4551 \\ -0.7164 & 0.4403 & -0.3010 \\ -0.2979 & 0.1915 & -0.3132 \\ -0.2487 & 0.2657 & 0.7401 \\ -0.2245 & -0.2145 & 0.0577 \\ -0.5338 & -0.7110 & 0.2335 \end{bmatrix}$$

which is approximately X, and equal to X when we include more digits of precision in the entries of Y, X, T, and $\hat{S}$.

3.4 Projection Mean Algorithm

Using these pairs of inversely-related maps, we describe an iterative algorithm to compute the *projection mean* for a collection of points on either the Grassmann manifold or Stiefel manifold. Figure 3.3 depicts the concept of a manifold mean, which is repeated here to contribute to a more self-contained chapter. The center white point defines the particular shifted tangent space, and thus the set of shifted tangent vectors resulting from projecting the set of given manifold points to the shifted tangent space. Essentially, $[Y]$ (resp Y) is the projection mean of a set of points on $Gr(n,p)$ (resp $St(n,p)$) if their scaled images on $T_{[Y]}$ (resp T_Y) sum to be the origin. To find such a point, we begin with a guess, $[\mu]$ (resp μ), for the mean and then use $\tilde{\Pi}_{\mathcal{G},[\mu]}$ (resp $\Pi_{\mathcal{S},\mu}$) to project the manifold points to the tangent space $T_{[\mu]}$ (resp T_μ). We scale the tangent vectors by the manifold distances, and then average them within the tangent space to get a vector δ. Mapping δ to the manifold using $\tilde{\Pi}^{-1}_{\mathcal{G},[\mu]}$ (resp $\Pi^{-1}_{\mathcal{S},\mu}$) yields an updated mean iterate $[\mu]$ (resp μ). We repeat until $\|\delta\|_F < \epsilon$ for some ϵ, which implies that $[\mu]$ (resp μ) is "centered". The algorithms we develop are Algorithm 6 and Algorithm 7.

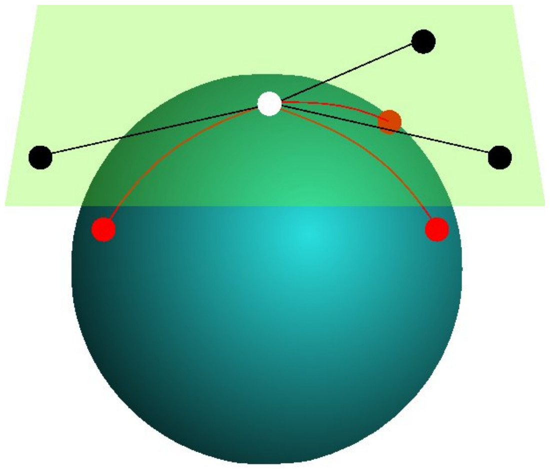

Fig. 3.3: Three points on the matrix manifold and their associated tangent vectors. Upon convergence of the algorithm the center white point represents the mean subspace. (The points are represented in 3D for purposes of illustration. They actually correspond to points on a matrix manifold.)

Algorithm 6 Calculate $[\mu_{proj}]$ on $Gr(n,p)$

```
Require: $[X_i] \in Gr(n,p), 1 \leq i \leq C$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  $\mu \Leftarrow X_1$
  repeat
    $\delta \Leftarrow \frac{1}{C}\sum_{i=1}^{C} \frac{\tilde{\Pi}_{\mathcal{G},[\mu]}([X_i])}{\|\tilde{\Pi}_{\mathcal{G},[\mu]}([X_i])\|_F} d_{\mathcal{G}}([\mu],[X_i])$
    $\mu \Leftarrow \tilde{\Pi}^{-1}_{\mathcal{G},[\mu]}(\delta)$
  until $\|\delta\|_F < \epsilon$
  $\mu_{proj} \Leftarrow \mu$
```

Algorithm 7 Calculate μ_{proj} on $St(n,p)$

```
Require: $X_i \in St(n,p), 1 \leq i \leq C$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  $\mu \Leftarrow X_1$
  repeat
    $\delta \Leftarrow \frac{1}{C} \sum_{i=1}^{C} \frac{\Pi_{\mathcal{S},\mu}(X_i)}{\|\Pi_{\mathcal{S},\mu}(X_i)\|_F} d_{\mathcal{S}}(\mu, X_i)$
    $\mu \Leftarrow \Pi_{\mathcal{S},\mu}^{-1}(\delta)$
  until $\|\delta\|_F < \epsilon$
  $\mu_{proj} \Leftarrow \mu$
```

4. FLAG MEAN: A GEOMETRIC VARIABLE-DIMENSION SUBSPACE AVERAGE

Given a set of subspaces, perhaps of varying dimension, of an ambient space, we develop an pair of algorithms to compute a mean subspace of flexible dimension. The distinction between the pair of formulations is the choice of metric in the optimization problem. We call these means *Flag Means*, due to the nested relationship demonstrated by flag means of increasing dimension. The connection between the flag mean and the Karcher mean allows valuable comparison. In particular, the Karcher mean is guaranteed to exist for a local neighborhood of points on the Grassmann manifold, which are fixed dimension subspaces. For added theoretical insight, we draw upon the connection to Multiset Canonical Correlation Analysis.

4.1 Motivation

As previously outlined, in computational data analysis it is common to appeal to established geometric theory to develop algorithms and process data. The Grassmann manifold and Stiefel manifold are matrix manifolds which have found use as a setting in which to classify and make comparisons between large data sets. Given a cluster of points on these manifolds, we have described procedures to find a single point, on the manifold, which represents the cluster. These means can be used to reduce the cost of classification algorithms or to aid in clustering tasks. More generally, given a collection of subspaces, perhaps of of different dimensions, as opposed to points on a particular manifold, we investigate how to compute a mean representative.

4.2 Mathematical Background

A favorable setting for data organization and classification is the Grassmann manifold. The Grassmann manifold $Gr(n,p)$ is the set of all p-dimensional subspaces of $\mathbb{R}^n$. Orthonormal matrix representatives for points on the Grassmannian are identified to perform computations. More precisely, a $n \times p$ matrix Y, $Y^TY = I$, is a representative for the p-dimensional subspace spanned by the columns of Y, denoted $[Y]$. Given a cluster of N points on $Gr(n,p)$, $\{[X_i]\}_{i=1}^N$, with $X_i^TX_i = I$, we may compute a mean, $[\mu] \in Gr(n,p)$. One approach is to compute the Karcher mean, $[\mu_{KM}]$, using the iterative procedure found in [6]. The Karcher mean depends upon principal angles. Given two p-dimensional subspaces $[X]$ and $[Y]$ of $\mathbb{R}^n$, there are p principal angles $0 \le \theta_1 \le \theta_2 \le ... \le \theta_p \le \frac{\pi}{2}$ between $[X]$ and $[Y]$. The principal angles may be computed as the inverse cosine of the singular values of X^TY [8]. Intuitively, $[\mu_{KM}]$ is identified as the p-dimensional subspace of $\mathbb{R}^n$, given by $[\mu_{KM}] = \underset{[\mu]}{\operatorname{argmin}} \sum_{i=1}^{N} d([\mu],[X_i])^2$, where we measure the distance $d([X],[Y])$ as the square root of the sum of the squares of the principal angles between the subspaces $[X]$ and $[Y]$. The Karcher mean iterative algorithm is guaranteed to converge for a cluster of points located within a certain convergence radius.

On the other hand, we consider forming a new mean, called the flag mean. Suppose that we are given N subspaces of $\mathbb{R}^n$, of any, perhaps variable, dimension. Let $\{X_i\}_{i=1}^N$, $X_i^TX_i = I$, be orthonormal matrices whose column spaces are the N given subspaces, denoted $\{[X_i]\}_{i=1}^N$. We consider two formulations of the flag mean, based on two distinct optimization problems. In the first, we work with the cosines of the principal angles. In the second, we utilize the squares of the cosines of the principal angles. In each case, we generate a q-dimensional flag mean, $1 \le q \le r$, where r is the dimension of the union of the $\{[X_i]\}$. We compute the flag mean by forming an orthonormal basis of q vectors. The name “flag mean” arises from the fact that the q-dimensional flag mean is a proper subspace of the $(q+1)$-dimensional flag mean, resulting in a nested sequence of flag means. This sequence is a flag, by definition.

4.3 Flag Mean Formulation 1

4.3.1 Finding the First Flag Mean Vector

Appealing to the concept of center of mass, we seek a unit vector $\hat{u}^{(1)}$ that minimizes the sum of the distances between $\hat{u}^{(1)}$ and each of the subspaces $[X_i]$. Note that the 'hat' notation denotes a unit vector. Hence we write

$$\begin{aligned} &\underset{u}{\text{minimize}} \quad \sum_{i=1}^{N} d(u, [X_i]) \\ &\text{subject to} \quad u^T u = 1. \end{aligned} \tag{4.3.1}$$

As it is common to use principal angles to measure distance between subspaces, we let $d(u, [X_i]) = \theta_i$ where θ_i is the principal angle between the line containing u and $[X_i]$. In other words, θ_i is the solution of the problem

$$\begin{aligned} &\underset{k_i}{\text{minimize}} \quad \arccos u^T k_i \\ &\text{subject to} \quad k_i^T k_i = 1, k_i \in [X_i]. \end{aligned}$$

The updated optimization problem from (4.3.1) would be

$$\begin{aligned} &\underset{u}{\text{minimize}} \quad \sum_{i=1}^{N} \theta_i \\ &\text{subject to} \quad u^T u = 1. \end{aligned} \tag{4.3.2}$$

However, (4.3.2) does not have a unique solution $\hat{u}^{(1)}$ in many cases. In place of (4.3.2), we consider maximizing the sum of the cosines of the principal angles. Thus, we want to find the $\hat{u}^{(1)}$ that achieves

$$\begin{aligned} &\underset{u,k_i}{\text{maximize}} \quad \sum_{i=1}^{N} u^T k_i \\ &\text{subject to} \quad u^T u = 1, k_i^T k_i = 1, k_i \in [X_i]. \end{aligned} \tag{4.3.3}$$

This updated problem eliminates many of the ambiguities in solution vector. However, given two subspaces $[X_1]$ and $[X_2]$ equal to the x and y axis, respectively, in $\mathbb{R}^2$, the selection of either diagonal would solve (4.3.3). We observe that orthogonality and sign issues remain obstacles to solvability.

We note that $k_i \in [X_i]$ implies that $k_i = X_i\alpha_i$ for some p-dimensional vector α_i. Also, $1 = k_i^T k_i = \alpha_i^T X_i^T X_i \alpha_i = \alpha_i^T \alpha_i$, so we have

$$\begin{aligned} \underset{u,\alpha_i}{\text{maximize}} \quad & \sum_{i=1}^{N} u^T X_i \alpha_i \\ \text{subject to} \quad & u^T u = 1, \alpha_i^T \alpha_i = 1. \end{aligned}$$

Writing a Lagrange function for this optimization problem, with Lagrange multipliers λ and $\{\mu_i\}_{i=1}^N$, we have

$$\Lambda(u, \alpha_1, ..., \alpha_N, \lambda, \mu_1, ..., \mu_N) = \sum_{i=1}^{N} u^T X_i \alpha_i - \lambda(u^T u - 1) - \sum_{i=1}^{N} \mu_i(\alpha_i^T \alpha_i - 1). \qquad (4.3.4)$$

We compute the following vector partial derivatives:

$$\frac{\partial \Lambda}{\partial u} = \sum_{i=1}^{N} X_i \alpha_i - 2\lambda u \qquad (4.3.5)$$

$$\frac{\partial \Lambda}{\partial \alpha_i} = X_i^T u - 2\mu_i \alpha_i, 1 \le i \le N \qquad (4.3.6)$$

$$\frac{\partial \Lambda}{\partial \lambda} = u^T u - 1$$

$$\frac{\partial \Lambda}{\partial \mu_i} = \alpha_i^T \alpha_i - 1, 1 \le i \le N.$$

We seek a stationary point for Λ, so we set the partials equal to 0. From the relations given by (4.3.5) and (4.3.6) we can develop a two-phase iterative procedure to first, find the $\{\alpha_i\}$ given iterate u_k, and second, to find updated u_{k+1} from these $\{\alpha_i\}$. We solve for α_i in the equation resulting from (4.3.6) to get

$$\alpha_i = \frac{X_i^T u}{2\mu_i}, 1 \leq i \leq N.$$

Although μ_i is unknown, we can circumvent its value by appealing to the fact that α_i is a unit vector. First notice that $\mu_i = \frac{1}{2}\cos\theta_i > 0$. Therefore, regardless of scalar μ_i,

$$\alpha_i = \frac{X_i^T u}{\|X_i^T u\|_2}, 1 \leq i \leq N.$$

Next we solve for u in the resulting equation from (4.3.5) to get

$$u = \frac{\sum_{i=1}^{N} X_i \alpha_i}{2\lambda}.$$

Again, we can bypass λ by realizing that u is unit length and $\lambda = \sum_{i=1}^{N} \frac{1}{2} u^T X_i \alpha_i = \sum_{i=1}^{N} \mu_i > 0$, so

$$u = \frac{\sum_{i=1}^{N} X_i \alpha_i}{\|\sum_{i=1}^{N} X_i \alpha_i\|_2}.$$

It follows that, given an initial guess for $\hat{u}^{(1)}$, $\hat{u}_0$, we can iteratively converge to $\hat{u}^{(1)}$ using Algorithm 8. Note that the 'hat' notation is used to imply normalization. Convergence to the globally optimal $\hat{u}^{(1)}$ is assured when the given subspaces occupy a single 2^n-tant (quadrant, octant, etc) in $\mathbb{R}^n$, thus avoiding orthogonality and local maxima. This is reasonable to assume for one dimensional subspaces generated via digital photographs, for example.

4.3.2 Finding the Remaining $q - 1$ Flag Mean Vectors

In order to build dimension for the q-dimensional flag mean, we determine q unit vectors that are mutually orthogonal. Suppose we have already collected $m - 1$ mutually orthogonal unit

Algorithm 8 Calculate $\hat{u}^{(1)}$ for Flag Mean 1

```
Require: $[X_i], 1 \leq i \leq N$ subspaces of $\mathbb{R}^n$; guess $\hat{u}_0 \in \mathbb{R}^n$; $\epsilon > 0$
Ensure: $X_i^T X_i = I$
  repeat
    $\alpha_i \Leftarrow X_i^T \hat{u}_k,\ 1 \leq i \leq N$
    $u_{k+1} \Leftarrow \sum_{i=1}^{N} X_i \hat{\alpha}_i$
  until $\|\hat{u}_{k+1} - \hat{u}_k\|_2 < \epsilon$
  $\hat{u}^{(1)} \Leftarrow \hat{u}_{k+1}$
```

vectors $\{\hat{u}^{(1)}, \hat{u}^{(2)}, ..., \hat{u}^{(m-1)}\}$, $\hat{u}^{(i)T}\hat{u}^{(j)} = \delta_{ij}$. The computation of $\hat{u}^{(m)}$ is a slight variation of Algorithm 8. The only modification necessary to the Lagrangian in (4.3.4) is to add the constraint that u is orthogonal to each $\hat{u}^{(j)}$, $1 \leq j < m$. The resulting Lagrangian, including Lagrange multipliers λ, $\{\mu_i\}_{i=1}^{N}$, and $\{\delta_j\}_{j=1}^{m-1}$, is

$$\begin{aligned}\Lambda(u, \alpha_1,..., \alpha_N, \lambda, \mu_1, ..., \mu_N, \delta_1, ...\delta_{m-1}) = \\ \sum_{i=1}^{N} u^T X_i \alpha_i - \lambda(u^T u - 1) - \sum_{i=1}^{N} \mu_i(\alpha_i^T \alpha_i - 1) - \sum_{j=1}^{m} \delta_j u^T \hat{u}^{(j)}.\end{aligned} \tag{4.3.7}$$

The partial derivatives of Λ remain the same as before, except

$$\frac{\partial \Lambda}{\partial u} = \sum_{i=1}^{N} X_i \alpha_i - 2\lambda u - \sum_{j=1}^{m-1} \delta_j \hat{u}^{(j)}. \tag{4.3.8}$$

Setting (4.3.8) equal to 0, solving for u, and letting $s = \sum_{i=1}^{N} X_i \alpha_i$ yields

$$u = \frac{s - \sum_{j=1}^{m-1} \delta_j \hat{u}^{(j)}}{2\lambda}. \tag{4.3.9}$$

To solve for δ_l, $1 \leq l \leq m-1$, we multiply on the left by $\hat{u}^{(l)T}$. Due to the requirements of orthogonality and unit length, we obtain

$$\delta_l = \hat{u}^{(l)T} s.$$

Substituting in (4.3.9), we find that

$$u = \frac{s - \sum_{j=1}^{m-1} \hat{u}^{(j)T} s \hat{u}^{(j)}}{2\lambda}. \tag{4.3.10}$$

Pairing (4.3.10) with (4.3.6) and recognizing that $\lambda > 0$ leads to the iterative procedure, Algorithm 9.

Algorithm 9 Calculate $\hat{u}^{(m)}$ for Flag Mean 1

```
Require: $[X_i], 1 \leq i \leq N$ subspaces of $\mathbb{R}^n$; guess $\hat{u}_0 \in \mathbb{R}^n$; $\epsilon > 0$; $1 < m \leq q$
Ensure: $X_i^T X_i = I$
  repeat
    $\alpha_i \Leftarrow X_i^T \hat{u}_k,\ 1 \leq i \leq N$
    $s \Leftarrow \sum_{i=1}^{N} X_i \hat{\alpha}_i$
    $u_{k+1} \Leftarrow (s - \sum_{j=1}^{m-1} \hat{u}^{(j)T} s \hat{u}^{(j)})$
  until $\|\hat{u}_{k+1} - \hat{u}_k\|_2 < \epsilon$
  $\hat{u}^{(m)} \Leftarrow \hat{u}_{k+1}$
```

4.3.3 Newton's Method Solution

There are numerous other iterative algorithms that can be used to determine a critical point for (4.3.4). One of these algorithms is Newton's method, which is implemented here to find a zero for the partial derivatives of (4.3.4). Here we only consider seeking $\hat{u}^{(1)}$, although the analysis could be extended to find $\hat{u}^{(m)}$, $m > 1$, by substituting the partials of (4.3.7). We let

$$\Theta = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \\ \lambda \\ u \\ \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}, \Delta\Theta = \begin{bmatrix} \Delta\mu_1 \\ \vdots \\ \Delta\mu_N \\ \Delta\lambda \\ \Delta u \\ \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_N \end{bmatrix}, F(\Theta) = \begin{bmatrix} X_1\alpha_1 + \cdots + X_N\alpha_N - 2\lambda u \\ X_1^T u - 2\mu_1\alpha_1 \\ \vdots \\ X_N^T u - 2\mu_N\alpha_N \\ u^T u - 1 \\ \alpha_1^T\alpha_1 - 1 \\ \vdots \\ \alpha_N^T\alpha_N - 1 \end{bmatrix}.$$

We want to use the approximation $F(\Theta + \Delta\Theta) \approx F(\Theta) + F'(\Theta)\Delta\Theta$ to iteratively find a zero for F. We have

$$F'(\Theta) = \begin{bmatrix} 0 & \dots & 0 & -2u & -2\lambda I_n & X_1 & \dots & X_N \\ -2\alpha_1 & \dots & 0 & 0 & X_1^T & -2\mu_1 I_p & \dots & 0 \\ & \ddots & & & & & \ddots & \\ 0 & \dots & -2\alpha_N & 0 & X_2^T & 0 & \dots & -2\mu_N I_p \\ 0 & \dots & 0 & 0 & 2u^T & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 2\alpha_1^T & \dots & 0 \\ \vdots & & & & & & \ddots & \\ 0 & \dots & 0 & 0 & 0 & 0 & \dots & 2\alpha_N^T \end{bmatrix}.$$

Given a guess for Θ, Θ_0, we can solve the linear system $F'(\Theta_0)\Delta\Theta_0 = -F(\Theta_0)$ for $\Delta\Theta_0$. Then we update $\Theta_1 = \Theta_0 + \Delta\Theta_0$ and iterate until $F(\Theta_k)$ is less than some tolerance. The u from within the final Θ_k is the first flag mean vector, $\hat{u}^{(1)}$. This procedure, like any nonconvex Newton's method, is sensitive to initial value Θ_0.

4.3.4 Iterative Eigenproblem Solution

Another iterative algorithm to determine a critical point for (4.3.4) involves an iterative eigenvector problem. Recall that solving for α_i in the equation resulting from (4.3.6) yields

$$\alpha_i = \frac{X_i^T u}{2\mu_i}, 1 \leq i \leq N. \tag{4.3.11}$$

Also recall that from (4.3.5),

$$\sum_{i=1}^{N} X_i \alpha_i = 2\lambda u. \tag{4.3.12}$$

Substituting (4.3.11) into (4.3.12) gives

$$\sum_{i=1}^{N} \frac{X_i X_i^T}{4\mu_i} u = \lambda u.$$

This motivates an iterative eigenvector problem. We may begin with a set $\{\mu_i\}$, perhaps determined based on computation of the principal angles between the given subspaces and a sensibly chosen central vector, for which the first flag mean 2 vector is a good candidate (see 4.4). Then we compute the largest eigenvector of $\sum_{i=1}^{N} \frac{X_i X_i^T}{4\mu_i}$, which becomes $\hat{u}_1$ when scaled to unit length. We may repeat computation of the $\{\mu_i\}$ based on principal angles, and proceed to find largest eigenvector $\hat{u}_2$ of the updated matrix. We observe convergence to a vector, $\hat{u}^{(1)}$.

Making a reverse substitution to remove u yields a second eigenvector problem. By stacking the equations given by the substitution, and letting $\alpha = [\alpha_1^T \dots \alpha_N^T]^T$, we have

$$[\frac{X_1}{4\mu_1} \cdots \frac{X_N}{4\mu_N}]^T [X_1 \dots X_N]\alpha = \lambda\alpha.$$

It is interesting that eigenvectors α are not, in general, such that each of the α_i is unit length. However, upon convergence to a flag vector, we find that indeed, $\|\alpha_i\| = 1$ for all i.

4.3.5 Discriminative Canonical Correlations Solution

We consider a final iterative algorithm that is motivated by the work of Kim et al [15]. This method requires the SVD. Discriminative Canonical Correlations (DCC) produces a transformation T that separates between class instances while simultaneously condensing within class instances. The DCC iterative algorithm for producing T is maleable to fit our task. We know that the cosine of the principal angle between the line containing $\hat{u}$ and $[X_i]$ can be computed via the SVD. Consider the thin SVD

$$X_i^T \hat{u} = U_i \Sigma_i V_i$$

where Σ_i and V_i are 1×1. Then by [8],

$$\cos\theta_i = \Sigma_i = V_i U_i^T X_i^T \hat{u}.$$

As a result, (4.3.3) can be written as

$$\begin{aligned} \underset{u}{\text{maximize}} \quad & \sum_{i=1}^{N} (V_i X_i U_i)^T u \\ \text{subject to} \quad & u^T u = 1. \end{aligned} \tag{4.3.13}$$

Notice that the maximum of (4.3.13) will be achieved when u points in the same direction as $\sum_{i=1}^{N} V_i X_i U_i$. In other words, the maximizer is

$$\hat{u} = \sum_{i=1}^{N} V_i X_i U_i. \tag{4.3.14}$$

The corresponding iterative algorithm requires a guess for $\hat{u}_0$, from which the SVD of each $X_i^T \hat{u}_0$ may be computed. Then we compute $\hat{u}_1$ from (4.3.14), and repeat. We find that this algorithm converges to $\hat{u}^{(1)}$.

4.4 Flag Mean Formulation 2

4.4.1 Standard Flag Mean 2

Now we consider a variation of the optimization problem in (4.3.3), in which we substitute the squares of the cosines of the principal angles. Again, we write θ_i for the principal angle between the line containing u and $[X_i]$. Then we aim to solve

$$\begin{aligned} &\underset{u}{\text{maximize}} \quad \sum_{i=1}^{N} \cos^2 \theta_i \\ &\text{subject to} \quad u^T u = 1. \end{aligned} \tag{4.4.1}$$

In the following Proposition, we verify that $\cos^2 \theta_i = u^T X_i X_i^T u$.

Proposition 4.4.1. $\cos^2 \theta_i = u^T X_i X_i^T u$

Proof. Let X_i be an $n \times p$ orthonormal basis for $[X_i]$, and let $u \in \mathbb{R}^n$ be unit length. Consider the SVD decomposition $u^T X_i = U \Sigma V^T$. We know that $U = \pm 1$ and Σ is $1 \times n$ with $\Sigma_{11} = \cos \theta_i$ according to [8]. Then

$$\begin{aligned} u^T X_i X_i^T u &= U \Sigma V^T V \Sigma^T U \\ &= U^2 \cos^2 \theta_i \\ &= \cos^2 \theta_i. \end{aligned}$$

□

Because of Proposition 4.4.1, equivalent to (4.4.1) we have the following optimization problem

$$\begin{aligned} &\underset{u}{\text{maximize}} \quad u^T (\sum_{i=1}^{N} X_i X_i^T) u \\ &\text{subject to} \quad u^T u = 1. \end{aligned} \tag{4.4.2}$$

Letting $A = \sum_{i=1}^{N} X_i X_i^T$, this translates to the Lagrangian

$$L(u, \lambda) = u^T A u - \lambda(u^T u - 1).$$

Taking partial derivatives we have:

$$\frac{\partial L}{\partial u} = 2Au - 2\lambda u$$

$$\frac{\partial L}{\partial \lambda} = u^T u - 1.$$

Thus we solve $Au = \lambda u$ for eigenvectors $\hat{u}^{(1)}, ..., \hat{u}^{(q)}$, which are the eigenvectors corresponding to the q largest eigenvalues in descending order. The flag mean of dimension q is the subspace spanned by these q eigenvectors. See Algorithm 10. We note that since A is symmetric, these q eigenvectors of A form an orthonormal set. This second formulation of the flag mean, as with the first formulation, fails when there is certain orthogonality between the given subspaces. On the other hand, the cosine squared cost function leads to a robust solution that does not depend upon a seed vector for an iterative method.

Algorithm 10 Calculate $\{u^{(m)}\}_{m=1}^{q}$ for Flag Mean 2

```
Require: $[X_i], 1 \leq i \leq N$ subspaces of $\mathbb{R}^n$; $1 \leq q \leq r$
Ensure: $X_i^T X_i = I$
  $A \Leftarrow \sum_{i=1}^{N} X_i X_i^T$
  $u^{(m)} \Leftarrow m$-th largest eigenvector of $A$, $1 \leq m \leq q$
```

We remark that the computation of the eigenvectors of A becomes intractable for large n. By appealing to the SVD, we realize that we can solve an equivalent reduced size problem. Let $\mathcal{X} = [X_1, X_2, \ldots, X_N]$, and note that $A = \mathcal{X}\mathcal{X}^T$. Then by the thin SVD, $\mathcal{X} = U\Sigma V^T$, which implies $A = U\Sigma^2 U^T$. But then the Np left singular vectors of $\mathcal{X}$ are equal to the Np largest eigenvectors of A. When $Np < n$, computing the thin SVD of $\mathcal{X}$ is faster than computing the eigenvectors of A.

4.4.2 *Weighted Flag Mean 2*

The ability to compute a weighted flag mean offers advantages under certain circumstances. For instance, if we have a variable level of confidence in the given subspaces, we may opt to scale the contribution of each subspace according to reliability. Alternatively, we could weight the contribution of each subspace according to the dimension of the subspace, perhaps to increase the impact of larger subspaces.

The second formulation of the flag mean allows for straightforward weighted mean calculations. Recall that this flag mean is based upon the eigenvectors of $A = \sum_{i=1}^{N} X_i X_i^T$. Let $\{c_i\}_{i=1}^{N}$ be weights that are selected to correspond to the N given subspaces. Then we let

$$B = \sum_{i=1}^{N} c_i X_i X_i^T$$

and compute the eigenvectors of B, the sum of weighted summands. The weighted flag mean of dimension q is the subspace spanned by the q eigenvectors corresponding to the q largest eigenvalues, in descending order. See Algorithm 11. Our remark about computational tractability for large n using the thin SVD of $\mathcal{X}$ applies here for $\mathcal{X} = [\frac{X_1}{\sqrt{c_1}}, \frac{X_2}{\sqrt{c_2}}, \ldots, \frac{X_N}{\sqrt{c_N}}]$.

Algorithm 11 Calculate $\{u^{(m)}\}_{m=1}^{q}$ for Weighted Flag Mean 2

```
**Require:** $[X_i], 1 \leq i \leq N$ subspaces of $\mathbb{R}^n$; $1 \leq q \leq r$; weights $c_i, 1 \leq i \leq N$
**Ensure:** $X_i^T X_i = I$
  $B \Leftarrow \sum_{i=1}^{N} c_i X_i X_i^T$
  $u^{(m)} \Leftarrow m$-th largest eigenvector of $B$, $1 \leq m \leq q$
```

4.4.3 *Connection to MCCA*

The flag mean 2 formulation is intimately connected to Multiset Canonical Correlation Analysis (MCCA).

Multiset CCA

Via et al [27] and Kettenring [13], among others, consider multiset extensions to Canonical Correlation Analysis (CCA). They work with data matrices, which we relabel $D_1, \ldots, D_N$, each matrix representing a different data set. In what follows we assume the matrices have been replaced with orthonormal bases $X_1, \ldots, X_N$.

Recall that for two data sets we may seek canonical vectors $k_1 \in R(X_1)$ and $k_2 \in R(X_2)$ in order to solve

$$\begin{aligned} \underset{k_1,k_2}{\text{maximize}} \quad & k_1^T k_2 \\ \text{subject to} \quad & k_1^T k_1 + k_2^T k_2 = 1. \end{aligned} \tag{4.4.3}$$

Using the multiset formulation, this becomes

$$\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} k_i^T k_j$$

subject to the constraints

$$k_i \in R(X_i)$$

and

$$\sum_{i=1}^{N} k_i^T k_i = 1.$$

Expressing k_i as a linear combination, we write

$$k_i = X_i \alpha_i$$

and form the Lagrangian

$$L(\alpha) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_i^T X_i^T X_j \alpha_j - \lambda\Big(\sum_{i=1}^{N} \alpha_i^T \alpha_i - 1\Big)$$

If we define the block column vector α as

$$\alpha = [\alpha_1^T, \ldots, \alpha_N^T]^T$$

and

$$\mathcal{X} = [X_1, \ldots, X_N]$$

then the Lagrangian takes on the form

$$L(\alpha) = \alpha^T(\mathcal{X}^T\mathcal{X} - I)\alpha - \lambda(\alpha^T\alpha - 1).$$

Solving the Lagrangian gives

$$\mathcal{X}^T\mathcal{X}\alpha = (\lambda + 1)\alpha.$$

Thus the set of vectors

$$k_i = X_i\alpha_i$$

obtained via the eigenvector problem are the canonical vectors of the multiset CCA.

For the subsequent sets of canonical vectors it can be shown that the above argument extends to

$$k_i^{(m)} = X_i\alpha_i^{(m)}$$

where $\alpha^{(m)}$ is the eigenvector of $\mathcal{X}^T\mathcal{X}$ associated with the m^{th} largest eigenvalue.

Singular Value Decomposition

Consider the thin SVD of the bases matrix $\mathcal{X}$, i.e.,

$$\mathcal{X}\alpha^{(m)} = \sigma^{(m)}\hat{u}^{(m)}$$

and

$$\mathcal{X}^T \hat{u}^{(m)} = \sigma^{(m)} \alpha^{(m)}$$

for each $1 \leq m \leq (Np)$. This is also written

$$\mathcal{X} = \mathcal{U} \Sigma \mathcal{A}^T$$

where $\mathcal{U} = [\hat{u}^1 | \ldots | \hat{u}^{(Np)}]$ and $\mathcal{A} = [\hat{\alpha}^{(1)} | \ldots | \hat{\alpha}^{(Np)}]$. We observe that the canonical vectors $k_i^{(m)} = X_i \alpha_i^{(m)}$ in multiset CCA are found by solving for $\mathcal{A}$, the right singular vectors of $\mathcal{X}$.

Flag Mean 2

We have observed that the eigenvectors $\{\hat{u}^{(m)}\}$ that form our flag mean come from the solutions to the eigenvector problem

$$\mathcal{X}\mathcal{X}^T u = \lambda u.$$

So, as we have already observed, these vectors $\{\hat{u}^{(m)}\}$ are actually the left singular vectors of $\mathcal{X}$.

We now have an interesting connection between multiset CCA and our flag vectors $\{\hat{u}^{(m)}\}$ from the equation

$$\hat{u}^{(m)} = \frac{\mathcal{X} \alpha^{(m)}}{\sigma^{(m)}}.$$

From this it follows that

$$\hat{u}^{(m)} = \frac{1}{\sigma^{(m)}} \sum_i X_i \alpha_i^{(m)}$$

or

$$\hat{u}^{(m)} = \frac{1}{\sigma^{(m)}} \sum_i k_i^{(m)}.$$

In other words, each flag mean 2 vector is a scaling of the average of the canonical vectors in multiset CCA.

Connection to CCA-MAXVAR

We note that although $\hat{u}$ did not appear to be of any central interest to authors of the multiset CCA paper [27], it was of interest to Kettenring (1971) [13] who called it z. He was interested in finding z and y_i to

$$\text{minimize} \quad \sum_{i}^{N} \|z - a_i y_i\|^2$$

where

$$y_i = D_i f_i$$

with the constraint that $\|y_i\| = 1$.

This optimization looks very different from ours, and yet it has a similar objective: to find a vector in the middle of a collection of data matrices. Via shows that the multiset CCA produces the solution for z as

$$z = \frac{\mathcal{D}\alpha}{N}$$

with $\mathcal{D} = [D_1, D_2, \ldots, D_N]$.

Thus, if one were to use orthonormal bases X_i in place of data matrices D_i, then Kettenring's method for computing z is the same as ours.

Cosines to Basis Vectors

We examine an interesting alternate interpretation of flag mean 2, involving a different set of squares of cosines. Remember that the flag mean vectors are determined to maximize the sum of the squares of the principal angles:

$$\sum_{i=1}^{N} \cos^2 \theta_i = \sum_{i=1}^{N} u^T X_i X_i^T u.$$

Notice that

$$u^T X_i X_i^T u = \sum_{j=1}^{p} (X_{ij}^T u)^2$$

where X_{ij} is the j^{th} column of the i^{th} orthonormal basis matrix. But then we have

$$\sum_{ij} (X_{ij}^T u)^2 = \sum_{ij} \cos^2 \phi_{ij}$$

where ϕ_{ij} is the angle between u and the jth basis vector of the ith orthonormal basis. Summarizing, our optimization problem, and thus the geometric interpretation of $\hat{u}$, can be expressed in terms of $\{\theta_i\}$ or $\{\phi_{ij}\}$.

Novel Aspects of Flag Mean 2

In comparing our work to MCCA, we may identify a number of fresh contributions. To begin with, we work with bases X_i from the beginning, instead of data sets D_i. We solve the eigenvector problem $\mathcal{X}\mathcal{X}^T u = \lambda u$, instead of $\mathcal{X}^T \mathcal{X} \alpha = \lambda \alpha$. Our novel form for the optimization problem, in terms of squares of cosines of principal angles or basis vector angles, provides additional insight into the geometry of the data fitting. We provide an interesting connection between our work and MCCA using the SVD of $\mathcal{X}$, and provide a different proof of the fact that $\hat{u}$ is the average of the canonical vectors. Finally, we also observe the flag structure of the vectors of the flag mean 2.

5. APPLICATIONS OF MATRIX MANIFOLD MEANS

5.1 Normal Mean Experimental Results

5.1.1 KTH Action Data

As an illustration of the normal mean, we consider the KTH action video data set [21]. This data set consists of 600 video files, based on 25 subjects, 6 actions, and 4 scenarios. The resolution of each grayscale image in the videos is 120×160, meaning 19200 pixels per image. We compute a Grassmann manifold normal mean, a Stiefel manifold normal mean, and a Karcher mean of points generated from the video of subject 1 performing the action of handwaving in scenario 1. First we normalize by subtracting the mean of video images 1 through 15 from each image. Then we form five orthonormal matrices, $\{X_1, ..., X_5\}$, $X_i \in St(19200, 3)$ and $[X_i] \in Gr(19200, 3)$, based on these mean-subtracted images, according to the following procedure. We raster-scan images 1 through 3, concatenate them into a matrix M_1 of size 19200×3, and find an orthonormal basis $X_1 = UV^T$ for M_1 using the thin SVD, $M_1 = USV^T$. Likewise, we advance from image 4 through image 6, images $7-9$, images $10-12$ and images $13-15$ to form the remaining X_i.

For all the mean computations recorded below, we used $\epsilon = 10^{-4}$. The three reshaped column vectors of the Grassmannian normal mean of $\{[X_i]\}_{i=1}^5$ are displayed in Figure 5.1. Algorithm 2 took 91 iterations to converge. The three reshaped column vectors of the Stiefel normal mean of $\{X_i\}_{i=1}^5$ are displayed in Figure 5.2. Algorithm 3 took 28 iterations to converge. The three reshaped column vectors of the Karcher mean of $\{[X_i]\}_{i=1}^5$ are displayed in Figure 5.3. The Karcher mean algorithm, Algorithm 1, took 78 iterations to converge. We remark that another visualization tool is to stack the three photographs comprising a particular 3-dimensional mean and view as an RGB color photograph.

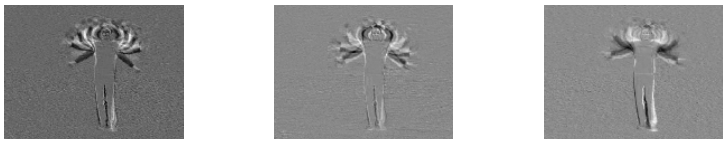

Fig. 5.1: Reshaped Grassmannian normal mean column vectors from KTH action data.

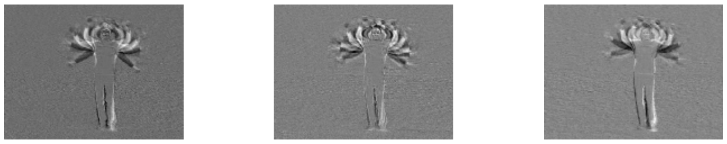

Fig. 5.2: Reshaped Stiefel normal mean column vectors from KTH action data.

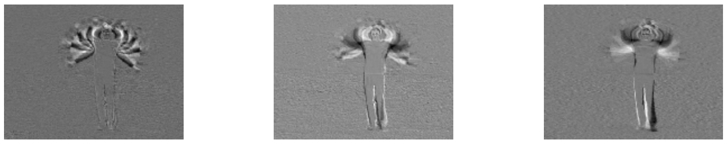

Fig. 5.3: Reshaped Karcher mean column vectors from KTH action data.

5.1.2 CSU PAL Faces

We now draw attention to the advantages of a Stiefel mean versus a Grassmannian mean, by applying our algorithms to color photographs. We examine RGB photos of faces selected from video sequences generated by the Colorado State University Pattern Analysis Lab. In particular, we take five images from the video of the first subject engaged in protocol 2, smiling, under the lighting condition 2, which included frequent stark change in light source. The resolution of each image is 1440×1080, meaning 1555200 pixels in each of the three color sheets. We raster-scan each color sheet and concatenate horizontally to obtain five 1555200×3 matrices, $M_1, ..., M_5$. We form an orthonormal basis matrix X_i, $[X_i] \in Gr(1555200, 3)$, for the column space of each of the M_i using the thin SVD $M_i = X_i S V^T$. We also form an orthonormal basis Y_i for M_i by using the Gram-Schmidt procedure [23], so that $Y_i \in St(1555200, 3)$.

Based on $\epsilon = 10^{-4}$, we calculate the Grassmannian normal mean of the $\{[X_i]\}_{i=1}^5$ using Algorithm 2 and the Stiefel normal mean of the $\{Y_i\}_{i=1}^5$ using Algorithm 3, and note rapid convergence. To allow the means, as well as the X_i and Y_i, to be displayed as images, we perform an affine transformation to scale the entries to be in the standard 0 to 255 range, and reshape to the original image dimensions. The scaled and reshaped $\{X_i\}_{i=1}^5$ are shown in Figure 5.4, while the $\{Y_i\}_{i=1}^5$ are shown in Figure 5.5. From Figure 5.6, we observe that the Stiefel mean preserves some color information. In contrast, the Grassmannian mean has lost this information. This matches our intuition, as points on the Stiefel manifold are particular orthonormal bases, while points on the Grassmann manifold are subspaces, each represented by an equivalence class of orthonormal bases.

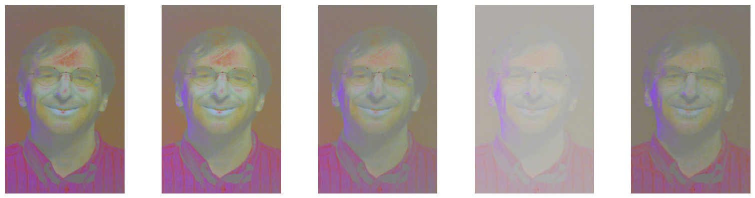

Fig. 5.4: Scaled and reshaped Grassmannian point representatives $\{X_i\}_{i=1}^5$ from CSU PAL faces data.

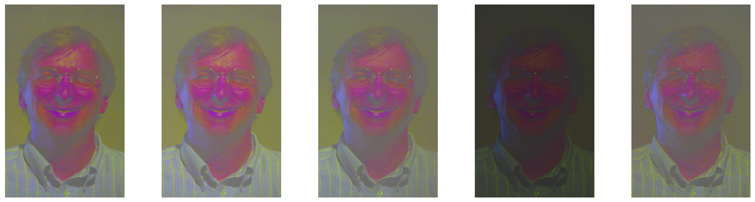

Fig. 5.5: Scaled and reshaped Stiefel points $\{Y_i\}_{i=1}^5$ from CSU PAL faces data.

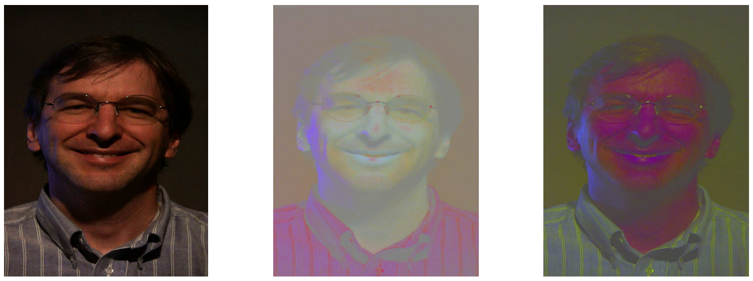

Fig. 5.6: Normal means for CSU PAL faces data. Left: One of the five original images. Center: Grassmannian mean. Right: Stiefel mean. We observe that the blue of the shirt and the red of the face are preserved better in the Stiefel mean.

5.2 Flag Mean Experimental Results

5.2.1 Mean of Two Intersecting Planes

As an example, we compute the flag mean of dimension $q = 2$ of two intersecting planes in $\mathbb{R}^3$. The first plane, P_1, is the xy-plane, represented by the orthonormal matrix

$$Y_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The second plane, P_2, is defined as the span of the vectors $[1, 0, 0]$ and $[0, \frac{1}{2}, \frac{\sqrt{3}}{2}]$. We use the orthonormal matrix

$$Y_2 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

to represent P_2.

Upon application of the flag mean 1 algorithm to this pair of 2-dimensional subspaces of $\mathbb{R}^3$, the first flag mean vector is $\hat{u}^{(1)} = [1, 0, 0]$. This result is to be expected because P_1 and P_2 share this direction, so there is a minimum principal angle sum of 0 (see (4.3.1)). The second flag mean vector is $\hat{u}^{(2)} = [0, \frac{\sqrt{3}}{2}, \frac{1}{2}]$. This result is also accurate, as $\hat{u}^{(2)}$ lies at the minimizing angle of exactly $\frac{\pi}{6}$ radians to both the xy-plane and P_2. In Figure 5.7 we see P_1 depicted in dark blue, P_2 depicted in green, and the flag mean plane (the span of the vectors $\hat{u}^{(1)}$ and $\hat{u}^{(2)}$) depicted in red. Application of the flag mean 2 algorithm yields precisely the same mean plane.

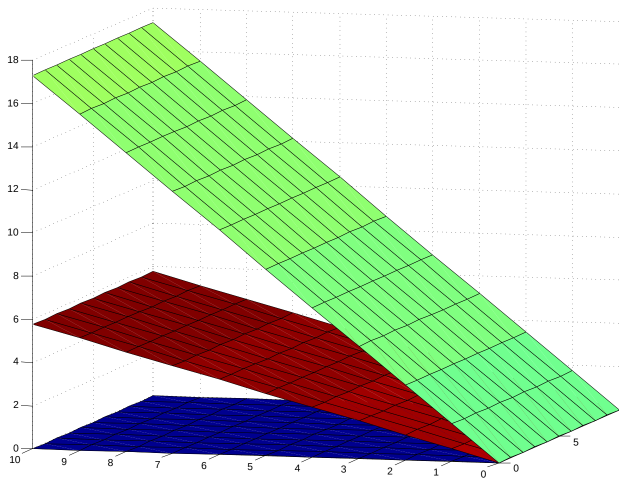

Fig. 5.7: The flag mean plane of the two intersecting planes, P_1 and P_2, is depicted in red.

5.2.2 Mean for Four Image Planes

We construct four 2-dimensional image subspaces in $\mathbb{R}^{1120000}$. Each subspace is the span of two vectors, obtained by raster-scanning images from the PAL Face Database. We select subject 1 and pair an image of subject 1 with variable illumination but fixed pose with an image of a distinct subject with the identical pose. Figure 5.8 displays the 8 (pre-raster-scanned) grayscale face images, with subspace pairings running vertically.

We proceed to compute flag mean 1 with $q = 8$. We make the guess of $\hat{u}^{(0)} = \frac{1}{\sqrt{1120000}}e$ where e is a tall vector of 1's. The resulting sequence of 8 flag mean vectors is computed in 24.96 seconds, and is shown in Figure 5.9. Observe that the first flag mean vector lies

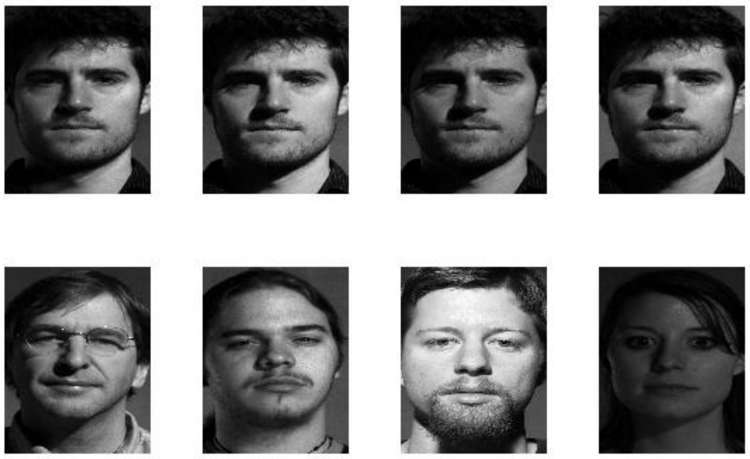

Fig. 5.8: Original subspace pairings run vertically

in the common subspace direction of subject 1. The subsequent flag mean vectors display attributes of the other subjects as well as subject 1. Flag mean 2 is admissible if we use the reduced size SVD solution, as the large eigenvector problem is intractable. Using the economy version takes only 2.12 seconds. The result is shown in Figure 5.10. For comparison, the Karcher mean algorithm took 4 iterations or 11.70 seconds and the Grassmann manifold normal mean algorithm took 5 iterations or 26.42 seconds on this face data set.

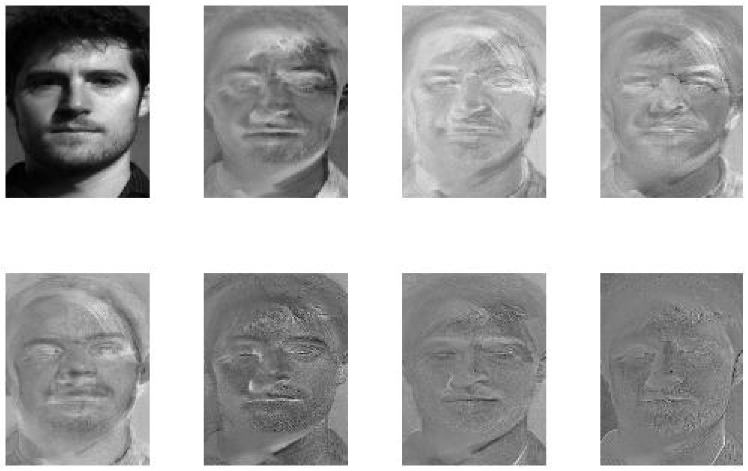

Fig. 5.9: Flag mean 1 vectors 1-8 run left to right, top to bottom. Notice the clarity of subject 1 in $\hat{u}^{(1)}$.

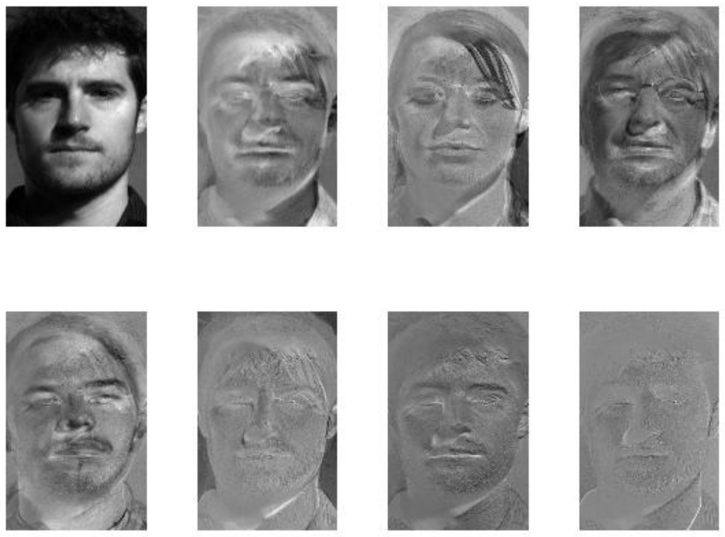

Fig. 5.10: Flag mean 2 vectors 1-8 run left to right, top to bottom. Again we can see sharply the identity of subject 1 in $\hat{u}^{(1)}$.

5.3 Computational Complexity Analysis

It behooves us to compare the computational complexities of the mean algorithms. We begin by recording the computational cost of operations relevant to our algorithms, such as matrix multiplication and the thin SVD. These are compiled in Table 5.1.

Tab. 5.1: Computational costs for revelant operations

Computation	**Complexity (worst case)**
$n \times p$ by $p \times p$ matrix multiplication	$O(np^2)$
$p \times n$ by $n \times p$ matrix multiplication	$O(np^2)$
$p \times p$ matrix inverse	$O(p^3)$
$n \times p$ matrix thin SVD	$O(np^2)$
Solve linear system with A $p^2 \times p^2$	$O(p^4)$ (reducible because of sparsity)
Solve $n \times n$ eigen problem	$O(n^3)$
Add two $n \times p$ matrices	$O(np)$

Now suppose that we have N manifold points, each represented by a matrix of size $n \times p$. One may verify that the computational complexities of each of the algorithms are as displayed in Table 5.2. In general, the Karcher mean and Grassmannian normal mean are comparable in speed, while the Stiefel normal mean is slower and the flag means are faster.

Tab. 5.2: Summary of matrix manifold mean methods. N is the number of subspaces, n is the dimension of the ambient space, and p is the dimension of the subspaces

Algorithm	**Environment**	**Computational Complexity**
Normal Mean	Grassmann	1 iter: $O(Nnp^2) + O(Np^3)$
Normal Mean	Stiefel	1 iter: $O(Nnp^2) + O(Np^4)$
Projection Mean	Stiefel	TBD
Flag (cos sq)	subspaces	Entire flag mean: $O(n^3)$ or $O(n(Np)^2)$
Flag (cos)	subspaces	1 iter for 1 flag mean vector: $O(Nnp)$
Karcher Mean	Grassmann	1 iter: $O(Nnp^2) + O(Np^3)$

5.4 Mean Benefit Comparisons

The novelty and advantages of these new means when compared to the Karcher mean are manifold. The normal mean provides a mean on both the Grassmann and the Stiefel manifold, and permits the F-normal mean variant. The flag mean can be computed over varying dimension subspaces, while the Karcher mean requires fixed dimension subspaces. We can generate an arbitrary dimension flag mean, while the Karcher mean is a subspace of dimension matching the given subspaces. The flag mean has flag structure. In other words, there is an intrinsic ordering to the flag mean orthonormal basis, while the Karcher mean orthonormal basis does not invite such interpretation. The flag mean 2 formulation has a closed form eigenvector solution, versus an iterative solution. In addition, the flag mean objective function involves cosines of principal angles in place of principal angles themselves, which decreases sensitivity to outliers. For these reasons, and many others, the contribution of these novel mean algorithms is highly significant.

6. CONCLUSION

We furnish a summary of the findings of this research. We have confidently developed the normal mean on the Grassmann and Stiefel manifolds. We have derived a map from a point in the tangent space of the Grassmann manifold to the closest Grassmannian point. This could be coupled with a projection map that is well-defined on Grassmannian points to produce a projection mean algorithm. We have offered evidence that the computation of a projection mean on the Stiefel manifold reduces to exhaustively solving a continuous algebraic Riccati equation, and selecting the minimum norm solution. Our work has generated two versions of the flag mean, one of which utilizes principal angles and the other utilizes the *square* of the principal angles. The former is solved via an iterative method, sensitive to initial condition, while the latter is an eigenvector problem. A weighted flag mean was considered.

We make a few observations about our results. The success of the normal mean was tied to the unique point of intersection between a normal space and a tangent space for both the Grassmann and Stiefel manifold. The intersection of the normal space and the Grassmann manifold or Stiefel manifold is a family of orthogonal matrices, which increases problem difficulty. For the Grassmann manifold, the ideal member of this family can be selected but there is no accompanying projection map. For the Stiefel manifold, we must generate the entire set of solutions for a particular quadratic equation, which is demonstrated to be a non-trivial, but approachable, continuous algebraic Riccati equation. In this way, it is clear that the normal mean has a simpler derivation than the projection mean. Despite the difficulties of the projection mean, we expect to soon develop practical projection mean algorithms, first on the Stiefel, and then on the Grassmann manifold. Experiments show successful and sensible results using both of the flag mean methods. In general, the orientation of bases and initialization vector are major factors in the outcome of the first flag mean. We hope to adjust the flag mean iterative method to be robust to initialization. The second flag mean method is stable, and provides an enlightening insight into the core theory of multiple-set canonical correlation analysis.

BIBLIOGRAPHY

[1] P. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

[2] P. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, Princeton, NJ, 2008.

[3] I. Arnold, W.F. and A. Laub. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proceedings of the IEEE*, 72(12):1746 – 1754, dec. 1984.

[4] C. Baker. *Riemannian manifold trust-region methods with applications to eigenproblems.* PhD thesis, Florida State University, 2008.

[5] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: Software for numerical algebraic geometry. Available at http://www.nd.edu/∼sommese/bertini.

[6] E. Begelfor and M. Werman. Affine invariance revisited. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2087–2094, Washington, DC, USA, 2006. IEEE Computer Society.

[7] J. Beveridge, B. Draper, J. Chang, M. Kirby, and C. Kley, H.and Peterson. principal angles separate subject illumination spaces. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):351–356, 2009.

[8] A. Bjorck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comp.*, 27:579–594, 1973.

[9] J. Chang, M. Kirby, H. Kley, C. Peterson, B. Draper, and J. Beveridge. Recognition of digital images of the human face at ultra low resolution via illumination spaces. In *Proceedings of the 8th Asian conference on Computer vision-Volume Part II*, pages 733–743. Springer-Verlag, 2007.

[10] Y. Chikuse. Procrustes analysis on some special manifolds. statistical inference and data analysis (tokyo, 1997). *Comm. Statist. Theory Methods*, 28:885–903, 1999.

[11] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl*, 20:303–353, 1998.

[12] J. Hamm. *Subspace-Based Learning with Grassmann Kernels.* PhD thesis, University of Pennsylvania, 2008.

[13] J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

[14] T.-K. Kim, J. Kittler, and R. Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *European Conference on Computer Vision (ECCV*, pages 251–262, 2006.

[15] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE*, 29(6):1005–1018, 2007.

[16] A. Laub. A schur method for solving algebraic riccati equations. *Automatic Control, IEEE Transactions on*, 24(6):913 – 921, dec 1979.

[17] Y. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 833–839, June 2010.

[18] B. Mondal, S. Dutta, and R. Heath. Quantization on the grassmann manifold. *IEEE Trans. on Signal Processing*, 55(8):4208–4216, 2007.

[19] I. Rahman, I. Drori, V. Stodden, D. Donoho, and P. Schroder. Multiscale representations for manifold-valued data. *Multiscale Model. Simul.*, 4(4):1201–1232, 2005.

[20] P. H. Schoneman. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.

[21] C. Schouldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *International Conference on Pattern Recognition*, 2004.

[22] M. Shub. Some remarks on dynamical systems and numerical analysis. In *Proc. VII ELAM (L. Lara-Carrero and J. Lewowicz, eds.), Equinoccio, U. Simon Bolivar, Caracas*, pages 69–92, 1986.

[23] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM: Society for Industrial and Applied Mathematics, 1997.

[24] 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.

[25] K. Turner. Cone fields and topological sampling in manifolds with bounded curvature. *arXiv:1112.6160v1*, 2012.

[26] K. Varshney and A. Willsky. Linear dimensionality reduction for margin-based classification: high-dimensional data and sensor networks. *IEEE Trans. Signal Processing*, 59:2496–2512, 2011.

[27] J. Va, I. Santamara, and J. Prez. A learning algorithm for adaptive canonical correlation analysis of several data sets. *Neural Networks*, 20(1):139 – 152, 2007.

[28] T. Wang and P. Shi. Kernel grassmannian distances and discriminant analysis for face recognition from image sets. *Pattern Recognition Letters*, 30(13):1161–1165, 2009.

[29] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *The Journal of Machine Learning Research*, 4:913–931, 2003.

Appendix A

MATLAB CODES

A.1 Calculate Distance on Grassmann Manifold

```
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                       dG function
% Input: representatives, A and B, for points [A] and [B] on Grassmann
%        manifold
% Output: Distance, d, between [A] and [B], as measured by the square
%         root of the sum of the squares of the principal angles
%
%%%%%%%%%%%%%%%%%%%%%%%%%%


function d = dG(A,B)

[U1,S1,V1] = svd(A,0);
[U2,S2,V2] = svd(B,0);
cosV = svd(U1'*U2);  %vector of cosines of principal angles

d = 0;
for i=1:length(cosV)
    mycos = min(1,cosV(i));
    d = d + acos(mycos)^2;
end
d = sqrt(d);
```

A.2 Calculate Distance on Stiefel Manifold

```
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                       dS function
% Input: points A and B on the Stiefel manifold
% Output: Distance, d, between A and B, as measured by the
%         2-norm of the difference matrix
%
%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function d = dS(A,B)

d = sqrt(trace((A-B)'*(A-B)));
```

A.3 Calculate Karcher Mean of Grassmannian Points

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        GmuKM function
% Input: tensor, M, of matrices representing points on a Gr
% Output: representative, mu, for the Karcher mean, [mu], on Gr
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function mu = GmuKM(M)

maxiter = 1000;
eps = 10^(-4); %control how small norm(delta) must be

numMat = size(M,3);
onMat = zeros(size(M));

%obtain orthonormal bases
for i=1:numMat
   [U,S,V] = svd(M(:,:,i),0);
   onMat(:,:,i) = U;
end

%initialization of mu
mu=onMat(:,:,1);

%compute first delta
delta = zeros(size(M,1),size(M,2));
for i=1:numMat
    delta = delta+GLOG(mu,onMat(:,:,i));
end
delta = 1/numMat*delta;

%iterate to find KM
iter = 0;
while norm(delta,'fro')>=eps &&  iter<maxiter
    iter = iter+1;
    mu = GEXP(mu,delta);  %map delta to the manifold: new mu
    delta = zeros(size(M,1),size(M,2));
    for i=1:numMat
      delta = delta+GLOG(mu,onMat(:,:,i));
    end
    delta = 1/numMat*delta; %ave corresp tangent v's: new delta
end
display(['KM took ' num2str(iter) ' iters to converge'])

%print the minimized quantity
TotalDist = 0;
for i =1:numMat
    TotalDist = TotalDist + dG(onMat(:,:,i),mu)^2;
end
display(['Sum sqrd dists btw muKM and manifold pts: ' TotalDist])
```

A.4 Subroutine for Karcher Mean: Compute Log

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
% Extracted from Begelfor
%
%                        GLOG function
% Input: representative, Y, for point of tangency [Y] on Grassmann
%        manifold, and representative X for [X],
%        another point on the Grassmannian
% Output: T on the tangent space to Grassmannian at [Y],
%          norm(T) is equal to the distance from [X] to [Y]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function T = GLOG(Y,X)

%[U,S,V] = svd((eye(size(Y,1))-Y*Y')*X*inv(Y'*X),0); %orig idea
%[U,S,V] = svd((X-Y*(Y'*X))*inv(Y'*X),0);
quantity = X/(Y'*X)-Y; %shortcut
[U,S,V] = svd(quantity,0);

theta = diag(atan(diag(S)));
T = U*theta*V';
```

A.5 Subroutine for Karcher Mean: Compute Exp

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
% Extracted from Begelfor
%
%                        GEXP function
% Input: representative, Y, for point of tangency [Y] on Grassmann
%        manifold, and vector T in tangent space to manifold at [Y]
% Output: X, a representative for a point, [X], on Grassmannian
%          whose distance to [Y] is equal to norm(T)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function X = GEXP(Y,T)

[U,S,V] = svd(T,0);
X = Y*V*diag(cos(diag(S)))+U*diag(sin(diag(S)));
```

A.6 Calculate Normal Mean of Grassmannian Points

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                       GmuN function
% Input: tensor, M, of matrices representing points on a Gr
% Output: representative, mu, for the normal mean, [mu], on Gr
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function mu = GmuN(M)

maxiter = 1000;
eps = 10^(-4);
numMat = size(M,3);
n = size(M(:,:,1),1);
p = size(M(:,:,1),2);

%obtain orthonormal bases
for i=1:numMat
    [U,S,V] = svd(M(:,:,i),0);
    M(:,:,i)=U;
end
%initialize mu by guessing one of the given points
mu = M(:,:,1);
%find the first delta
delta = zeros(n,p);
for i=1:numMat
    delta = delta+R_Ginv(mu,M(:,:,i));
end
delta = 1/numMat*delta;

iter = 0;
while norm(delta,'fro')>=eps && iter < maxiter
    iter = iter+1
    mu = R_G(mu,delta);  %map delta to the manifold: new mu
    %compute the new delta
    delta = zeros(n,p);
    for i=1:numMat
      delta = delta+R_Ginv(mu,M(:,:,i));
    end
    delta = 1/numMat*delta;  %ave tangent v's: new delta
end
display(['G normal mean: ' num2str(iter) ' iters to converge'])

%print the quantity that the KM minimizes
TotalDist = 0;
for i =1:numMat
    TotalDist = TotalDist + dG(M(:,:,i),mu)^2;
end
display(['Sum sqrd dists btw muGN and manifold pts: ' TotalDist])
```

A.7 Subroutine for Grassmann Normal Mean: Compute Retraction

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        R_G function
% Input: representative, Y, for point of tangency [Y] on Gr
%        manifold, and vector T in tangent space to manifold at [Y]
% Output: X, the closest orthonormal matrix to T+Y
%          (a representative for the closest point on the Gr, [X])
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function X = R_G(Y,T)

[U,S,V] = svd(T+Y,0); %thin svd
X = U*V';
```

A.8 Subroutine for Grassmann Normal Mean: Compute Retraction Inverse

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        R_Ginv function
% Input: Representatives for [Y], the point of tangency on Gr,
%        and [X], another point on the Grassmannian
% Output: T on tangent space to Grassmannian at [Y],
%          via normal retraction inverse map applied to [X]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function T = R_Ginv(Y,X)
```

```
T = X/(Y'*X)-Y;

%check to ensure T is a tangent vector
error = norm(Y'*T,'fro');
if error>10^-3
    display('did not land on Grass Tangent plane');
end

%to allow for scaling vector
l = norm(T,'fro');
if l < 10^(-6)  %essentially zero vector
    T = T*0;
else
    pow = 1; %adjust power to control magnitude of tangent vector
    scal = dG(Y,X)^pow;  %scale length to power of manifold dist
    T = T/l*scal;
end
```

A.9 Calculate Normal Mean of Stiefel Points

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        SmuN function
% Input: tensor, M, of matrices representing points on a Stiefel
% Output: mu, the normal mean on the Stiefel
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function mu = SmuN(M)

n = size(M(:,:,1),1);
p = size(M(:,:,1),2);
numMat = size(M,3);
maxiter = 1000;
eps = 10^(-4);  %how small norm(delta) needs to be to terminate

%obtain orthonormal bases
 for i =1:numMat
    [U,S,V] = svd(M(:,:,i),0);
    M(:,:,i) = U*V'; %closest orthonormal basis
 end

%initial guess of mu
mu = M(:,:,1);

%obtain first delta
delta = zeros(n,p);
for i=1:numMat
```

```
    delta = delta+R_Sinv(mu,M(:,:,i));
end
delta = 1/numMat*delta; %initial delta

iter = 0;
while norm(delta,'fro')>=eps && iter<maxiter
    iter = iter+1;
    mu = R_S(mu,delta);  %map delta to the manifold: new mu
    delta = zeros(n,p);
    for i=1:numMat
      delta = delta+R_Sinv(mu,M(:,:,i));
    end
    delta = 1/numMat*delta; %ave tangent v's: new delta
end
display(['Stiefel normal mean: ' num2str(iter) ' iters to converge.'])

%print the quantity that the KM minimizes
TotalDist = 0;
for i =1:numMat
    TotalDist = TotalDist + dG(M(:,:,i),mu)^2;
end
display(['Sum sqrd dists btw muSN and manifold pts: ' TotalDist])
```

A.10 Subroutine for Stiefel Normal Mean: Compute Retraction

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        R_S function
% Input: point of tangency, Y, on Stiefel manifold,
%        and vector T in tangent space to manifold at Y
% Output: X, the closest point on the Stiefel manifold to T+Y
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function X = R_S(Y,T)

[U,S,V] = svd(T+Y,0);
X = U*V';
```

A.11 Subroutine for Stiefel Normal Mean: Compute Retraction Inverse

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                       R_Sinv function
% Input: Y, the point of tangency on Stiefel manifold,
%        and X, another point on the Stiefel
% Output: T on tangent space to Stiefel at Y, via
%         normal retraction inverse map applied to X
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function T = R_Sinv(Y,X)

nr = size(Y,1);
nc = size(Y,2);
M = Y'*X;
S = zeros(nc);
A = zeros(nc^2);
b = [];

rowofA = 1;
for i=1:nc
    for j=i:nc
        if i==j;
            A(rowofA,(i-1)*nc+1:i*nc) = M(j,:);
            b = [b; 1];
            rowofA = rowofA +1;
        else
            A(rowofA,(i-1)*nc+1:i*nc) = M(j,:);
            A(rowofA,(j-1)*nc+1:j*nc) = M(i,:);
            b = [b; 0];
            rowofA = rowofA +1;

            symCond = zeros(nc);
            symCond(i,j)=1;
            symCond = symCond - symCond';
            A(rowofA,:) = symCond(:);
            b = [b; 0];
            rowofA = rowofA+1;
        end
    end
end

S=A\b;
S = reshape(S,nc,nc);
S=S';
T=X*S-Y;

%error checking
```

```
if norm(Y'*T+T'*Y,'fro')>10^-4
    display('did not land on Stief Tangent plane');
end

%To allow for scaling of vector
l = norm(T,'fro');
if l < 10^(-6)
    T = T*0;
else
    eps = 1;
    T = T/l*eps*dS(X,Y);
end
```

A.12 Calculate Flag Mean Formulation 1 of Subspaces

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                        muFLAG1 function
% Input: array, M, of matrices with column spaces defining
%        subspaces of R^n and
%        dimension d <= dim(span(union of the column spaces))
% Output: representative, mu, for the 1-norm flag mean
%         of dim d, [mu]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function mu = muFLAG1(M,d)

numsubsp = length(M);
n = size(M{1},1); %ambient space dim
maxIter = 1000;
progTol = 10^-6; %designates some movement

%obtain orthonormal bases
for i = 1:numsubsp
    X = M{i};
    [X,S,V] = svd(X,0);
    M{i} = X;
end

%find each of the d flag mean vectors iteratively
mu = zeros(n,d);
for vect = 1:d
    u = ones(n,1)/sqrt(n); %guess
    progress = 100;
    iter = 0;
    while progress >= progTol && iter<maxIter
        sum=zeros(n,1);
        for i = 1:numsubsp
```

```
            temp = M{i}'*u;
            alpha=temp/norm(temp);
            sum = sum+M{i}*alpha;
        end
        unew = sum;
        for prev = 1:vect-1
            prevvect = mu(:,prev);
            unew = unew - prevvect'*sum*prevvect;
        end
        unew = unew/norm(unew);
        progress = acos(abs(unew'*u)); %angle btw iterates: progress
        u = unew;
        iter = iter + 1;
    end
    mu(:,vect) = u;
    display(['Vect ' int2str(vect)  ': ' num2str(iter) ' iters'])
end
```

A.13 Calculate Flag Mean Formulation 2 of Subspaces

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Justin Marks, Colorado State University, June 2012
%
%                       muFLAG2 function
% Input: array, M, of matrices with column spaces defining
%        subspaces of R^n
%        and dimension d <= dim(span(union of the column spaces))
% Output: representative, mu, for the 2-norm flag mean
%          of dim d, [mu]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

function mu = muFLAG2(M,d)

numMat = length(M);
n = M{1};
n = size(n,1);
basisSum = zeros(n);

%obtain orthonormal bases
for i=1:numMat
   [U,S,V] = svd(M{i},0);
   basisSum = basisSum + U*U';
end

[mu, eigvals] = eigs(basisSum);
mu = mu(:,1:d);
```