arXiv:2203.04437v1 [stat.ML] 8 Mar 2022

THE FLAG MEDIAN AND FLAGIRLS

Nathan Mankovich, Emily J. King, Chris Peterson, and Michael Kirby
Department of Mathematics
Colorado State University
Fort Collins, Colorado
{nmank, emily.king, christopher2.peterson, and michael.kirby}@colostate.edu

ABSTRACT

Finding prototypes (e.g., mean and median) for a dataset is central to a number of common machine learning algorithms. Subspaces have been shown to provide useful, robust representations for datasets of images, videos and more. Since subspaces correspond to points on a Grassmann manifold, one is led to consider the idea of a subspace prototype for a Grassmann-valued dataset. While a number of different subspace prototypes have been described, the calculation of some of these prototypes has proven to be computationally expensive while other prototypes are affected by outliers and produce highly imperfect clustering on noisy data. This work proposes a new subspace prototype, the flag median, and introduces the FlagIRLS algorithm for its calculation. We provide evidence that the flag median is robust to outliers and can be used effectively in algorithms like Linde-Buzo-Grey (LBG) to produce improved clusterings on Grassmannians. Numerical experiments include a synthetic dataset, the MNIST handwritten digits dataset, the Mind's Eye video dataset and the UCF YouTube action dataset. The flag median is compared the other leading algorithms for computing prototypes on the Grassmannian, namely, the ℓ_2-median and to the flag mean. We find that using FlagIRLS to compute the flag median converges in 4 iterations on a synthetic dataset. We also see that Grassmannian LBG with a codebook size of 20 and using the flag median produces at least a 10% improvement in cluster purity over Grassmannian LBG using the flag mean or ℓ_2-median on the Mind's Eye dataset.

1 Introduction

The mean and median are basic methods for calculating central prototypes from a probability distribution. The median is commonly more robust to outliers than the mean. Generalizations of such prototypes to Euclidean space can be formulated as a solution to an optimization problem. Suppose we have a set of points in Euclidean space, $\mathscr{X} = \{\mathbf{x}_i\}_{i=1}^p \subset \mathbb{R}^n$ which we would like to represent as a prototype $\mathbf{y}$ via solving

$$\arg\min_{\mathbf{y}\in A} \sum_{i=1}^{p} \|\mathbf{x}_i - \mathbf{y}\|_2^q. \tag{1}$$

The solution to (1) for $A = \mathbb{R}^n$ and $q = 2$ is called the centroid, which may be viewed as the generalization of the mean. In fact, the centroid is the component-wise mean of the vectors in $\mathscr{X}$, $\sum_{i=1}^p \mathbf{x}_i / p$. Generalizations of the median involve solving (1) when $q = 1$. When $A = \mathscr{X}$, the solution is called the medoid, while when $A = \mathbb{R}^n$, the solution is called the geometric median. The geometric median inherits the robustness to outliers from the median without being required to be a point in the dataset; however, it is not as straightforward to compute as a centroid since that calculation is not simply a least squares problem. An iterative algorithm for approximating a geometric median is the Weiszfeld algorithm [1]; each iteration of this algorithm is a weighted centroid problem. Thus, the Weiszfeld algorithm falls into a class known as Iteratively Reweighted Least Squares algorithms (IRLS). These Euclidean prototypes are used as a statistic for a dataset and in common machine learning algorithms like k-means and nearest centroid classification.

Not all datasets are best represented using points in Euclidean space. Specifically, image or video datasets are sometimes better represented using subspaces, i.e., as points on a Grassmannian. For example, the smallest principal angle between two subspaces has proven powerful for modeling illumination spaces [2]. Hyperspectral data

may fail to be linearly separable in Euclidean space but separate linearly on the Grassmannian [3]. Hence, it is potentially useful to find versions of the Euclidean prototypes on the Grassmannian. A logical generalization of prototypes from Euclidean space to the Grassmannian is to replace the Euclidean 2-norm in the optimization problems for the centroid, medoid and geometric median with a distance or dissimilarity between subspaces. The centroid is generalized using the geodesic distance in [4] and the chordal distance in [5]. To the extent of our research, we have have not found a generalization of the medoid. However, the geometric median has been generalized using the geodesic distance and is called the ℓ_2-median in [6, 7]. Prototypes like these have been used alone as a method to classify emotion in images [8], as a step in a k-means type algorithm [9] and in feature extraction [7].

Popular machine learning techniques, like dictionary learning, have been adapted to Riemannian manifolds [10]. Jayasumana et. al. consider learning on the Grassmannian (and Riemannian manifolds in general) with RBF kernels and advocate for chordal distance (sometimes referred to as the projection norm) kernels on the Grassmannian because chordal distance generates a positive definite Gaussian kernel [11]. More recently, Cherian et. al. use kernalized Grassmannian pooling for activity recognition [12]. Methods for Riemannian optimization like Riemannian SVRG have gained popularity alongside this surge of interest in Riemannian learning [13]. Even more uses for subspaces in computer vision and machine learning can be found in, e.g., [6, 14, 5, 15, 16, 9, 8].

In this paper we propose the flag median, a prototype which is a generalization of the geometric median to the Grassmannian using the chordal distance. We solve the flag median optimization problem using the novel FlagIRLS algorithm. The FlagIRLS is an IRLS algorithm on the Grassmannian that solves a weighted flag mean problem at each iteration similar to the way an iteration of the Weiszfeld algorithm solves a weighted centroid problem. We conduct experiments with the flag median, ℓ_2-median and the flag mean on synthetic datasets, the MNIST handwritten digits dataset [17], the DARPA (Defense Advanced Research Projects Agency) Mind's Eye dataset used in [9] and the UCF YouTube action dataset [18]. In these examples we find that the FlagIRLS algorithm tends to converge quickly. We show that the flag median appears to be more robust to outliers than the flag mean and ℓ_2-median, and produces the highest cluster purities in the LBG algorithm [19].

2 Background

2.1 Introduction to the Grassmannian

For the purposes of this paper, the Grassmannian manifold (a.k.a. "the Grassmannian"), denoted $\mathrm{Gr}(k,n)$, is the manifold whose points correspond to the k dimensional subspaces of $\mathbb{R}^n$. We will represent a point in $\mathrm{Gr}(k,n)$ using a tall $n \times k$ real matrix $\mathbf{X}$ with orthonormal columns. The point on $\mathrm{Gr}(k,n)$ determined by $\mathbf{X}$ is the column space of $\mathbf{X}$ and is denoted $[\mathbf{X}]$. Thus if $\mathbf{X}$ and $\mathbf{Y}$ have the same column space then they determine the same point $[\mathbf{X}] = [\mathbf{Y}]$ on $\mathrm{Gr}(k,n)$.

In order to allow more flexibility in our generalization of optimization problems to subspaces, we work with points that are not all necessarily on the same Grassmannian manifold but are in the same ambient space. Suppose we have a set of subspaces of n-dimensional space, $\{[\mathbf{X}_1],[\mathbf{X}_2],\ldots,[\mathbf{X}_p]\}$, where $[\mathbf{X}_i] \in \mathrm{Gr}(k_i,n)$. We want to find an r-dimensional subspace of $\mathbb{R}^n$, $[\mathbf{Y}^*] \in \mathrm{Gr}(r,n)$, that is in some sense the center of these points, i.e., that $[\mathbf{Y}^*]$ is a solution to

$$\arg\min_{[\mathbf{Y}]\in\mathrm{Gr}(r,n)} \sum_{i=1}^{p} d([\mathbf{X}_i],[\mathbf{Y}]) \tag{2}$$

where d measures dissimilarity between its arguments.

Principal angles between subspaces are a common dissimilarity measure that is invariant to orthogonal transformations [20, 5]. Take $[\mathbf{X}],[\mathbf{Y}] \in \mathrm{Gr}(k,n)$. The ith smallest principal angle between $[\mathbf{X}]$ and $[\mathbf{Y}]$, $\theta_i([\mathbf{X}],[\mathbf{Y}]) \in [0,\pi/2]$ is defined as the solution to (3) [20].

$$\begin{aligned} \cos\theta_i([\mathbf{X}],[\mathbf{Y}]) &= \max_{\mathbf{x}\in[\mathbf{X}]} \max_{\mathbf{y}\in[\mathbf{Y}]} \mathbf{x}^T\mathbf{y} = \mathbf{x}_i^T\mathbf{y}_i \\ \text{Subject to } \mathbf{x}^T\mathbf{x} &= \mathbf{y}^T\mathbf{y} = 1 \\ \mathbf{x}^T\mathbf{x}_j &= \mathbf{y}^T\mathbf{y}_j = 0 \text{ for } j = 1,2,\ldots,i-1 \end{aligned} \tag{3}$$

Now let $\theta([\mathbf{X}],[\mathbf{Y}]) \in \mathbb{R}^k$ be the vector of principal angles between $[\mathbf{X}]$ and $[\mathbf{Y}]$. The geodesic distance on $\mathrm{Gr}(k,n)$ is $\|\theta([\mathbf{X}],[\mathbf{Y}])\|_2$ and the chordal distance on $\mathrm{Gr}(k,n)$ is $\|\sin(\theta([\mathbf{X}],[\mathbf{Y}]))\|_2$ [21]. We can calculate these quantities when $[\mathbf{X}] \in \mathrm{Gr}(k,n)$, $[\mathbf{Y}] \in \mathrm{Gr}(r,n)$ where $k \neq r$ by setting the last $\max(k,r) - \min(k,r)$ entries of $\theta([\mathbf{X}],[\mathbf{Y}]) \in \mathbb{R}^{\max(k,r)}$ to 0.

2.2 Geodesic Distance Prototypes

The Euclidean mean and geometric median have been translated to the Grassmannian using the geodesic distances. The mean on the Grassmannian using geodesic distance (the solution to (4) for $q = 2$) is called the Karcher mean and the geometric median using geodesic distance (the solution to (4) for $q = 1$) is called the ℓ_2-median.

$$\arg\min_{[\mathbf{Y}]\in \mathrm{Gr}(r,n)} \sum_{i=1}^{p} \|\theta([\mathbf{X}_i],[\mathbf{Y}])\|_2^q. \tag{4}$$

The Karcher mean and the ℓ_2-median are only computable in the case where all subspaces are of equal dimensions (e.g., $r = k_1 = k_2 \cdots = k_p$) and [9] use examples to show that the available algorithms to compute these prototypes are slow. The most common algorithm for finding the solution to the Karcher mean was discovered by Karcher [4] and Fletcher et. al. [7] show we can find the ℓ_2-median using a Weiszfeld-type algorithm. [9] show that the Karcher mean is not only slow to compute, but also produces lower cluster purities than the ℓ_2-median in their LBG clustering example so we choose not to use the Karcher mean as a prototype in our experiments (Section 5).

For context, the Weiszfeld algorithm for vectors in $\mathbb{R}^n$ is stated in Algorithm 1.

Algorithm 1: Weiszfeld Algorithm in $\mathbb{R}^n$

Data: $\{\mathbf{x}_i\}_{i=1}^p \subset \mathbb{R}^n$
Result: The geometric median $\mathbf{y} \in \mathbb{R}^n$
while *not converged* **do**
  $w_i = \frac{p}{\|\mathbf{x}_i - \mathbf{y}\|_2} \left(\sum_{k=1}^{p} \frac{1}{\|\mathbf{x}_k - \mathbf{y}\|_2} \right)^{-1}$;
  $\mathbf{y} \leftarrow \sum_{i=1}^{p} \frac{w_i \mathbf{x}_i}{p}$;
end

Note that each iteration of the Weiszfeld algorithm (Algorithm 1) is the solution to the least squares problem (1) (with $A = \mathbb{R}^n$ and $q = 2$) for the weighted vectors $w_i \mathbf{x}_i$. The weights, w_i, come from the fact that the geometric median $\mathbf{y}$ satisfies (presuming $\mathbf{y} \notin \mathscr{X}$)

$$\mathbf{y} = \left(\sum_{i=1}^{p} \frac{\mathbf{x}_i}{\|\mathbf{x}_i - \mathbf{y}\|_2} \right) \Big/ \left(\sum_{k=1}^{p} \frac{1}{\|\mathbf{x}_k - \mathbf{y}\|_2} \right).$$

Fletcher et. al. solve (4) for $q = 1$ by generalizing this approach to Riemannian manifolds.

In Section 5, we use the unweighted Weiszfeld-type algorithm from Fletcher et. al. with geodesic distance to calculate the ℓ_2-median on the Grassmannian. Let d be the maximum distance between points in the dataset and let δ be the convergence parameter. We define $N_{d,\delta}$ as the number of iterations of one run of our implementation. The complexity of our implementation of this algorithm in Section 5 is $O\left(npk^2 N_{d,\delta}\right)$.

2.3 The Flag Mean Prototype

Draper et. al. [5] present the flag mean as an average of subspaces of different dimensions using the squared chordal distance. The optimization problem for the flag mean is

$$\arg\min_{[\mathbf{Y}]\in \mathrm{Gr}(r,n)} \sum_{i=1}^{p} \|\sin(\theta([\mathbf{X}_i],[\mathbf{Y}]))\|_2^2. \tag{5}$$

This flag mean determines not only a point on $\mathrm{Gr}(r,n)$, it determines a point on various flag manifolds. A flag manifold is a manifold whose points represent a flag of subspaces $[\mathbf{S}_1] \subset [\mathbf{S}_2] \subset \cdots \subset [\mathbf{S}_r] = \mathbb{R}^n$. If we let $s_i = \dim([\mathbf{S}_i])$, then we say the flag is of type $s_1, s_2, \ldots, s_r$. For more details on flag manifolds, see [22].

We will refer to a flag mean in this paper as the point on $Gr(r,n)$ determined by the flag. Let $[\mathbf{Y}]$ be the flag mean of $\{[\mathbf{X}_i]\}_{i=1}^k$. Let $\mathbf{y}_i$ be the ith column of $\mathbf{Y}$, the orthonormal matrix representation of $[\mathbf{Y}]$. Then the r^{th} "real" flag mean is the point on the flag manifold of type $\{1,2,\ldots,r,n\}$ defined as in (6).

$$[\![\mathbf{Y}]\!] = \mathrm{span}\{\mathbf{y}_1\} \subset \mathrm{span}\{\mathbf{y}_1,\mathbf{y}_2\} \subset \cdots \subset \mathrm{span}\{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_r\} \subset \mathbb{R}^n \tag{6}$$

The point on $Gr(r,n)$ determined by the flag is $\text{span}\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r\} \subset \mathbb{R}^n$.

Draper et. al. [5] show that we can calculate the flag mean by utilizing the singular value decomposition (SVD) of the matrix $[\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p]$. The flag mean, as a point on $Gr(r,n)$, is the span of the r left singular vectors of the corresponding to the r largest singular values.

The complexity of this algorithm is $O\left(n\left(\sum_{i=1}^{p} k_i\right)^2\right)$. Marks [16] suggests calculating weighted flag means in his dissertation. This weighted flag mean calculation will be used as an iteration of the FlagIRLS algorithm introduced in Section 3.2.

3 Flag Median

The translation of the geometric median to the Grassmannian using chordal distance is called the flag median. The optimization problem for this novel prototype is in (7).

$$\arg\min_{[\mathbf{Y}]\in Gr(r,n)} \sum_{i=1}^{p} \|\sin\theta([\mathbf{X}_i],[\mathbf{Y}])\|_2 \tag{7}$$

We call this the flag median since, using FlagIRLS, $[\mathbf{Y}]$ actually is a flag of subspaces rather than a single r dimensional subspace of n dimensional space. This flag median is indeed a median (similar to the geometric median) because it minimizes the chordal distance rather than the squared chordal distance problem in (5) that is solved by the flag mean.

3.1 Derivation

In this section we show that the FlagIRLS algorithm can be used to approximate the flag median. The algorithm derived in this section revolves around weighted flag means of $\{[\mathbf{X}_i]\}_{i=1}^{p}$. For the rest of this paper we will denote the weight of the subspace $[\mathbf{X}_i]$ as w_i.

Notice that the flag median optimization problem in (7) involves a sum of two norms of the vector of sines of principal angles and the flag mean optimization problem in (5) involves squared two norms of the same vector. So, in other words, we are deriving an algorithm, analogous to Weiszfeld and IRLS in the Euclidean setting, that approximates solutions to the 2-norm problem by iteratively solving squared 2-norm problems. So the FlagIRLS algorithm 2 provides an "iterative reweighted least squares" method for approximating the flag median.

Let us begin by translating the flag median problem from (7) to an optimization problem over matrices with orthonormal columns. The eigenvalues of $\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}$ are the entries in the vector $\cos^2(\theta([\mathbf{X}_i],[\mathbf{Y}]))$. Using properties of trace we can show $\text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}) = \sum_{j=1}^{m_i}\cos^2\theta_j([\mathbf{X}_i],[\mathbf{Y}])$ [20]. This allows us to rewrite the flag median problem from (7) as the matrix optimization problem in (8) where $m_i = \min(r, k_i)$.

$$\min_{\substack{\mathbf{Y}\in\mathbb{R}^{n\times r}\\ \mathbf{Y}^T\mathbf{Y}=I}} \sum_{i=1}^{p}\left(m_i - \text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y})\right)^{1/2} \tag{8}$$

We formulate a Lagrangian from this problem using Λ as a symmetric matrix of Lagrange multipliers with entries λ_{ij} in (9).

$$\begin{aligned}\mathscr{L}(\mathbf{Y},\Lambda) = &\sum_{i=1}^{p}\left(m_i - \text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y})\right)^{1/2}\\ &-\langle\Lambda, \mathbf{Y}^T\mathbf{Y} - I\rangle\end{aligned} \tag{9}$$

We then calculate (10) the Lagrangian with respect to the jth column of $\mathbf{Y}$, namely $\mathbf{y}_j$, and set it equal to 0.

$$\mathbf{y}_j^T\sum_{i=1}^{p}\frac{-1}{\left(m_i - \text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y})\right)^{1/2}}\mathbf{X}_i\mathbf{X}_i^T\mathbf{y}_j = 2\lambda_{jj} \tag{10}$$

Now define the matrix $\mathbf{X}$

$$\mathbf{X} = \left[w_1\mathbf{X}_1, w_2\mathbf{X}_2, \cdots, w_p\mathbf{X}_p\right] \tag{11}$$

where $w_i = \left(\frac{1}{m_i - \text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y})}\right)^{1/4}$.

Combining the information in (10) and the matrix $\mathbf{X}$ in (11), we see that $\mathbf{Y}$ must be r left singular vectors of $\mathbf{X}$ when $[\mathbf{Y}]$ to be the flag median of $\{[\mathbf{X}_i]\}_{i=1}^p$.

Now let us consider an iterative algorithm with the jth iteration of the form $\mathbf{Y}_{j+1} = \text{Flag Mean}\left(\left\{w_i^{(j)}\mathbf{X}_i\right\}_{i=1}^p\right)$ where $w_i^{(j)} = \left(\frac{1}{m_i - \text{tr}(\mathbf{Y}_j^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_j)}\right)^{1/4}$. This algorithm will be formalized in Section 3.2. For this type of algorithm, we desire $\mathbf{Y}_{j+1}$ to be an approximation of the flag median of the dataset $\{\mathbf{X}_i\}_{i=1}^p$. We will now show that the columns of $\mathbf{Y}_{j+1}$ should be chosen as the left singular vectors of $\mathbf{X}$ associated with the r largest singular values and therefore the update using the flag mean of $\left\{w_i^{(j)}\mathbf{X}_i\right\}_{i=1}^p$ is the correct update choice.

Let $\mathbf{U}^*$ be some matrix consisting of r left singular vectors of $\mathbf{X}$. To determine $\mathbf{U}^*$ where

$$\mathbf{U}^* = \arg\min \sum_{i=1}^{p} \left(q - \text{tr}(\mathbf{U}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{U})\right)^{1/2}. \tag{12}$$

we solve the optimization problem

$$\max_{\substack{\mathbf{U}\in\mathbb{R}^{n\times r}\\ \mathbf{U}^T\mathbf{U}=\mathbf{I}}} \text{tr}\left(\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U}\right) \tag{13}$$

which requires the columns of $\mathbf{U}^*$ to be the left singular vectors of $\mathbf{X}$ associated with the largest singular values. So we take our update to be $\mathbf{Y}_{j+1} = \mathbf{U}^*$.

3.2 The FlagIRLS Algorithm

We use an iteratively reweighted least squared flag mean algorithm structure to solve (7). We will call the weight for subspace $\mathbf{X}_i$, w_i. A concern with these weights arises when the denominator of a weight is zero. For the flag median objective function, the denominator is zero when $[\mathbf{Y}]$ is a subspace of $[\mathbf{X}]$ or vise versa. To avoid singularities, we added a small quantity ϵ to the denominator. The w_i for the flag median problem is in (14)

$$w_i = \left(\frac{1}{m_i - \text{tr}(\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}) + \epsilon}\right)^{1/4} \tag{14}$$

We use these weights, along with the flag mean, in the FlagIRLS algorithm as described in Algorithm 2.

Algorithm 2: The FlagIRLS algorithm. See (14) for the algorithm weights. We assume the columns of $\mathbf{U}$ are sorted from the left singular vector associated with the smallest to the largest singular values of $\mathbf{X}$.

Input: A set of orthonormal subspace representatives $\{\mathbf{X}_i\}_{i=1}^p$ for $\{[\mathbf{X}_i] \in \text{Gr}(k_i, n)\}_{i=1}^p$
Output: An orthonormal subspace representative $\mathbf{Y}$ for the flag median $[\mathbf{Y}] \in \text{Gr}(r, n)$
while *not converged* **do**
  assign each w_i;
  $\mathbf{X} \leftarrow \left[w_1\mathbf{X}_1 | w_2\mathbf{X}_2 | \cdots | w_p\mathbf{X}_p\right]$;
  $\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{X}$ %calculate the SVD ;
  $\mathbf{Y} \leftarrow \mathbf{U}[:, 1:r]$ %first r columns of $\mathbf{U}$;
end

An important note is that FlagIRLS is an iterative weighted flag mean algorithm so the outputs of this algorithm come from the left singular vectors of $\mathbf{X}$. We take r singular vectors associated with the r largest singular values of $\mathbf{X}$, i.e., the first r columns of $\mathbf{U}$. However, there are n columns of $\mathbf{U}$, so FlagIRLS actually outputs a flag of subspaces $[\mathbf{U}[:,1]] \subset [\mathbf{U}[:,:2]] \subset \cdots \subset [\mathbf{U}[:,:n]]$. This flag is used to distinguish between different prototypes in Section 5.2 with MNIST digits.

4 Limitations

The main limitation of calculating the flag median is the speed of FlagIRLS. This requires that we take the thin SVD of $\mathbf{X} \in \mathbb{R}^{n\times pk}$ every iteration in FlagIRLS. The complexity of the FlagIRLS algorithm is the complexity of the flag

mean times the number of iterations of the algorithm, i.e., $O\left(nN_\delta\left(\sum_{i=1}^{p} k_i\right)^2\right)$ where N_δ is the number of iterations and δ is the convergence parameter.

Another current limitation of this work is the lack of proven mathematical guarantees for the flag median and the FlagIRLS algorithm. Although, for all our examples, FlagIRLS converges to a local minimum of the flag median problem, we have not worked out the mathematical theory to find the conditions where FlagIRLS converges. We also still need to determine the conditions where an iteration of FlagIRLS is a contraction mapping. On a larger scale, given a dataset of subspaces of $\mathbb{R}^n$, we have yet to determine where the flag median problem is convex. Section 3.1 shows that FlagIRLS is a logical algorithm for finding the flag median, but further development of the mathematical theory would give us more intuition about which datasets are good for FlagIRLS, how to initialize FlagIRLS and overall provide the user with a better understand of rates of convergence. Currently, the FlagIRLS algorithm is run with a number of different initializations to verify convergence.

5 Experiments

In this section we carry out experiments with synthetic data, the MNIST handwritten digits dataset [17], the Mind's Eye dataset [9] and the UCF YouTube action dataset [18]. The goal is to compare the flag median to the flag mean and the ℓ_2-median and establish the efficiency of FlagIRLS. For all of this section we use FlagIRLS to compute the flag median and the Weiszfeld-type algorithm from [7] for the ℓ_2-median.

The convergence criteria for our implementation of FlagIRLS is as follows. We terminate the algorithm when objective function values of consecutive iterates of FlagIRLS are less than $\delta = 10^{-11}$, or if the ith iteration resulted in an increasing objective function value. In the former case, we output the $(i-1)$st iterate. For our weights in all examples we run the FlagIRLS algorithm with $\epsilon = 10^{-7}$.

The convergence criteria of our implementation of the Weiszfeld-type algorithm from [7] to calculate the ℓ_2-median is similar to the Flag IRLS convergence criteria. We terminate the algorithm when when objective function values of consecutive iterates are less than $\delta = 10^{-11}$.

Both FlagIRLS and the Weiszfeld-type algorithm are terminated when we have exceeded 1000 iterations. FlagIRLS never exceeds 1000 iterations in our examples.

5.1 Synthetic Data

We begin with two experiments on a dataset consisting of 10 points from $\mathrm{Gr}(3,20)$ and 10 points from $\mathrm{Gr}(5,20)$. A representative for a point on $\mathrm{Gr}(k,n)$ is sampled in two steps. The first step is to sample an $n \times k$ matrix from a uniform distribution on $[-.5,.5)$, $\mathscr{U}[-.5,.5)$. We then do the QR decomposition of this matrix to get a point on $\mathrm{Gr}(k,n)$. We perform two experiments on this dataset: The first experiment verifies convergence of FlagIRLS, and the second experiment compares the convergence rate of FlagIRLS to Grassmannian gradient descent.

For the first experiment, we run 100 trials of FlagIRLS with different random initializations. For each of these trials, we verify that we have converged by checking 100 points near the FlagIRLS algorithm output. Given one algorithm output, $[\mathbf{X}] \in \mathrm{Gr}(3,20)$, we sample the entries of $\mathbf{Y} \in \mathbb{R}^{20\times 3}$ from $\mathscr{U}[-0.5,0.5)$ and check the objective function value at the first 3 columns of $\mathbf{Q}$ where $\mathbf{Q}$ comes from the QR decomposition of the matrix $\mathbf{X} + 0.00001\mathbf{Y}$. We call these points "test points" for the algorithm output. We say the FlagIRLS algorithm for flag median converged when all the objective function values of the test points are less than or equal to the objective function value for the algorithm output. In this experiment, we find that 100% of the FlagIRLS trials converge.

We now show an example with the same dataset where we run FlagIRLS and Grassmannian gradient with 100 random intializations to compute the flag median. The results of this experiment are in Figure 1. For this example, Grassmannian gradient descent is implemented with a step size of 0.01. We find that FlagIRLS converges in fewer iterations than Grassmannian gradient descent for the flag median problem.

For our next example, we use a dataset of 200 points on $\mathrm{Gr}(6,100)$. The points are sampled by first fixing a "center" point for the dataset, $[\mathbf{X}_*]$. We do this by taking a random 100×6 matrix with entries from $\mathscr{U}[-.5,.5)$. We then take $\mathbf{X}_*$ as the first 6 columns of $\mathbf{Q}$ from the QR decomposition of this random matrix. The 200 points in the dataset are now calculated via the following steps. For each point, we generate $\mathbf{Z}$ by sampling a random 100×6 matrix with entries sampled from $\mathscr{U}[-.5,.5)$ and scaling it by 0.01. We then take the point determined by the first 6 columns of $\mathbf{Q}$ from the QR decomposition of $\mathbf{X}_* + \mathbf{Z}$.

We then run our FlagIRLS and Weiszfeld-type algorithm implementations with 20 random initializations to calculate the flag median and the ℓ_2-median respectively. For the random initializations, we initialize FlagIRLS and

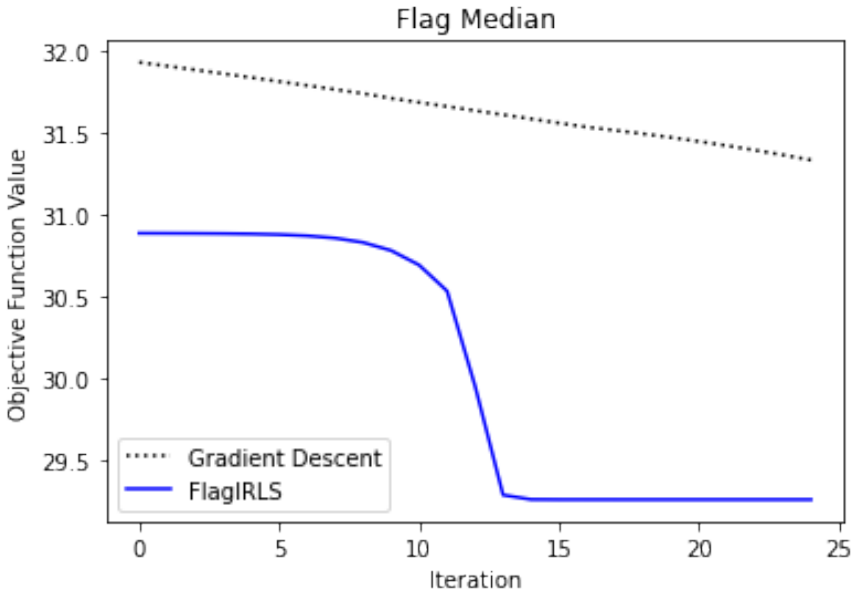

Figure 1: The mean objective function values over 100 trials with different random initializations. FlagIRLS converges in fewer iterations than gradient descent for the flag median problem for the synthetic dataset.

Algorithm/ Initialization	Mean Iterations
FlagIRLS/random	4.55 ± 0.50
Weiszfeld-Type/ datapoint	795.50 ± 227.40
Weiszfeld-Type/ randomly	968.80 ± 94.49

Table 1: The mean number of iterations until convergence of 20 random initalizations of FlagIRLS and the Weiszfeld-type algorithm on a dataset of 200 points on $\mathrm{Gr}(6,100)$. FlagIRLS is converges in far fewer iterations than the Weiszfeld-type algorithm and also sports a much lower standard deviation in the number of iterations.

the Weiszfeld-type algorithm at the same point. The results of this experiment are in Table 1. We terminate the Weiszfeld-type algorithm after 1000 iterations regardless of convergence. So perhaps, many of the high iteration runs of Weiszfeld still did not converge even after 1000 iterations.

Now we will use a dataset that consists of 200 points on $\mathrm{Gr}(3,20)$. These points consist of a cluster of 180 points centered around the subspace $[\mathbf{X}_*]$ and 20 outlier points. The points from the 180-point cluster are sampled by the following process. We calculate a fixed "center" point for the dataset, $[\mathbf{X}_*]$, by taking a 20×3 matrix with entries from $\mathscr{U}[-.5,.5)$. We then take $\mathbf{X}_*$ as the first 3 columns of $\mathbf{Q}$ from the QR decomposition of this random matrix. We then calculate the points in the cluster via the following steps. The first step is to generate $\mathbf{Z}$ by sampling a random 20×3 matrix with entries sampled from $\mathscr{U}[-.5,.5)$ and scaling it by 0.01. The second step generates one point in the 180 point cluster as the first 3 columns of $\mathbf{Q}$ from the QR decomposition of $\mathbf{X}_* + \mathbf{Z}$. A point from the set of outlier 20 points is the first 3 columns of the QR decomposition of a random 20×3 matrix with entries sampled from $\mathscr{U}[-.5,.5)$.

Table 2 shows the results of calculating the flag median, ℓ_2-median and flag mean of this dataset and then computing the chordal distance between $[\mathbf{X}_*]$ and the three different prototypes. Notice the flag median is the least affected by the outliers, the ℓ_2-median is twice as affected and the flag mean is ten times more affected by the outliers.

Algorithm	Chordal Distance
Flag Median	0.0017
ℓ_2-median	0.0022
Flag Mean	0.0128

Table 2: The chordal distance between the algorithm result and $[\mathbf{X}^*]$.

Note: for Table 2, FlagIRLS converges to the flag median in one iteration.

5.2 MNIST Handwritten Digits dataset

The MNIST digits dataset is a set of 28×28 single band images of handwritten digits [17]. We represent an MNIST handwritten digit using an element of $\mathrm{Gr}(1,784)$ to by taking one image, vectorizing it, then dividing the resulting vector by its norm.

For our first example, we see how the flag median, ℓ_2-median and flag mean prototypes are classified by a MNIST-trained 3-layer neural network. This trained neural network classifier has a 97% test accuracy on the MNIST test dataset. We generate our datasets for this experiment by to taking 20 examples of the digit 1 and i examples of the digit 9 from the MNIST training dataset. We let $i = 0,1,2,3,\ldots,19$ and this results in 20 datasets. For each of these datasets, we calculate the flag median, ℓ_2-median and flag mean, then predict the class of each of these prototypes by passing each through the neural network classifier. In Figure 2 we plot the predicted class of each prototype for each dataset by the trained neural network. For this figure, we choose to use the random initialization that resulted in the best predictions of the ℓ_2-median.

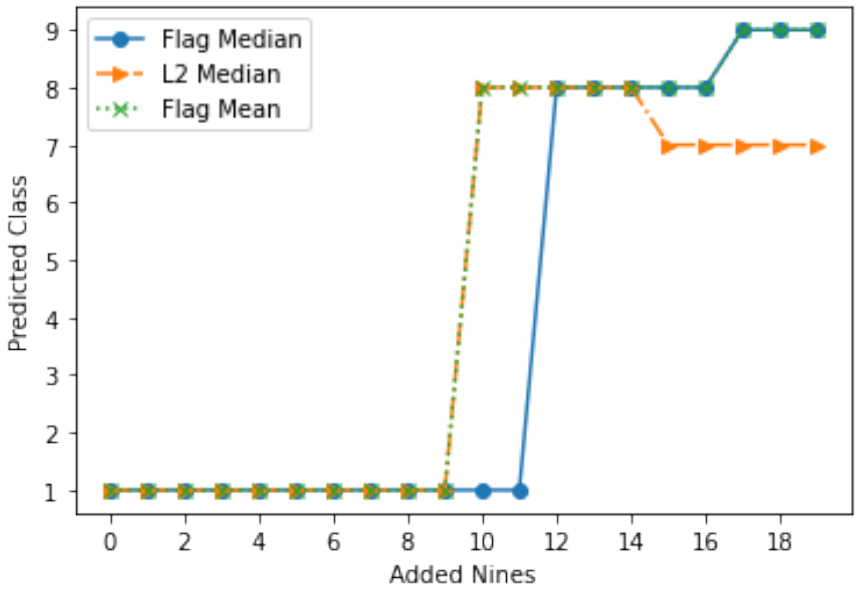

Figure 2: The neural network predicted class of the prototype for the dataset with 20 examples of 1's and i examples of 9's with $i = 0,1,2,\ldots,19$.

The ℓ_2-median and flag mean are misclassified with $i = 9$ added examples of 9's whereas the flag median is still classified correctly for $i = 10$ and $i = 11$ added examples 9's. Therefore the flag median is the most robust prototype to outliers in this experiment. The common misclassification as 8 is likely due to the fact that the 1's tend to be at an angle, so when averaged, they tend to look like fuzzy 8's, especially when some 9's have been introduced to the dataset. Also, the ℓ_2-median of a dataset of 20 1's with 15 to 19 9 digits is misclassified as a 7. This is likely a result of the different angled 1's and the introduction of the examples of 9's adding the top of the digit 7.

Now we use Multi Dimensional Scaling (MDS) [23] to visualize the movement of the prototypes of an MNIST dataset that is poisoned with outliers. For this experiment, we use 20 examples of 7's and $i = 0,2,4,6,8$ examples of 6's. This results in 5 different subspace datsets formed from examples from the MNIST training dataset. We then calculate the flag median, ℓ_2-median and the flag mean. We generate a distance matrix for all the examples of 6's and 7's along with the exemplars from each dataset using the geodesic distance and pass the distance matrix through a MDS algorithm to visualize relationships between these subspaces in two dimensions. This example is in Figure 3.

Notice that the flag mean is moving the most as we add examples of 6's and the ℓ_2-median is moving similarly to the flag mean. The flag median moves substantially less than the other prototypes and therefore is the least affected prototype by the added examples of 6's.

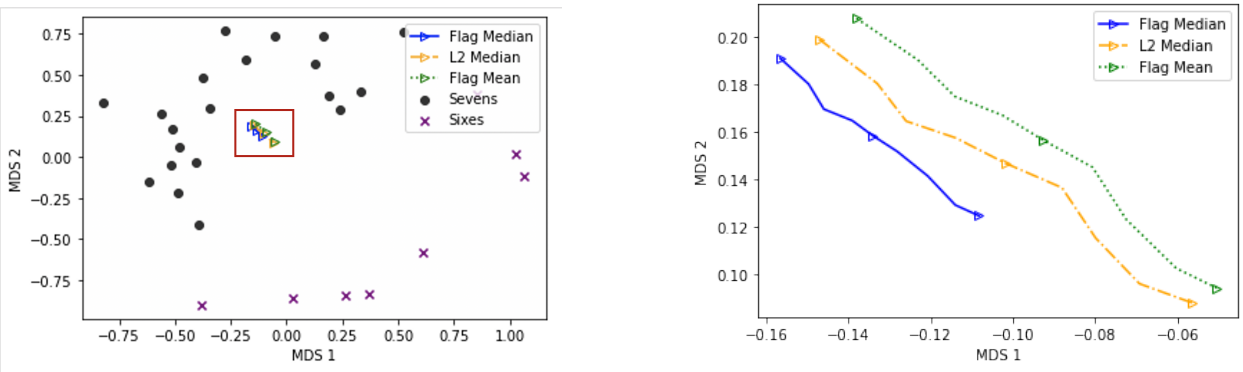

Figure 3: MDS embedding of flag median, ℓ_2-median and flag mean with points as one dimensional subspaces. We have 20 examples of 7's and i examples of 6's. Each triangle represents a prototype for $i = 0, 4, 8$. The furthest left triangle is the prototype for the dataset with $i = 0$ examples of 6's and the furthest right triangle is the prototype for the dataset with $i = 8$ examples of 6's. The lower image is a zoomed in version of the interior of the red box in the upper image to clarify the difference between the exemplars.

We now compute the $r = 5$-dimensional flag median and flag mean of a dataset with 20 examples of 7's with $i = 8$ 6's. We plot each of the reshaped columns of the matrix representative of these prototypes in Figure 4. Notice that the flag mean is more affected by examples of 6's than the flag median. This is particularly noticeable in the final column (dimension 5) where there is a clear 6 in the image for flag mean whereas the 6 is not clear in the flag median.

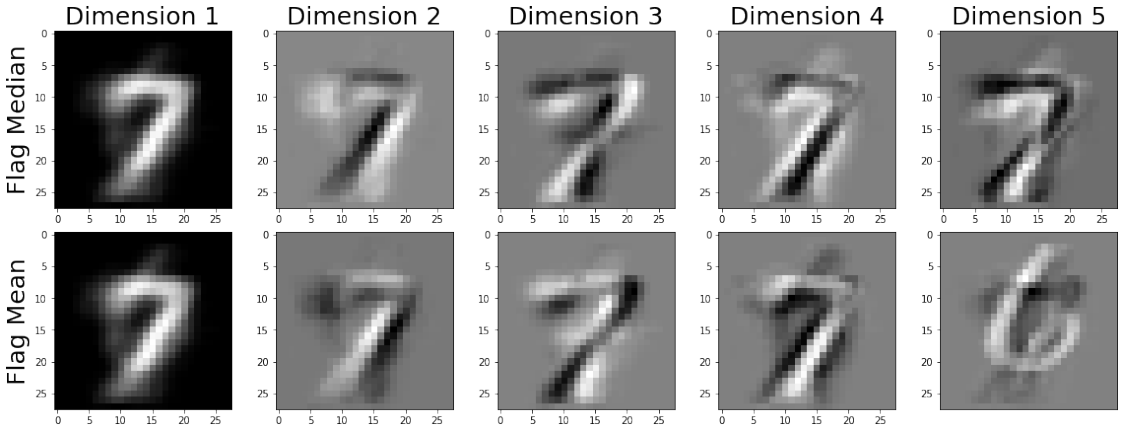

Figure 4: Each column of the matrix representative for flag median and flag mean on the dataset with 20 examples of 7's and $i = 8$ examples of 6's.

5.3 Mind's Eye dataset

The Mind's Eye dataset is a set of grey-scale outdoor video clips that are centered on moving objects (mainly humans) and have a subtracted background. Each video clip consists of 48 frames, each rescaled to a size of 32×32 pixels. We use the preprocessed data from the k-means experiment from Marrinan et. al. [9]. These data and the scripts for the preprocessing can be accessed at https://www.cs.colostate.edu/~vision/summet. There are 77 labels of the video clips for the action of the centered object in the video. A video clip is represented on $\mathrm{Gr}(48, 1024)$ by the span of the 1024×48 matrix formed by vectorizing and horizontally stacking each frame.

For this example, we use subspaces (points in $\mathrm{Gr}(48, 1024)$) that represent clips with action labels bend, follow, pickup, ride-bike and run. There are 27 examples of bend, 32 of follow, 27 of pickup, 17 of ride-bike and 24 of run. We run the Linde-Buzo-Grey (LBG) algorithm [19, 24] to cluster these data with different sized codebooks (numbers of centers) and prototype calculation using the flag median, ℓ_2-median and flag mean. In the LBG

algorithm, we calculate distance using chordal distance. For each number of centers, we run 10 trials with different LBG initializations. The results are in Figure 5.

We note that the flag median produces the highest cluster purities for 8, 12, 16 and 20 clusters. In all of the previous experiments we found that the flag median is more robust to outliers which may be the key factor in the success of the flag median prototype LBG implementation. We also note that the ℓ_2-median and the flag mean LBG implementations have similar cluster purities for each of the codebook sizes. Again, this is consistent with the similar behavior the ℓ_2-median and the flag mean MNIST experiments (see Figures 2 and 3).

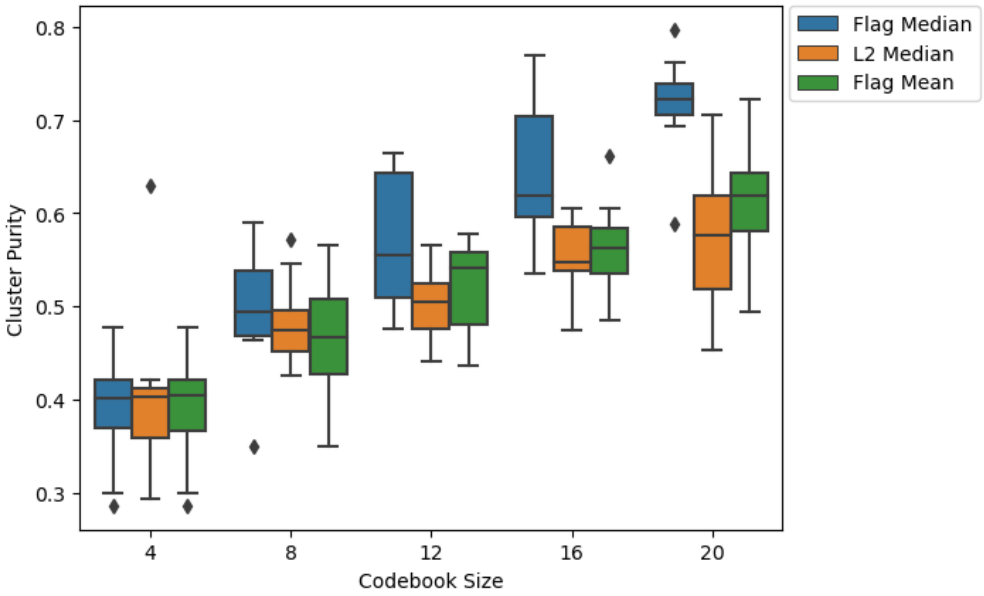

Figure 5: An LBG implementation on the Mind's Eye dataset. The results of 3 different implementations of LBG for codebook sizes $4, 8, 12, 16$ and 20. The flag median is competitive with the ℓ_2-median and flag mean for a size 4 codebook and outperforms ℓ_2-median and flag mean for codebook sizes $8, 12, 16, 20$.

5.4 UCF YouTube dataset

Our final dataset is a subset UCF YouTube Action dataset [18]. This dataset contains 11 categories of actions. For each category, the videos are grouped into groups with common features. For this expeeriment, we take approximately one example from each group within an action category. Specifically our dataset consists of 23 examples of basketball shooting, 22 of biking/cycling, 25 of diving, 24 of golf swinging, 24 of horse back riding, 24 of soccer juggling, 23 of swinging, 24 of tennis swinging, 24 of trampoline jumping, 22 of volleyball spiking, and 24 of walking with a dog. Since these RGB videos are quite large, we convert them to greyscale. Then we generate a matrix for each video whose columns are vectorizations of each frame. Finally, we perform the QR decomposition of each video and take the first 10 columns of $\mathbf{Q}$ to be it's representative on the Grassmannian.

We then run subspace LBG with 48 dimensional flag mean and the flag median. The results are in Figure 6. We choose to omit the ℓ_2-median LBG implementation since the Weiszfeld-type algorithm since it can only compute a 10 dimensional prototype. We run our LBG implementations with 10 trials for each of the following codebook sizes: $4, 8, 12, 16$ and 20. We see the flag median LBG implementation out preform the flag mean LBG implementation in all trials.

6 Conclusion

In this paper we presented a new prototype, the flag median, for clusters of points on the Grassmannian. We propose the FlagIRLS algorithm to approximate solutions to the flag median optimization problem. We run experiments comparing the flag median, flag mean, and the ℓ_2-median. In our experiments, we find the FlagIRLS generally converges faster than gradient descent. In addition, we discover that the flag median is the most robust to outliers and produces higher cluster purities than the flag mean and ℓ_2-median algorithms.

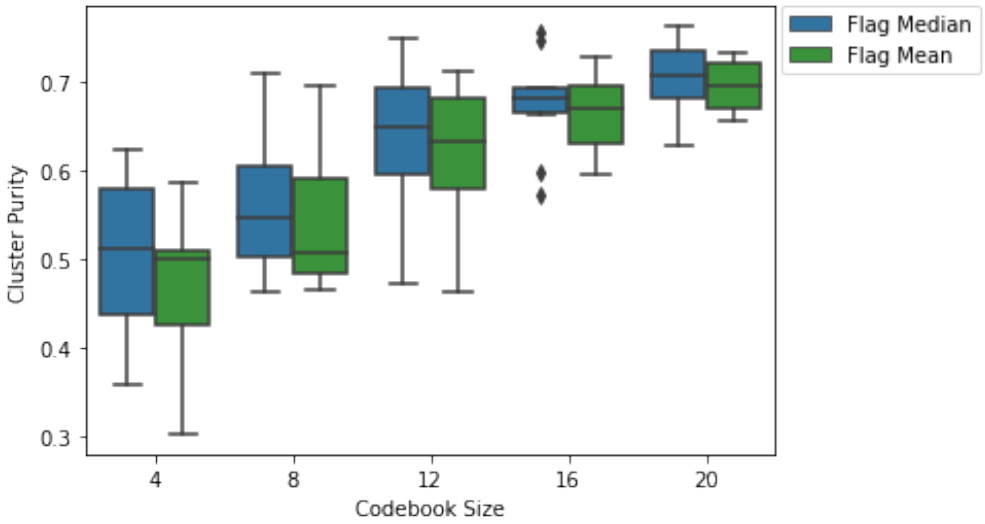

Figure 6: An LBG implementation on the YouTube dataset. The results of 2 different implementations of LBG for codebook sizes 4, 8, 12, 16 and 20. The flag median outperforms flag mean for all codebook sizes.

Future work with the flag median and FlagIRLS could involve machine learning or add details to the mathematical theory. For machine learning, the flag median can be used as a step in a subspace k-means algorithm, Grassmannian n-shot learning or any other machine learning algorithm in which calculating an "average" is a step. Most likely these types of algorithms will be useful for classifying images and videos. In terms of mathematics, we would like to find domain on which the flag median problem is convex and proofs for the convergence rates of FlagIRLS is an open problem. There are potential connections between this flavor of optimization problem and frame theory; so further investigation in this direction could prove useful. Finally, the flag median could be generalized to other spaces such as Stiefel manifolds.

Acknowledgement: This work was partially supported by National Science Foundation award NSF-ATD 1830676.

References

[1] Endre Weiszfeld. Sur le point pour lequel la somme des sistances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937. 1

[2] J Ross Beveridge, Bruce A Draper, Jen-Mei Chang, Michael Kirby, Holger Kley, and Chris Peterson. Principal angles separate subject illumination spaces in YDB and CMU-PIE. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):351–363, 2008. 1

[3] Sofya Chepushtanova and Michael Kirby. Sparse Grassmannian embeddings for hyperspectral data representation and classification. *IEEE Geoscience and remote sensing letters*, 14(3):434–438, 2017. 2

[4] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977. 2, 3

[5] Bruce Draper, Michael Kirby, Justin Marks, Tim Marrinan, and Chris Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014. 2, 3, 4

[6] Khurrum Aftab, Richard Hartley, and Jochen Trumpf. Generalized Weiszfeld algorithms for Lq optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):728–745, 2014. 2

[7] P Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on Riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1):S143–S152, 2009. 2, 3, 6

[8] Jiayao Zhang, Guangxu Zhu, Robert W Heath Jr, and Kaibin Huang. Grassmannian learning: Embedding geometry awareness in shallow and deep learning. *arXiv preprint arXiv:1808.02229*, 2018. 2

[9] Tim Marrinan, J Ross Beveridge, Bruce Draper, Michael Kirby, and Chris Peterson. Finding the subspace mean or median to fit your need. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1082–1089, 2014. 2, 3, 6, 9

[10] Yuchen Xie, Baba C Vemuri, and Jeffrey Ho. Dictionary learning on Riemannian manifolds. In *MICCAI workshop on STMI*, volume 1800, page 1800, 2012. 2

[11] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on Riemannian manifolds with gaussian RBF kernels. *IEEE PAMI*, 37(12):2464–2477, 2015. 2

[12] Anoop Cherian, Suvrit Sra, Stephen Gould, and Richard Hartley. Non-linear temporal subspace representations for activity recognition. In *Proceedings of the IEEE CVPR*, pages 2197–2206, 2018. 2

[13] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *NIPS*, 29:4592–4600, 2016. 2

[14] Daniel J Bates, Brent R Davis, Michael Kirby, Justin Marks, and Chris Peterson. The max-length-vector line of best fit to a set of vector subspaces and an optimization problem over a set of hyperellipsoids. *Numerical Linear Algebra with Applications*, 22(3):453–464, 2015. 2

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

[16] Justin D Marks. *Mean Variants on Matrix Manifolds*. PhD thesis, Colorado State University, 2012. 2, 4

[17] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 2, 6, 8

[18] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003. IEEE, 2009. 2, 6, 10

[19] Yoseph Linde, Andres Buzo, and Robert Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980. 2, 9

[20] Ake Bjorck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973. 2, 4

[21] John H Conway, Ronald H Hardin, and Neil JA Sloane. Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Mathematics*, 5(2):139–159, 1996. 2

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

[23] Joseph B Kruskal. *Multidimensional scaling*. Number 11. Sage, 1978. 8

[24] Shannon Stiverson, Michael Kirby, and Chris Peterson. Subspace quantization on the grassmannian. In *International Workshop on Self-Organizing Maps*, pages 251–260. Springer, 2019. 9