



Self-organizing mappings on the flag manifold with applications to hyper-spectral image data analysis

Xiaofeng Ma¹ · Michael Kirby¹ · Chris Peterson¹

Received: 15 May 2020 / Accepted: 11 December 2020
© The Author(s) 2021

Abstract

A flag is a nested sequence of vector spaces. The type of the flag encodes the sequence of dimensions of the vector spaces making up the flag. A flag manifold is a manifold whose points parameterize all flags of a fixed type in a fixed vector space. This paper provides the mathematical framework necessary for implementing self-organizing mappings on flag manifolds. Flags arise implicitly in many data analysis contexts including wavelet, Fourier, and singular value decompositions. The proposed geometric framework in this paper enables the computation of distances between flags, the computation of geodesics between flags, and the ability to move one flag a prescribed distance in the direction of another flag. Using these operations as building blocks, we implement the SOM algorithm on a flag manifold. The basic algorithm is applied to the problem of parameterizing a set of flags of a fixed type.

Keywords Self-organizing mappings · SOM · Flag manifolds · Geodesic · Visualization

1 Introduction

Self-Organizing Mappings (SOMs) were introduced as a means to *see* data in high-dimensions [8–11]. This competitive learning algorithm effectively transports the notion of proximity in the data space to proximity in the index space (which may in turn be endowed with its own geometry). As a tool, SOMs have been widely applied and extended [5]. The goal of the SOM algorithm is to produce a topology preserving mapping from a high-dimensional space to a low-dimensional space in the sense that points that are neighbors in the high-dimensional space are also represented as neighbors in the low-dimensional index space.

The geometric framework of the vanilla version of the SOM algorithm is Euclidean space. In this setting, the

distance between points is simply the standard 2-norm of the vector difference. The movement of a center toward a pattern takes place on a line segment in the ambient space. The only additional ingredient to the algorithm is a metric on the index space. Some additional treatments are needed when data are living on a high-dimensional manifold rather than Euclidean space. In [20], the author proposed a modification of the Self-organizing map algorithm to learn the manifold structure in the high-dimensional observation coordinates. Motivated by the subspace approach to data analytics we proposed a version of SOM using the geometric framework of the Grassmannian [3, 17–19]. This subspace approach has proven to be effective in settings where you have a collection of subspaces built up from a set of patterns drawn from a given family. Given that one can compute distances between points on a Grassmann manifold and that one can move one point in the direction of another, it is possible to transport the SOM algorithm on Euclidean space to an SOM algorithm on a Grassmannian [7, 14].

An interesting structure that generalizes Grassmannians and encodes additional geometry in data is known as the *flag manifold*. The points of a flag manifold parameterize the flags of a given type. Thus, a single point on a flag manifold corresponds to a sequence of nested subspaces.

✉ Michael Kirby
Michael.Kirby@Colostate.Edu

Xiaofeng Ma
Ma@math.colostate.edu

Chris Peterson
peterson@math.colostate.edu

¹ Department of Mathematics, Colorado State University,
Fort Collins, Colorado, USA

As an example, the wavelet transform applied to a data vector produces a sequence of approximations that live in nested scaling subspaces [6]. The nested sequence of scaling subspaces is a flag and corresponds to a single point on an appropriate flag manifold. As a second example, consider an ordered basis, $v_1, v_2, \dots, v_k$ for a set of data produced, for instance, as the output of a principal component analysis computation. The ordered basis induces the flag $S_1 \subset S_2 \subset \dots \subset S_k$ where S_i is the span of $v_1, \dots, v_i$. Again, this nested sequence of vector spaces is a flag thus corresponds to a point on a flag manifold. In this paper we extend SOM to perform a topology preserving mapping on points that correspond to nested subspaces such as those arising, for instance, from ordered bases or wavelet scaling spaces. To accomplish this, we show how to compute the distance between two points on a flag manifold and demonstrate how to move a flag a prescribed distance in the direction of another. Given these building blocks, we illustrate how one may extend SOM to the geometric framework of a flag manifold.

This paper provides an extension to [13]. The outline of this paper is as follows: In Sect. 2, we provide a formal definition of the flag manifold and illustrate with concrete examples. In Sect. 3, we introduce the numerical representation of flag manifolds. Here we indicate explicitly how distances can be computed between flags, and further, how a flag can be moved in the direction of another flag. In Sect. 4 we put the pieces together to realize the SOM algorithm on flag manifolds. We demonstrate the algorithm with a preliminary computational example. Section 5 consists of a numerical example utilizing the algorithm. Finally, in Sect. 6 we summarize the results of the paper and point toward future directions of research.

2 Introduction to flag manifold with data analysis examples

In this section, we introduce the basics of the flag manifold, fix some terminology and notation, and provide examples of its appearance in the context of data analysis.

A *flag* of subspaces in $\mathbb{R}^n$ is a nested sequence of subspaces $\{0\} \subset V_1 \subset V_2 \subset \dots \subset V_d = \mathbb{R}^n$. The signature or type of the flag refers to the dimensions of the V_i . There are two standard ways to encode this dimension information. One way is as the sequence $(\dim V_1, \dim V_2, \dots, \dim V_d)$. A second way to encode this dimension information is as the sequence $(\dim V_1, \dim V_2 - \dim V_1, \dim V_3 - \dim V_2, \dots, \dim V_d - \dim V_{d-1})$. In this paper, we will use this second encoding for the type of flag. We let $FL(n_1, n_2, \dots, n_d)$ denote the *flag manifold* whose points parameterize all flags of type $(n_1, n_2, \dots, n_d)$. Thus, a point on this flag manifold corresponds to a nested sequence of

subspaces $\{0\} \subset V_1 \subset V_2 \subset \dots \subset V_d = \mathbb{R}^n$ with the dimension of V_i equal to $n_1 + \dots + n_i$. As a special case, a flag of type $(1, 1, \dots, 1)$ is referred to as a full flag and $FL(1, 1, \dots, 1)$ denotes the manifold whose points parameterize all full flags in $\mathbb{R}^n$. Figure 1 illustrates the nested structure of the first three low-dimensional elements comprising a full flag in $\mathbb{R}^n$.

A flag of type $(k, n-k)$ is simply a k -dimensional subspace of $\mathbb{R}^n$ (which can be considered as a point on the Grassmann manifold $Gr(k, n)$). Hence $FL(k, n-k) = Gr(k, n)$. The Grassmannian-SOM algorithm is developed in [7, 14]. The idea that the flag manifold is a generalization of the Grassmann manifold will be utilized later to introduce the geodesic formula on the flag manifold. The nested structure inherent in a flag shows up naturally in the context of data analysis.

1. Wavelet analysis: Wavelet analysis and its associated multiresolution representation produces a nested sequence of vector spaces that approximate data with increasing resolution [2, 15, 16]. Each *scaling* subspace V_j is a dilation of its adjacent neighbor V_{j+1} in the sense that if $f(x) \in V_j$ then a reduced resolution copy $f(x/2) \in V_{j+1}$. The scaling subspaces are nested

$$\dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset \dots$$

and in the finite-dimensional setting can be considered as a point on a flag manifold. In Fig. 2, we visualize points on the geodesic between the flags associated with Daubechies2 (Haar) and Daubechies4 as they are applied to a particular image of size 32×32 . To be more specific, for each timestamp t ($0 \leq t \leq 1$), we have a flag corresponding to a point on the geodesic between Daubechies2 and Daubechies4. Using the flag corresponding to one of these time steps, we can transform an MNIST image (considered as a 32×32 matrix) by multiplying on both the left and the right by the projection matrices associated to each subspace in the flag. In this figure, each row is showing how this transformation affects the 32×32 MNIST image while morphing along this geodesic. Each column is a visualization of the nested scaling subspaces, i.e., a 4-dimensional scaling subspace living in an 8-dimensional scaling subspace living in a 16-dimensional

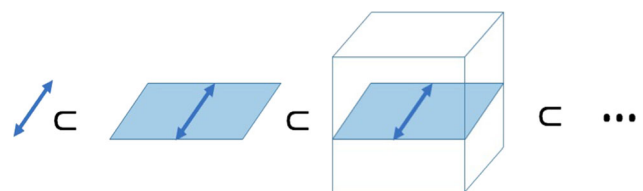
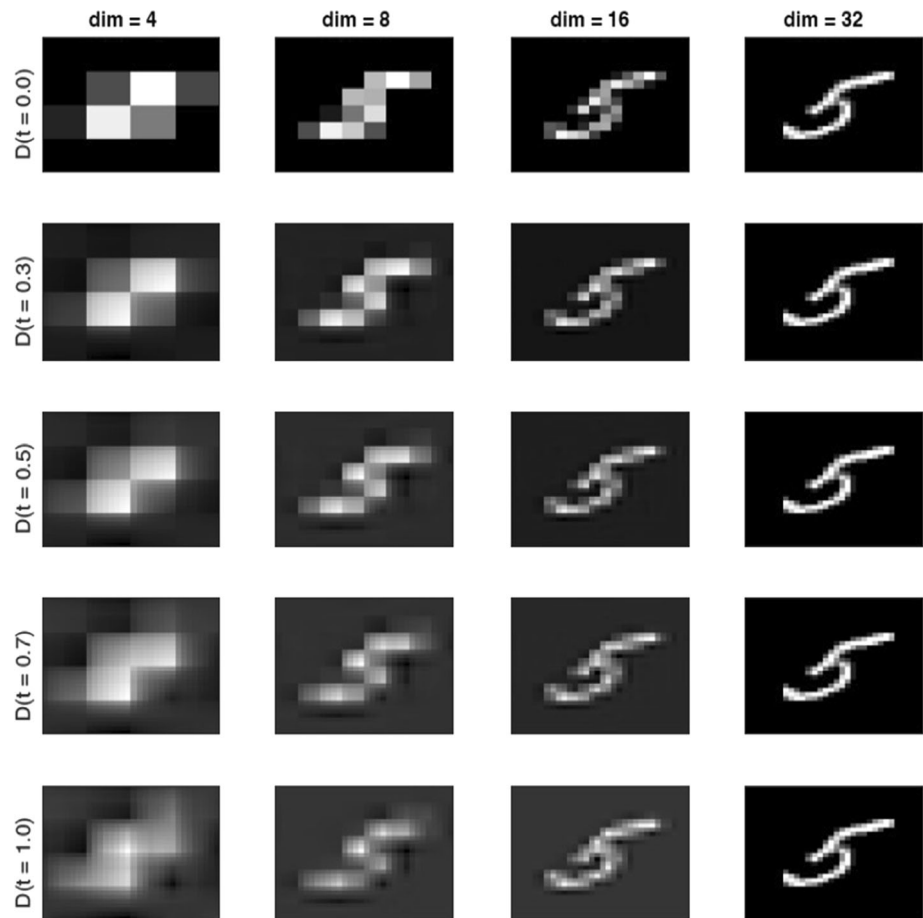


Fig. 1 Illustration of a nested sequence of subspaces corresponding to a point on the flag manifold $FL(1, 1, \dots, 1)$

Fig. 2 A visualization of the geodesic between the flags associated with Daubechies2 (Haar) and Daubechies4 transform matrix of size 32×32



scaling subspace living in a 32-dimensional ambient space. Note that the last column remains constant for all t since it recovers the original MNIST image.

2. SVD basis of a real data matrix: Let $X \in \mathbb{R}^{n \times k}$ be a real data matrix consisting of k samples in $\mathbb{R}^n$. Let $U\Sigma V^T = X$ be the thin SVD of X . The columns of the n -by- d orthonormal matrix U is an ordered basis for the column span of X . This basis is ordered by the magnitude of the singular values of X . This order provides a straightforward way to associate to U a point on a flag manifold. If $U = [u_1 | u_2 | \dots | u_d]$ then the nested subspaces $\text{span}([u_1]) \subsetneq \text{span}([u_1 | u_2]) \subsetneq \dots \subsetneq \text{span}([u_1 | \dots | u_d]) \subsetneq \mathbb{R}^n$ is a flag of type $(1, 1, \dots, 1, n-d)$ in $\mathbb{R}^n$. After we introduce the distance metric on the flag manifold in Sect. 3.2, one could consider computing the distance between two flags, perhaps derived from a thin SVD of two different data sets, which takes the order of the bases into consideration.

3 Numerical representation and geodesics

A point in the vector space $\mathbb{R}^n$ can be naturally represented by an $n \times 1$ vector. For a more abstract object like a Grassmann or flag manifold, we need a way to represent points in such a way that we can do computations. In this section, we describe how we can represent points and we describe how to determine and express geodesic paths between points. Note that in this paper we are using $\exp$ and $\log$ to denote the matrix exponential and the matrix logarithm.

3.1 Flag manifold

The flag manifold $FL(n_1, n_2, \dots, n_d)$ is a manifold whose points parameterize the set of all flags of type $(n_1, n_2, \dots, n_d)$. The presentation in [4] describes how to view the Grassmann manifold $Gr(k, n)$ as the quotient

manifold $O(n)/O(k) \times O(n-k)$ where $O(n)$ denotes the orthogonal group and $O(k) \times O(n-k)$ denotes the block diagonal matrix with elements from $O(k)$ in the first block and elements from $O(n-k)$ in the second block. If we let $SO(n)$ denote the special orthogonal group and $S(O(k) \times O(n-k))$ denote the subgroup of $O(k) \times O(n-k)$ consisting of matrices having determinant 1, then an equivalent description of $Gr(k, n)$ is as the quotient manifold $SO(n)/S(O(k) \times O(n-k))$. In the same way, $FL(n_1, n_2, \dots, n_d)$ is the quotient manifold $SO(n)/S(O(n_1) \times O(n_2) \times \dots \times O(n_d))$ where $n_1 + n_2 + \dots + n_d = n$. Let $P \in SO(n)$ be an n -by- n special orthogonal matrix, the equivalence class $[P]$, representing a point on the flag manifold, is the set of special orthogonal matrices

$$[P] = \left\{ P \begin{pmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & P_d \end{pmatrix} : P_i \in O(n_i), \prod_i \det(P_i) = 1 \right\}.$$

A manifold closely related to $FL(n_1, n_2, \dots, n_d)$ is the fully oriented flag manifold $FL_O(n_1, n_2, \dots, n_d) = SO(n)/SO(n_1) \times SO(n_2) \times \dots \times SO(n_d)$. There is a natural map $\phi : FL_O(n_1, n_2, \dots, n_d) \rightarrow FL(n_1, n_2, \dots, n_d)$. This map is surjective and is a 2^{d-1} cover of $FL(n_1, n_2, \dots, n_d)$. Thus, the inverse image of a point in the flag manifold is a collection of 2^{d-1} points in the fully oriented flag manifold.

It is well known that the geodesic paths on $SO(n)$ are given by exponential flows $P(t) = P \exp(t\mathbf{A})$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is any skew symmetric matrix and $P(0) = P$. The geodesics on $SO(n)$, i.e., $P(t) = P \exp(t\mathbf{A})$, continue to be geodesics on $FL(n_1, n_2, \dots, n_d)$ as long as they are perpendicular to the orbits generated by $S(O(n_1) \times O(n_2) \times \dots \times O(n_d))$, which requires further constraints on the tangent vector $\mathbf{A}$. $FL(n_1, n_2, \dots, n_d)$ is a quotient manifold of $SO(n)$. Let $[P] \in FL(n_1, n_2, \dots, n_d)$. The tangent space to $SO(n)$ at P , $T_P SO(n)$, can be decomposed into a vertical space V_P and a horizontal space H_P . The vertical space is the set of vectors in the tangent space corresponding to motions flowing along the equivalence class $[P]$ at P . The horizontal space is defined as the orthogonal (with respect to the Euclidean metric) complement of the vertical space in $T_P SO(n)$. The Euclidean metric is defined as a function $d : T_P SO(n) \times T_P SO(n) \rightarrow \mathbb{R}$:

$$d(U, V) = \text{Tr}(U^T V) \\ = \text{vec}(U)^T \text{vec}(V)$$

Intuitively, the vectors in the vertical space can be thought of as the set of velocity vectors which preserve the equivalence class, while the vectors in the horizontal space modify the equivalence class. Therefore, tangent vectors to geodesics need to be further constrained to the horizontal space. If V is a tangent vector to $FL(n_1, n_2, \dots, n_d)$, then there is a horizontal vector $\bar{V} \in H_P$ which represents V uniquely, which gives a numerical/matrix representation to the tangent vectors.

The vertical space at a point P is the set of matrices

$$V_P = \left\{ P \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & A_d \end{pmatrix} \right\},$$

where A_i is a n_i -by- n_i skew symmetric matrix. The horizontal space H_P is the set of matrices which are orthogonal to the vertical space and living in $T_P SO(n)$. Consider the following set of equations

$$\text{Tr} \left(\Delta^T P \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & A_d \end{pmatrix} \right) = 0 \\ \Delta = P\mathbf{A}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $A_i \in \mathbb{R}^{n_i \times n_i}$ are skew symmetric matrices. By solving the above system of equations, we can conclude that the horizontal space at P is the set of matrices

$$H_P = \left\{ P \begin{pmatrix} \mathbf{0}_{n_1} & & & * \\ & \mathbf{0}_{n_2} & & \\ & & \ddots & \\ -*^T & & & \mathbf{0}_{n_d} \end{pmatrix} \right\}$$

where $\mathbf{0}_{n_i}$ denotes an $n_i \times n_i$ matrix of zeros. This leads one to conclude that the geodesic paths on $FL(n_1, n_2, \dots, n_d)$ are exponential flows:

$$P(t) = P \exp(t\tilde{\mathbf{C}}) \quad (1)$$

where $\tilde{\mathbf{C}}$ is any skew symmetric matrix of the form

$$\tilde{C} = \begin{pmatrix} \mathbf{0}_{n_1} & & * \\ & \mathbf{0}_{n_2} & \\ & & \ddots \\ -*^T & & \mathbf{0}_{n_d} \end{pmatrix}, \mathbf{0}_{n_i} = \mathbf{0}^{n_i \times n_i}.$$

3.2 Skew symmetric matrix determines a geodesic between two points

By Eq. (1), one may trace out the geodesic path on a flag manifold emanating from P in the direction of $\tilde{C}$. In this section, we utilize Eq. (1) to solve the inverse problem:

Given two points $Q_1, Q_2 \in SO(n)$, whose equivalence classes $[Q_1], [Q_2]$ represent flags of type $(n_1, n_2, \dots, n_d)$, the goal is to obtain a factorization

$$Q_2 = Q_1 \cdot \exp(H) \cdot M \quad (2)$$

for H and M where H and M are constrained to be of the form

$$H = \begin{pmatrix} \mathbf{0}_{n_1} & & * \\ & \mathbf{0}_{n_2} & \\ & & \ddots \\ -*^T & & \mathbf{0}_{n_d} \end{pmatrix} \text{ and } M = \begin{pmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & M_d \end{pmatrix}$$

where H is skew symmetric, $M_i \in O(n_i)$, and $M \in SO(n)$. However, related to the covering map mentioned above, this factorization has multiple solutions. The expression $Q(t) = Q_1 \exp tH$, as t varies from 0 to 1, traces out a geodesic path between Q_1 and $Q_1 \exp(H)$. The length of the geodesic path between Q_1 and $Q_1 \exp(H)$ can be computed as a function of the eigenvalues of H which can be simplified to the expression

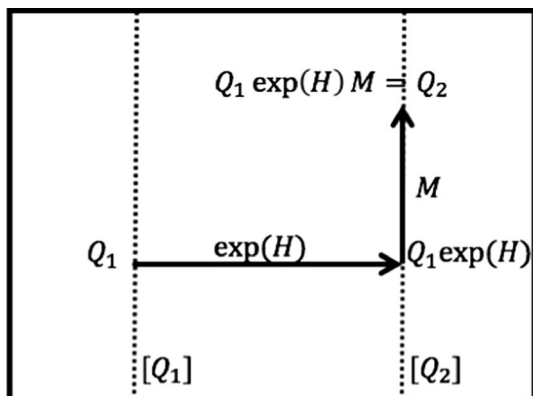


Fig. 3 Illustration of Eq. (2). The vertical lines represent the equivalence classes $[Q_1]$ and $[Q_2]$, respectively. Q_1 is mapped to an element in $[Q_2]$ by right multiplication with $\exp(H)$ which is then sent to Q_2 by multiplying with M

$$\text{Length of Path} = \sqrt{\frac{1}{2} \text{trace}(H^T H)}. \quad (3)$$

For additional details on this formula, see [4, 21]. If one starts with two special orthogonal matrices Q_1 and Q_2 , one can consider their equivalence classes $[Q_1], [Q_2]$ as two points on a flag manifold. In order to compute their distance apart on the flag manifold $FL(n_1, n_2, \dots, n_d)$, we consider the inverse image of $[Q_1]$ and $[Q_2]$ under the map ϕ . The inverse image of each determines 2^{d-1} points on $FL_O(n_1, n_2, \dots, n_d)$. The algorithms that we propose in the next section compute the shortest length of a geodesic between a point in the inverse image of $[Q_1]$ and a point in the inverse image of $[Q_2]$. The length of this shortest geodesic is the distance between $[Q_1]$ and $[Q_2]$ as points on $FL(n_1, n_2, \dots, n_d)$.

Equation (2) can be interpreted in the following way. First, we map Q_1 to a representative in $[Q_2]$ via the geodesic determined by the velocity matrix H . Second, we map this element in $[Q_2]$ to Q_2 via the matrix M . Figure 3 is a pictorial illustration of the idea behind Eq. (2).

For $FL(k, n-k)$, i.e., the Grassmannian $Gr(k, n)$, one can solve for H analytically. See [4] for details.

For the more general case of computing the length of the geodesic between $[Q_1]$ and $[Q_2]$ (as shown in Fig. 3), we will present an iterative algorithm to obtain a numerical approximation of H and M in Sect. 3.3. Before we proceed to the algorithm, let us further simplify Eq. (2) by letting $Q = Q_1^T Q_2$. This allows us to rewrite (1) as

$$Q = \exp(H) \cdot M \quad (4)$$

Here we define $\mathcal{W}$ as the vector space of all n -by- n skew symmetric matrices. Let $\mathbf{p} = (n_1, n_2, \dots, n_d)$. We define $\mathcal{W}_{\mathbf{p}}$ to be the set of all block diagonal skew symmetric matrices of type $\mathbf{p}$ and its orthogonal complement $\mathcal{W}_{\mathbf{p}}^{\perp}$ in $\mathcal{W}$, i.e.,

$$\mathcal{W}_{\mathbf{p}} = \left\{ G \in \mathcal{W} \mid G = \begin{pmatrix} G_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & G_d \end{pmatrix} \right\}, \quad (5)$$

$$\mathcal{W}_{\mathbf{p}}^{\perp} = \left\{ H \in \mathcal{W} \mid H = \begin{pmatrix} \mathbf{0}_{n_1} & & * \\ & \ddots & \\ -*^T & & \mathbf{0}_{n_d} \end{pmatrix} \right\}. \quad (6)$$

where by definition, $G_i \in \mathbb{R}^{n_i \times n_i}$ is skew symmetric for each i . Instead of solving Eq. (4) directly, we propose to solve the following alternative equation:

$$Q = \exp(H) \cdot \exp(G) \quad (7)$$

where $G \in \mathcal{W}_{\mathbf{p}}$ and $H \in \mathcal{W}_{\mathbf{p}}^{\perp}$. However, it is important to note that $\exp(G)$ will produce an element in $SO(n_1) \times SO(n_2) \times \cdots \times SO(n_d)$. As a consequence, in these computations we implicitly work on the fully oriented flag manifold $SO(n)/SO(n_1) \times SO(n_2) \times \cdots \times SO(n_d)$. The fact that there is the natural 2^{d-1} to 1 map from the fully oriented flag manifold to the flag manifold means we must compute values for H and G for many different representatives. As the output of the algorithm (Algorithm 2), we must pick the “optimal” H giving the shortest distance arising from this map.

3.3 Iterative alternating algorithm

The idea of the Iterative Alternating algorithm is straightforward. Given an initial guess $G^{[0]} \in \mathcal{W}_{\mathbf{p}}$, since Q and $G^{[0]}$ are known, we can solve for H numerically. Let $\hat{H} = \log(Q \cdot \exp(G^{[0]}))^T$. Since $\hat{H}$ is generally not of the desired form (i.e., $\hat{H} \notin \mathcal{W}_{\mathbf{p}}^{\perp}$), we project $\hat{H}$ onto $\mathcal{W}_{\mathbf{p}}^{\perp}$ to obtain an update for H . This projection has the effect of zeroing out entries in a certain pattern in $\hat{H}$. We let $H^{[1]} = \text{Proj}_{\mathcal{W}_{\mathbf{p}}^{\perp}}(\hat{H})$. Then we start updating G . Let $\hat{G} = \log(\exp(H^{[1]})^T Q)$ then project $\hat{G}$ onto $\mathcal{W}_{\mathbf{p}}$ to obtain an update for G . This projection has the effect of zeroing out entries in a pattern complementary to what we did to obtain an update for H . We let $G^{[1]} = \text{Proj}_{\mathcal{W}_{\mathbf{p}}}(\hat{G})$. Now iterate this process obtaining $H^{[2]}$ then $G^{[2]}$ and continue until the values stabilize. The pseudo code of our Iterative Alternating algorithm is presented in Algorithm 1 and Algorithm 2.

Algorithm 1: Iterative Alternating algorithm

```

Input Data:  $X_1, X_2 \in SO(n)$ ,  $\mathbf{p} = (n_1, n_2, \dots, n_d)$ 
Output Data:  $H^*, G^*$ 
Define:  $d(H) = \sqrt{\frac{1}{2} \text{Tr}(H^T H)}$ 

1 Function FlagLog( $X_1, X_2, \mathbf{p}$ ):
2    $Q = X_1^T X_2$ 
3    $\{Q_i\}_{i=1}^{2^{d-1}} = \text{generateQi}(Q, \mathbf{p})$ 
4   for  $Q$  in  $\{Q_i\}_{i=1}^{2^{d-1}}$  do
5     for  $i = 1, \dots, m$  do
6       Generate random  $G^{[0]}$ 
7        $k = 0$ 
8       while  $k \leq \text{itermax}$  and  $\text{err} < \epsilon$  do
9          $k = k + 1$ 
10         $H^{[k]} = P_H(\log(Q \exp(-G^{[k-1]})))$ 
11         $G^{[k]} = P_G(\log(\exp(-H^{[k]})Q))$ 
12         $\text{err} = \|Q - \exp(H) \exp(G)\|_F$ 
13      end
14      if  $d^* > d(H^{[k]})$  then
15         $d^*, H^*, G^* = d(H^{[k]}), H^{[k]}, G^{[k]}$ 
16      end
17    end
18    return  $H^*, G^*$ 
    
```

Algorithm 2: Fully-oriented flag representations

```

1 Function generateQi( $Q, \mathbf{p}$ ):
2   colHeader =  $[0, \text{cumsum}(n_1, n_2, \dots, n_{d-1})] + 1$ 
3    $m = \text{length}(\text{colHeader})$ 
4    $n = \text{floor}(d/2)$ 
5    $i = 1$ 
6    $Q_i = Q$ 
7   for  $j = 1 : n$  do
8      $C = \text{nchoosek}(\text{colHeader}, 2*j)$ 
9     for  $k = 1 : \text{size}(C, 1)$  do
10       $i = i + 1$ 
11       $Q_i = Q$ 
12       $Q_i(:, C(k,:)) = -Q_i(:, C(k,:))$ 
13    end
14  end
15  return  $\{Q_i\}_{i=1}^{2^{(d-1)}}$ 
    
```

We walk through two examples as an illustration of the numerical computation of the geodesic formula and distance between two points on a flag manifold. Here two types of flag manifolds are utilized to illustrate how the geometry of a Grassmann manifold differs from that of a related flag manifold.

Algorithm 3: Flag SOM algorithm

Input Data: $\{X^{(\mu)} \in SO(n)\}_{\mu=1}^P, \{C_i^{[0]}\}_{i \in \mathcal{I}}, M \in \mathbb{N}, \mathbf{p} = (n_1, n_2, \dots, n_d),$
 $\sigma_0 \in \mathbb{R}, \epsilon_0 \in \mathbb{R}$
Output Data: Updated SOM network
Define: $d(H) = \sqrt{\sum_{k=1}^l \lambda_k^2}$ where $\pm i\lambda_k$'s are eigenvalues of H .

```

1 Function flagSOM():
2    $Q = X_1^T X_2$ 
3    $\{Q_i\}_{i=1}^{2^{d-1}} = \text{generateQi}(Q, \mathbf{p})$ 
4   for  $m = 1, 2, \dots, M$  do
5      $\mu = \text{randi}(P)$ 
6     for  $i \in \mathcal{I}$  do
7        $H_i^m, G_i^m = \text{FlagLog}(X^{(\mu)}, C_i^m)$ 
8        $s = \text{svd}(H_i^m)$ 
9        $d_i^m = \sqrt{(s^T s)/2}$ 
10    end
11     $i_m^* = \arg \min_i \{d_i^m\}$ 
12    for  $i \in \mathcal{I}$  do
13       $\text{idxDist} = \|i - i_m^*\|_2$ 
14       $\epsilon_m = \epsilon_0(1 - m/M)$ 
15       $\sigma_m = \sigma_0(1 - m/M)$ 
16       $t_m = \epsilon_m \cdot \exp(-\text{idxDist}^2 / \sigma_m^2)$ 
17       $C_i^{m+1} = C_i^m \exp(t_m H_i^m) \exp(G_i^m)$ 
18    end
19  end
20  return  $\{C_i^M\}_{i \in \mathcal{I}}$ 
    
```

Let

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and let} \quad Y = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}$$

be two data matrices. Let $X = Q_1 R_1$ and $Y = Q_2 R_2$ be the full QR-decomposition of X and Y . Here we look at two different flag structures:

1. Flag manifold of type $\mathbf{p} = (2, 2)$: Let $Q = Q_1^T Q_2$. The initial G_0 (and any other G_i in the iterative procedure) should be of the form

$$G_i = \begin{pmatrix} 0 & g_1 & 0 & 0 \\ -g_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_2 \\ 0 & 0 & -g_2 & 0 \end{pmatrix}.$$

The output velocity matrix H (and any other H_i in the iterative procedure) should be of the form

$$H_i = \begin{pmatrix} 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \\ -h_1 & -h_3 & 0 & 0 \\ -h_2 & -h_4 & 0 & 0 \end{pmatrix}.$$

The unique singular values of output H are $\lambda_1 = 1.0172$ and $\lambda_2 = 0.5536$. The geodesic distance is therefore $d([Q_1], [Q_2]) = \sqrt{\lambda_1^2 + \lambda_2^2} = 1.1581$. One thing to note is that $FL(2, 2)$ is equivalent to $Gr(2, 4)$. It is easy to verify that λ_1, λ_2 are exactly the principal angles between X and Y .

2. Flag manifold of type $\mathbf{p} = (1, 1, 2)$: For this example, the G_i 's and H_i 's should be of the form

$$G_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_1 \\ 0 & 0 & -g_1 & 0 \end{pmatrix}$$

and

$$H_i = \begin{pmatrix} 0 & h_1 & h_2 & h_4 \\ -h_1 & 0 & h_3 & h_5 \\ -h_2 & -h_3 & 0 & 0 \\ -h_4 & -h_5 & 0 & 0 \end{pmatrix},$$

respectively. The unique singular values of output H are $\lambda_1 = 1.0469$, $\lambda_2 = 0.5404$ and the geodesic distance

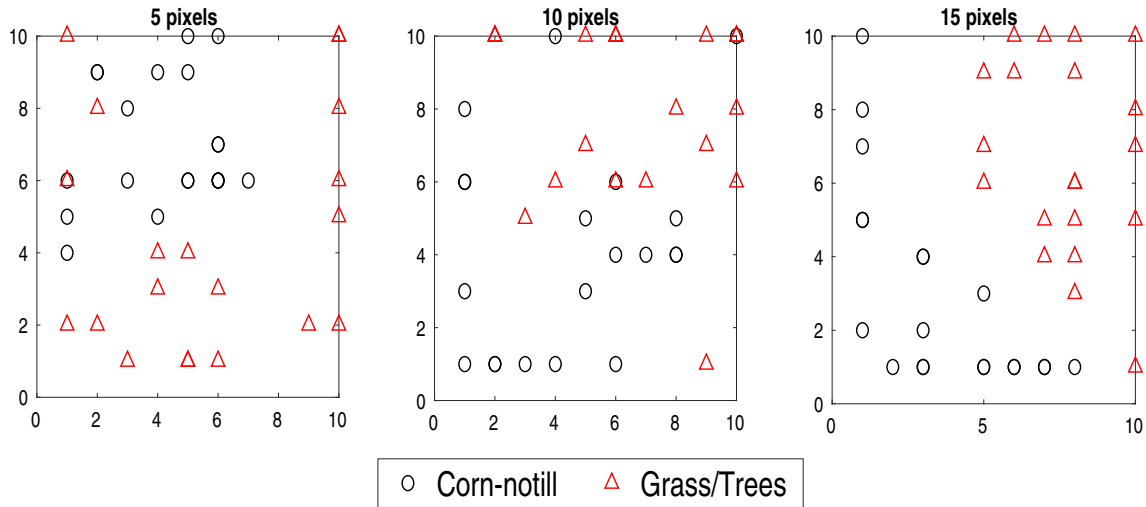


Fig. 4 Flag-SOM visualization results of Corn-notill and Grass/Trees. Left: 5 pixels used to form the SVD basis. Middle: 10 pixels are used to form the SVD basis. Right: 15 pixels are used to form the SVD basis

is therefore $d([Q_1], [Q_2]) = 1.1782$. The geodesic distance is larger than the previous example since we have imposed more structure in this example.

4 SOM on flag manifolds

In this section we extend the SOM algorithm to the setting of flag manifolds. The general setting of SOM starts with a set of training data $x^{(\mu)}$ with $\mu = 1, \dots, p$ and an initial set of randomized centers $\{C_i\}$ where the subscript i is associated to the label of the low-dimensional index a_i . The standard SOM center update equation is given by,

$$C_i^{m+1} = C_i^m + \epsilon_m h(d(a_i, a_{i^*}))(X - C_i^m).$$

The superscript m is indicating the m -th iteration in the SOM algorithm. Here i^* is the winning center of data point X , i.e.,

$$i^* = \arg \min \|X - C_i\|_2.$$

We also set the localization function as the standard

$$h(s) = e^{-s^2/\sigma^2}$$

and d is the metric which induces the geometry on the index set. Here we mainly focus on the simple one,

$$d(a_i, a_j) = \|a_i - a_j\|_2$$

where the indices are enumerated by subscript, i.e., the index set contains $a_1, a_2, \dots, a_N$. On the flag manifold, points are no longer living in a Euclidean space thus cannot be moved using the standard update equation. For a given data point X from a flag manifold of type

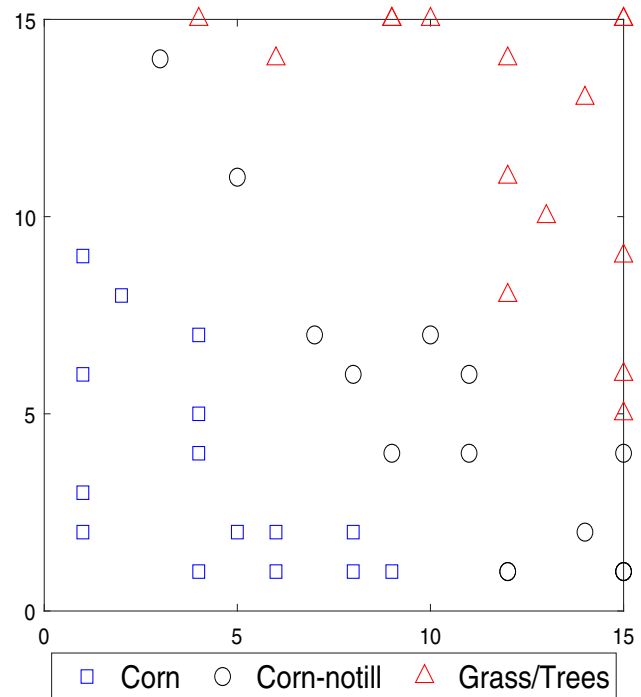


Fig. 5 Flag-SOM visualization results of Corn-notill, corn and Grass/Trees with only 5 bands selected (bands:100, 125, 149, 206, 207). We used 15 pixels to form the SVD basis

$\mathbf{p} = (n_1, n_2, \dots, n_d)$, we identify the winning center, from the set of all nested subspaces of type $\mathbf{p}$ which represent centers $\{C_i\}$, that is closest via

$$i^* = \arg \min_i d_g(X, C_i)$$

where d_g is defined in Eq. (3). To move the centers toward the nested subspace pattern X according to the SOM update

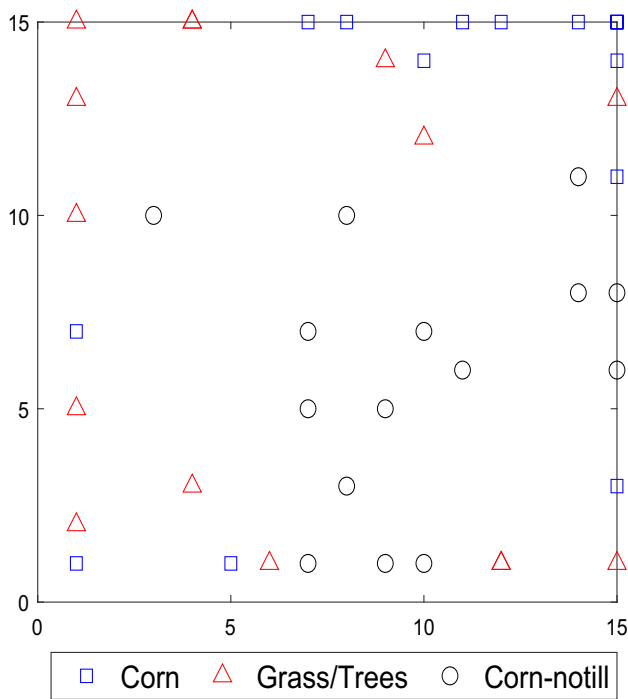


Fig. 6 Grassmann-SOM visualization results of Corn-notill, corn and Grass/Trees with only 5 bands selected (bands:100, 125, 149, 206, 207). We used 15 pixels to form the SVD basis

we compute the geodesic, using the Iterative Alternating algorithm described in Algorithm 1 and 2, between each center C_i and nested subspace pattern X .

Our localization term now becomes

$$t_n = \epsilon_n h_n(d(a_i, a_i^*)).$$

We now take

$$h_n(s) = \exp(-s^2/\sigma_n^2)$$

where $\epsilon_n = \epsilon_0(1 - n/T)$ and $\sigma_n = \sigma_0(1 - n/T)$. The centers thus change along the geodesic by moving from $C_i(0)$ to $C_i(t)$ where t is adjusted for the step size. The algorithm for SOM on a flag manifold is summarized in Algorithm 3.

5 Numerical experiment

In this section, to illustrate the proposed method for visualizing real world data, we implement it on the well known Indian Pine hyper-spectral image data set [12]. This data set was collected over an agricultural area in Northwestern Indiana in 1992. It consists of 145×145 pixels by 220 bands from $0.4\mu\text{m}$ to $2.4\mu\text{m}$. This data set has been previously studied within the context of band selection [1],

which will be utilized here to show the advantage of the flag manifold as a refined version of the Grassmann manifold. For the illustration of the algorithm, we consider a two-class problem and a three-class problem. We preprocessed data via mean centering, i.e., each pixel is subtracted by the mean value of the pixels for the whole scene (spectrum). In this application we selected only 5 bands (hence the ambient space is $\mathbb{R}^5$) to form 5×5 ordered orthogonal SVD basis matrices. Each SVD basis represents a data point within a specific class. The number of pixels (with the same class labels) required to form a robust SVD basis is also explored in this experiment.

For the two class problem we initialized the centers for flag-SOM by selecting 100 5×5 orthogonal matrices at random, corresponding to a 10×10 integer lattice. This was done by computing the singular value decomposition of matrices of size 5×5 from the uniform distribution. We also assemble 15 $5 \times k$ matrices Y_i from both classes, which results in constructing 30 data points U_i living on the flag manifold $Fl(1, 1, 3)$. Here U_i is the ordered set of left singular vectors of the corresponding data matrix Y_i , i.e., $U_i \Sigma_i V_i^T = \text{SVD}(Y_i)$. In Fig. 4, we observe that as we increase the number of pixels (k) used to form SVD bases, more robust and clear separation is achieved via flag-SOM between two classes, namely Corn-notill and Grass/Trees. When $k = 15$ pixels are used to construct SVD bases, there is a linear separation between two classes. With smaller values of k (e.g. $k = 5$ or $k = 10$), we observe a lack of linear separability.

In Fig. 5, we see the results of flag-SOM on the three class problem when the data points reside on $Fl(1, 1, 3)$. 225 centers (5×5 orthogonal matrices) are randomly generated as the centers of the 15×15 integer lattice. 15 SVD bases from each class is generated as described previously. We observe in Fig. 5 that with only 5 bands selected from 220 bands, we still obtain an excellent clustering with all three classes. Here we also measure the quality of the flag-SOM by computing the topographic error. First, we define two centers to be adjacent in their index space if their indices has distance 1 (Note that indices are defined on the integer grid). We obtain a topographical error of 0.22. If we relax the definition of adjacency by allowing the surrounding 8 nodes on the integer grid to be considered as adjacent, the topographical error becomes 0.04.

For a numerical comparison of SOM visualizations, we introduce a distortion error to measure the separation and compactness for the class distributions on the SOM grid, in our case, the integer grid. Let $a \in \mathbb{R}^2$ be the coordinates of the winning centers on the integer grid, which belong to

one of the k classes $\{S_i\}_{i=1}^k$. Let c_i be the mean coordinate of winning center for each class, i.e.,

$$c_i = \frac{1}{n_i} \sum_{a \in S_i} a.$$

The visualization distortion error is defined as

$$D = \sum_{i=1}^k \sum_{a \in S_i} \|a - c_i\|^2.$$

In Fig. 6, we demonstrate the Grassmannian-SOM on the same data set for the purpose of comparison. We observe that with low ambient dimension, the Grassmannian-SOM shows poor separation on the 2D grid with a distortion error of 2023. The flag-SOM visualization obtains well separated classes with a much lower distortion error of 981. Note that the Grassmannian-SOM suffers from the low ambient dimension while flag-SOM is still separating classes well thanks to the refined structure of the flag manifold.

6 Conclusions and future work

We have presented algorithms for Self-Organizing Mappings on flag manifolds. Techniques for computing the key ingredients of the SOM on flags are determining distances between flags and moving one flag a prescribed distance in the direction of another flag. The algorithm was tested on a sample problem that involves computing an ordering of points on a flag manifold. The flag-SOM algorithm has been demonstrated on hyper-spectral image data, in which case the algorithm organizes the hyper-spectral image data in the index space and separates 5×5 SVD bases when only 5 out of 220 bands are utilized.

Note that we have yet to explore the impact of the flag structure for the flag-SOM algorithm. Searching for an optimal flag structure has the potential to improve the visualization results.

Acknowledgements We thank the National Science Foundation for support CCF-BSF:CIF 1712788, and NSF-ATD 1830676.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless

indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Chepushtanova S, Gittins C, Kirby M (2014) Band selection in hyper-spectral imagery using sparse support vector machines, Algorithms and Technologies for Multispectral, Hyper-spectral, and Ultraspectral Imagery XX (Miguel Velez-Reyes and Fred A. Kruse, eds.), Proceedings of SPIE, vol 9088
2. Daubechies I (1992) Ten lectures on wavelets, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA
3. Draper Bruce, Kirby Michael, Marks Justin, Marrinan Tim, Peterson Chris (2014) A flag representation for finite collections of subspaces of mixed dimensions. Linear Algebra Appl 451:15–32
4. Edelman Alan, Arias TA, Smith ST (1998) The geometry of algorithms with orthogonality constraints. SIAM J Matrix Anal Appl 20(2):303–353
5. Kaski Samuel, Kangas Jari, Kohonen Teuvo (1998) Bibliography of self-organizing map (SOM) papers: 1981–1997. Neural Comput Surv 1(3&4):1–176
6. Kirby M (2001) Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns. Wiley, Hoboken
7. Kirby M, Peterson C (2017) Visualizing data sets on the Grassmannian using self-organizing mappings. In: 2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM), June pp 1–6
8. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol. Cybern. 43:59
9. Kohonen Teuvo (1990) The self-organizing map. Proc IEEE 78(9):1464–1480
10. Kohonen Teuvo (1998) The self-organizing map. Neurocomputing 21(1):1–6
11. Kohonen Teuvo (2013) Essentials of the self-organizing map. Neural Netw 37:52–65
12. Landgrebe D (1992) Aviris nw indiana's indian pines 1992 data set
13. Ma Xiaofeng, Kirby Michael, Peterson Chris (2019) Self-organizing mappings on the flag manifold. International workshop on self-organizing maps. Springer, Berlin, pp 13–22
14. Ma X, Kirby M, Peterson C, Scharf L (2019) Self-organizing mappings on the Grassmannian with applications to data analysis in high-dimensions. Neural Comput Appl 1–12
15. Mallat S (1989) Multiresolution approximations and wavelet orthonormal bases of L^2 . Trans. Am. Math. Soc. 315:69–87
16. Mallat S (1989) A theory of multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. 315:69–87
17. Marrinan Tim, Beveridge JR, Draper B, Kirby M, Peterson C (2013) Flag manifolds for the characterization of geometric structure in large data sets, vol 2015. Numerical Mathematics and Advanced Applications-ENUMATH. Springer, Berlin, pp 457–465
18. Marrinan T, Draper B, Beveridge JR, Kirby M, Peterson C (2014) Finding the subspace mean or median to fit your need. In:

- Computer vision and pattern recognition (CVPR), IEEE, pp 1082–1089
19. Marrinan T, Beveridge JR, Draper B, Kirby M, Peterson C (2016) Flag-based detection of weak gas signatures in long-wave infrared hyperspectral image sequences. In: Proceedings SPIE defense + security, international society for optics and photonics, pp 98401N–98401N
 20. Similä Timo (2005) Self-organizing map learning nonlinearly embedded manifolds. *Inf Vis* 4(1):22–31
 21. Ke Ye, Ken Sze-Wai Wong, Lek-Heng Lim (2019) Optimization on flag manifolds, Preprint: <https://arxiv.org/pdf/1907.00949.pdf>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.