Self-Organizing Mappings on the Flag Manifold

Xiaofeng Ma, Michael Kirby, and Chris Peterson

Colorado State University, Fort Collins CO 80523, USA
xiaofeng.ma@rams.colostate.edu,michael.kirby@colostate.edu,
christopher2.peterson@colostate.edu

Abstract. A flag is a nested sequence of vector spaces. The type of the flag is determined by 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 particular 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 techniques for instance in 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 [7–10]. This competitive learning algorithm effectively transports the notion of proximity in the data space to proximity in the index space; this may in turn be endowed with its own geometry. This tool has now been widely applied and extended [4]. The goal of the SOM algorithm is to produce a topology preserving mapping in the sense that points that are neighbors in 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 towards 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.

Motivated by the subspace approach to data analytics we proposed a version of SOM using the geometric framework of the Grassmannian [15, 2, 16, 14]. 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 one can compute distances between points on a Grassmannian, and 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 [6, 11].

An interesting structure that generalizes Grassmannians and encodes additional geometry in data is known as the *flag manifold*. Intuitively, a point on a flag manifold is a set of nested subspaces. So, for example, given a data vector, a wavelet transform produces a set of approximations that live in nested scaling subspaces [5]. The nested sequence of scaling subspaces is a flag and corresponds to a single point on an appropriate flag manifold. Alternatively, an ordered basis, $v_1, v_2, \dots, v_k$ for a set of data produced by principal component analysis induces the flag $S_1 \subset S_2 \subset \cdots \subset S_k$ where S_i is the span of $v_1, \dots, v_i$. 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 is outlined as follows: In Section 2 we provide a formal definition of the flag manifold and illustrate with concrete examples. In Section 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 Section 4 we put the pieces together to realize the SOM algorithm on flag manifolds. We demonstrate the algorithm with a preliminary computational example. Finally, in Section 5 we summarize the results of the paper and point towards future directions of research.

2 Introduction to flag manifold with data analysis examples

Let us first introduce the flag manifold. A $flag$ of type $(n_1, n_2, \dots, n_d; n)$ is a nested sequence of subspaces in $\mathbb{R}^n$ where $\{0\} \subsetneq V_1 \subsetneq V_2 \subsetneq \cdots \subsetneq V_d = \mathbb{R}^n$, $dim\, V_j = \Sigma_{i=1}^{j} n_i$ and $n_1 + n_2 + \cdots + n_d = n$. We let $FL(n_1, n_2, \dots, n_d; n)$ denote the *flag manifold* whose points parameterize all flags of type $(n_1, n_2, \dots, n_d; n)$. As a special case, the flag of type $(1, 1, \cdots, 1; n)$ is referred to as a full flag and $FL(1, 1, \cdots, 1; n)$ is the full flag manifold 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 $(k, n-k; n)$ 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 $FL(k, n-k; n) = Gr(k, n)$. The Grassmannian-SOM algorithm is developed in [6, 11]. 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.

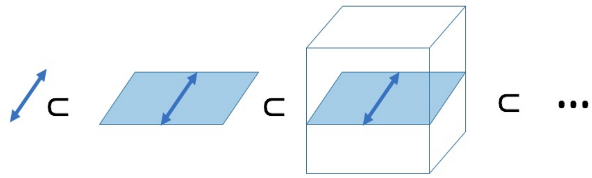

Fig. 1: Illustration of a nested sequence of subspaces corresponding to a point on the flag manifold $FL(1, 1, \cdots, 1; n)$.

1. Wavelet analysis: Wavelet analysis and its associated multiresolution representation produces a nested sequence of vector spaces that approximate data with increasing resolution [12, 13, 1]. 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

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

   and in the finite dimensional setting can be considered as a point on a flag manifold. The flag SOM algorithm provides a means to visualize relationships in a collection of discrete wavelet transforms and organize the corresponding sequences of nested subspaces in a coherent manner via a low-dimensional grid.
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 $U = [u_1|u_2|\ldots|u_d]$ then the nested subspaces $span([u_1]) \subsetneq span([u_1|u_2]) \subsetneq \cdots \subsetneq span([u_1|\cdots|u_d]) \subsetneq \mathbb{R}^n$ is a flag of type $(1, 1, \ldots, 1, n-d; n)$ in $\mathbb{R}^n$. After we introduce the distance metric on the flag manifold in Section 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 basis 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 $FL(n_1, n_2, \ldots, n_d; n)$ consists of the set of all flags of type $(n_1, n_2, \ldots, n_d; n)$. The presentation in [3] describes how to view the Grassmann manifold $Gr(k, n)$ as the quotient manifold $SO(n)/S(O(k)\times O(n-k))$. Similarly, we can view $FL(n_1, n_2, \ldots, n_d; n)$ as the quotient manifold $SO(n)/S(O(n_1) \times O(n_2) \times \cdots \times O(n_d))$ where $n_1 + n_2 + \cdots + n_d = n$. Let $P \in SO(n)$ be an n-by-n orthogonal matrix, the equivalence class $[P]$, representing a point on the flag manifold, is the set of orthogonal matrices

$$[P] = \left\{ P \begin{pmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & P_d \end{pmatrix} : P_i \in O(n_i)\,,\; n_1 + n_2 + \cdots + n_d = n \right\}.$$

It is well known that the geodesic paths on $SO(n)$ are given by exponential flows $Q(t) = Q\exp(t\mathbf{A})$ where $\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ is any skew symmetric matrix and $Q(0) = Q$. Viewing $FL(n_1, n_2, \ldots, n_d; n)$ as a quotient manifold of $SO(n)$, one can show that geodesics on $SO(n)$ continue to be geodesics on $FL(n_1, n_2, \ldots, n_d; n)$ as long as they are perpendicular to the orbits generated by $S(O(n_1) \times O(n_2) \times \cdots \times O(n_d))$ (for a derivation on a Grassmann manifold, see [11]). This leads one to conclude that the geodesic paths on $FL(n_1, n_2, \ldots, n_d; n)$ are exponential flows:

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

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

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

3.2 Geodesic and distance between two points on flag Manifold

By Equation (1), one may trace out the geodesic path on a flag manifold emanating from P in the direction of $\tilde{\mathbf{C}}$. In this section we utilize Equation (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 $(n_1, n_2, \ldots, n_d; n)$, obtain a factorization

$$Q_2 = Q_1 \cdot \exp(H) \cdot M \tag{2}$$

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

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

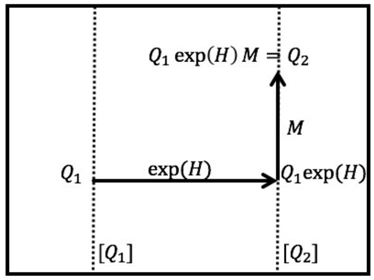

Fig. 2: Illustration of Equation (2). The vertical lines represents the equivalence classes $[Q_1]$ and $[Q_2]$ respectively. Q_1 is mapped to an element in $[Q_2]$ by right multiplication with $\exp(H)$ which is then sent to Q_2 by multiplying with M.

where H is skew symmetric, $M_i \in O(n_i)$, and $M \in SO(n)$. The **distance** between $[Q_1]$ and $[Q_2]$ along the geodesic given by H is

$$d([Q_1],[Q_2]) = \sqrt{\Sigma_{i=1}^{l} \lambda_i^2} \tag{3}$$

where the λ_i's are the distinct singular values of H.
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 2 is a pictorial illustration of the idea behind Equation (2). For $FL(k, n-k; n)$ i.e. the Grassmannian $Gr(k,n)$, one can solve for H analytically. Please see [3] for details. For the more general case, we will present an iterative algorithm to obtain a numerical approximation of H and M in Section 3.3. Before we proceed to the algorithm, let us further simplify Equation (2) by letting $Q = Q_1^T Q_2$. This allows us to rewrite (1) as

$$Q = \exp(H) \cdot M \tag{4}$$

Here we define $\mathcal{W}$ as the vector space of all n-by-n skew symmetric matrices. Let $\mathbf{p} = (n_1, n_2, \ldots, n_d; n)$. 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.

$$\mathcal{W}_{\mathbf{p}} = \{G \in \mathcal{W} \mid G = \begin{pmatrix} G_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & G_d \end{pmatrix}\}, \quad \mathcal{W}_{\mathbf{p}}^{\perp} = \{H \in \mathcal{W} \mid H = \begin{pmatrix} \mathbf{0}_{n_1} & & * \\ & \ddots & \\ -*^T & & \mathbf{0}_{n_d} \end{pmatrix}\}.$$

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

$$Q = \exp(H) \cdot \exp(G) \tag{5}$$

Algorithm 1: Iterative Alternating Exp-Log Algorithm

Input Data: Load matrices $Q_1, Q_2 \in SO(n)$ from the flag manifold, desired flag type $\mathbf{p} = (n_1, n_2, \ldots, n_d; n)$, initial $G_0 \in \mathcal{W}_{\mathbf{p}}$

Output Data: Optimal skew symmetric matrices $H \in \mathcal{W}_{\mathbf{p}}^{\perp}$,$G \in \mathcal{W}_{\mathbf{p}}$ such that $Q_1^T Q_2 = exp(H) \cdot exp(G)$

Result: Geodesic path and geodesic distance between $[Q_1]$ and $[Q_2]$

Initialization: Let $Q = Q_1^T Q_2$, set initial $d_H = \infty$

Define: Geodesic distance associated to H: $d_H = \sqrt{\Sigma_{i=1}^{l} \lambda_i^2}$ where λ_i's are distinct singular values of H.

```
for i = 1, ..., m do
    Generate initial randomized G_0
    for k = 1, ..., l do
        Step 1: H_k = Proj_H(log(Q · exp(G_{k-1})^T))
        Step 2: G_k = Proj_G(log(exp(H_k)^T Q))
    end
    if current H is associated to a smaller d_H then
        Update d_H
        Set current H and G as our output
    else
        continue
    end
end
```

where $G \in \mathcal{W}_{\mathbf{p}}$ and $H \in \mathcal{W}_{\mathbf{p}}^{\perp}$. It is important to note that in these computations, we are implicitly working on the fully oriented flag manifold $SO(n)/SO(n_1) \times SO(n_2) \times \cdots \times SO(n_d)$. There is a natural 2^{d-1} to 1 map from the fully oriented flag manifold to the flag manifold. As the output of the algorithm, we must pick the "optimal" H with 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 ($\hat{H} \notin \mathcal{W}_{\mathbf{p}}^{\perp}$), we project $\hat{H}$ onto $\mathcal{W}_{\mathbf{p}}^{\perp}$ to obtain the updated H. This projection zeros out certain select entries in $\hat{H}$, which is denoted by $H_1 = \mathrm{Proj}_{\mathcal{W}_{\mathbf{p}}^{\perp}}(\hat{H})$. Then we start updating G. Let $\hat{G} = \log(\exp(H_1)^T Q)$ we project $\hat{G}$ onto $\mathcal{W}_{\mathbf{p}}$ which zeros out other select entries, i.e. $G_1 = \mathrm{Proj}_G(\hat{G})$. Now then iterate this process until it converges. The pseudo code of our Iterative Alternating algorithm is presented in Algorithm 1.

We walk through two examples as an illustration of the numerical computation of the geodesic formula and distance between two points from the flag manifold. Here two types of flag manifold are utilized to illustrate the differ-

Algorithm 2: Flag Manifold Self-Organizing Mapping

Input Data: Load class labeled orthonormal data matrices $\{X_i\} \in R^{n\times k}$ for $i = 1, \cdots, P$ such that $X_i^T X_i = I_k$ where X_i is the nested subspace of interest, k is the dimension of the most outer subspace of interest, n is the dimension of the data, i is the matrix index. Let $\{Y_i\}$ for $i = 1, \cdots, P$ be the corresponding label set.
Output Data: Final centers and indices of each data subspace.
Result: Representation of points on $FL(n_1, n_2, \cdots, n_d; n)$ as indices of SOM centers.
Initialization: Complete the orthogonal complement of X_i by computing the QR-decompostion, i.e., $X_i = Q_i R_i$, such that $Q_i \in SO(n)$. Select number of centers and the structure of low dimensional index set. Initialize centers $\{C_i\}$ so that $C_i \in SO(n)$
Define: Geodesic distance d_g on the flag manifold
Step 1: Present a random point(nested sequence of subspaces) to the network.
Step 2: Move all the centers C_i proportionally towards the presented nested sequence of subspaces along the appropriate geodesic.
Step 3: Repeat until convergence.

ent geometry between a Grassmann and a flag manifold. Let $X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ and $Y = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}$ be two data matrices of interest. 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 $\mathbf{p} = (2, 2; 4)$: Let $Q = Q_1^T Q_2$ and the initial G_0(or any 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(or any H_i) 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 $\lambda_1 = 1.0172$, $\lambda_2 = 0.5536$ and 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 $FL(2, 2, ; 4)$ 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 $\mathbf{p} = (1, 1, 2; 4)$: 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 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 $x^{(\mu)}$ $\mu = 1, \cdots, p$ 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 $a_1, a_2, \cdots, a_N$. In the following example, we use $a_1 = (1, 7)$, $a_2 = (1, 6)$,$a_3 = (1, 5)$,$\cdots$, $a_{49} = (7, 1)$. 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 $\mathbf{p} = (n_1, n_2, \cdots, n_d; n)$, 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 Equation (3). To move the centers towards the nested subspace pattern X according to the SOM update we compute the geodesic, using the Iterative Alternating algorithm described in Algorithm 1, between each center C_i and nested subspace pattern X.

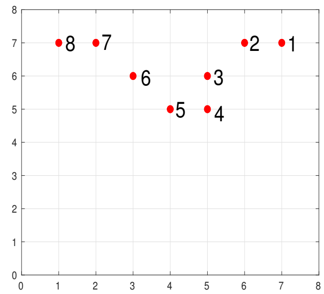

Fig. 3: 8 data points are sampled uniformly along the geodesic curve in $FL(1, 1, 2; 4)$ and labelled in the ascending order from 1 to 8. We observe that 7×7 2D lattice index sorts the label/order of the data points and preserves the geometry of this geodesic line.

Our localization term now becomes

$$t = \epsilon_n h(d(a_i, a_{i^*})).$$

We now take

$$h(s) = \exp(-s^2/\sigma^2)$$

where $\epsilon_n = \epsilon_0(1/\ln(\mathrm{e} + n - 1))$. 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 2.

4.1 Numerical Experiment

In this section we present an illustrative example concerning a straight line on a flag manifold. We select an initial position $Q \in SO(n)$ from a flag manifold of type $\mathbf{p} = (1, 1, 2; 4)$ and a velocity matrix $H \in \mathcal{W}_{\mathbf{p}}^{\perp}$. Then a set of 8 points are uniformly sampled along the geodesic path emanating from P in the direction H, i.e.,

$$X_i = Q \cdot \exp(t_i H),\ i = 0, 1, 2, \cdots, 7$$

where $t_i = 0.1i$. We employ the 7×7 2D grid index set described above for this example. In Figure (3), we see that the square lattice index set captures the geometry of the points living on a geodesic in the high dimensional flag manifold and sorts the labels (subscripts of data points X_i's) in the right order.

5 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. In future work we will explore the application of this flag SOM algorithm to real-world data sets.

References

1. Ingrid Daubechies, *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1992.
2. 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** (2014), 15–32.
3. Alan Edelman, Tomás A Arias, and Steven T Smith, *The geometry of algorithms with orthogonality constraints*, SIAM journal on Matrix Analysis and Applications **20** (1998), no. 2, 303–353.
4. Samuel Kaski, Jari Kangas, and Teuvo Kohonen, *Bibliography of self-organizing map (SOM) papers: 1981–1997*, Neural computing surveys **1** (1998), no. 3&4, 1–176.
5. M. Kirby, *Geometric data analysis: An empirical approach to dimensionality reduction and the study of patterns*, Wiley, 2001.
6. M. Kirby and C. Peterson, *Visualizing data sets on the Grassmannian using self-organizing mappings*, 2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM), June 2017, pp. 1–6.
7. T. Kohonen, *Self-organized formation of topologically correct feature maps*, Biol. Cybern. **43** (1982), 59.
8. Teuvo Kohonen, *The self-organizing map*, Proceedings of the IEEE **78** (1990), no. 9, 1464–1480.
9. ———, *The self-organizing map*, Neurocomputing **21** (1998), no. 1, 1–6.
10. ———, *Essentials of the self-organizing map*, Neural Networks **37** (2013), 52–65.
11. Xiaofeng Ma, Michael Kirby, Chris Peterson, and Louis Scharf, *Self-organizing mappings on the grassmannian with applications to data analysis in high-dimensions*, submitted Neural Computing and Applications (2018).
12. S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* l^2, Trans. Amer. Math. Soc. **315** (1989), 69–87.
13. ———, *A theory of multiresolution signal decomposition: the wavelet representation*, IEEE Trans. Pattern Anal. Mach. Intell. **315** (1989), 69–87.
14. Tim Marrinan, J Ross Beveridge, Bruce Draper, Michael Kirby, and Chris Peterson, *Flag manifolds for the characterization of geometric structure in large data sets*, Numerical Mathematics and Advanced Applications-ENUMATH 2013, Springer, 2015, pp. 457–465.
15. Tim Marrinan, Bruce Draper, J Ross Beveridge, Michael Kirby, and Chris Peterson, *Finding the subspace mean or median to fit your need*, Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1082–1089.
16. Timothy Marrinan, J. Ross Beveridge, Bruce Draper, Michael Kirby, and Chris Peterson, *Flag-based detection of weak gas signatures in long-wave infrared hyperspectral image sequences*, Proceedings SPIE Defense + Security, International Society for Optics and Photonics, 2016, pp. 98401N–98401N.