

## Sparse Exact PGA on Riemannian Manifolds

Monami Banerjee<sup>1</sup>, Rudrasis Chakraborty<sup>1</sup>, and Baba C. Vemuri<sup>1</sup>

<sup>1</sup>Department of CISE, University of Florida, FL 32611, USA

<sup>1</sup>{monamie.b, rudrasischa, baba.vemuri}@gmail.com

### Abstract

*Principal Component Analysis (PCA) is a widely popular dimensionality reduction technique for vector-valued inputs. In the past decade, a nonlinear generalization of PCA, called the Principal Geodesic Analysis (PGA) was developed to tackle data that lie on a smooth manifold. PGA suffers from the same problem as PCA in that, in both the methods, each Principal Component (PC) is a linear combination of the original variables. This makes it very difficult to interpret the PCs especially in high dimensions. This led to the introduction of sparse PCA (SPCA) in the vector-space input case. In this paper, we present a novel generalization of SPCA, called sparse exact PGA (SEPGA) that can cope with manifold-valued input data and respect the intrinsic geometry of the underlying manifold. Sparsity has the advantage of not only easy interpretability but also computational efficiency. We achieve this by formulating the PGA problem as a minimization of the projection error in conjunction with sparsity constraints enforced on the principal vectors post isomorphic mapping to  $R^m$ , where  $m$  is the dimension of the manifold on which the data reside. Further, for constant curvature smooth manifolds, we use analytic formulae for the projection error leading to an efficient solution to the SEPGA problem. We present extensive experimental results demonstrating the performance of SEPGA in achieving very good sparse principal components without sacrificing the accuracy of reconstruction. This makes SEPGA accurate and efficient in representing manifold-valued data.*

### 1. Introduction

Principal Component Analysis (PCA) [14] is a widely used data-processing and dimensionality reduction technique with numerous applications in Science and Engineering. Given an  $N \times m$  data matrix, where  $N$  is the number of  $m$ -dimensional data vectors, PCA seeks a linear combination of the original variables such that the data variance is maximized along these derived variables. Each Principal Component (PC) is a linear combination of all the  $m$  vari-

ables where each coefficient in the combination is typically nonzero. This makes each PC difficult to interpret as each PC is a linear combination of all of the  $m$  variables. This is a serious drawback of PCA (see [31] for more discussion on this issue). In order to overcome this drawback, researchers [31, 15] proposed a sparse PCA (sPCA) where each principal vector is sparse, and hence not only the dimension of the data gets reduced but the number of explicitly used variables are also small. The effectiveness of sparse principal vectors can be demonstrated with a simple example. Let the data points lie in  $\mathbf{R}^3$  and we want to choose two principal vectors, so a PCA algorithm can choose a set of two orthogonal vectors in  $\mathbf{R}^3$ , say  $\{v_1, v_2\}$ , such that the data variance in the subspace,  $V$ , spanned by  $\{v_1, v_2\}$  is maximized. Let the set of  $\{v_1, v_2\}$  for which the data variance in  $V$  is maximized be denoted by  $U$ . If  $V$  is spanned by say  $(0, 1, 0)$  and  $(1, 0, 0)$ , we want these two vectors to be selected as the principal vectors instead of some  $\{v_1, v_2\} \in U$ . This is because, the former are most natural and also sparse. In section 2, we present another toy example for further illustration.

PCA further requires data to lie in a vector space. However, with the advent of new sensing technologies and higher compute power, manifold-valued data have become widely popular in many fields including Computer Vision, Medical Imaging and Machine Learning. This motivated researchers to generalize PCA to the manifold-valued data setting. A nonlinear version of PCA, called the principal geodesic analysis (PGA) was introduced in [9] for data lying on a Riemannian Manifold. Since the objective function of PGA is nonlinear, Fletcher et al., proposed a linearized version [9]. This linearized version, henceforth referred to as PGA, though computationally efficient, lacks in performance accuracy when the data variance is high. This motivated researchers [25, 4] to solve the objective function of PGA exactly, instead of solving a linearized version of the objective function as in [9]. But, the exact objective function formulation in [25], hereafter referred to as exact PGA, involves solving a hard optimization problem which makes exact PGA infeasible even for moderate data size or dimension. Recently, authors in [4] presented a solution to

the exact objective function of PGA, hereafter referred to as ePGA, on manifolds with constant sectional curvature. They provided analytic formulae for the projection operation in the exact PGA formulation of [25].

Several other variants of PGA exist in literature and we present a few representative methods here. On the special Lie group,  $SO(3)$ , authors in [21] computed the exact principal geodesics. Geodesic PCA (GPCA) [13, 12] solves a different optimization function namely, optimizing the projection error along the geodesics. The authors in [12] minimize the projection error instead of maximizing variance in geodesic subspaces. GPCA does not use a linear approximation, but it is restricted to manifolds where a closed form expression for the geodesics exists. More recently, a probabilistic version of PGA called PPGA was presented in [30], which is a nonlinear version of PPCA [26]. Recently, Hauberg [11] generalized the concept of principal curves to Riemannian manifolds and proposed this as a good alternative to PGA for representing the manifold valued data. He argues that principal curves can be more useful than principal geodesics which are required to pass through the mean of the data. Finally, none of the above variants of PGA attempt to compute the solution to the exact PGA problem defined in [25], nor do they address the sparsity of the principal vectors.

Since a general Riemannian manifold lacks vector space structure and hence does not have a canonical basis, sparsity is not well defined. In this work, we first define sparsity of principal vectors on a Riemannian manifold. An obvious and approximate way to perform sparse PGA is by linearizing the manifold locally, as was done in [9], and then apply sPCA on that linearized space. This can be easily done on the Riemannian manifold by first lifting every data point on to the tangent space anchored at the Fréchet mean (FM) [10],  $\mu$ , of the data points and then performing sPCA on the tangent space. Henceforth, we will refer to this formulation as sPGA, which obviously has the same drawback as PGA [9], i.e., for data with large variance the performance will be quite poor. We demonstrate this low performance accuracy of sPGA via experiments in section 3. In this paper, we define a sparse exact PGA algorithm, hereafter referred to as SEPGA, which solves the objective function on the Riemannian manifold without the linearization step described above. We achieve the sparsity of principal vectors residing in the tangent space (anchored at the Fréchet mean of the given data) of the manifold by enforcing the  $\ell_1$ -norm constraint on the isomorphically mapped principal vectors to  $R^m$ , where  $m$  is the dimension of the manifold on which the input data reside. Note that the sparsity constraint can not meaningfully be enforced in the tangent space prior to the aforementioned isomorphic mapping to  $R^m$ . We present an example in Fig. 1 to demonstrate the usefulness of our sparse PG formulation for better interpretation of the data.

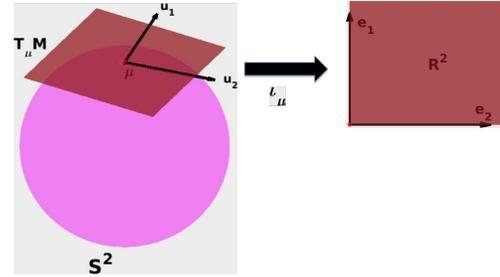


Figure 1. Sparse PGA interpretation

Let  $\mathbf{u}_1 = (1/\sqrt{2}, 1/\sqrt{2}, 0)^t$  and  $\mathbf{u}_2 = (-1/\sqrt{2}, 1/\sqrt{2}, 0)^t$ . Further, let  $\{\mathbf{u}_1, \mathbf{u}_2\}$  form a basis of  $T_\mu \mathbf{S}^2$ , where  $\mu = (0, 0, 1)^t$ . Let  $\mathbf{e}_1 = (1, 0)^t$  and  $\mathbf{e}_2 = (0, 1)^t$ , and  $\iota_\mu$  map  $\mathbf{u}_1 \mapsto \mathbf{e}_1$  and  $\mathbf{u}_2 \mapsto \mathbf{e}_2$ . In this example, we use two methods to define sparsity, first we add a sparsity constraint on the principal vector (which lies in  $T_\mu \mathbf{S}^2$ ) and second we add a sparsity constraint on the isomorphically mapped principal vector. Let, the first method return  $(1, 0, 0)^t$  as the principal vector while the second method returns say,  $(1/\sqrt{2}, 1/\sqrt{2}, 0)^t$  as the principal vector (after applying  $\iota_\mu$  on  $(1/\sqrt{2}, 1/\sqrt{2}, 0)^t$  we get  $\mathbf{e}_1 = (1, 0)^t$ ). Although the first principal vector has more zeros but if we expand  $(1, 0, 0)^t$  in the chosen basis of  $T_\mu \mathbf{S}^2$ , i.e.,  $(1, 0, 0)^t = 1/\sqrt{2} \mathbf{u}_1 + (-1/\sqrt{2}) \mathbf{u}_2$ , we can see that none of the components along any of the basis vector is zero, i.e., we can not infer which dimension of the data has more influence on the principal vector. But, the principal vector returned by the second method is  $(1/\sqrt{2}, 1/\sqrt{2}, 0)^t = 1 \mathbf{u}_1 + 0 \mathbf{u}_2$ , i.e., we can easily say that the second dimension of the data has no influence on the principal vector. This example demonstrates that more zeros in the principal vector (in the tangent space of the manifold) does not imply that the vector is more interpretable. This is because, in contrast to the Euclidean space, where we have a canonical basis, the manifold lacks any such canonical basis in its tangent space. Hence, a principal vector is more interpretable if it is “close” to the chosen basis of the tangent space.

Our SEPGA method overcomes the drawback of PGA and provides a set of principal components each of which is sparse. We will demonstrate the performance of the SEPGA method in section 2 via comparisons to PGA [9], sPGA and ePGA [4]. In this comparison, we report the average reconstruction error [25], average expressed variance and the average sparsity measure (defined in section 3).

The rest of the paper is organized as follows. In Section 2, we present the formulation and the algorithm of our SEPGA method. We present experimental results for the SEPGA algorithm along with comparisons to PGA and ePGA in Section 3. Section 3 also contains a comparative analysis with the sPGA algorithm. Finally, in section 4, we draw conclusions.

## 2. Sparse Exact PGA

Let  $(\mathcal{M}^m, g)$  be a complete Riemannian manifold where  $m = \dim(\mathcal{M})$  is equipped with a Riemannian metric  $g$  [8]. Let,  $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbf{R}$  be the distance function induced by  $g$  on  $\mathcal{M}$  given by,  $d(x_i, x_j) = g_{x_i}(\text{Log}_{x_i} x_j, \text{Log}_{x_i} x_j)$ , where  $\text{Log}$  is the Riemannian inverse exponential map (Note that the completeness assumption of manifold ensures that  $\text{Log}$  map is defined on the entire manifold, but for a manifold which is not geodesically complete, within a geodesic ball of appropriate radius, the  $\text{Log}$  map is well defined). Consider the data  $X = \{x_i\}_{i=1}^N \subset \mathcal{M}$  within a geodesic ball of sufficient radius such that Fréchet mean (FM) [10],  $\mu$ , exists and is unique [1]. The goal of Principal Geodesic Analysis (PGA) is to find a set of  $r < m$  orthogonal basis vectors of  $T_\mu \mathcal{M}$  (called principal vectors),  $\{\mathbf{v}_j\}_{j=1}^r$  such that the data variance along the geodesic submanifold spanned by these  $r$  vectors is maximized [9]. An alternative definition of PGA [25] involves minimizing the reconstruction error,  $\sum d^2(x_i, \hat{x}_i)$ , where  $\hat{x}_i$  is the  $i^{\text{th}}$  reconstructed data point in the principal submanifold spanned by the basis vectors  $\{\mathbf{v}_j\}_{j=1}^r$ . It can be easily shown that on a manifold with zero sectional curvature (canonically isomorphic to Euclidean space), this alternative formulation leads to exactly the same solution as one obtained from PGA [9, 4]. But, on a general Riemannian manifold these two formulations are not equivalent, but nonetheless it serves as an alternative formulation [25]. We will use this alternative formulation to define PGA on a Riemannian manifold since it uses the intrinsic geometry of the Riemannian manifold under consideration and does not consider a linearization as in the PGA formulation in [9]. The principal vectors  $\{\mathbf{v}_j\}$  are defined recursively by,

$$\mathbf{v}_j = \arg \min_{\|\mathbf{v}\|=1, \mathbf{v} \in V_{j-1}^\perp} \frac{1}{N} \sum_{i=1}^N d^2(x_i, \Pi_{S_j}(x_i)) \quad (1)$$

$$S_j = \text{Exp}_\mu(\text{span}\{V_{j-1}, \mathbf{v}_j\}) \quad (2)$$

where,  $V_{j-1} = \{\mathbf{v}_1, \dots, \mathbf{v}_{j-1}\}$ .  $S_j$  is the submanifold spanned by  $V_j = \{V_{j-1}, \mathbf{v}_j\}$ , and  $\Pi_{S_j}(x)$  is the point in  $S_j$  closest to  $x \in \mathcal{M}$ .

### 2.1. Various Forms of the Projection Operator

We now present a method to approximate  $\Pi_{S_j}(x)$  on a general manifold  $\mathcal{M}$  and present exact analytic forms derived in [4] for  $\Pi_{S_j}(x)$  in the case of constant curvature manifolds. Let  $\hat{x} = \Pi_{S_j}(x)$ , then  $\hat{x}$  can be expressed as  $\text{Exp}_\mu\left(\sum_j c(x, \mathbf{v}_j) \mathbf{v}_j\right)$  where the coefficient function  $c : \mathcal{M} \times T_\mu \mathcal{M} \rightarrow \mathbf{R}$  can be defined as:

$$c(x, \mathbf{v}_j) = \text{sgn}(g_\mu(\mathbf{v}_j, \text{Log}_\mu x)) d(\mu, \Pi_{\text{span}\{\mathbf{v}_j\}}(x)) \quad (3)$$

where  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$  returns the closest point of  $x$  on the geodesic of dim-1 submanifold spanned by  $\mathbf{v}_j$ . Note

that, we used  $\text{sgn}(g_\mu(\mathbf{v}_j, \text{Log}_\mu x))$  to define  $c(x, \mathbf{v}_j)$  as a signed distance, since the coefficient,  $c(x, \mathbf{v}_j)$ , can be negative as well. Since, on a general Riemannian manifold,  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$  is the solution of a hard optimization problem [25], hence, we approximate  $c(x, \mathbf{v}_j)$  as follows:

$$c(x, \mathbf{v}_j) = g_\mu(\text{Log}_\mu x, \mathbf{v}_j) \quad (4)$$

But, for a constant curvature manifold, we can use the exact analytic form of  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$  by using the closed form expressions for the projection operator derived in a recently published work [4]. It is well known in Differential Geometry that any nonzero constant curvature  $m$ -dim manifold is isomorphic to either the hypersphere,  $\mathbf{S}^m$ , or the hyperbolic manifold  $\mathbf{H}^m$ . Therefore, we only need to consider these two cases. For the sake of completeness, we will restate these closed form expressions for the projection operator,  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$ , in the case where  $x \in \mathbf{S}^m$  or  $x \in \mathbf{H}^m$ .

On  $\mathbf{S}^m$ ,

$$\begin{aligned} \Pi_{\text{span}\{\mathbf{v}_j\}}(x) = & \cos\left(\arctan\left(\frac{\langle \mathbf{v}_j, x \rangle / \langle x, \mu \rangle}{|\mathbf{v}_j|}\right)\right) \mu \\ & + \sin\left(\arctan\left(\frac{\langle \mathbf{v}_j, x \rangle / \langle x, \mu \rangle}{|\mathbf{v}_j|}\right)\right) \mathbf{v}_j / |\mathbf{v}_j| \end{aligned} \quad (5)$$

where  $\langle \cdot, \cdot \rangle$  is the standard inner product on  $\mathbf{R}^{m+1}$ . Analogously, we can define  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$  on  $\mathbf{H}^m$  as

$$\Pi_{\text{span}\{\mathbf{v}_j\}}(x) = \cosh(a)\mu + \sinh(a)\mathbf{v}_j/|\mathbf{v}_j| \quad (6)$$

where,  $a = \tanh^{-1}\left(-\frac{\langle \mathbf{v}_j, x \rangle_H / \langle x, \mu \rangle_H}{|\mathbf{v}_j|}\right)$ . Here  $\langle x, y \rangle_H$  is

defined as follows:  $\langle x, y \rangle_H = -x_1 y_1 + \sum_{i=2}^{m+1} x_i y_i$ . Note that, both  $\langle x, y \rangle$  and  $\langle x, y \rangle_H$  are defined on the ambient space  $\mathbf{R}^{m+1}$ . Equipped with these closed form expressions for  $\Pi_{\text{span}\{\mathbf{v}_j\}}(x)$  on constant curvature manifolds, we can compute  $c(x, \mathbf{v}_j)$  analytically.

### 2.2. Sparse PGA Formulation and Algorithm

In Principal Component analysis (PCA), each principal vector is a linear combination of all the basis vectors and moreover all the coefficients are in general nonzero. For example, given a data in  $\mathbf{R}^3$ , suppose  $(a, b, c)^t$  (for some non zero  $a, b, c$  such that  $a^2 + b^2 + c^2 = 1$ ) and  $(d, f, 0)^t$  (with non zero  $d, f$  with  $d^2 + f^2 = 1$ ) both capture the maximum variance. Then both  $(a, b, c)^t$  and  $(d, f, 0)^t$  are the first principal vectors but  $(d, f, 0)^t$  is sparse, hence more preferable. The reason for  $(d, f, 0)^t$  being preferable over non-sparse principal vectors is that, it is easy to interpret the data using  $(d, f, 0)^t$  as the principal vector, i.e., we can say that the third dimension of the data is redundant in capturing the maximum variance of the data as,  $(d, f, 0)^t =$

$d\mathbf{e}_1 + f\mathbf{e}_2 + 0\mathbf{e}_3$ . Since,  $(a, b, c)^t = a\mathbf{e}_1 + b\mathbf{e}_2 + c\mathbf{e}_3$ , where  $\mathbf{e}_i$  are canonical basis and  $a, b, c$  are non-zero, the principal vector  $(a, b, c)^t$  does not reveal much about the structure of the data. So, the goal of sparse PCA is to find principal vectors that have more zero components, i.e., to find vectors that are “close” to being the canonical basis vectors. We refer the readers to [31] for a detailed discussion regarding the usefulness of sparse principal vectors. The usefulness of sparse PCA motivates us to seek a generalization in a general Riemannian manifold setting, i.e., to find “sparse” principal geodesics. Unfortunately, as a general Riemannian manifold,  $\mathcal{M}$ , lacks vector space structure, there is no canonical basis, i.e., “sparsity” is not well-defined. But notice that on  $\mathcal{M}$ , tangent space at any point is isomorphic to  $\mathbf{R}^m$ , where  $m$  is the dimension of  $\mathcal{M}$ . The principal vectors,  $\{\mathbf{v}_i\}$  lie on the tangent space at  $\mu, T_\mu\mathcal{M}$ , so we define a principal vector  $\mathbf{v}_i$  to be sparse if and only if  $\iota_\mu(\mathbf{v}_i)$  is sparse, where  $\iota_\mu$  is an isomorphism from  $T_\mu\mathcal{M}$  to  $\mathbf{R}^m$ . Hence, given an isomorphism  $\iota_\mu$ , our goal is to find a set of  $r$  principal vectors ( $r$  corresponding principal geodesics) that are isomorphic to the canonical basis of  $\mathbf{R}^m$  (or “close” to being canonical) such that they minimize the reconstruction error.

As an example, consider the following toy data given in Fig. 2. Here, we randomly generated 500 points (with  $\mu = (0, 0, 1)^t$ )

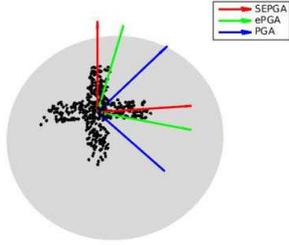


Figure 2. Sparse PGA for a toy data set.

along two orthogonal geodesics on  $\mathbf{S}^2$  with added random noise. Note that  $T_\mu\mathcal{M} \cong \mathbf{R}^2$ . Let us choose the basis of  $T_\mu\mathcal{M}$  to be  $(0, 1, 0)^t$  and  $(1, 0, 0)^t$ . Let us define  $\iota_\mu$  to be  $\iota_\mu((0, 1, 0)^t) = \mathbf{e}_1$  and  $\iota_\mu((1, 0, 0)^t) = \mathbf{e}_2$  (Note that in order to define isomorphism between vector spaces, it is enough to define the mapping on the basis vectors). The principal geodesics are computed using PGA [9], Exact PGA (ePGA) [4] and the proposed method of sparse PGA (SEPGA). It is evident from the figure, that SEPGA yields a set of vectors  $(0, 1, 0)^t$  and  $(1, 0, 0)^t$ , which are isomorphic to  $\mathbf{e}_1$  and  $\mathbf{e}_2$  respectively. The other two PGA methods fail to do so, as expected. However, it should be pointed out that all of these three methods give perfect reconstruction of the data, as the principal vectors obtained by all three methods span the same subspace.

By choosing appropriate  $\iota_\mu$ , we will get different variants of sparse PGA, as suggested below.

- If we don’t know the structure of the data on  $\mathcal{M}$ , we can choose an arbitrary basis of  $T_\mu\mathcal{M}$ , and hence  $\iota_\mu$  (by Algorithm 1), then get the sparse principal vectors with respect to the chosen  $\iota_\mu$ . In all of our experiments,

we follow this procedure.

- If we know that the data has sparse principal vectors in the ambient space, e.g., the synthetic example in Fig. 2 where we know the sparse principal vectors are the two canonical basis of  $\mathbf{R}^3$ ,  $(0, 1, 0)^t$  and  $(1, 0, 0)^t$  respectively, we can choose  $\iota_\mu$  to be the identity map (Note that, here the identity map is defined using the embedding of  $\mathbf{S}^2$  in  $\mathbf{R}^3$ ).
- One can choose the principal vectors given by [9, 4] as the basis of  $T_\mu\mathcal{M}$  and hence choose  $\iota_\mu$ . Then, the algorithms in [9, 4] are special cases of our sparse formulation. In other words, we can generate the same principal vectors as given by [9, 4] without removing the sparsity constraint.

In the rest of this section, we will develop the SEPGA algorithm. Using the above hypothesis and notations, the SEPGA objective function is given below:

$$\arg \min_{\{\mathbf{v}_j\}_{j=1}^r \subset T_\mu\mathcal{M}} E = \underbrace{\frac{1}{N} \sum_{i=1}^N d^2(x_i, \hat{x}_i)}_{e_1} + \lambda_1 \underbrace{\sum_{j=1}^r \|\iota_\mu(\mathbf{v}_j)\|_1}_{e_2} \quad (7)$$

$$\text{subject to, } \underbrace{g_\mu(\mathbf{v}_j, \mathbf{v}_k)}_{e_3} = \delta_{jk}, \quad \forall j, k$$

where,

$$\hat{x}_i = \text{Exp}_\mu \left( \sum_j c(x_i, \mathbf{v}_j) \mathbf{v}_j \right), \quad \forall i \quad (8)$$

Here,

$$c(x_i, \mathbf{v}_j) = \begin{cases} \text{sgn}(g_\mu(\mathbf{v}_j, \tilde{\mathbf{v}}_i)) d(\mu, \Pi_{ij}), & \mathbf{S}^m (\text{Eq.5}) \\ \text{sgn}(g_\mu(\mathbf{v}_j, \tilde{\mathbf{v}}_i)) d(\mu, \Pi_{ij}), & \mathbf{H}^m (\text{Eq.6}) \\ g_\mu(\tilde{\mathbf{v}}_i, \mathbf{v}_j), & \text{other } \mathcal{M} (\text{Eq.4}) \end{cases}$$

Where,  $\tilde{\mathbf{v}}_i = \text{Log}_\mu x_i$  and  $\Pi_{ij} = \Pi_{\text{span}\{\mathbf{v}_j\}}(x_i)$ . The tangent space,  $T_x\mathcal{M}$  is isomorphic to  $\mathbf{R}^m$  for all  $x \in \mathcal{M}$ . Let,  $\iota_x : T_x\mathcal{M} \xrightarrow{j_x} V_m \xrightarrow{\iota} \mathbf{R}^m$  be an isomorphism, where  $V_m$  is an  $m$ -dimensional vector space. So,  $\iota_x = j_x \circ \iota$ . Note that, below, we will show that  $j_x$  is an isometry (with a pull back inner product on  $V_m$ ) but  $\iota$  is not, so in general,  $\iota_x$  is not an isometry. We define a pull back inner product,  $\tilde{g}$  on  $V_m$  as follows:

$$\tilde{g}(\tilde{u}, \tilde{v}) := \left( (\iota_x^{-1})^* g \right) (\tilde{u}, \tilde{v}) = g \left( \iota_x^{-1}(\tilde{u}), \iota_x^{-1}(\tilde{v}) \right) \quad (9)$$

where,  $\tilde{u}, \tilde{v} \in V_m$ . As  $\iota_x$  is an isomorphism, hence  $\text{Ker} = \emptyset$ , so,  $\tilde{g}$  is positive definite, which makes  $\tilde{g}$  a valid inner product on  $V_m$ . On,  $\text{SPD}(m)$ ,  $j_x(\mathbf{v}) = \text{vec}(\mathbf{v})$ , where “vec”

is the standard vectorization operation on matrices, and  $\iota_x(\mathbf{v}) = \text{upper}(\mathbf{v})$ , where  $\text{upper}$  is the vectorized upper-diagonal (including the diagonal elements) of  $\mathbf{v}$ . On  $\mathbf{S}^m$ ,  $j_x$  is the identity map, e.g., let  $m = 2$ ,  $x = (0, 0, 1)^t$ ,  $\mathbf{v} = (a, b, 0)^t$ ,  $a, b \in \mathbf{R}$ , then  $\iota_x(\mathbf{v}) = (a, b)^t$ . In Algorithm 1, we give an algorithm to get  $\iota_x$  for  $\mathcal{M}$  where  $x \in \mathcal{M}$ .

---

**Algorithm 1:** The algorithm to find  $\iota_x$

---

**Input:**  $x \in \mathcal{M}$

**Output:**  $\iota_x$

- 1 Identify  $T_x \mathcal{M}$  with  $V_m$  by  $j_x$ ,  $m$  is the dimension of  $\mathcal{M}$ ;
  - 2 Choose  $m$  orthogonal basis,  $\{\mathbf{u}_i\}_{i=1}^m$  of  $V_m$  using Gram-Schmidt orthogonalization;
  - 3 Define a linear map  $\iota_\mu$  which maps  $\mathbf{u}_i$  to  $\mathbf{e}_i$ , for all  $i$ , where  $\mathbf{e}_i$  is the canonical basis of  $\mathbf{R}^m$ ;
  - 4 As we define  $\iota_\mu$  to be linear, we can get  $\iota_x(\mathbf{v})$  for the given  $\mathbf{v} \in T_x \mathbf{M}$ ;
- 

In the objective function in Eq. 7, the error term  $e_1$  represents the average reconstruction error and the term  $e_2$  induces sparsity on  $\iota_\mu(\mathbf{v}_j)$ . The constraint  $e_3$  ensures that the principal vectors  $\{\mathbf{v}_j\}$  are orthonormal and mutually orthogonal (the  $\delta$  is the Kronecker-delta function).  $\lambda_1$  is the regularizing constant and we used a cross validation scheme to select the appropriate values for these parameters. Further details on this is presented in the experimental section. We used an augmented Lagrangian formulation to solve the objective function. The unconstrained objective using the augmented Lagrangian is given below where  $\lambda_{jk}$ ,  $\lambda_2$  are the Lagrange multipliers.

$$\begin{aligned} \arg \min_{\substack{\{\mathbf{v}_j\}_{j=1}^r \\ \subset T_\mu \mathcal{M}}} E &= \frac{1}{N} \sum_{i=1}^N d^2(x_i, \hat{x}_i) + \lambda_1 \sum_{j=1}^r \|\iota_\mu(\mathbf{v}_j)\|_1 \\ &+ \frac{\lambda_2}{2} \sum_{jk} (g_\mu(\mathbf{v}_j, \mathbf{v}_k) - \delta_{jk})^2 \\ &- \sum_{jk} \lambda_{jk} (g_\mu(\mathbf{v}_j, \mathbf{v}_k) - \delta_{jk}) \end{aligned} \quad (10)$$

Now, we optimize this unconstrained objective function using a gradient descent technique. The detailed algorithm is given in Algorithm 2. To illustrate the computation of the gradient of the objective function,  $E$ , with respect to  $\{\mathbf{v}_j\}$ , we now present this for two commonly encountered manifolds in Computer Vision namely,  $\mathbf{S}^m$  and  $\text{SPD}(m)$  (i.e., manifold of  $m$ -dimensional symmetric positive definite matrices).

**Computation of the gradient on  $\mathbf{S}^m$**

$$\frac{\partial e_1}{\partial \iota_\mu(\mathbf{v}_j)} = \frac{1}{N} \sum_{i=1}^N \left( \text{Log}_{x_i} \hat{x}_i \right)^t c(x_i, \mathbf{v}_j) I_m + \mathbf{v}_j \frac{\partial c(x_i, \mathbf{v}_j)}{\partial \mathbf{v}_j} \quad (11)$$

where,

$$\frac{\partial c(x_i, \mathbf{v}_j)}{\partial \iota_\mu(\mathbf{v}_j)} = \text{sgn}(\mathbf{v}_j^t \text{Log}_\mu x_i) \frac{-\mu^t}{\sqrt{1 - (\mu^t \Pi_{ij})^2}} \frac{\partial \Pi_{ij}}{\partial \mathbf{v}_j} \quad (12)$$

and

$$\begin{aligned} \frac{\Pi_{ij}}{\partial \iota_\mu(\mathbf{v}_j)} &= \frac{\mu^t x_i}{((\mu + \mathbf{v}_j)^t x_i)^2} \left\{ \cos \left( \arctan \left( \frac{\mathbf{v}_j^t x_i}{\mu^t x_i} \right) \right) \mathbf{v}_j - \right. \\ &\left. \sin \left( \arctan \left( \frac{\mathbf{v}_j^t x_i}{\mu^t x_i} \right) \right) \mu \right\} x_i^t + \sin \left( \arctan \left( \frac{\mathbf{v}_j^t x_i}{\mu^t x_i} \right) \right) I_m \end{aligned} \quad (13)$$

$$\frac{\partial e_2}{\partial \iota_\mu(\mathbf{v}_j)} = \text{sgn}(\mathbf{v}_j^t) \quad (14)$$

We can obtain an analogous formula on  $\mathbf{H}^m$  for  $\frac{\partial e_1}{\partial \iota_\mu(\mathbf{v}_j)}$  that can be derived using Eqs. [6, 11]. The formula for  $\frac{\partial e_2}{\partial \mathbf{v}_j}$  will be same as in Eq. 14.

**Computation of the gradient on  $\text{SPD}(m)$**

On  $\text{SPD}(m)$ , we use the  $\text{GL}(m)$  invariant metric, where  $\text{GL}(m)$  denotes the  $m$ -dimensional general linear group. Let,  $x \in \text{SPD}(m)$ ,  $\mathbf{u}, \mathbf{v} \in T_x \text{SPD}(m)$ . Then, the  $\text{GL}(m)$  invariant metric is defined by,  $g_x(\mathbf{u}, \mathbf{v}) = g_{h_x h^t}(\mathbf{u} h, \mathbf{v} h^t)$ ,  $\forall h \in \text{GL}(m)$  [8]. On  $\text{SPD}(m)$ , the  $\text{GL}(m)$  invariant inner product takes the form

$$g_x(\mathbf{u}, \mathbf{v}) = \text{tr}(x^{-1/2} \mathbf{u} x^{-1} \mathbf{v} x^{-1/2}). \quad (15)$$

Note that,  $T_x \text{SPD}(m)$  can be identified with the space of symmetric matrices. Hence, in Eq. 7,  $\{\mathbf{v}_j\}$  are symmetric matrices when the underlying manifold is  $\text{SPD}(m)$ . Now, we are ready to give the expression for the gradients:

$$\begin{aligned} \frac{\partial e_1}{\partial \iota_\mu(\mathbf{v}_j)} &= \frac{2}{N} \sum_{i=1}^N \iota_\mu \left( \mu^{-1/2} \text{Log}_{x_i} \hat{x}_i \mu^{-1/2} \right)^t \left( \mu^{-1/2} \right)^{\otimes 2} \\ &\left( c(x_i, \mathbf{v}_j) (I_m \otimes I_m) + \frac{\partial c(x_i, \mathbf{v}_j)}{\partial \iota_\mu(\mathbf{v}_j)} \right) \end{aligned} \quad (16)$$

where,

$$\frac{\partial c(x_i, \mathbf{v}_j)}{\partial \iota_\mu(\mathbf{v}_j)} = \iota_\mu \left( \mu^{-1/2} \text{Log}_\mu x_i \mu^{-1/2} \right)^t \left( \mu^{-1/2} \right)^{\otimes 2} \quad (17)$$

where,  $(\mu^{-1/2})^{\otimes 2} = (\mu^{-1/2} \otimes \mu^{-1/2})$ .

$$\frac{\partial e_2}{\partial \iota_\mu(\mathbf{v}_j)} = \text{sgn}(\iota_\mu(\mathbf{v}_j)^t) \quad (18)$$

---

**Algorithm 2:** The SEPGA algorithm

---

**Input:**  $X = \{x_i\}_{i=1}^N \subset \mathcal{M}$ ,  $\lambda_1, \lambda_2, \eta > 0$ ,  $\epsilon > 0$   
**Output:**  $\{\mathbf{v}_j\}_{j=1}^r \subset T_\mu \mathcal{M}$

- 1 Compute the FM,  $\mu$  of  $X$ ;
- 2 Find an isomorphism,  $\iota_\mu$ , using Algorithm 1 ;
- 3 Randomly initialize  $\{\mathbf{v}_j\}_{j=1}^r$  to be unit vectors on  $T_\mu \mathcal{M}$ ;
- 4 For constant curvature manifold,  $\mathbf{S}^m(\mathbf{H}^m)$ , use Eq. 5(6) to compute  $c(x_i, \mathbf{v}_j)$  and for a general smooth manifold, use Eq. 4;
- 5  $\hat{x}_i \leftarrow \text{Exp}_\mu \left( \sum_j c(x_i, \mathbf{v}_j) \mathbf{v}_j \right)$ ,  $\forall i$  using Eq. 8;
- 6 Compute the objective function,  $E$ , using Eq. 7;
- 7  $E^{\text{old}} \leftarrow E$ , and  $flag \leftarrow 1$ ;
- 8 Initialize  $\lambda_{jk}$  ;
- 9 **while**  $flag = 1$  **do**
- 10     Compute  $\frac{\partial E}{\partial \iota_\mu(\mathbf{v}_j)}$ ,  $\forall j = 1, \dots, r$  ;
- 11      $\mathbf{v}_j^{\text{new}} \leftarrow \iota_\mu^{-1} \left( \iota_\mu(\mathbf{v}_j) - \eta \frac{\partial E}{\partial \iota_\mu(\mathbf{v}_j)} \right)$ ,  $\forall j = 1, \dots, r$ ;
- 12      $\mathbf{v}_j^{\text{new}} \leftarrow \iota_\mu^{-1} (\iota_\mu(\mathbf{v}_j))$ ,  $\forall j = 1, \dots, r$ ;
- 13     Recompute  $\hat{x}_i \leftarrow \text{Exp}_\mu \left( \sum_j c(x_i, \mathbf{v}_j^{\text{new}}) \mathbf{v}_j^{\text{new}} \right)$ ,  $\forall i$  using Eq. 8;
- 14     Recompute the objective function,  $E$ , using Eq. 7;
- 15     **if**  $\| \frac{\partial E}{\partial \iota_\mu(\mathbf{v}_j^{\text{new}})} \| < \epsilon$ ,  $\forall j = 1, \dots, r$  **then**
- 16          $flag \leftarrow 0$ ;
- 17     **end**
- 18     **if**  $E < E^{\text{old}}$  **then**
- 19          $\lambda_{jk} \leftarrow \lambda_{jk} - \lambda_2 (g_\mu(\mathbf{v}_j^{\text{new}}, \mathbf{v}_k^{\text{new}}) - \delta_{jk})$ ,  $\forall j, k$ ;
- 20          $\mathbf{v}_j \leftarrow \mathbf{v}_j^{\text{new}}$ ,  $\forall j$ , and  $E^{\text{old}} \leftarrow E$ ;
- 21     **else**
- 22          $\eta \leftarrow 0.9 \eta$ ;
- 23     **end**
- 24 **end**

---

### 3. Experimental Results

In this section, we present experimental results demonstrating the comparative performance of SEPGA with PGA [9], sPGA and ePGA [4]. For sPGA, we used the publicly available code for the algorithm in [23] to run sPCA in the tangent space. For ePGA, we used the code available online for the algorithm in [3]. For exact PGA on general manifolds, we used the code available online provided by authors of [24]. This implementation however is not scalable to data of moderate size and dimensions. Further, it requires the computation of the Hessian of the objective, which is computationally expensive. Hence, for the real data applications that we present in this section, we were unable to report the results for the exact PGA algorithm. Though one can use a Sparse matrix version of the exact PGA code, along with efficient parallelization to make the exact PGA algorithm suitable for moderately large data, we would like to point out that since our algorithm does not need such modifications, it clearly gives SEPGA an additional advantage over exact PGA from a computational efficiency perspective.

Performance comparisons between SEPGA, PGA, sPGA and ePGA were achieved using the *average reconstruction error* measure i.e., the term,  $e_1$  in Eq. 7. We also report

the *expressed variance* (denoted by  $evar$ ) defined as : the ratio of the variance captured by the PGs to the data variance. Hence, the value is in the interval  $[0, 1]$ , with 1 being the best possible  $evar$  value. We also measured the *average sparsity* (denoted by  $\varsigma$ ) which is defined as the average percentage of the components in the isomorphically mapped principal vectors  $\leq 1E - 4$ . Note that, for sPGA, ePGA and PGA we took  $\iota_\mu$  to be identity. For ePGA and PGA, the average sparsity over all experiments is  $< 5\%$ , hence we do not report the average sparsity for these two methods. We report the *average reconstruction error*,  $e_1$ , value as obtained by sPGA while approximately maintaining the same sparsity (percentage) as given by SEPGA. The computation time required by SEPGA is comparable to the other PGA methods. We have used a cross validation scheme to compute  $\lambda_1$  and  $\lambda_2$  values in SEPGA. All the experiments reported here were performed on a desktop with a single 3.33 GHz Intel-i7 CPU with 24 GB RAM.

#### 3.1. Comparative performance of SEPGA on OASIS data [19]

In this section, we present the comparative results on publicly available OASIS data [19]. This dataset contains T1-MR brain scans of subjects with ages ranging between 18 to 96 including individuals with early Alzheimer’s Disease. We randomly picked 4 brain scans from within each decade, totaling 36 brain images. From each brain scan, we segmented the corpus callosum (CC) region. Then, we constructed two data representations from each of the CC shapes as follows. (1) We take the boundary of the CC shape and map it to  $\mathbf{S}^{24575}$  using the Schrödinger distance transform (SDT) feature [7]. (2) We selected a set of 250 landmark points on each CC shape and map these points on to the Kendall shape space [16] of dimension 250 which is a *complex projective space* isomorphic to  $\mathbf{S}^{499}$ . The performance of all the four methods are shown in Table 1.

From the results in Table 1, we can see that SEPGA gives very good sparsity without sacrificing the reconstruction error,  $e_1$ , and the expressed variance. sPGA performs very poorly in terms of expressed variance. Although ePGA maximizes the expressed variance, SEPGA gives a better competitive expressed variance along sparse principal vectors. We now present some reconstruction results in Figure 3, which depicts the point cloud representation of the CC shape boundary, using the principal geodesics given by SEPGA. The first row contains the original CC shapes of varying ages (age of the subjects is shown in the last row). The second and third row contain the reconstructed shapes using 30 and 50 (sparse) principal components, respectively. It is evident from the figure that using only 50 principal components (the original data dimension in the Kendall shape space is 250), SEPGA gives a nearly perfect reconstruction ( $e_1 \approx 1E - 16$ ).

$r$	SEPGA			sPGA			PGA		ePGA	
	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$evar$	$e_1$	$evar$
1	<b>0.03 (0.97)</b>	55.23 (58.00)	<b>0.36 (0.29)</b>	0.05 (1.14)	20.23 (3.80)	0.00 (0.00)	0.04 (1.10)	0.23 (0.02)	<b>0.03 (0.97)</b>	<b>0.36 (0.30)</b>
5	<b>0.01 (0.65)</b>	69.38 (62.68)	0.69 (0.78)	0.04 (1.14)	20.19 (23.92)	0.00 (0.00)	0.02 (0.91)	0.61 (0.15)	<b>0.01 (0.65)</b>	<b>0.76 (0.87)</b>
10	<b>0.00 (0.52)</b>	55.98 (63.95)	0.93 (0.86)	0.044 (1.10)	21.25 (18.26)	0.00 (0.02)	0.01 (0.69)	0.70 (0.31)	<b>0.00 (0.52)</b>	<b>0.99 (0.90)</b>

Table 1. Comparison results on OASIS data using SDT (Kendall shape space) features

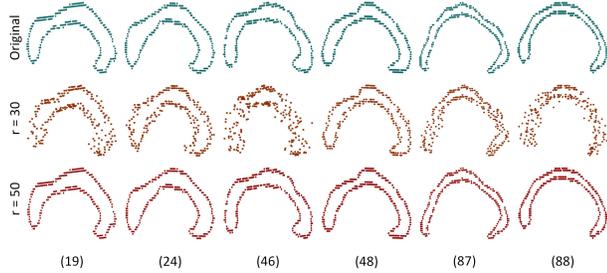


Figure 3. Reconstruction of OASIS data using sparse PGs (the figures depict CC boundary as a point cloud)

$r$	SEPGA			sPGA			PGA		ePGA	
	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$evar$	$e_1$	$evar$
1	<b>0.09</b>	59.34	<b>0.19</b>	0.11	43.34	0.00	0.10	0.17	<b>0.09</b>	<b>0.19</b>
5	<b>0.06</b>	63.84	<b>0.51</b>	0.11	50.12	0.00	<b>0.06</b>	0.47	<b>0.06</b>	<b>0.51</b>
10	<b>0.04</b>	65.20	0.68	0.11	48.92	0.00	0.05	0.64	<b>0.04</b>	<b>0.70</b>

Table 2. Comparison results on Gator Bait data using SDT features (on  $\mathbf{S}^{29999}$ )

### 3.2. Comparative performance of SEPGA on Gator Bait data [20]

In this section, we performed principal geodesic analysis on Gator Bait data [20]. This dataset is a 2D point set data of 100 fish images with varying subspecies. From each image, we extract the SDT feature [7] to map each data set on to  $\mathbf{S}^{29999}$ . The comparative performance of SEPGA with sPGA, PGA and ePGA is reported in Table 2.

From the results reported in Table 2, we can see that SEPGA and ePGA perform equally well in terms of  $e_1$  and  $evar$  respectively. In terms of sparsity, though sPGA gives sparse principal vectors, it fails to capture the principal direction along which the data variance is maximized, as is evident from the  $evar$  value.

### 3.3. Comparative performance of SEPGA on Yale face data [17]

In this section, we used data from the *YaleExtendedB* [17] face recognition dataset to perform the principal geodesic analysis. This data contains 16128 face images taken from 28 human subjects with varying pose and illumination conditions. This data is suited for classification purposes. We performed PGA for each human subject after extracting features from his/ her face images. Now, we measure the average reconstruction error within a class. A small average within class reconstruction error implies a good classification using principal geodesics as features, as shown in [5]. For person  $i$ , construct the SIFT descriptor

matrix,  $\mathcal{F}^{ij}$  [18] with his/ her  $j^{th}$  face image. We compute the covariance matrix  $\mathcal{C}^{ij}$  with the first  $d$  principal vectors of  $\mathcal{F}^{ij}$ .  $\mathcal{C}^{ij}$  is a  $d \times d$  matrix which is symmetric and positive semidefinite. We add  $\epsilon I_d$  to this matrix with small  $\epsilon > 0$  to make each  $\mathcal{C}^{ij}$  positive definite. Now, from this data, we choose  $n = 125$ ,  $d = 25$  to map each face image on to the  $\text{SPD}(25)$  manifold. Here, we performed principal geodesic analysis for each subject and reported the average performance.

$r$	SEPGA			sPGA			PGA	
	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$evar$
1	<b>0.00</b>	79.34	<b>0.69</b>	<b>0.00</b>	96.16	0.00	<b>0.00</b>	0.53
5	<b>0.00</b>	81.54	<b>0.78</b>	<b>0.00</b>	96.54	0.00	<b>0.00</b>	0.65
10	<b>0.00</b>	80.91	<b>0.98</b>	<b>0.00</b>	97.57	0.00	<b>0.00</b>	0.73

Table 3. Comparison results on YaleExtendedB data (on  $\text{SPD}(25)$ )

In this experiment, we could not compare with ePGA since  $\text{SPD}(m)$  is not a constant curvature manifold. From the results reported in Table 3, we can see that the average reconstruction error using only the first principal component is approximately zero which suggests that one principal component is sufficient to almost exactly reconstruct the images of a person. This is true for SEPGA, sPGA and PGA. But, SEPGA achieves this almost perfect reconstruction using only 21% of the original dimension which is 325 in this case! This indicates a significant amount of savings in storage which is a consequence of the sparsity in our SEPGA algorithm making it more attractive for use than the competing PGA methods. This highly sparse representation clearly gives evidence of a very ‘‘tight cluster’’ structure, which yields good classification accuracy, as was also reported in [5]. sPGA gives better sparse principal vectors, but it performs poorly in terms of expressed variance.

### 3.4. Comparative performance of SEPGA on KTH action recognition data [22]

In this section, we continue with our demonstration of the benefits of SEPGA in comparison to sPGA and PGA on the *KTH* action recognition data sets [22]. This dataset contains 2200 action videos taken from 25 human subjects performing 6 actions in 4 scenarios. Similar to subsection 3.3, we compute PGs for a fixed action and then report the average error, sparsity and expressed variance. From each video, we extract 25 frames and compute the HOG features [6] from each frame. Here, for  $i^{th}$  action and  $j^{th}$  video of this action sequence, we map the data on to the  $\text{SPD}(25)$

manifold. The results are reported in Table 4.

$r$	SEPGA			sPGA			PGA	
	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$evar$
1	<b>0.00</b>	27.38	<b>0.76</b>	0.35	16.00	0.00	0.34	0.63
5	<b>0.00</b>	71.59	<b>0.89</b>	0.42	60.16	0.00	0.34	0.65
10	<b>0.00</b>	70.15	<b>0.98</b>	0.53	70.32	0.00	0.30	0.73

Table 4. Comparison results on KTH action data (on SPD(25))

From the results, we can see that with just the first principal component, SEPGA yields perfect reconstruction while maintaining an expressed variance of 0.76. Both PGA and sPGA perform poorly in terms of the reconstruction error. sPGA gives comparable sparsity but does not yield a good  $evar$ . So, as before, these results also suggest the usefulness of SEPGA which yields high level of sparsity without affecting either the reconstruction error or the expressed variance.

### 3.5. Comparative performance of SEPGA on HCP data [27]

In this section, we report performance results on the Human Connectome Project (HCP) data. Here all the subjects in the cohort were scanned on a dedicated 3 Tesla (3T) scanner. We analyzed the high-quality curated diffusion MR imaging data (publicly available) obtained from over 840 healthy adults from the WU-Minn consortium [27]. We obtained diffusion tensor images (DTI) from the dMRI data by non-linear fitting of the tensors to the diffusion weighted ( $b = 1000$  s/mm<sup>2</sup>) images. These DTI images were spatially normalized using DTI-TK [29] which is a non-linear diffeomorphic registration and template estimation pipeline that can directly operate on the diffusion tensors using a log-Euclidean framework. Seventeen major white matter pathways were obtained by registering the publicly available IIT white matter atlas [28] to the HCP template using the ANTS software [2]. We analyzed data from the fornix and cingulum bundles which form a region of interest (ROI) of 228 voxels. From each voxel we extracted  $3 \times 3$  SPD matrices to obtain a product manifold of SPD(3) matrices of dimension ( $228 \times 6 = 1368$ ).

$r$	SEPGA			sPGA			PGA	
	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$evar$
1	<b>2.92</b>	28.96	<b>0.52</b>	4.68	10.53	0.29	4.68	0.29
3	<b>1.43</b>	59.23	<b>0.82</b>	2.90	34.12	0.51	2.88	0.52
5	<b>0.28</b>	74.37	<b>0.95</b>	1.01	49.08	0.72	1.02	0.77

Table 5. Comparison results on HCP data.

The results are reported in Table 5, which shows better performance of SEPGA over its competitors.

### 3.6. Comparative performance of SEPGA and sPGA on Synthetic Data

In section 1, we described a simple extension of sPCA to Riemannian manifolds by doing sPCA on  $T_\mu \mathcal{M}$ . Though this formulation is easier to solve, it has the same disadvantage as PGA, i.e., on data with large variance the perfor-

mance is lacking. In this section, we compare the performance of SEPGA with sPGA on synthetic data to explicitly demonstrate this lacking behavior of sPGA.

$r$	var.	SEPGA			sPGA		
		$e_1$	$\varsigma(\%)$	$evar$	$e_1$	$\varsigma(\%)$	$evar$
2	0.006	<b>0.00</b>	50.00	<b>0.99</b>	<b>0.00</b>	66.67	0.51
2	0.718	<b>0.04</b>	50.00	<b>0.98</b>	0.11	50.00	0.13

Table 6. Comparative performance on synthetic data (on  $\mathbf{S}^2$ ).

In Table 6, we report comparative results of SEPGA vs. sPGA on two synthetic datasets on  $\mathbf{S}^2$ . As we seek only 2 principal components on  $\mathbf{S}^2$ , i.e.,  $r = 2$ , the maximum achievable sparsity is 50%, i.e., 2 out of 4 components are nonzero (exactly 1) for SEPGA. For, sPGA, the maximum achievable sparsity is 66.67% (4 out of 6 components are 0), since  $\iota_\mu$  is the identity map. For data with small variance both SEPGA and sPGA achieve maximum sparsity, while SEPGA gives almost perfect expressed variance. But with a moderately large data variance, equal to 0.718, sPGA yields a poor reconstruction error  $e_1$ , and SEPGA achieves much higher sparsity in comparison to sPGA. Note that, here the value of  $\varsigma$  is same for both sPGA and SEPGA, but for SEPGA the maximum achievable sparsity is 50%, while for sPGA it is 66.67%.

## 4. Conclusions

In this paper, we presented a novel algorithm, called SEPGA, for performing sparse exact PGA using the projection error cost function optimization proposed in [25]. Sparsity in the context of SEPGA has the same advantages over PGA as does the vector space version namely, SPCA over PCA. More explicitly, SEPGA provides a sparse representation of the principal components computed. This sparsity has the advantage that it provides the most “natural” basis in the principal submanifold to represent the data. Moreover, SEPGA overcomes the drawbacks of the linear approximation in PGA. Through an extensive set of experiments, we demonstrated the performance of SEPGA in comparison to PGA and ePGA. SEPGA achieves nearly same or better accuracy as the competing methods but with a high level of sparsity. We also proposed a sparse version of the PGA algorithm, namely the sPGA, which performs comparably to SEPGA for data with a small variance. However, for moderate data variance, sPGA exhibits the same lacking behavior as that of PGA.

**Acknowledgements:** This research was funded in part by the NSF grants IIS-1525431 and IIS-1724174 to BCV.

## References

- [1] B. Afsari. Riemannian  $L^p$  center of mass: Existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139(2):655–673, 2011.
- [2] B. B. Avants, N. Tustison, and G. Song. Advanced normalization tools (ANTS). *Insight J*, 2:1–35, 2009.

- [3] R. Chakraborty. <http://www.cise.ufl.edu/~rudrasis/>, 2013.
- [4] R. Chakraborty, D. Seo, and B. C. Vemuri. An efficient exact-pga algorithm for constant curvature manifolds. <http://arxiv.org/abs/1603.03984/>, 2016.
- [5] R. Chakraborty and B. C. Vemuri. Recursive frechet mean computation on the grassmannian and its applications to computer vision. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.
- [7] Y. Deng, A. Rangarajan, et al. A Riemannian framework for matching point clouds represented by the Schrodinger distance transform. In *IEEE CVPR*, pages 3756–3761, 2014.
- [8] M. P. do Carmo Valero. *Riemannian geometry*. 1992.
- [9] P. Fletcher, C. Lu, et al. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE TMI*, pages 995–1005, 2004.
- [10] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'institut Henri Poincaré*, pages 215–310, 1948.
- [11] S. Hauberg. Principal curves on riemannian manifolds. 2015.
- [12] S. Huckemann, T. Hotz, and A. Munk. Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, 2010.
- [13] S. Huckemann and H. Ziezold. Principal component analysis for riemannian manifolds, with an application to triangular shape spaces. *Advances in Applied Probability*, pages 299–319, 2006.
- [14] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [15] I. T. Jolliffe. Rotation of principal components: choice of normalization constraints. *Journal of Applied Statistics*, 22(1):29–35, 1995.
- [16] D. Kendall. A survey of the statistical theory of shape. *Stat. Science*, pages 87–99, 1989.
- [17] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE TPAMI*, 27(5):684–698, 2005.
- [18] D. G. Lowe. Object recognition from local scale-invariant features. In *CVPR*, volume 2, pages 1150–1157, 1999.
- [19] D. S. Marcus, T. H. Wang, et al. OASIS: cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, pages 1498–1507, 2007.
- [20] A. Rangarajan. <https://www.cise.ufl.edu/~anand/publications.html>.
- [21] S. Said, N. Courty, N. Le Bihan, and S. J. Sangwine. Exact principal geodesic analysis for data on so (3). In *15th European Signal Processing Conference (EUSIPCO-2007)*, pages 1700–1705. EURASIP, 2007.
- [22] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36, 2004.
- [23] K. Sjöstrand, L. H. Clemmensen, R. Larsen, and B. Ersbøll. Spasm: A matlab toolbox for sparse statistical modeling. *Journal of Statistical Software Accepted for publication*, 2012.
- [24] S. Sommer. <https://github.com/nefan/smanifold>, 2014.
- [25] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In *Computer Vision–ECCV 2010*, pages 43–56. Springer, 2010.
- [26] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
- [27] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.
- [28] A. Varentsova, S. Zhang, and K. Arfanakis. Development of a high angular resolution diffusion imaging human brain template. *NeuroImage*, 91:177–186, 2014.
- [29] H. Zhang, P. A. Yushkevich, D. C. Alexander, et al. Deformable registration of diffusion tensor MR images with explicit orientation optimization. *MIA*, 10(5):764–785, 2006.
- [30] M. Zhang and P. T. Fletcher. Probabilistic principal geodesic analysis. In *Advances in Neural Information Processing Systems*, pages 1178–1186, 2013.
- [31] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.