Dimensionality Reduction on the SPD Manifold: A Comparative Study of Linear and Non-Linear Methods

Amal Araoud[1], Enjie Ghorbel[1,2][a] and Faouzi Ghorbel[1][b]

[1]*National School of Computer Science (ENSI), CRISTAL Laboratory, GRIFT Group, Manouba University, Tunisia*

[2]*Interdisciplinary Centre of Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg*

Keywords: Riemannian Manifolds, Riemannian Geometry, Symmetric Positive Definite (SPD) Matrices, Dimensionality Reduction, Non-Euclidean Geometry.

Abstract: The representation of visual data using Symmetric Positive Definite (SPD) matrices has proven effective in numerous computer vision applications. Nevertheless,the non-Euclidean nature of the SPD space poses a challenge, especially when dealing with high-dimensional data. Conventional dimensionality reduction methods have been typically designed for data lying in linear spaces, rendering them theoretically unsuitable for SPD matrices. For that reason, considerable efforts have been made to adapt these methods to the SPD space by leveraging its Riemannian structure. Despite these advances, a systematic comparison of conventional, i.e., linear and revisited, i.e., non-linear dimensionality reduction methods applied to SPD data according to their distribution remains lacking. In fact, while geometry-aware dimensionality reduction methods are highly relevant, the convexity of the SPD space may hinder their performance. This study addresses this gap by evaluating the performance of both linear and non-linear dimensionality reduction techniques within a binary classification scenario. For that purpose, a synthetically generated dataset exhibiting different class distribution configurations (distant, slight overlap, strong overlap) is used. The obtained results suggest that non-linear methods offer limited advantages over linear approaches. According to our analysis, this outcome may be attributed to two primary factors: the convexity of the SPD space and numerical issues.

1 INTRODUCTION

Symmetric Positive Definite (SPD) matrices are nonlinear mathematical entities that have shown great potential in the field of computer vision (Pennec et al., 2006; Tuzel et al., 2006; Harandi et al., 2012; Jayasumana et al., 2015). They have been used as representations for several visual classification tasks such as image classification (Chen et al., 2020) and action recognition (Ghorbel et al., 2018). Nonetheless, handling high-dimensional SPD matrices is tricky, as it induces a high computational complexity. To handle this issue, dimensionality reduction methods which aim at projecting high-dimensional data into a lower-dimensional space while preserving essential information might be employed. Conventional methods such as Principal Component Analysis (PCA) (Hotelling, 1933) are mainly linear, which means that they have been introduced for data lying in linear spaces. Although the space of SPD matrices is known to be non-Euclidean, conventional methods can be practically by flattening SPD matrices. However, such a process has been widely criticized in the literature as it is not theoretically sound (Harandi et al., 2018; Pennec et al., 2006; Tuzel et al., 2008; Jayasumana et al., 2015). Indeed, this would contribute to breaking the geometric structure of SPD matrices, potentially resulting to a physically implausible reduction, i.e., lower-dimensional matrices that are not SPD. To address this issue, probabilistic dimensionality reduction methods leveraging advanced distance measures have demonstrated improved classification accuracy compared to traditional approaches (Drira et al., 2012).

Moreover, recent advances in differential geometry have led to the development of specialized dimensionality reduction techniques, which account for the Riemannian structure of SPD (Harandi et al., 2018; Fletcher et al., 2004).These methods have shown promise in preserving the manifold intrinsic geometry while effectively reducing the dimension of SPD matrices. However, a comprehensive comparative analy-

[a] https://orcid.org/0000-0002-6878-0141

[b] https://orcid.org/0000-0002-6364-1089

Araoud, A., Ghorbel, E. and Ghorbel, F.
Dimensionality Reduction on the SPD Manifold: A Comparative Study of Linear and Non-Linear Methods.
DOI: 10.5220/0013183500003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 805-812
ISBN: 978-989-758-737-5; ISSN: 2184-433X

sis between traditional linear methods and non-linear approaches in the context of SPD matrices remains underexplored. Despite the theoretical soundness of non-linear methods, their effectiveness as compared to linear approaches is not guaranteed as numerical issues may arise. Furthermore, understanding the classification performance of these techniques under different data distribution configurations is of wide interest.

In this paper, the primary objective is therefore to compare the performance of both linear and non-linear dimensionality reduction techniques for data on the SPD manifold. Specifically, we evaluate two linear and two non-linear methods, namely, the classical PCA, a standard Convolution Neural Network (CNN)-based Autoencoder, Tangent PCA, and Harandi's method (Hotelling, 1933; Wang et al., 2023; Fletcher et al., 2004; Harandi et al., 2018), respectively, using synthetically generated data in a binary classification setup. Moreover, we consider different label-related distribution configurations, ranging from well-separated to strongly intertwined distributions.

Through this comparative analysis, we aim to provide deeper insights into the strengths and limitations of linear and non-linear approaches, contributing to a broader understanding of how dimensionality reduction techniques can be adapted to non-Euclidean spaces. Our findings are intended to inform the selection of appropriate methods for high-dimensional data on Riemannian manifolds, particularly in scenarios where SPD matrices play a central role.

Our findings suggest that linear and non-linear dimensionality reduction methods yield comparable performance on SPD matrices. This might be explained by two facts, namely, (1) **the convexity of the SPD cone**: the regions corresponding to SPD matrices with a minimum eigen value largely greater (interior of the cone) are not highly impacted by non-linearity, and (2) **the existence of numerical issues**: the regions that are the most impacted by the non-linearity are the ones close to the cone boundaries, representing the matrices with a minimum value close to 0. However, it is known that such SPD matrices are often subject to numerical issues when applying the logarithmic map (Ghorbel et al., 2018). This highlights the need for considering the convex structure of SPD matrices as well as potential numerical issues in non-linear dimensionality reduction techniques.

The remainder of this paper is organized as follows: **Section 2** discusses the mathematical preliminaries, focusing on the geometry of SPD manifolds and key operations like the exponential and logarithmic maps. **Section 3** details the proposed evaluation protocol, namely, the tested dimensionality reduction techniques and the considered data distribution configurations. **Section 4** details the experimental results and analysis, while **Section 5** concludes this work with a summary of findings and potential directions for future research.

2 PRELIMINARIES: THE RIEMANNIAN SPACE OF SPD MATRICES

A real symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is said to be Symmetric Positive Definite (SPD) if for all non-zero vectors $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^\top \mathbf{A}\mathbf{v} > 0$, implying that all eigenvalues of $\mathbf{A}$ are strictly positive. The space of SPD matrices denoted as $\mathcal{S}_n^{++}$ is therefore composed of $n \times n$ SPD matrices. Hence, the space $\mathcal{S}_n^{++}$ is non-linear and forms the interior of convex cone in the $\frac{n(n+1)}{2}$-dimensional Euclidean space delimited by the symmetric semi-positive definite matrices. As a consequence, traditional linear methods for operations such as averaging, classification, and dimensionality reduction are unsuitable. To account for this, the space $\mathcal{S}_n^{++}$ is mostly endowed with a Riemannian metric, resulting in a Riemmanian manifold. The latter is a differentiable manifold, equipped with a smoothly varying inner product on each tangent space. The tangent space at any point on the manifold consists of the set vectors tangent to all possible curves passing through that point. The Riemannian metric enables defining key geometric notions such as angles between curves and the lengths of curves.

In this section, we review several key concepts related to operations on the space of SPD matrices. We start by giving the logarithmic and exponential maps of the SPD space. Popular metrics for SPD matrices, such as the Affine-Invariant Riemannian Metric (AIRM) and the Log-Euclidean Metric are then recalled. Finally, we review the concept of the Frechet mean, which generalizes the notion of average to curved spaces.

2.1 Logarithmic and Exponential Maps on the SPD Manifold

In Riemannian geometry, the logarithmic and the exponential maps are use to map non-Euclidean SPD matrices to its tangent space and vice versa. Hence, this enables performing linear operations in the tangent space before projecting the results back onto the manifold.

Given an SPD matrix $\mathbf{P} \in \mathcal{S}_n^{++}$ and a reference point $\mathbf{X} \in \mathcal{S}_n^{++}$, the logarithmic map $\log_{\mathbf{X}} : \mathcal{S}_n^{++} \rightarrow$

$T_{\mathbf{X}}\mathcal{S}_n^{++}$ project an SPD matrix to its tangent space $T_{\mathbf{X}}\mathcal{S}_n^{++}$ at the point $\mathbf{X}$ as follows,

$$\log_{\mathbf{X}}(\mathbf{P}) = \mathbf{X}^{1/2}\log(\mathbf{X}^{-1/2}\mathbf{P}\mathbf{X}^{-1/2})\mathbf{X}^{1/2}, \quad (1)$$

where $\log(\cdot)$ denotes the matrix logarithm.

Conversely, the exponential map $\exp_{\mathbf{X}} : T_{\mathbf{X}}\mathcal{S}_n^{++} \to \mathcal{S}_n^{++}$ is defined to map a tangent vector at $\mathbf{X}$ back to the manifold as follows

$$\exp_{\mathbf{X}}(\mathbf{V}) = \mathbf{X}^{1/2}\exp(\mathbf{X}^{-1/2}\mathbf{V}\mathbf{X}^{-1/2})\mathbf{X}^{1/2}, \quad (2)$$

where $\mathbf{V} \in T_{\mathbf{X}}\mathcal{S}_n^{++}$ is the tangent vector, and $\exp(\cdot)$ denotes the matrix exponential.

2.2 Metrics and Divergences on SPD Manifolds

Several distance metrics have been developed for the SPD manifold. Each of these metrics respects the non-Euclidean structure of the manifold and is suited for different computational and statistical tasks. The most commonly used metrics include:

- **The Affine-Invariant Riemannian Metric (AIRM):** introduced in (Pennec et al., 2006). The AIRM computes the distance between two SPD matrices **P** and **Q** as follows,

$$d_{\text{AIRM}}(\mathbf{P},\mathbf{Q}) = \|\log(\mathbf{P}^{-1/2}\mathbf{Q}\mathbf{P}^{-1/2})\|_F \quad (3)$$

  where $\|\cdot\|_F$ denotes the Frobenius norm. One of the appealing properties of AIRM is its invariance under affine transformations, making it robust in applications where invariance to scaling or linear transformations is important.

- **The Stein Divergence:** has been proposed in (Cherian et al., 2013) and is computed between two SPD matrices **P** and **Q** as follows,

$$d_{\text{Stein}}(\mathbf{P},\mathbf{Q}) = \log\det\left(\frac{\mathbf{P}+\mathbf{Q}}{2}\right) - \frac{1}{2}\log\det(\mathbf{P}\mathbf{Q}) \quad (4)$$

  , This divergence is suitable for large-scale problems where computational efficiency is crucial.

- **Jeffrey's Divergence:** It is another useful divergence measure for SPD matrices and is defined between two SPD matrices **P** and **Q** as follows as,

$$d_{\text{Jeffrey}}(\mathbf{P},\mathbf{Q}) = \frac{1}{2}\left(\text{tr}(\mathbf{P}^{-1}\mathbf{Q}) + \text{tr}(\mathbf{Q}^{-1}\mathbf{P})\right) - n, \quad (5)$$

  where $\text{tr}(\cdot)$ is the trace operator and n is the dimensionality of the SPD matrices. Unlike AIRM, it may not always capture the full geometry of the SPD manifold, but it is computationally attractive for certain applications.

- **Log-Euclidean Metric:**
  This metric simplifies computations by treating the manifold of SPD matrices as a flat space after applying the matrix logarithm. The distance between two SPD matrices **P** and **Q** is given by,

$$d_{\text{LEM}}(\mathbf{P},\mathbf{Q}) = \|\log(\mathbf{P}) - \log(\mathbf{Q})\|_F, \quad (6)$$

  where $\|\cdot\|_F$ denotes the Frobenius norm.
  While LEM is computationally simpler and faster to compute than AIRM, it may not preserve certain affine-invariant properties, making it less suitable for tasks requiring such invariance (Arsigny et al., 2007).

2.3 Frechet Mean on the SPD Manifold

The Frechet mean, introduced by Frechet (Fréchet, 1948), generalizes the concept of averaging from Euclidean spaces to Riemannian manifolds. For a set of symmetric positive definite (SPD) matrices $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k\}$, the Frechet mean $\mathbf{P}_\mu$ is defined as the matrix that minimizes the expected sum of squared distances under a specified metric, such as the Affine-Invariant Riemannian Metric (AIRM).

Mathematically, the Frechet mean can be expressed as,

$$\mathbf{P}_\mu = \arg\min_{\mathbf{P}} \sum_{i=1}^{k} d(\mathbf{P},\mathbf{P}_i)^2, \quad (7)$$

where $d(\cdot,\cdot)$ denotes the geodesic distance between SPD matrices. The computation of $\mathbf{P}_\mu$ typically relies on iterative optimization techniques, like gradient descent.

This intrinsic mean provides a robust statistical measure within the context of SPD matrices, effectively capturing the manifold geometric structure.

3 PROPOSED EVALUATION PROTOCOL

This section outlines the proposed evaluation protocol. We begin by reviewing the linear and non-linear dimensionality reduction methods considered in this work, followed by a detailed description of the various data distribution configurations employed in our experiments.

3.1 Linear Dimensionality Reduction Methods

3.1.1 Classical PCA

The traditional PCA approach involves the following steps:

- **Step 1 (Centering the Data).** The first step is to center the data by calculating the mean μ of the dataset. Then, for each data point $\mathbf{X}_i$, subtract the mean μ to center the data around the origin as follows,
$$\mathbf{X}'_i = \mathbf{X}_i - \mu. \tag{8}$$
- **Step 2 (Calculate the Covariance Matrix of the Centered Data).** The covariance $\mathbf{C}$ of the centered data is computed as follows,
$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}'_i(\mathbf{X}'_i)^\top, \tag{9}$$
where N is the number of data points.
- **Step 3 (Eigen Decomposition).** Perform eigen decomposition of the covariance matrix $\mathbf{C}$ to obtain the principal components. The covariance matrix $\mathbf{C}$ can be decomposed as,
$$\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{U}^\top, \tag{10}$$
where $\mathbf{U}$ contains the eigenvectors (principal directions), and $\mathbf{S}$ contains the eigenvalues.
- **Step 4 (Data Projection).** Step 4 (Data projection): Projecting the Data $\mathbf{X}'_i$ after obtaining the principal components onto the principal directions as described below,
$$\mathbf{Y}_i = \mathbf{V}^\top\mathbf{X}'_i \tag{11}$$
where $\mathbf{Y}_i$ represents the data in the new reduced space. In our experiments, the SPD matrices are flattened to match the linear requirements of the traditional PCA.

3.2 Autoencoder on SPD Manifolds

As an alternative to PCA, a deep-learning based strategy called auto-encoder has been introduced in (Vincent et al., 2008) It aims at learning a lower-dimensional latent representation by first encoding the data in a lower dimensional space and then reconstructing it through a decoder. The parameters of the encoder and decoder are then learned by optimizing a reconstruction error.

Autoencoder Architecture. The architecture that is used in this paper is designed as follows (Hinton and Salakhutdinov, 2006):

- **Input Layer.** Takes flattened SPD matrices as input vectors. In our experiments, we consider only 3x3 matrices. Hence, the dimension of the input vector is equal to 9.
- **Encoding Layers.** Four consecutive layers compress the input vectors down to the target dimension of 2. This is done progressively, with dimensions reducing from $9 \to 8 \to 6 \to 4 \to 2$. A ReLU activation function is used at each layer to introduce non-linearity.
- **Decoding Layers.** The decoding layers symmetrically reconstruct the compressed data back to its original dimension. The process involves increasing the data from $2 \to 4 \to 6 \to 8 \to 9$ dimensions without using an activation function at the output layer.

Training Process. The Autoencoder is trained using the Mean Squared Error (MSE) as loss function, which measures the reconstruction error. The Adam optimizer is applied, and the network is trained over 100 epochs with a batch size of 32.

Dimensionality Reduction Results. After training, the encoder part of the network is used to reduce the SPD matrices to two dimensions. This compressed representation forms the input for subsequent classification tasks.

3.3 Non-Linear Dimensionality Reduction Methods

3.3.1 Tangent PCA

Classical PCA does not account for the non-linear geometry of the SPD space. Tangent PCA (tPCA) (Fletcher and Joshi, 2004) addresses this limitation by projecting data onto the tangent space of the manifold at a reference point, typically the Frechet mean. The tangent space serves as a linear approximation of the manifold around this reference point, enabling the application of standard linear techniques like PCA in this locally flat space. The results are then interpreted in the context of the original manifold. Below, we detail the steps for implementing tPCA for Symmetric Positive Definite (SPD) matrices:

- **Step 1 (Compute the Frechet Mean).** Given a set of SPD matrices $\{X_i\}_{i=1}^{N}$, compute the Frechet mean μ using the iterative algorithm described earlier.
- **Step 2 (Map to Tangent Space).** For each SPD matrix X_i, map it to the tangent space at μ using the logarithmic map: $Y_i = \log(\mu^{-1/2}X_i\mu^{-1/2})$.

- **Step 3 (Apply PCA).** Perform PCA on the set of tangent vectors $\{Y_i\}_{i=1}^{N}$ to obtain the principal components and reduce the dimensionality.
- **Step 4 (Map Back to the Manifold).** Project the reduced data back to the SPD manifold using the exponential map: $Z_i = \mu^{1/2} \exp(Y_i) \mu^{1/2}$.

3.3.2 Harandi's Method

In our work, we employ the supervised dimensionality reduction method proposed by Harandi et al. (Harandi et al., 2018), which projects high-dimensional SPD matrices onto a lower-dimensional SPD manifold while preserving class-specific structures. This approach learns a transformation matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$, where $m < n$, to map each $\mathbf{X} \in \mathcal{S}_{++}^{n}$ into a lower-dimensional SPD matrix $\mathbf{W}^T\mathbf{X}\mathbf{W} \in \mathcal{S}_{++}^{m}$. The lower-dimensional representations retains critical information for classification tasks. The mapping function $f_{\mathbf{W}}(X) = \mathbf{W}^T\mathbf{X}\mathbf{W}$ ensures that the transformed matrices remain SPD, while the optimization process aims to minimize intra-class distances and maximize inter-class distances. To achieve this, affinity functions $g_w(\mathbf{X}_i, \mathbf{X}_j)$ (for within-class similarity) and $g_b(X_i, X_j)$ (for between-class dissimilarity) are used to define an overall affinity function $a(\mathbf{X}_i, \mathbf{X}_j) = g_w(\mathbf{X}_i, \mathbf{X}_j) - g_b(\mathbf{X}_i, \mathbf{X}_j)$. This affinity function drives the loss function $L(\mathbf{W})$, which is optimized over the Grassmanian manifold $G(m,n)$, ensuring that $\mathbf{W}$ satisfies the unitary constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}_m$, preserving full-rank properties. The distances between SPD matrices are measured using metrics such as the Affine-Invariant Riemannian Metric (AIRM), Stein divergence, or Jeffrey divergence.

3.4 Data Generation Methodology

SPD matrices are generated through a controlled process that begins by sampling points from a uniform distribution within the tangent space of the SPD manifold. The tangent space, being Euclidean, allows for easier manipulation of data before projecting it back onto the SPD manifold using exponential map. This ensures the resulting symmetric matrices are valid SPD matrices, adhering to the geometric constraints of the manifold. The process is mathematically defined as,

$$\mathbf{A}_i = \exp(\mathbf{S}_i), \tag{12}$$

where

$$\mathbf{S}_i = \frac{1}{2}(\mathbf{M} + \mathbf{M}^T), \quad \text{and} \quad \mathbf{M}_{i,j} \sim \mathcal{U}(a,b). \tag{13}$$

Here, $\mathbf{S}_i$ is a symmetric matrix sampled from a uniform distribution between bounds a and b, and exp denotes the matrix exponential. By varying the bounds a and b, we control the variability and distribution of the generated matrices.

Configurations of SPD Data Distributions. We define three distinct configurations to evaluate the methods under varying degrees of overlap between two distributions (each one corresponding to one class) by replacing a and b from Equation (13):

- **Case 1 - Distant Distributions.**

  ```
  - Distribution 1: a=1,b=2
  - Distribution 2: a=0,b=1
  ```

  The clusters of SPD matrices are well-separated, making this configuration ideal for testing dimensionality reduction in clear-cut classification scenarios.

- **Case 2 - Moderately Overlapping Distributions.**

  ```
  - Distribution 1:  a=0,b=1
  - Distribution 2:  a=0.25,b=1.25
  ```

  In this case, the distributions exhibit partial overlap, providing a moderately challenging classification task with reduced separability.

- **Case 3 - Highly Overlapping Distributions.**

  ```
  - Distribution 1:  a=0 , b=1
  - Distribution 2:  a=0.125 , b=1.125
  ```

  This setup represents the most complex scenario, with significant overlap between distributions, posing substantial challenges for dimensionality reduction and classification.

For each configuration, 1000 data points per distribution are generated, yielding a balanced dataset of 2000 matrices for training and evaluation. This ensures statistical significance and provides a comprehensive basis for assessing performance across different methods.

Data Visualization. To provide a more intuitive illustration of the data generation process, we visualize the three SPD matrix distributions after projecting them into a two-dimensional space as shown in Figure 1. The figure illustrates the degree of separation or overlap across the three configurations. In case 1, the clusters are clearly separated, while in the moderately overlapping and highly overlapping cases, the degree of entanglement becomes progressively more pronounced. These visualizations highlight how varying the configuration affects the separability of data, providing an intuitive understanding of the challenges posed by each scenario.

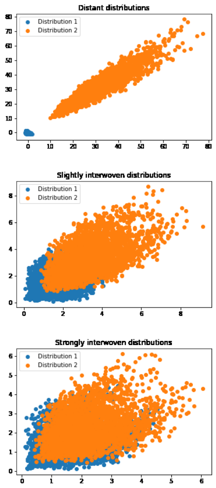

Figure 1: Visualization of the three distribution cases: (1) Distant, (2) Slightly interwoven, and (3) Strongly interwoven.

4 EXPERIMENTS AND RESULTS

4.1 Experimental Setup

This section outlines the experimental setup used to evaluate the classification performance of dimensionality reduction techniques under different configuration.

For classification, we employ the k-Nearest Neighbors (k-NN) algorithm on datasets after applying each dimensionality reduction technique. The classification performance is evaluated using accuracy, precision, recall, and F1-score with each experiment repeated 10 times to compute the mean and the standard deviation. The dataset is split into 80% for training and 20% for testing. We leverage the *Geomstats* library for the generation and manipulation of SPD matrices, which provides a robust framework for geometric learning and processing on Riemannian manifolds (Miolane et al., 2020). Additionally, for testing Harandi's method, we use the official code released by the authors (Harandi et al., 2018; Boumal et al., 2014).

5 RESULTS AND DISCUSSION

5.1 Method Comparison

Table 1 presents the classification performance of the tested methods under the three considered data distribution cases. Note that in this table, Harandi's approach is based on the Log-Euclidean distance. In case 1, where the distributions are distant, all methods achieve 100% accuracy. In case 2, the results are comparable across different methods where the distributions are slightly interwoven. Specifically, tangent PCA and the Autoencoder show slightly higher performance, with mean accuracies of 88.20% and 89.42%, respectively. On the other hand, classic PCA and Harandi's approach reach 88.03% and 86.69% of mean accuracies, respectively. This slight discrepancy indicates that, although all methods are effective, tangent PCA appears to slightly outperform other approaches. In case 3, where the distributions are strongly interwoven, the performance of the methods significantly decreases. Tangent PCA exhibits a mean accuracy of 69.33%, while Classic PCA, the Autoencoder and Harandi's method record mean accuracies of 69.40%, 70.80%, and 65.67%, respectively. Figure 2 shows the 2D visualization of the SPD matrices after applying the various dimensionality reduction techniques, demonstrating that the distribution configurations are preserved, highlighting the need for approaches that can better handle overlapping distributions.

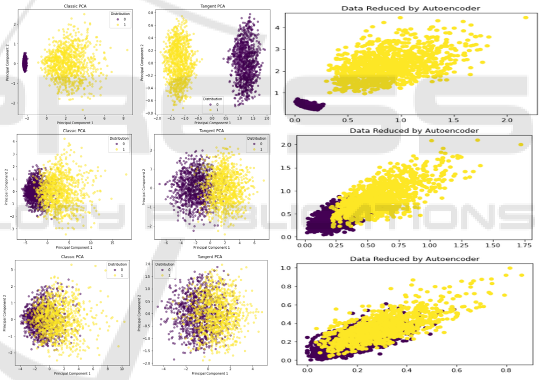

Figure 2: 2D Visualization of SPD matrices after dimensionality Reduction.

5.2 Impact of the Metrics on Harandi's Method

In Table 2, we present the results of using Harandi's method with different metrics across the three configuration cases. The values for AIRM, Stein, Jeffreys, Log-Euclidean, and Euclidean show a significant decrease in performance in Cases 2 and 3, highlighting the challenges associated with highly overlapping distributions. For instance, the AIRM value for Case 3 drops to 0.656, indicating a loss of information when projecting the data into tangent space. Overall, the results are stable for the different metrics.

Table 1: Comparison of Method Performance Across Different Cases.

Case	Method	Mean Accuracy (%)	Standard Deviation (σ)
Case 1: Distant Distributions	Tangent PCA	100.00	-
	Classic PCA	100.00	-
	Autoencoder	100.00	-
	Harandi's Method	100.00	-
Case 2: Slightly Interwoven	Tangent PCA	88.20	1.09
	Classic PCA	88.03	1.56
	Autoencoder	89.42	1.67
	Harandi's Method	86.69	1.86
Case 3: Strongly Interwoven	Tangent PCA	69.33	1.71
	Classic PCA	69.40	1.99
	Autoencoder	70.80	1.13
	Harandi's Method	65.67	3.10

Table 2: Performance of Harandi's Method Across Different Metrics.

Case	AIRM		Stein		Jeffreys		Log-Euclidean		Euclidean	
	Mean (%)	Std. Dev (σ)	Mean (%)	Std. Dev (σ)	Mean (%)	Std. Dev (σ)	Mean (%)	Std. Dev (σ)	Mean (%)	Std. Dev (σ)
Case 1	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
Case 2	86.56	1.47	**86.69**	1.86	86.66	1.48	86.03	1.60	85.74	1.61
Case 3	65.10	3.42	65.47	2.95	65.49	2.24	**65.67**	3.10	62.49	3.58

5.3 Classification Error According to the Eigenvalue Range

In this section, we analyze the classification error based on the minimal eigenvalue ranges of the Symmetric Positive Definite (SPD) matrices. The goal is to explore the relationship between the geometry of the SPD space, particularly the structure of its convex cone, and the performance of dimensionality reduction methods, specifically classic PCA and tangent PCA. By segmenting the data into minimal eigenvalue ranges, we aim to evaluate how these methods perform across different regions of the SPD space, particularly in areas closer to the cone boundaries (where non-linearity is more pronounced) versus more internal regions that are almost linear due to the convexity of the SPD space.

To perform this analysis, we first calculate the minimal eigenvalue of each SPD matrix, which indicates how ”close” a matrix is to the boundary of the convex cone. Matrices with smaller minimal eigenvalues are closer to the boundary, where non-Euclidean curvature is stronger. Conversely, matrices with a larger minimal eigenvalue reside in regions where the geometry of the space is locally Euclidean. Then, we report in Figure 3 a histogram including the classification error according to the range of minimum eigenvalues for case 2 and case 3.

The classification error is generally lower in the bins corresponding to the largest minimal eigenvalues, indicating that classic PCA performs better in regions of the cone where the geometry is almost linear. This might be explained by the fact that the space SPD is convex. But, surprisingly, classic PCA slightly outperforms tangent PCA in regions with very low minimal eigenvalues. This might be due to the fact that numerical issues may occur when computing the matrix logarithm of SPD matrices with very small eigenvalues. Tangent PCA, on the other hand, outperforms its classic counterpart for moderately low minimal eigenvalues.

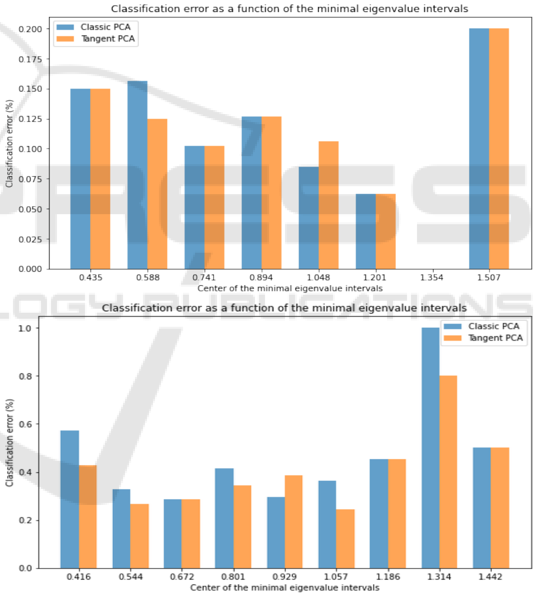

Figure 3: Classification Errors as a Function of Minimal Eigenvalues Intervals.

In conclusion, while classic PCA excels in certain scenarios, tangent PCA's performance is notably influenced by the geometric properties of the data. This study highlights the importance of understanding the underlying structure of data distributions when selecting dimensionality reduction techniques, particularly in complex scenarios where traditional methods may struggle.

6 CONCLUSION

In this study, we explored various dimensionality reduction techniques for Symmetric Positive Definite (SPD) matrices, including both linear and non-linear approaches. The results highlight the lack of robustness of existing methods in handling overlapping distributions in a classification context. Interestingly, linear and non-linear methods showed similar performance with SPD matrices. Two possible explanations could be: the convexity of the SPD space and the numerical issues raised by the logarithmic calculation.

In future work, a deeper analysis of these methods according to the local geometry of the SPD space is needed to discard or validate these hypotheses. Investigating dimensionality reduction in non-convex spaces is also extremely relevant. Finally, we aim to extend the dimensionality reduction methods for SPD matrices to more complex configurations, such as highly overlapping distributions.

REFERENCES

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2007). Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):328–347.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. (2014). Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459.

Chen, K.-X., Ren, J.-Y., Wu, X.-J., and Kittler, J. (2020). Covariance descriptors on a gaussian manifold and their application to image set classification. *Pattern Recognition*, 107:107463.

Cherian, A., Sra, S., Banerjee, A., and Papanikolopoulos, N. (2013). Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2161–2174.

Drira, W., Neji, W., and Ghorbel, F. (2012). Dimension reduction by an orthogonal series estimate of the probabilistic dependence measure. In *ICPRAM (1)*, pages 314–317.

Fletcher, P. T. and Joshi, S. (2004). Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In Sonka, M., Kakadiaris, I. A., and Kybic, J., editors, *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 87–98, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fletcher, P. T., Lu, C., Pizer, S. M., and Joshi, S. C. (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23:995–1005.

Fréchet, M. R. (1948). Les éléments aléatoires de nature quelconque dans un espace distancié.

Ghorbel, E., Boonaert, J., Boutteau, R., Lecoeuche, S., and Savatier, X. (2018). An extension of kernel learning methods using a modified log-euclidean distance for fast and accurate skeleton-based human action recognition. *Computer Vision and Image Understanding*, 175:32–43.

Harandi, M., Salzmann, M., and Hartley, R. (2018). Dimensionality reduction on spd manifolds: The emergence of geometry-aware methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):48–62.

Harandi, M. T., Sanderson, C., Hartley, R. I., and Lovell, B. C. (2012). Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *European Conference on Computer Vision*.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2015). Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(12):2464–2477.

Miolane, N., Brigant, A. L., Mathe, J., Hou, B., Guigui, N., Thanwerdas, Y., Heyder, S., Peltre, O., Koep, N., Zaatiti, H., Hajri, H., Cabanes, Y., Gerald, T., Chauchat, P., Shewmake, C., Kainz, B., Donnat, C., Holmes, S., and Pennec, X. (2020). Geomstats: A python package for riemannian geometry in machine learning.

Pennec, X., Fillard, P., and Ayache, N. (2006). A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66.

Tuzel, O., Porikli, F., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 589–600, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tuzel, O., Porikli, F., and Meer, P. (2008). Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1713–1727.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Wang, R., Wu, X.-J., Xu, T., Hu, C., and Kittler, J. (2023). U-spdnet: An spd manifold learning-based neural network for visual classification. *Neural Networks*, 161:382–396.