

## Article

# Dimensionality Reduction of SPD Data Based on Riemannian Manifold Tangent Spaces and Isometry

Wenxu Gao <sup>1</sup>, Zhengming Ma <sup>1,\*</sup>, Weichao Gan <sup>1</sup> and Shuyu Liu <sup>2</sup>

<sup>1</sup> School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou 510006, China; gaowx3@mail2.sysu.edu.cn (W.G.); ganwch5@mail2.sysu.edu.cn (W.G.)

<sup>2</sup> Public Experimental Teaching Center, Sun Yat-sen University, Guangzhou 510006, China; ljie@mail.sysu.edu.cn

\* Correspondence: isszm@mail.sysu.edu.cn

**Abstract:** Symmetric positive definite (SPD) data have become a hot topic in machine learning. Instead of a linear Euclidean space, SPD data generally lie on a nonlinear Riemannian manifold. To get over the problems caused by the high data dimensionality, dimensionality reduction (DR) is a key subject for SPD data, where bilinear transformation plays a vital role. Because linear operations are not supported in nonlinear spaces such as Riemannian manifolds, directly performing Euclidean DR methods on SPD matrices is inadequate and difficult in complex models and optimization. An SPD data DR method based on Riemannian manifold tangent spaces and global isometry (RMTSISOM-SPDDR) is proposed in this research. The main contributions are listed: (1) Any Riemannian manifold tangent space is a Hilbert space isomorphic to a Euclidean space. Particularly for SPD manifolds, tangent spaces consist of symmetric matrices, which can greatly preserve the form and attributes of original SPD data. For this reason, RMTSISOM-SPDDR transfers the bilinear transformation from manifolds to tangent spaces. (2) By *log* transformation, original SPD data are mapped to the tangent space at the identity matrix under the affine invariant Riemannian metric (AIRM). In this way, the geodesic distance between original data and the identity matrix is equal to the Euclidean distance between corresponding tangent vector and the origin. (3) The bilinear transformation is further determined by the isometric criterion guaranteeing the geodesic distance on high-dimensional SPD manifold as close as possible to the Euclidean distance in the tangent space of low-dimensional SPD manifold. Then, we use it for the DR of original SPD data. Experiments on five commonly used datasets show that RMTSISOM-SPDDR is superior to five advanced SPD data DR algorithms.



**Citation:** Gao, W.; Ma, Z.; Gan, W.; Liu, S. Dimensionality Reduction of SPD Data Based on Riemannian Manifold Tangent Spaces and Isometry. *Entropy* **2021**, *23*, 1117. <https://doi.org/10.3390/e23091117>

Academic Editor: Cristóbal López

Received: 7 July 2021

Accepted: 23 August 2021

Published: 27 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** dimensionality reduction; tangent space; symmetric positive; definite matrices; isometry

## 1. Introduction

The past few decades have witnessed the rapid development of machine vision, in which machine learning methods based on different mathematical platforms take various forms of image inputs. The most common form is the vector data flattened by a two-dimensional gray image or multi-channel color image [1–3]. This method is rough and simple on data processing and, although it has achieved a certain effect on some models, its problem is destroying the inherent position information and geometry structure of images. Another typical form is to keep the two-dimensional or higher-dimensional tensor data of images. There are lots of learning methods that take this data form as inputs [4–6] and the mathematical platform they all use involves tensor algebra. In addition, the convolutional neural network [7,8], as a branch of deep learning, has attained great achievements in processing tensor form for image classification. Though the position and statistics information of original image pixels are well maintained with tensor data as inputs, these methods cannot excavate the inherent geometric relations from the manifolds on which the data are located. On the other hand, taking the convolution neural network as an example, the excessively high computational complexity is also a drawback of the methods using tensor

data. The last form is symmetric positive definite (SPD) data. Due to their ability to capture higher-order statistical information and excellent feature expression capacity, SPD data have been widely concerned and applied. Examples of mature applications include joint covariance descriptors for image sets and video frames [9–11], region covariance matrices for posture analysis [12,13], pedestrian detection [14,15] and texture discrimination [16–18]. Although SPD data provide many advantages and good properties for machine vision tasks, the applications are faced with certain challenges:

1. SPD manifold is not a vector space and does not have the concepts of metric and inner product itself, so the traditional algorithms based on Euclidean space are not feasible here. It is not only unreasonable but also inadequate to apply the principles of Euclidean spaces to analyze SPD manifolds, for example, to directly use the Frobenius inner product to measure the similarity and dissimilarity between two SPD matrices.
2. If the original dimensionality of SPD data is relatively high, there must be lots of redundant information, which not only affects the calculation speed, but may also affect the discrimination effect. In addition, it is well known that high-dimensional spaces can lead to curse of dimensionality, which is embodied in the exponential increase in the geometric volume of the space as the dimensionality increases, making available data extremely sparse. Further, sparsity is a problem for any method that requires statistical significance. Moreover, the high-dimensional feature space is of little significance to the distance metric. Since most classifiers rely on Euclidean distance or Mahalanobis distance, classification performance decreases with the increase in dimensionality.

In order to overcome challenge (1), we can equip the SPD manifold with a Riemannian metric to explain the geometric properties of SPD matrices by the Riemannian geometry. The similarity and dissimilarity between SPD samples can be effectively calculated by the derived corresponding geodesic distance under the assigned Riemannian metric. Some outstanding studies [19–22] have achieved satisfactory results in this regard. Another treatment method is to embed SPD manifolds to reproducing kernel Hilbert spaces (RKHS) by kernel tricks, then adopt the kernel-based learning methods [21,23,24]. For problem (2), the general solution is dimensionality reduction (DR), or low-dimensional embedding.

This paper proposes a DR algorithm of SPD data based on Riemannian manifold tangent spaces and isometry (RMTSISOM-SPDDR). Different from existing supervised DR methods, our algorithm is unsupervised, which greatly saves the costs due to the fact that the labelling process for supervised learning methods is bound to consume time and effort. In summary, our contributions lie in the followings:

- (1) By using Riemannian manifold geometry, high-dimensional SPD data are mapped to the tangent space at the identity matrix, which makes linear operations possible and preserves the data form and attributes of original SPD data to the maximum extent.
- (2) We embedded the mapped data into a simple low-dimensional tangent space through bilinear transformation, so as to effectively avoid the problems caused by curse of dimensionality such as over-fitting and high calculation costs.
- (3) The bilinear transformation is determined by the proposed isometric criterion to maintain the distance relationship before and after DR. This is an unsupervised process, no manual labeling is required, which greatly saves labor and time costs.

The structure of this paper is as follows: In the Section 2, we introduce relevant preliminary knowledge needed for this paper, including notations, basic concepts of Riemannian manifolds. In the Section 3, some related works are briefly summarized. The Section 4 is our main part; in this section, we explain the proposal and solution of RMTSISOM-SPDDR in detail, with the algorithm flow and complexity analysis. The Section 5 mainly carries on comparative analysis of some advanced algorithms and introduces the connections and differences between them and RMTSISOM-SPDDR. In Section 6, sufficient experiments are conducted to verify the superiority of RMTSISOM-SPDDR. In Section 7, we make a summary of the whole paper.

## 2. Notations and Preliminaries

### 2.1. Notations

The basic notations in the paper are presented in Table 1.

**Table 1.** Notations and corresponding descriptions in this paper.

Notations	Descriptions
$T_p(M)$	The tangent space of the differential manifold $M$ at $p$
$G_p(M)$	The differentiable function germ of the differential manifold $M$ at $p$
$Sym_{++}^D$	The entire $D \times D$ SPD matrices
$Sym^D$	The entire $D \times D$ symmetric matrices
$X$	Capital letter denotes an SPD matrix
$\Phi$	Greek capital letter denotes a tangent vector of an SPD manifold

### 2.2. Preliminaries

Riemannian manifolds are based on the concepts of differential manifolds and tangent spaces. For the content of differential manifolds and tangent spaces, please refer to the Appendix A. A generalized differential manifold is a topological space, but there is no distance metric or linear operations between its elements. Tangent spaces of a differential manifold support linear operations, but lack of distance metric between tangent vectors. In order to support machine learning, the differential manifold must be equipped with a Riemannian metric to make it a Riemannian manifold.

The Riemannian metric is a symmetric, positive definite, smooth second order tensor field of a differential manifold. Let  $\Gamma_2(T_p(M))$  represent the second order tensor space of the tangent space  $T_p(M)$ , where  $\Gamma_2(M) = \cup_{p \in M} \Gamma_2(T_p(M))$ . Then the second order tensor field is defined as  $\omega_2 : M \rightarrow \Gamma_2(M)$ , for any  $p \in M$ ,  $\omega_2(p) \in \Gamma_2(T_p(M))$ . If  $\omega_2$  satisfies the following properties, for any  $\Phi_p, \Theta_p \in T_p(M)$

$$(1) \text{ symmetric: } \omega_2(p)(\Phi_p, \Theta_p) = \omega_2(p)(\Theta_p, \Phi_p)$$

(2) positive definite:

$$\omega_2(p)(\Phi_p, \Phi_p) \geq 0 \text{ and } \omega_2(p)(\Phi_p, \Phi_p) = 0 \Leftrightarrow \Phi = 0$$

(3) smooth:  $\omega_2$  is smooth on  $M$

Then,  $\omega_2$  is called a Riemannian metric on  $M$  and  $(M, \omega_2)$  a Riemannian manifold.

By utilizing the Riemannian metric  $\omega_2$ , the length of any smooth curve  $\rho : (-\varepsilon, \varepsilon) \rightarrow M$  on  $M$  can be defined as

$$L(\rho) = \int_{-\varepsilon}^{\varepsilon} \sqrt{\omega_2(\rho(t))(\rho'(t), \rho'(t))} dt \quad (1)$$

Then, the geodesic distance between any two points on  $M$  is further defined, for any  $p, q \in M$ , as

$$geoD(p, q) = \min\{L(\rho) | \rho : (-\varepsilon, \varepsilon) \rightarrow M, \rho(-\varepsilon) = p, \rho(\varepsilon) = q\} \quad (2)$$

Since the symmetric positive definite second order tensor field  $\omega_2(p)$  on the tangent space  $T_p(M)$  is the inner product of  $T_p(M)$ , once the Riemannian metric is defined, the inner product of the tangent space at each point of the Riemannian manifold is defined. Hence, tangent spaces of Riemannian manifolds are inner product spaces, which are finite-dimensional, namely, Hilbert spaces.

## 3. Related Works

In this section, we briefly review related works on discriminant analysis and dimensionality reduction (DR) of SPD manifolds. In recent years, the wide applications of multimedia and big data make SPD data occur in more and more occasions and attract attention of an extensive range. As mentioned in the Section 1, the applications of SPD data

face some challenges, the most prominent of which are caused by the high dimensionality, for example, the overfitting caused by the case that the sample size is far smaller than the data dimensionality, the increase of computational complexity and the information redundancy caused by high-dimensional data. All these problems seriously affect the effective and reliable applications of SPD data on various occasions. In order to effectually solve above problems, the processing methods of SPD data in almost all relevant studies can be categorized into three kinds. First, the original SPD manifold is embedded into RKHS by an implicit mapping to utilize various kernel learning methods. Second, in order to use the machine learning models built on Euclidean spaces or to perform linear operations on SPD, the original SPD manifold is flattened to a Euclidean space, so as to make use of Euclidean methods. Third, by learning a mapping, the original SPD manifold is directly projected into a lower-dimensional one. These three methods can be said to be the basic methods. Other methods are the derivation or combination of these methods.

There is a lot of research on embedding SPD data into RKHS. This method can not only overcome the defect that original SPD manifolds do not support linear operations, but also be applicable to different occasions by choosing different kernel functions due to the diversity of kernel functions. Wang et al. [24] proposed a discriminant analysis method of multiple manifolds. It first constructs image sets into an SPD manifold and a Grassmann manifold, then embeds the obtained SPD manifold and Grassmann manifold into RKHS by Riemannian kernel and projection kernel, respectively, and, finally, carries out weighted discriminant analysis on them. A kernel-based sparse representation of Lie group and dictionary learning were proposed in [21] and successfully applied in face recognition and texture analysis. Huang et al. [25] mapped the source Euclidean space and Riemannian manifold to a common Euclidean subspace, which is implemented by a corresponding higher-dimensional RKHS. Through this mapping, the problem of learning the metric cross two heterogeneous spaces can be reduced into the problem of learning the Euclidean metric in the target Euclidean space. One of the advantages of using the kernel methods mentioned above is the multiplicity and selectivity of the kernel. However, choosing an impertinent kernel often results in performance degradation. In [26], a kernel learning method is used to overcome the kernel selection problem of classification on the Riemannian manifold. It presents two guidelines for jointly learning the kernel and classifier with the multi-kernel learning method.

Flattening SPD data into Euclidean forms is also a very classical method. Based on a spatio-temporal covariance descriptor, Sanin et al. [27] proposed a new motion and gesture recognition method, as well as a weighted Riemannian locally preserving projection method. The method requires linear operations; the first thing to do in this work is to perform tangential transformation on the spatio-temporal covariance descriptor, namely SPD data. Vemulapalli et al. [28] focused on logarithmic Euclidean Riemannian geometry and proposed a data-driven method for learning the Riemannian metric of SPD matrices. Huang et al. [9] used neural network structure. In this work, the input SPD matrix is converted into a more ideal SPD matrix by the design of the bilinear mapping layer and the nonlinear activation function is applied to the new SPD matrix by the eigenvalue correction layer. Finally, the data of the hidden are mapped to a Euclidian space in the output layer. A common problem with all of these works is that they inevitably destroy the geometry of SPD manifolds. In order to overcome this limitation, [10] proposed a method, LEML, to directly learn the logarithm of SPD matrix. By learning a tangent mapping, LEML can directly convert the matrix logarithm from the tangent space to another tangent space with more distinguishable information. Under the tangential mapping framework, the new metric learning can be reduced to the optimization seeking a Mahalanobis-like matrix, which can be solved by traditional metric learning techniques. Then, a more concise and efficient method,  $\alpha$ -CML, was proposed based on the improvement of LEML in [29].

In many cases, SPD data are awfully high-dimensional, so how to effectively represent high-dimensional SPD data in low-dimensional space becomes particularly important. For SPD data, DR is usually performed by seeking a full column rank projection matrix.

In [18], a direct DR method from high-dimensional SPD manifolds to low-dimensional ones is proposed. The projection generates a low-dimensional SPD manifold with maximum distinguishing ability through the orthogonal constraint and weighted affiliative similarity metric based on the manifold structure. Under the framework of the neural network, Dong et al. [30] made use of two layers (2D full connection layer and symmetric cleaning layer) to embed original SPD data into a subspace with lower dimension and better differentiation. On the basis of PCA [31], established a common component analysis (CCA) for multiple SPD matrices. This method attempts to find a low-dimensional common subspace but ignores the Riemannian geometry of the SPD manifold. From the perspective of geometric structure, Harandi et al. [32] used bilinear transformation to realize a supervised DR algorithm of SPD data and another unsupervised one. The supervised method gathers samples of the same class and separates samples of different classes at the same time to obtain a low-dimensional embedding representation with strong discrimination. In consideration of the maximum variance unfolding (MVU) algorithm [33], the unsupervised method maps SPD samples to a lower-dimensional SPD manifold in line with the criterion of maximum variance. Under the guidance of the nonparametric estimation theory, ref. [34] proposed a DR method of SPD which makes use of estimated probability distribution function (PDF) to construct an affinity matrix so as to achieve the maximum separability with respect to categories from the perspective of probability.

#### 4. The Proposed Method

In this paper, the proposed dimensionality reduction (DR) method based on tangent spaces and isometry first maps the original SPD data to the tangent space at the identity matrix; then, the bilinear transformation between manifolds is transferred to the tangent spaces. In the tangent space after DR, Euclidean metric is adopted to measure the distance between the data points after DR, while the bilinear transformation is determined by the isometric criterion. Specifically, we aim to keep the distance between data points in the tangent space after DR as close as possible to the corresponding geodesic distance on the original SPD manifold. With this idea in mind, we then derive the model.

##### 4.1. The Isometric Transformation between SPD Manifolds and Their Tangent Spaces

SPD data are a kind of non-Euclidian data which can form a differential manifold under a certain structure. In particular, the tangent space of an SPD manifold is a symmetric matrix space, which can be regarded as the minimum linear extension set of the SPD manifold, that is, the minimum linear space containing the SPD matrices. In principle, machine learning algorithms developed in the symmetric matrix space are also suitable for its SPD subset.

As mentioned earlier, there is no distance metric on a differential manifold or its tangent spaces. A Riemannian metric must be endowed so as to obtain a Riemannian manifold. At present, there are several commonly used Riemannian metrics. The affine invariant Riemannian metric (AIRM) is adopted in this paper; for any  $X \in \text{Sym}_{++}^D$ , the inner product of the tangent space  $T_X(\text{Sym}_{++}^D)$  at point  $X$  is defined as follows:

$$\langle \Phi, \Theta \rangle_X = \left\langle X^{-1/2} \Phi X^{-1/2}, X^{-1/2} \Theta X^{-1/2} \right\rangle_F, \forall \Phi, \Theta \in T_X(\text{Sym}_{++}^D) \quad (3)$$

Particularly, the identity matrix is an SPD and the affine invariant inner product of its tangent space  $T_{I_D}(\text{Sym}_{++}^D)$  is as follows:

$$\langle \Phi, \Theta \rangle_{I_D} = \left\langle I_D^{-1/2} \Phi I_D^{-1/2}, I_D^{-1/2} \Theta I_D^{-1/2} \right\rangle_F = \langle \Phi, \Theta \rangle_F, \forall \Phi, \Theta \in T_X(\text{Sym}_{++}^D) \quad (4)$$

The Euclidean distance between tangent vectors derived from the inner product is

$$\|\Phi - \Theta\|_{I_D}^2 = \langle \Phi - \Theta, \Phi - \Theta \rangle_F, \forall \Phi, \Theta \in T_X(\text{Sym}_{++}^D) \quad (5)$$

The geodesic distance on the SPD manifold defined by the AIRM is

$$geoD(X, Y) = \|\log(Y^{-1/2}XY^{-1/2})\|_F, \forall X, Y \in Sym_{++}^D \tag{6}$$

Over here,  $\log : Sym_{++}^D \rightarrow Sym^D$ . For any  $X \in Sym_{++}^D$ , by eigen-decomposition, we can have  $X = Udiag(\lambda_1, \dots, \lambda_D)U^T$ . Hence,

$$\log(X) = Udiag(\log(\lambda_1), \dots, \log(\lambda_D))U^T \tag{7}$$

It is obvious that  $\log(X)$  has the same size as  $X$ . Besides, we note that the injection  $\log(\cdot)$  is monotonic, which does not change the order of eigenvalues. Hence, it realizes the transformation with the minimum change of properties compared with original data.

If we intend to transfer learning tasks from the original SPD manifold to its tangent space, we first need to transform original data from the manifold to one of its tangent spaces. This paper selects the tangent space at identity  $T_{I_D}(Sym_{++}^D)$  and  $\log$  transformation. For any  $X \in Sym_{++}^D$ ,

$$geoD(X, I_D) = \|\log(I_D^{-1/2}XI_D^{-1/2})\|_F = \|\log(X)\|_F \tag{8}$$

where  $\|\log(X)\|_F$  is the Euclidean distance between  $\log(X)$  and the origin point. Hence,  $\log$  is actually an isometric transformation; the geodesic distance between SPD data and the identity matrix is equal to the Euclidean distance between its tangent vector and the origin of the tangent space.

#### 4.2. The Bilinear Transformation between Tangent Spaces

This paper studies the DR of SPD data. At present, the bilinear transformation of SPD matrices is the most commonly used method in DR algorithms of SPD data:

$$f_W : Sym_{++}^D \rightarrow Sym_{++}^d$$

For  $X \in Sym_{++}^D$ ,

$$f_W(X) = W^T X W \in Sym_{++}^d \tag{9}$$

where  $W \in R^{D \times d}$ ,  $W^T W = I_d$ ,  $D \gg d$ .

Obviously,  $f_W$  is a transformation between two SPD manifolds. Since the SPD manifold is not a Euclidean space, it is difficult to directly solve  $f_W$ , no matter the modeling or optimization.

Due to the fact that the tangent space of an SPD manifold contains the SPD manifold itself, this paper proposes a DR model of bilinear transformation based on tangent spaces at identities of SPD manifolds:

$$g_W : T_{I_D}(Sym_{++}^D) \rightarrow T_{I_d}(Sym_{++}^d)$$

For any  $\Theta \in T_{I_D}(Sym_{++}^D)$ ,

$$g_W(\Theta) = W^T \Theta W \in T_{I_d}(Sym_{++}^d) \tag{10}$$

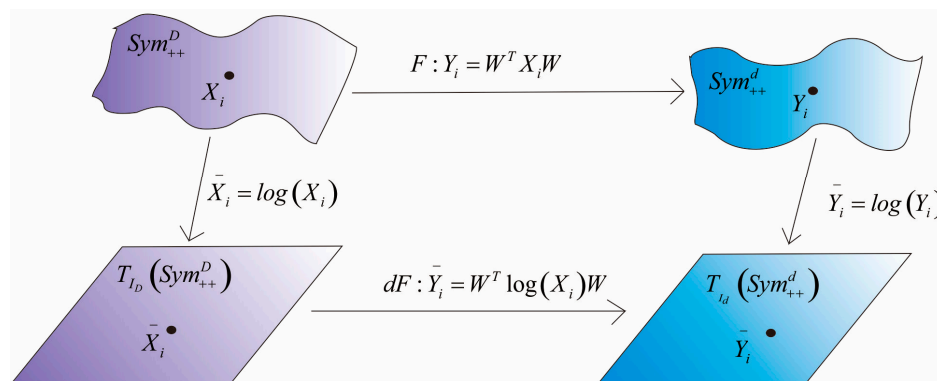
Over here,  $W \in R^{D \times d}$ ,  $W^T W = I_d$ ,  $D \gg d$ .

**Remark 1:**  $g_W$  is a transformation between tangent spaces of SPD manifolds, that is, a transformation between two Euclidean spaces. It is obviously much easier to build DR algorithms on Euclidean spaces.

**Remark 2:** If we just consider DR between tangent spaces, the constraint  $W^T W = I_d$  is unnecessary. Any matrix  $W \in R^{D \times d}$  can make  $g_W(\Theta)$  implement DR of the symmetric matrix  $\Theta$ .

However, our objective is to reduce data dimensionality for SPD matrices. By adding the constraint  $W^T W = I_d$ , it can be guaranteed that  $g_W|_{Sym_{++}^D} = f_W$  on the SPD manifold.

The advantage of the DR method based on tangent spaces of SPD manifolds proposed in this paper is that the DR between SPD manifolds is realized by the DR between Euclidean tangent spaces. A brief schematic diagram of the DR process is shown in Figure 1.



**Figure 1.** Schematic diagram of DR for SPD data. First, original SPD data are mapped to the tangent space at identity. Subsequently, the bilinear transformation between tangent spaces is determined in line with a specific criterion. Finally, the bilinear transformation learned from tangent spaces is taken to realize the DR of SPD data.

### 4.3. Dimensionality Reduction of SPD Data Based on Tangent Spaces of SPD Manifolds and Isometry

#### 4.3.1. Isometry Modeling

The DR scheme of SPD data based on tangent spaces of SPD manifolds and bilinear transformation described in Section 4.2 is only a framework. Moreover, it is necessary to determine the bilinear transformation matrix  $W$  according to a certain criterion. Different criteria generate different DR algorithms. In this paper, an isometric criterion is proposed to keep the distance between data points in the tangent space after DR close to the geodesic distance on the original SPD manifold as much as possible.

For a given set of data on an SPD manifold,

$$X_i \in Sym_{++}^D, i = 1, \dots, N$$

Since  $Sym_{++}^D$  is a Riemannian manifold, the geodesic distance between two SPD matrices  $X_i$  and  $X_j$  (under AIRM) is

$$g_{ij} = geoD(X_i, X_j) = \|\log(X_j^{-1/2} X_i X_j^{-1/2})\|_F, 1 \leq i, j \leq N \tag{11}$$

After  $\log$  transformation, the set of data is transformed to another set of data in  $T_{I_D}(Sym_{++}^D)$ :

$$\log(X_i) \in T_{I_D}(Sym_{++}^D), i = 1, \dots, N$$

Afterwards, by the bilinear transformation between  $T_{I_D}(Sym_{++}^D)$  and  $T_{I_d}(Sym_{++}^d)$ , we have

$$W^T \log(X_i) W \in T_{I_d}(Sym_{++}^d), i = 1, \dots, N$$

Because the tangent space  $T_{I_d}(Sym_{++}^d)$  is a Euclidean space, the Euclidean distance between any two tangent vectors  $W^T \log(X_i) W$  and  $W^T \log(X_j) W$  is

$$d_{ij} = \|W^T \log(X_i) W - W^T \log(X_j) W\|_F = \|W^T (\log(X_i) - \log(X_j)) W\|_F = \|W^T \Phi_{ij} W\|_F \tag{12}$$

where  $\Phi_{ij} = \log(X_i) - \log(X_j), 1 \leq i, j \leq N$ .

The so-called isometric criterion is that the geodesic distance  $g_{ij}$  is equal to the Euclidean distance  $d_{ij}$ .

$$\sum_{i=1}^N \sum_{j=1}^N (g_{ij} - d_{ij})^2 \xrightarrow{W^T W = I_d} \min \tag{13}$$

### 4.3.2. Objective Function

This subsection derives the objective function of the model.

For convenience, we construct two distance matrices, as follows:

$$G = \begin{bmatrix} g_{11} & \cdots & g_{1N} \\ \vdots & \ddots & \vdots \\ g_{N1} & \cdots & g_{NN} \end{bmatrix}, D = \begin{bmatrix} d_{11} & \cdots & d_{1N} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NN} \end{bmatrix} \tag{14}$$

According to the isometric criterion,  $W$  is selected to minimize the difference between the geodesic distance on the manifold and the distance in the tangent space at identity. Hence, the following model is obtained:

$$\hat{W} = \arg \min_W \|G - D\|_F^2 \quad s.t. \quad W^T W = I_d \tag{15}$$

For  $\forall g_{ij} \in G$  and  $\forall d_{ij} \in D$ , let  $\varepsilon_{ij} = (g_{ij} - d_{ij})^2 = g_{ij}^2 + d_{ij}^2 - 2g_{ij}d_{ij}$ ; we expect that  $g_{ij}$  and  $d_{ij}$  are as close as possible and, replacing  $g_{ij}$  with  $d_{ij}$ , we have  $\varepsilon_{ij} \approx g_{ij}^2 - d_{ij}^2$

Let

$$G^2 = \begin{bmatrix} g_{11}^2 & \cdots & g_{1N}^2 \\ \vdots & \ddots & \vdots \\ g_{N1}^2 & \cdots & g_{NN}^2 \end{bmatrix}, D^2 = \begin{bmatrix} d_{11}^2 & \cdots & d_{1N}^2 \\ \vdots & \ddots & \vdots \\ d_{N1}^2 & \cdots & d_{NN}^2 \end{bmatrix} \tag{16}$$

Then,

$$\hat{W} = \arg \min_W \|G^2 - D^2\|_F^2 \tag{17}$$

can be represented in another form:

$$\hat{W} = \arg \min_W \|G^2 - D^2\|_F^2 \tag{18}$$

Let the centering matrix be  $\Gamma$ ,

$$\Gamma = \left( I_D - \frac{1}{D} (ee^T) \right) \tag{19}$$

where  $e = [1, 1, \dots, 1]^T \in \mathbb{R}^D$ ,  $I_D$  is a  $D \times D$  identity matrix.

Use (19) to center the objective function and obtain

$$\begin{aligned} \hat{W} &= \arg \min_W \left\| -\frac{1}{2} \Gamma (G^2 - D^2) \Gamma \right\|_F^2 \\ &= \arg \min_W \left\| -\frac{1}{2} \Gamma G^2 \Gamma - \left( -\frac{1}{2} \Gamma D^2 \Gamma \right) \right\|_F^2 \\ &= \arg \min_W \left\| \rho_G - \rho_D \right\|_F^2 \end{aligned} \tag{20}$$

Over here,  $\rho_G = -\frac{1}{2} \Gamma G^2 \Gamma$ ,  $\rho_D = -\frac{1}{2} \Gamma D^2 \Gamma$ .

It can be proved that  $\rho_G$  and  $\rho_D$  are both inner matrices [35],  $d_{ij}^2 = (\rho_D)_{ii} + (\rho_D)_{jj} - 2(\rho_D)_{ij}$ .



According to the properties of linear algebra and matrix trace, Equation (20) can be further rewritten as

$$\hat{W} = \arg \min_W \left[ -2tr(\rho_G \rho_D^T) + \|\rho_G\|_F^2 + \|\rho_D\|_F^2 \right] \tag{21}$$

In consideration of

$$d_{ij} = \|\Theta_i - \Theta_j\|_F = \|W^T(\log(X_i) - \log(X_j))W\|_F \leq \|X_i - X_j\|_F \tag{22}$$

hence,

$$\|\rho_D\|_F^2 \leq \|\rho_S\|_F^2 \tag{23}$$

where

$$S^2 = \begin{bmatrix} s_{11}^2 & \dots & s_{1N}^2 \\ \vdots & \ddots & \vdots \\ s_{N1}^2 & \dots & s_{NN}^2 \end{bmatrix}, s_{ij} = \|X_i - X_j\|_F, \rho_S = -\frac{1}{2}\Gamma S^2 \Gamma \tag{24}$$

It can be seen that the objective function has an upper boundary:

$$\begin{aligned} \hat{W} &= \arg \min_W \left[ -2tr(\rho_G \rho_D^T) + \|\rho_G\|_F^2 + \|\rho_D\|_F^2 \right] \\ &\leq \arg \min_W \left[ -2tr(\rho_G \rho_D^T) + \|\rho_G\|_F^2 + \|\rho_S\|_F^2 \right] \end{aligned} \tag{25}$$

Since  $G^2$  and  $S^2$  are unrelated to  $W$ , we can further simplify the objective function as

$$\begin{aligned} \hat{W} &= \arg \min_W -2tr(\rho_G \rho_D^T) \\ &= \arg \min_W -2 \sum_{i=1}^N \sum_{j=1}^N (\rho_D)_{ij} (\rho_G)_{ij} \end{aligned} \tag{26}$$

In consideration of

$$2(\rho_D)_{ij} = \|\Theta_i\|_F^2 + \|\Theta_j\|_F^2 - \|\Theta_i - \Theta_j\|_F^2 \tag{27}$$

substitute (27) into (26),

$$\begin{aligned} \hat{W} &= \arg \min_W -2 \sum_{i=1}^N \sum_{j=1}^N \|\Theta_i\|_F^2 (\rho_G)_{ij} + \sum_{i=1}^N \sum_{j=1}^N \|\Theta_i - \Theta_j\|_F^2 (\rho_G)_{ij} \\ &= \arg \min_W -2 \sum_{i=1}^N \|\Theta_i\|_F^2 \sum_{j=1}^N (\rho_G)_{ij} + \sum_{i=1}^N \sum_{j=1}^N \|\Theta_i - \Theta_j\|_F^2 (\rho_G)_{ij} \end{aligned} \tag{28}$$

On account of

$$\sum_{j=1}^N (\rho_G)_{ij} = 0 \tag{29}$$

we have

$$\begin{aligned} \hat{W} &= \arg \min_W \sum_{i=1}^N \sum_{j=1}^N \|\Theta_i - \Theta_j\|_F^2 (\rho_G)_{ij} \\ &= \arg \min_W \sum_{i=1}^N \sum_{j=1}^N \|W^T \log(X_i)W - W^T \log(X_j)W\|_F^2 (\rho_G)_{ij} \\ &= \arg \min_W \sum_{i=1}^N \sum_{j=1}^N tr \left( W^T (\log(X_i) - \log(X_j)) W W^T (\log(X_i) - \log(X_j))^T W \right) (\rho_G)_{ij} \end{aligned} \tag{30}$$

So far, we have derived the objective function of the model and what we need to do next is optimizing the function.

#### 4.4. Solution to RMTSISOM-SPDDR

In this subsection, we solve the derived objective function. Obviously, this is a convex optimization problem, but this objective function does not correspond to a closed-form solution, so we are supposed to find its local optimal solution  $\hat{W}$  through an iterative method.

Let the  $t^{\text{th}}$  iteration yield  $\hat{W}_t$ , then the result of the  $t+1^{\text{th}}$  iteration can be expressed as

$$\begin{aligned}\hat{W}_{t+1} &= \underset{W}{\operatorname{argmin}} \sum_{i=1}^N \sum_{j=1}^N \operatorname{tr} \left( W^T (\log(X_i) - \log(X_j)) \hat{W}_t \hat{W}_t^T (\log(X_i) - \log(X_j))^T W \right) (\rho_G)_{ij} \quad \text{s.t. } W^T W = I_d \quad (31) \\ &= \underset{W}{\operatorname{argmax}} \operatorname{tr} (W^T L W)\end{aligned}$$

Over here,

$$L = - \sum_{i=1}^N \sum_{j=1}^N (\rho_G)_{ij} (\log(X_i) - \log(X_j)) \hat{W}_t \hat{W}_t^T (\log(X_i) - \log(X_j))^T \quad (32)$$

The eigenvalue decomposition of  $L$  is needed and  $\hat{W}_{t+1}$  is obtained by taking the eigenvectors corresponding to the maximum  $d$  eigenvalues. The entire algorithm flow of RMTSISOM-SPDDR is shown in Algorithm 1.

---

#### Algorithm 1. Procedures of RMTSISOM-SPDDR.

---

**Input:**  $N$  training samples  $X = \{X_1, X_2, \dots, X_N\}$ ,  $X_i \in \operatorname{Sym}_{++}^D$ , target dimension  $d$  and the maximum iteration times  $\mathit{maxIter}$ .

**Output:** bilinear transformation matrix  $W \in \mathbb{R}^{D \times d}$ .

1: Map SPD data into the corresponding tangent space at identity using Equation (7);

2: Construct Riemannian metric matrix using Equation (16);

3: Construct the inner product matrix  $\rho_G$  using Equation (20);

4: Initialize  $\hat{W}_0 = I_D(:, 1:d)$ , namely, the first  $d$  columns of  $I_D \in \mathbb{R}^{D \times D}$ ;

5: while  $t = 1: \mathit{maxIter}$

    Calculate  $L$  using Equation (32);

    Do eigen-decomposition of  $L$  to obtain the bilinear transformation matrix  $\hat{W}_t$ ;

end while

6: return  $\hat{W}_t$ .

---

#### 4.5. Complexity Analysis

In this subsection, we analyze the computational complexity of RMTSISOM-SPDDR. The complexity of the algorithm comes from five aspects—the mapping from the SPD manifold to its tangent space at identity, the construction of the Riemannian metric matrix, the calculation of the inner matrix  $\rho_G$ , the construction of  $L$  and the eigen-decomposition of  $L$ . First of all, the mapping from the SPD manifold to the tangent space at identity requires eigen-decomposition of each data point and taking the logarithm of its eigenvalues. This process can be realized by the built-in function `eig` in MATLAB and the time complexity of the implementation is  $\mathcal{O}(ND^3 \log(D))$ . Secondly, the construction of the Riemannian metric matrix requires calculating the geodesic distances of data pairs. Note, it is a symmetric matrix which needs to calculate  $N(N+1)/2$  times. Hence, the corresponding time complexity is  $\mathcal{O}(N(N+1)/2) = \mathcal{O}(\frac{1}{2}N^2 + \frac{1}{2}N) = \mathcal{O}(N^2)$ . The calculation of  $\rho_G$  involves, first, squaring each element of the Riemannian metric matrix, then multiplying it left and right by the centering matrix, so the time complexity of this procedure is  $\mathcal{O}(N^2 + N^4) = \mathcal{O}(N^4)$ . Besides, the construction of  $L$  requires four matrix multiplications and  $N^2$  matrix additions, the time complexity is  $\mathcal{O}(N^2(d^2D^4))$ . Finally, the complexity of the eigen-decomposition for  $L$  is  $\mathcal{O}(D^3 \log(D))$ .

Since these five processes are cascaded, the final time complexity of RMTSISOM-SPDDR is  $\mathcal{O}(ND^3 \log(D) + N^2 + N^4 + N^2(d^2D^4) + D^3 \log(D))$ .

## 5. Comparison with Other State-of-the-Art Algorithms

In this section, we introduce five state-of-the-art DR algorithms of SPD data, including PDDRSPD [34], PSSSD [36], DARG-Graph [37], SSPDDR [32] and LG-SPDDR [38], in summary. Here, we also point out the differences and connections between RMTSISOM-SPDDR and these algorithms.

These DR algorithms for comparison all belong to supervised learning and affinity, in which data labels are needed. In contrast, RMTSISOM-SPDDR is unsupervised and on the basis of global isometry. Except for LG-SPDDR, the DR of all other comparison algorithms is based on direct bilinear transformation between SPD matrices, while the DR of RMTSISOM-SPDDR is based on the bilinear transformation between tangent vectors of SPD matrices. LG-SPDDR utilizes exp transformation for the dimensional-reduced tangent vectors to map them back to a low-dimensional manifold, which means its framework is different from ours.

### 5.1. SSPDDR

In SSPDDR [32], a supervised DR method of SPD data is proposed, where the metric  $\delta(X_i, X_j)$  is given to measure the similarity among SPD data. Moreover, the bilinear transformation is employed to the DR process:

$$X \in \text{Sym}_{++}^D \Rightarrow W^T X W \in \text{Sym}_{++}^d$$

Over here,  $W \in R^{D \times d}$ ,  $W^T W = I_d$  and  $D \gg d$ . We can easily know that, if  $X$  is an SPD matrix,  $W^T X W$  is an SPD matrix as well.

In SSPDDR, the transformation matrix  $W$  can be determined as follows:

$$\sum_{j=1}^N \sum_{i=1}^N a(X_i, X_j) \delta(W^T X_i W, W^T X_j W) \xrightarrow{W^T W = I_d} \min \quad (33)$$

Over here,  $a(X_i, X_j) = a_w(X_i, X_j) - a_b(X_i, X_j)$  and

$$a_w(X_i, X_j) = \begin{cases} 1 & X_i \text{ and } X_j \text{ are adjacent and have the same label} \\ 0 & \text{others} \end{cases} \quad (34)$$

$$a_b(X_i, X_j) = \begin{cases} 1 & X_i \text{ and } X_j \text{ are adjacent but have different labels} \\ 0 & \text{others} \end{cases} \quad (35)$$

SSPDDR indicates that if  $X_i$  and  $X_j$  are adjacent and intraclass, their compactness is maximized in the process of DR, or else their separability is minimized. If  $X_i$  and  $X_j$  are not adjacent, their relationship will be neglected in DR.

### 5.2. PDDRSPD

In PDDRSPD [34], labeled SPD data  $\{(X_i, c_i) | i = 1, \dots, N\}$  are given, in which  $c_i$  is the label of  $X_i$ ,  $c_i \in \{1, \dots, L\}$ .

For each class, PDDRSPD firstly calculates its mean vector and covariance matrix. Subsequently, based on nonparametric estimation, it estimates the Gaussian density functions  $\{f_c(X) | c = 1, \dots, L\}$ .

The framework of PDDRSPD is similar to [32], but the affinity coefficients is quite different:

$$\sum_{j=1}^N \sum_{i=1}^N a(X_i, X_j) \delta(W^T X_i W, W^T X_j W) \xrightarrow{W^T W = I_d} \min \quad (36)$$

Over here  $a(X_i, X_j) = a_w(X_i, X_j) - a_b(X_i, X_j)$  and

$$a_w(X_i, X_j) = \begin{cases} \frac{(\lambda_w - 1) |f_c(X_i) - f_c(X_j)|}{\sum_{X \in N_w(X_i)} |f_c(X_i) - f_c(X)|} & X_j \in N_w(X_i) \\ 1 & i = j \\ 0 & \text{others} \end{cases} \quad (37)$$

where  $c$  represents the common label of  $X_i$  and  $X_j$ ,  $N_w(X_i)$  represents the set of  $\lambda_w$  samples belonging to the same class as  $X_i$  and farthest from  $X_i$ .

$$a_b(X_i, X_j) = \begin{cases} \frac{\lambda_b f_{c_j}(X_j)}{\lambda_s \sum_{X \in N_b(X_i)} f_{c_j}(X)} & c_j \in C_b(X_i) \\ 0 & \text{others} \end{cases} \quad (38)$$

Over here,  $C_b(X_i) = \{c_{i_1}, \dots, c_{i_{\lambda_b}}\}$  and

$$\underbrace{f_{c_{i_1}}(X_i) \geq \dots \geq f_{c_{i_{\lambda_b}}}(X_i)}_{\lambda_b} \geq \dots \geq f_{c_{i_L}}(X_i) \quad (39)$$

$N_b(X_j)$  is the set of  $\lambda_s$  SPD matrices belonging to the same class as  $X_j$  and corresponding to largest Gaussian density values.

### 5.3. LG-SPDDR

In LG-SPDDR [38], a DR method for SPD data is proposed as

$$X \in \text{Sym}_{++}^D \Rightarrow Y_W = \exp(W^T \log(X) W) \in \text{Sym}_{++}^d$$

where  $W \in R^{D \times d}$ ,  $D \gg d$ .

To be specific, LG-SPDDR is implemented by three procedures:

- The given SPD data  $X$  are firstly mapped from the original SPD manifold  $\text{Sym}_{++}^D$  to the corresponding tangent space  $T_{I_D}(\text{Sym}_{++}^D) = \text{Sym}^D$  and turn into  $\log(X) \in \text{Sym}^D$ ;
- Subsequently,  $\log(X)$  is dimensional-reduced by the bilinear transformation, transformed into another tangent space  $T_{I_d}(\text{Sym}_{++}^d) = \text{Sym}^d$  and turns into  $W^T \log(X) W \in \text{Sym}^d$ ;
- Finally, through the  $\exp$ -transformation,  $W^T \log(X) W$  is remapped back into a new low-dimensional SPD manifold  $\text{Sym}_{++}^d$  and turns into  $\exp(W^T \log(X) W) \in \text{Sym}_{++}^d$ .

LG-SPDDR determines the transformation matrix  $W$  as follows:

$$\sum_{j=1}^N \sum_{i=1}^N a(X_i, X_j) \delta(Y_i^W, Y_j^W) \xrightarrow{W \in R^{D \times d}} \min \quad (40)$$

Over here, the definition of affinity coefficients  $a(X_i, X_j)$  is similar to [32].

**Remark 3:** In spite of the claim that LG-SPDDR is based on Lie groups and Lie algebra, it seems to be irrelevant to the multiplication of Lie groups and the Lie bracket of Lie algebra.

### 5.4. PSSSD

PSSSD [36] is a supervised metric learning method. Both the bilinear transformation and label information are taken into account in the learning process. Its core idea is a framework for learnable metric, in which:

(1) A set of divergences measuring the similarity between two SPD matrices have been given:

$$\mathcal{D}^{(\alpha_m, \beta_m)}(X_i \| X_j) = \frac{1}{\alpha\beta} \log \left( \frac{\alpha_m (X_i X_j^{-1})^{\beta_m} + \beta_m (X_i X_j^{-1})^{\alpha_m}}{\alpha_m + \beta_m} \right) \tag{41}$$

$$\alpha_m \beta_m \neq 0, \alpha_m + \beta_m \neq 0$$

Over here,  $\alpha_m$  and  $\beta_m$  are all learnable,  $m = 1, \dots, M$ .

(2) It exploits multiple transformation matrices to map original SPD manifold to multiple submanifolds:

$$W_m \in R^{D \times d}, W_m^T W_m = I_d, m = 1, \dots, M$$

(3) To measure the difference between two SDP matrices  $X_i$  and  $X_j$ , the learnable metric is defined as

$$\mathcal{D}^\Theta(X_i \| X_j) = d_{ij}^T \Phi d_{ij}, d_{ij} = \begin{bmatrix} \mathcal{D}^{(\alpha_1, \beta_1)}(W_1^T X_i W_1 \| W_1^T X_j W_1) \\ \vdots \\ \mathcal{D}^{(\alpha_M, \beta_M)}(W_M^T X_i W_M \| W_M^T X_j W_M) \end{bmatrix} \in R^M \tag{42}$$

Over here,  $\Phi \in R^{M \times M}$  is a learnable SPD matrix. Hence, entire learnable parameters can be represented by

$$\Theta = \{\Phi, W_1, \dots, W_M, \alpha_1, \dots, \alpha_M, \beta_1, \dots, \beta_M\} \tag{43}$$

The metric  $\mathcal{D}^\Theta(\bullet \| \bullet)$  is finally specified as

$$\frac{1}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} y_{ij} \max(\mathcal{D}^\Theta(X_i, X_j) - \zeta_S, 0)^2 + \frac{1}{|\mathcal{H}|} \sum_{(i,j) \in \mathcal{H}} (1 - y_{ij}) \max(\zeta_H - \mathcal{D}^\Theta(X_i, X_j), 0)^2 \xrightarrow{\Theta} \min \tag{44}$$

where  $\mathcal{S}$  represents the set of data pairs having the same label and  $\mathcal{H}$  represents the set of data pairs having different labels;  $y_{ij} = \begin{cases} 1 & \text{If } X_i \text{ and } X_j \text{ are similar} \\ 0 & \text{otherwise} \end{cases}$ ,  $\zeta_S$  and  $\zeta_H$  are similarity thresholds.

The objective of LG-SPDDR indicates that, if  $X_i$  and  $X_j$  are similar to each other and belong to the same class, the metric  $\mathcal{D}^\Theta(X_i, X_j)$  is minimized. In contrast, if  $X_i$  and  $X_j$  are similar but belong to the different classes,  $\mathcal{D}^\Theta(X_i, X_j)$  is maximized.

Strictly speaking, PSSSD is essentially not a DR algorithm. To compare it with RMTSISOM-SPDDR, the matrix  $W = [W_1 \dots W_M] \in R^{D \times Md}$  is used for DR.

### 5.5. DARG-Graph

Let  $g = \mathcal{N}(\mu, \Sigma)$  be a Gaussian distribution,  $\mu \in R^D$  be its mean vector and  $\Sigma \in R^{D \times D}$  be its covariance matrix. In DARG-Graph, a number of metrics  $\delta(g_i, g_j)$  to measure the difference between two Gaussian distributions  $g_i$  and  $g_j$  are given.

In DARG-Graph [37], the DR of a Gaussian distribution is proposed as follows:

$$g = \mathcal{N}(\mu, \Sigma) \Rightarrow g^W = \mathcal{N}(W^T \mu, W^T \Sigma W)$$

where  $W \in R^{D \times d}$  and  $W^T W = I_d, D \gg d$ . We can easily know that  $g^W$  is a Gaussian distribution as well.

The transformation matrix  $W$  is specified as follows:

$$\sum_{j=1}^N \sum_{i=1}^N a(g_i, g_j) \delta(g_i^W, g_j^W) \xrightarrow{W^T W = I_d} \min \quad (45)$$

Over here,  $a(X_i, X_j) = a_w(X_i, X_j) - a_b(X_i, X_j)$  and

$$a_w(g_i, g_j) = \begin{cases} e^{-\frac{\delta(g_i, g_j)}{\sigma^2}} & \delta(g_i, g_j) < \varepsilon \text{ and have the same label} \\ 0 & \text{others} \end{cases} \quad (46)$$

$$a_b(g_i, g_j) = \begin{cases} e^{-\frac{\delta(g_i, g_j)}{\sigma^2}} & \delta(g_i, g_j) < \varepsilon \text{ but have different labels} \\ 0 & \text{others} \end{cases} \quad (47)$$

Note that RMTDISOM-SPDDR is a DR algorithm for SPD data, not for Gaussian distributions. When comparing our algorithm with DARG-Graph, we let vector  $\mu = 0$ .  $\Sigma$  in  $\mathcal{N}(\mu, \Sigma)$  is an SPD matrix. Hence, the DR of Gaussian distributions is converted to the DR of SPD data.

## 6. Experiments

In this section, we compare RMTSISOM-SPDDR with five state-of-the-art algorithms by evaluating the performance of algorithms on five commonly used benchmark datasets. This section will introduce the datasets and the setting of experimental parameters, present the experimental results and further discuss the effectiveness of our algorithm.

### 6.1. Datasets Description and Experiment Settings

**Dataset overview:** Since we conduct experiments on five datasets and the processing of different datasets is not the same, it is not appropriate to list all descriptions here. We explain the specific description of each dataset one by one in the following corresponding subsections.

**Parameter setting:** The parameters that need to be adjusted for RMTSISOM-SPDDR include the target dimension  $d$  of the SPD data after DR and the maximum iteration number *maxIter* of iterative optimization. For  $d$ , the specific setting is related to the dimensionality of original SPD manifolds on which different datasets are located. It can be observed from the different experimental charts how we set it. For *maxIter*, we uniformly set it as 50, in this paper, for three reasons. Firstly, in our experimental results, under different target dimensions on all datasets, RMTSISOM-SPDDR converged before 50 iterations consistently. Hence, too many iterations are uneconomical for the calculation cost. Secondly, it is insignificant to improve performance by more iterations, which may result in overfitting, making worse performance. Thirdly, although the optimization process iterated for 50 times, the optimal result may occur before the 50th iteration. Our program recorded the classification result of each iteration and we selected the optimal result as the final experimental result. In addition, the classifier used in the experiments was the kNN classifier. For other algorithms, parameter settings are all shown in Table 2.

**Table 2.** Parameter settings of five comparison algorithms.

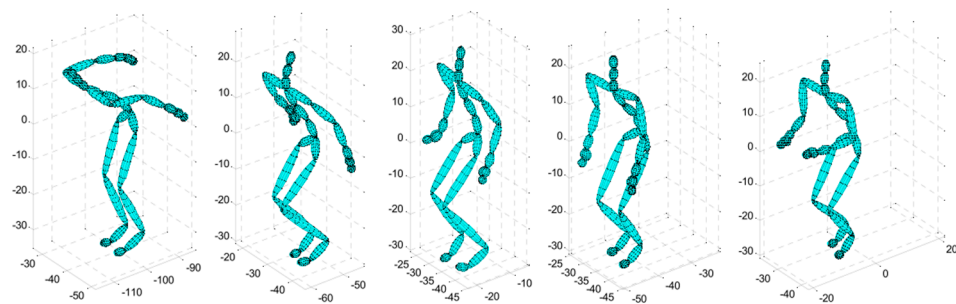
Methods	Parameter Settings
SSPDDR	None
PDDRSPD	$\lambda_w = 3, \lambda_b = 3, \lambda_s = 3$
DARG-Graph	$\sigma = 1$
LG-SPDDR	$k_w = 5, k_b = 20$
PSSSD	$\zeta_S = 1, \zeta_D = 10$

## 6.2. The Experiment on the Motion Dataset

Our first experiment was conducted on the Motion dataset. In this subsection, we successively introduce Motion and the experimental development and present results of RMTSISOM-SPDDR and other algorithms on this dataset. Subsections for other datasets keep the same narrative structure and we do not repeat this description.

### 6.2.1. The Description of Motion

Motion is derived from the human motion sequences in the HDM05 database [39], which was produced by a system based on optical mark technology. Experimenters worn a suit with 40–50 retro-reflective markers, which were tracked by 6–12 calibrated high-resolution cameras, resulting in clear and detailed motion capture data. The database consists of 14 kinds of motions, such as walking, running, jumping, kicking, etc. Partial samples of Motion are shown in Figure 2. Motion includes 306 image sets, each of which consists of 120 images. By the method in [24], they were made into  $93 \times 93$  SPD matrices. We adopted the same partition method as [32] and obtained SPD matrices for training and testing, respectively, with the ratio of 2:3.



**Figure 2.** Samples of the Motion dataset.

### 6.2.2. The Experimental Results on Motion

On Motion, we carried out classification experiments for the proposed algorithm (RMTSISOM-SPDDR) and other five comparison algorithms, respectively, and tested the convergence performance of RMTSISOM-SPDDR. Subsequent experiments on other datasets also followed the same development and we do not repeat this description in the other subsections.

From the classification results on Motion shown in Figure 3, we can see, under each target dimension selected, the classification accuracy of RMTSISOM-SPDDR is generally more than 70% and far ahead of all comparison algorithms. Especially under the target dimensions of 48, 43, 38 or 33, the accuracy of RMTSISOM-SPDDR is 6 percentage points higher than that of the comparison algorithm ranked second (LG-SPDDR). On the whole, the fluctuation of the classification accuracies of RMTSISOM-SPDDR under different target dimensions is relatively small compared with other algorithms.

To evaluate the convergence performance of RMTSISOM-SPDDR under different target dimensions on Motion, we obtained Figure 4. The loss function decreased the most in the second iteration, then became stable. Among these four curves, the convergence was the fastest when  $d = 43$  or  $d = 53$  and it basically converged at the fifth iteration. When  $d = 33$  or  $d = 63$ , the loss function continued to decline in 50 iterations, but the rangeability was very small after 25 iterations. It was unnecessary to continue iterations after 50 iterations. On the one hand, the decline of the loss function in continuous iterations was insignificant, compared with the calculation cost paid. On the other hand, excessive pursuit of the decline of the loss function may lead to overfitting. When the embedding dimension was equal to 33, the loss value fluctuated greatly, but the overall trend still declined. RMTSISOM-SPDDR still shows satisfactory convergence performance on Motion. It is worth noting that the lower the target dimension, the smaller the absolute value of the loss function, which means that we need less computation.

### Classification Accuracy on Motion Dataset

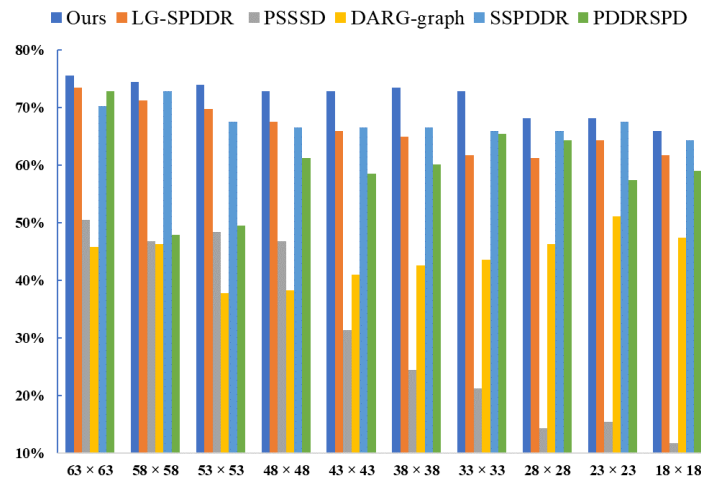


Figure 3. Classification accuracy on the Motion dataset. The horizontal axis represents the target dimension for DR and the vertical axis represents the classification accuracy.

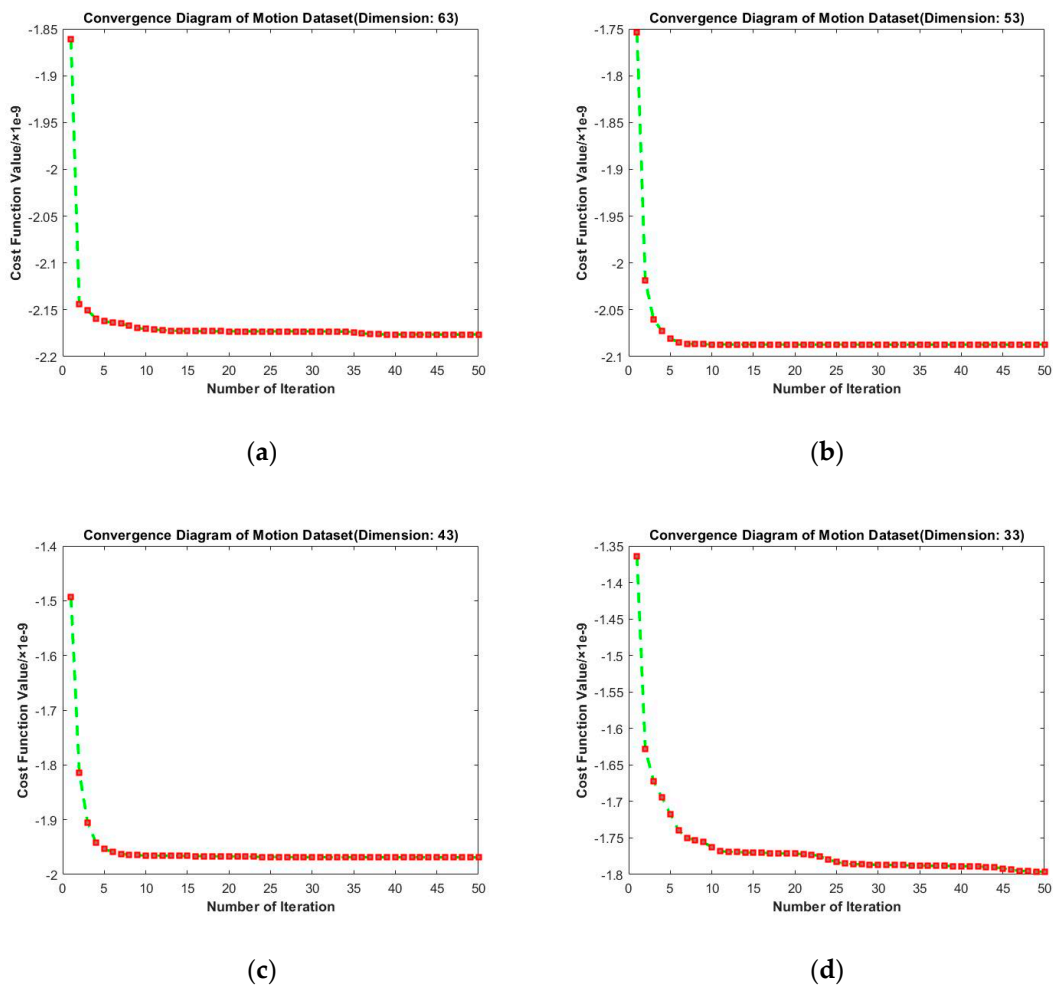


Figure 4. Convergence diagrams on the Motion dataset. The horizontal axis represents the number of iterations and the vertical axis represents the value of the loss function. (a–d) represent the results under different target dimensions (63, 53, 43, and 33) respectively.



### 6.3. The Experiment on the Traffic Dataset

#### 6.3.1. The Description of Traffic

The Traffic video dataset is derived from the UCSD traffic dataset [40], which contains 254 freeway video sequences recorded by a fixed camera from a freeway in Seattle. These sequences can be roughly divided into three traffic conditions (heavy, medium, light; see Figure 5). The three traffic conditions correspond to 44, 45 and 165 sequences, respectively. Each frame was converted to a grayscale image whose size was adjusted to  $125 \times 125$  pixels. We used HoG features [41] to represent frames of a traffic video. In this way, one feature vector corresponded to one frame. In order to acquire the SPD data needed, the feature vectors were normalized and corresponding covariances were calculated. In the experiments on Traffic, we selected 190 sequences for training and the remaining 64 sequences as the test set.



Figure 5. Samples of the Traffic dataset.

#### 6.3.2. The Experimental Results on Traffic

Figure 6 presents the results of the classification experiments on Traffic. The classification accuracies of RMTSISOM-SPDDR are not all the way ahead of other comparison algorithms. This is mainly reflected in the three dimensions of 45, 40 and 30, in which the accuracies of DARG are higher than ours. The possible reason might be that the scene is relatively complex, while DARG firstly adopts a Gaussian mixed model to extract data features, so as to obtain more discriminant information in high dimensions. However, in cases of low dimensions, the accuracies of DARG drop sharply, while ours still remain at a high level. For SPD data, we have already discussed the problems caused by their high dimensionality, so the DR of SPD data is the objective we are committed to. In such a case, RMTSISOM-SPDDR shows superiority in the data classification under low target dimensions, which not only meets the requirements of reducing data dimensionality, but also maintains high classification performance.

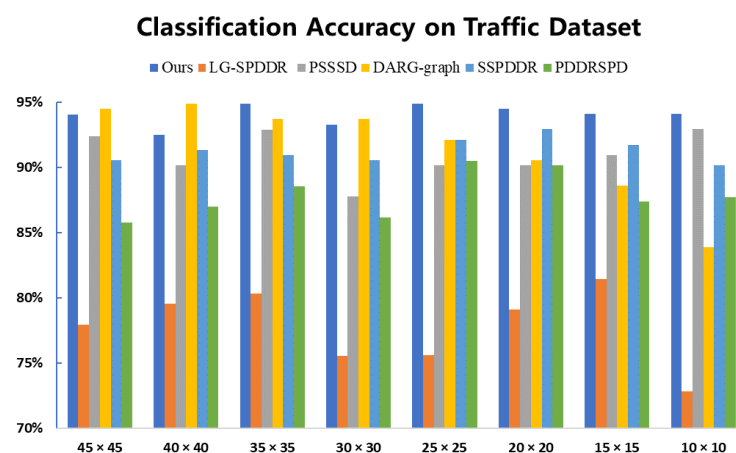
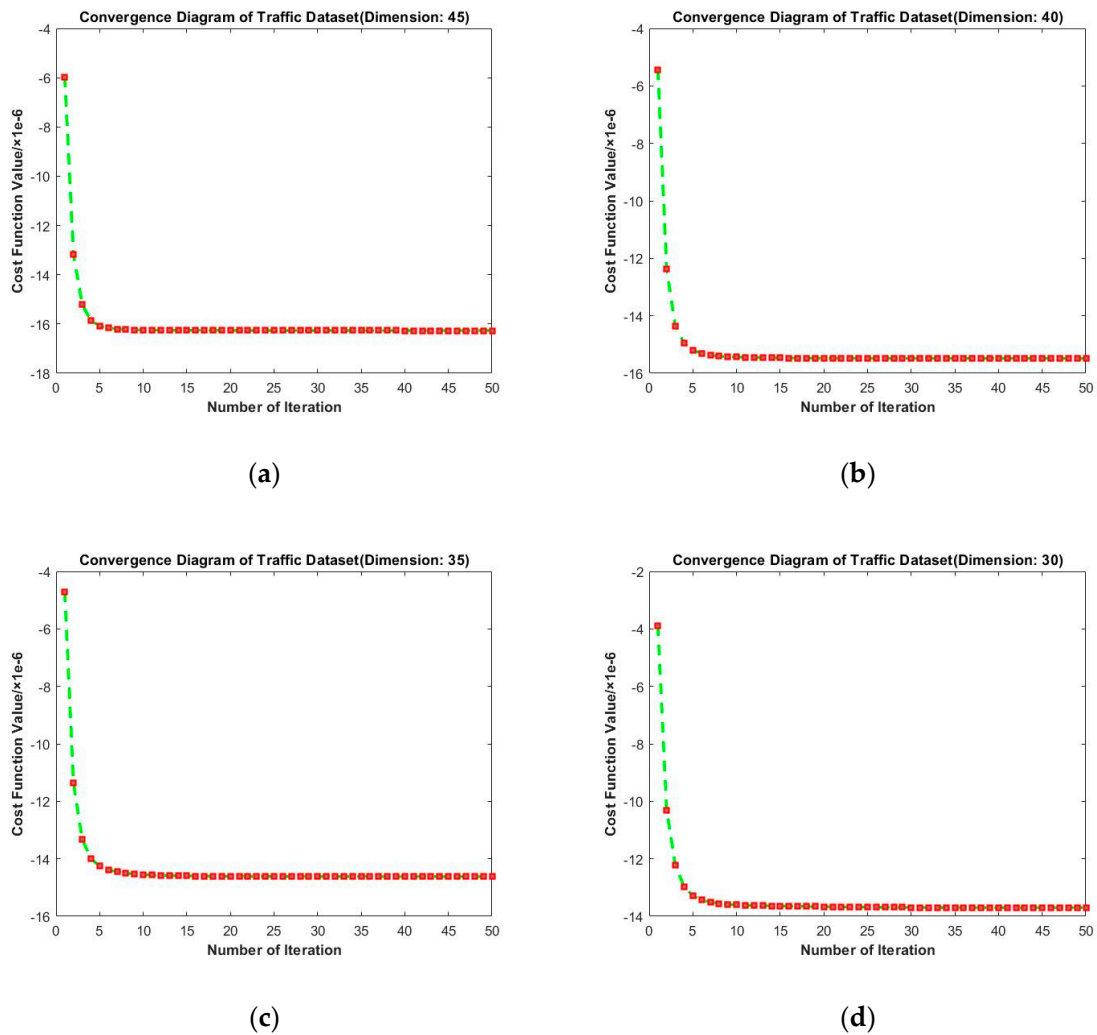


Figure 6. Classification accuracy on the Traffic dataset. The horizontal axis represents the target dimension for DR and the vertical axis represents the classification accuracy.

From Figure 7, the convergence performance of selected target dimensions is as follows: From the horizontal comparison, we find that the convergence curves under these four dimensions tended to be stable after the 10th iteration. It can be observed that RMTSISOM-SPDDR also converged fast on Traffic. Similarly, we find, from the horizontal comparison, that the loss function still decreased under different dimensions, which is reasonable.



**Figure 7.** Convergence diagrams on the Traffic dataset. The horizontal axis represents the number of iterations and the vertical axis represents the value of the loss function. (a–d) represent the results under different target dimensions (45, 40, 35, and 30) respectively.

#### 6.4. The Experiment on the ICT-TV Database

##### 6.4.1. The Description of ICT-TV

The ICT-TV Database [42] consists of two video datasets of TV series, namely, “Big Bang Theory” (BBT) and “Prison Break” (PB), which are shown in Figures 8 and 9. ICT-TV consists of facial video shots taken in these two TV series, namely, 4667 shots in BBT and 9435 shots in PB. The pictures in the shots are stored with the size of  $150 \times 150$  pixels. [30] were followed to preprocess these datasets. We adjusted all detected facial images to  $48 \times 60$  pixels, then adopted histogram equalization, so as to reduce the effect caused by light. All facial frames were flattened into vector representations, on which PCA was performed. Hence, a set of 100-dimensional vectors were obtained which were applied to generate symmetric positive definite kernel matrices. For classification, we utilized the facial videos of 5 main characters in BBT and 11 main characters in PB, respectively.



Figure 8. Samples of BBT Dataset.



Figure 9. Samples of PB Dataset.

#### 6.4.2. The Experimental Results on ICT-TV

Because the size of the datasets (BBT and PB) belonging to the ICT-TV database was excessively large, we segmented them to conduct experiments. We divided BBT (or PB) evenly into 10 sub-datasets. For each sub-dataset, training set and test set were divided from these samples with the ratio of 1:1. Finally, 10 experiments were conducted on these sub-datasets and took average results as final classification accuracies.

As can be seen from Figure 10, all algorithms showed high classification accuracies on BBT, while the result of RMTSISOM-SPDDR is higher than that of comparison algorithms in most dimensions. With the decrease in the target dimension, the performance of algorithms decreases. This may indicate that the lower the dimension is, the more information of the original data is lost, leading to a decline in classification.

### Classification Accuracy on BBT Dataset

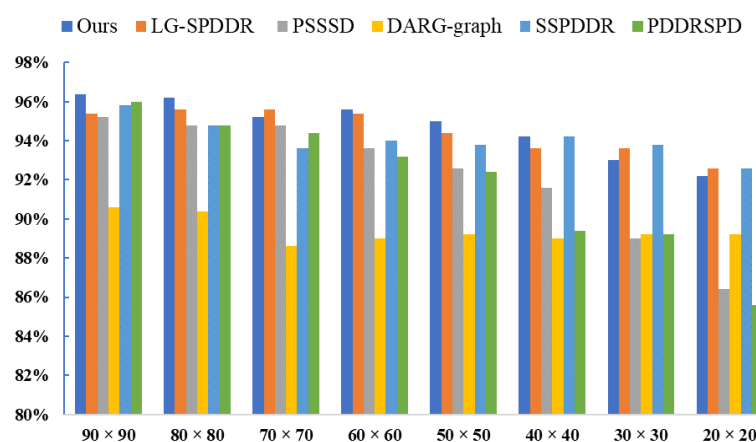
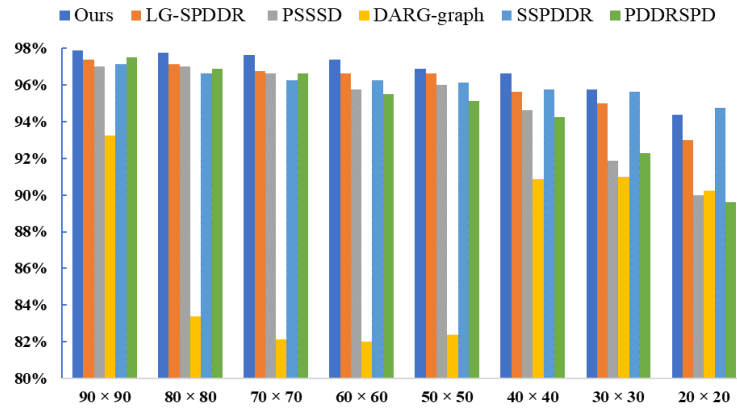


Figure 10. Classification accuracy on the BBT dataset. The horizontal axis represents the target dimension for DR and the vertical axis represents the classification accuracy.

The results of classification experiments on PB are shown in Figure 11. We can see that RMTSISOM-SPDDR is the most outstanding. This is mainly reflected in the fact that,

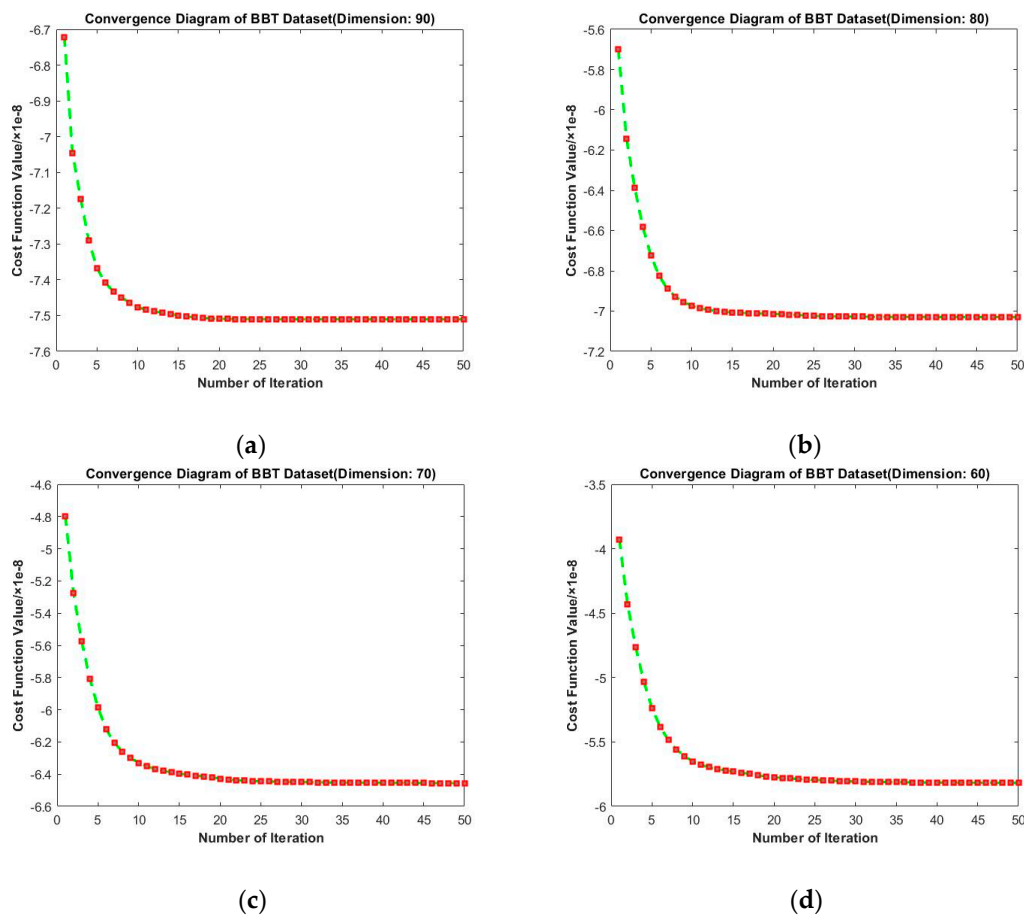
under most target dimensions, our algorithm achieves the highest classification accuracy, while in the case of  $d = 20$ , the accuracy of SSPDDR is the highest, followed by ours.

### Classification Accuracy on PB Dataset



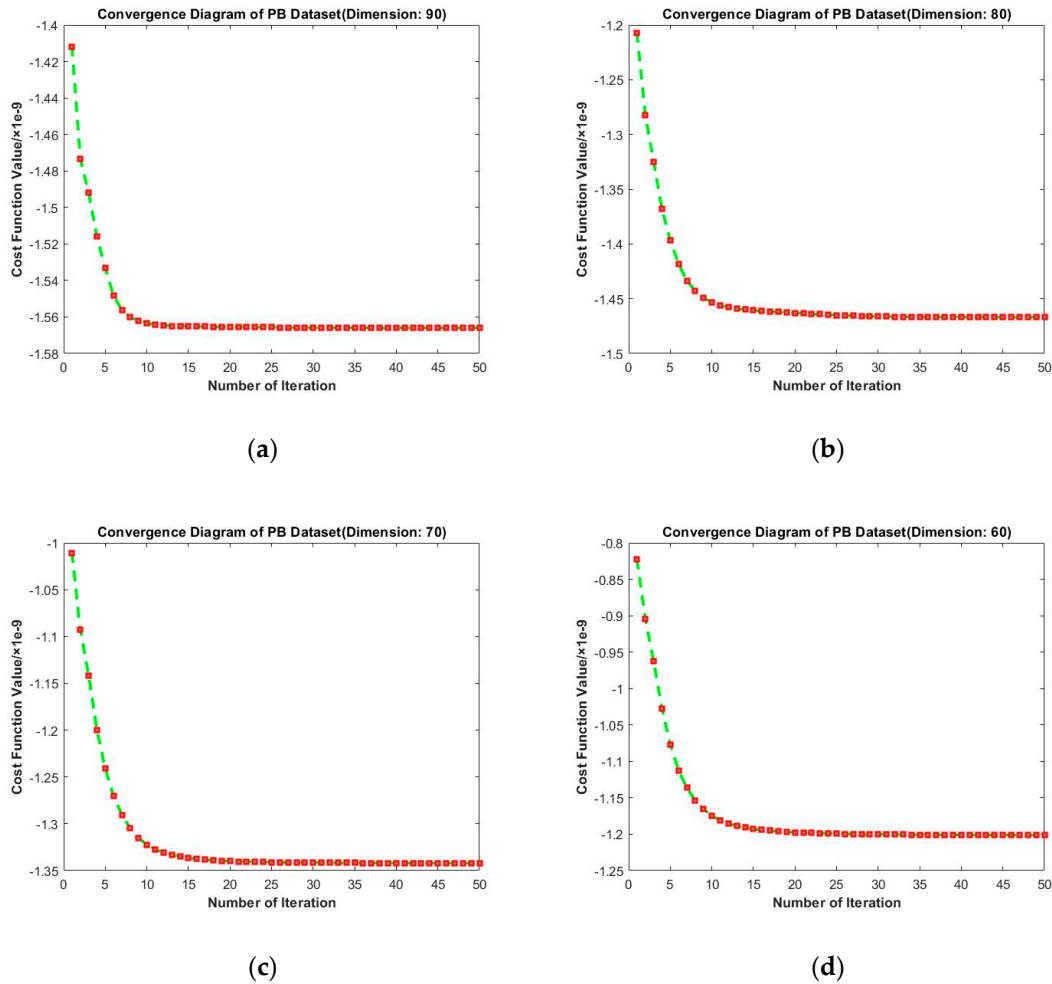
**Figure 11.** Classification accuracy on the PB dataset. The horizontal axis represents the target dimension for DR and the vertical axis represents the classification accuracy.

Convergence diagrams of RMTSISOM-SPDDR on BBT are shown in Figure 12. They indicate that our algorithm basically reflects preferable convergence performance. For each dimension, the curve was highly similar, generally leveling off after 15–20 iterations and there was almost no fluctuation.



**Figure 12.** Convergence diagrams on the BBT dataset. The horizontal axis represents the number of iterations and the vertical axis represents the value of the loss function. (a–d) represent the results under different target dimensions (90, 80, 70, and 60) respectively.

Convergence diagrams of RMTSISOM-SPDDR on PB are shown in Figure 13. It is similar to the cases on BBT. The longitudinal comparison indicates that fifteen iterations were enough to bring the loss function down to an approximately minimum level and the fluctuation since then was very little.



**Figure 13.** Convergence diagrams on the PB dataset. The horizontal axis represents the number of iterations and the vertical axis represents the value of the loss function. (a–d) represent the results under different target dimensions (90, 80, 70, and 60) respectively.

### 6.5. The Experiment on the ETH-80 Dataset

#### 6.5.1. The Description of ETH-80

ETH-80 is a benchmark image set for recognition [43]. It covers eight categories (cars, cups, horses, dogs, cows, tomatoes, peas, apples). One category is comprised of ten objects, each of which consists of 41 pictures taken from different angles of an object, adjusted to  $43 \times 43$  pixels. As shown in Figure 14, objects of the same category are different in terms of physical attributes, such as visual angle, color, texture distribution and shape. For this dataset, we made it into SPD data according to [24]. Specifically, let  $M_i$  be the mean image of the  $i^{th}$  image set,  $M_i = \frac{1}{n_i} \sum_{j=1}^{n_i} C_{ij}$ , where  $n_i$  is the number of images in the  $i^{th}$  image set,  $C_{ij}$  is the  $j^{th}$  image in the  $i^{th}$  image set. Then, the SPD data corresponding to the  $i^{th}$  image set can be calculated by  $X_i^* = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (C_{ij} - M_i)(C_{ij} - M_i)^T$ . Further, we added a correction term to  $X_i^*$  so as to guarantee its positive definiteness, that is,  $X_i = X_i^* + \frac{\text{tr}(X_i^*)}{\lambda} I$ ,  $\lambda = 10^5$ .

Each image set corresponds to an SPD matrix. In the experiment, we evenly divided image sets to training sets and test sets with the ratio of 1:1.



Figure 14. Samples of the ETH-80 dataset.

### 6.5.2. The Experimental Results on ETH-80

It can be seen, from Figure 15, that, under each dimension selected, the classification accuracy of RMTSISOM-SPDDR is generally higher than that of the comparison algorithms and remains at a high level (above 80%). When RMTSISOM-SPDDR reduces the original dimension to different target dimensions, the fluctuation of classification accuracy is relatively small and the dimension can be reduced as much as possible without losing performance. Comparatively speaking, the results of DARG-graph and SSPDDR fluctuate greatly, which means there is a trade-off between classification accuracy and target dimension.

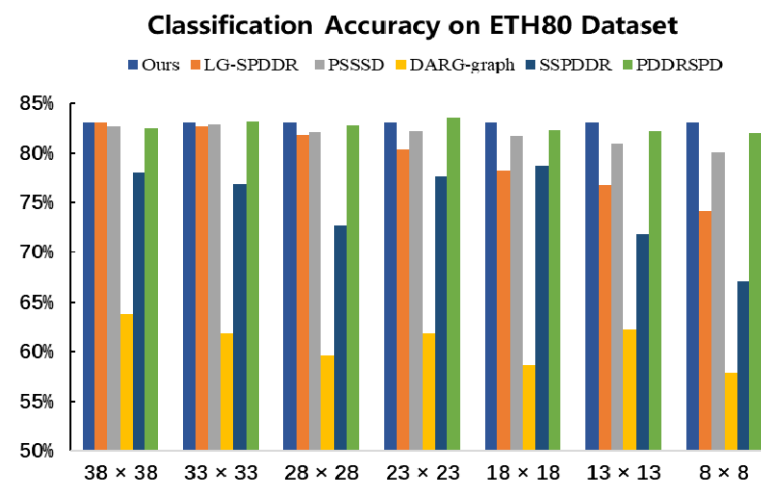
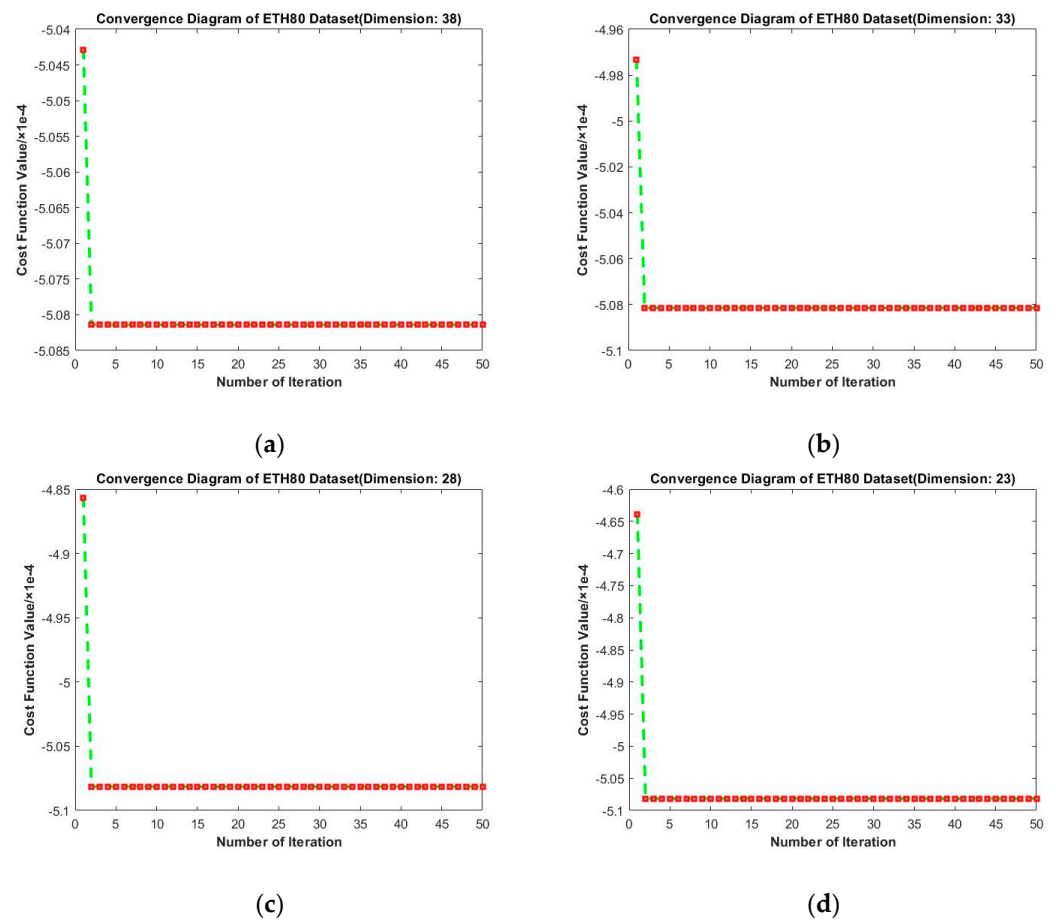


Figure 15. Classification accuracy on the ETH80 dataset. The horizontal axis represents the target dimension for DR and the vertical axis represents the classification accuracy.

Figure 16 presents results of convergence experiments on ETH-80. The longitudinal comparison shows that the value of loss function gradually decreases with the increase in iteration times. Surprisingly, on ETH80, the second iteration leads to the largest drop and subsequent fluctuations are so small to be ignored. This may be because the initial projection matrix for this dataset was already near the local optimal solution. Moreover, the depression of the optimal solution is relatively deep, so that the results of subsequent iterations do not deviate from the optimal solution. From the horizontal comparison, we find that the loss function still tends to decrease under different target dimensions. Hence, our algorithm consistently shows excellent performance.



**Figure 16.** Convergence diagrams on the ETH80 dataset. The horizontal axis represents the number of iterations and the vertical axis represents the value of the loss function. (a–d) represent the results under different target dimensions (38, 33, 28, and 23) respectively.

## 7. Discussions and Conclusions

- (1) As a non-Euclidean data form, SPD data have in-depth applications in machine learning. Compared with vector representation, SPD data can more effectively extract higher-order statistical information. Generally, in order to avoid the problems of high computational complexity and sparse distribution in high-dimensional SPD manifolds, we hope to reduce the dimensionality of SPD data while maintaining useful and representative information. However, the nonlinear manifold structure results in difficulties in learning tasks where linear operations are needed.
- (2) SPD data equipped with a Riemannian metric generally constitute a Riemannian manifold. For Riemannian manifolds, we have known that all tangent spaces are complete finite-dimensional Hilbert spaces, namely, Euclidean spaces. Hence, this paper transfers learning tasks from original SPD manifolds to their tangent spaces.

- (3) The tangent space of an SPD manifold is a symmetric matrix space. From the perspective of data form and attributes, it can be proved to be the minimum linear extension of the original manifold. Inspired by this, we map SPD data into the tangent space at identity by the isometric transformation ( $\log$ ).
- (4) A framework is proposed for SPD data dimensionality reduction (DR). RMTSISOM-SPDDR realizes the procedure through the bilinear transformation between tangent spaces at identity matrices. These tangent spaces are Hilbert spaces and the required bilinear transformation can be determined to a specific criterion and then taken as the DR transformation for the original SPD manifold.
- (5) This paper specifies the bilinear transformation by the global isometric criterion. The so-called isometric criterion means that the geodesic distance on the original high-dimensional SPD manifold is equal to the corresponding Euclidean distance in the tangent space of the low-dimensional SPD manifold. This preserves the distance relationship between data points well.
- (6) In comparison to many existing state-of-the-art algorithms, such as PDDRSPD [34], PSSSD [36], DARG-Graph [37], SSPDDR [32] and LG-SPDDR [38], there are three differences between them and our proposed method (RMTSISOM-SPDDR). First, most of these algorithms are based on the bilinear transformation between two manifolds, while the proposed RMTSISOM-SPDDR is based on the bilinear transformation between the tangent spaces of the two manifolds. The tangent spaces are finite Hilbert spaces, i.e., Euclidean spaces, which can support more complex DR models. Second, all of these algorithms utilize local analysis, while the proposed RMTSISOM-SPDDR is based on global isometry, providing a different perspective. Finally, all of these algorithms are supervised, while the proposed RMTSISOM-SPDDR is unsupervised.

**Author Contributions:** Conceptualization, W.G. (Wenxu Gao) and Z.M.; methodology, Z.M. and W.G. (Wenxu Gao); software, W.G. (Wenxu Gao); validation, Z.M., W.G. (Wenxu Gao) and S.L.; formal analysis, Z.M.; investigation, W.G. (Weichao Gan); resources, S.L.; data curation, Z.M.; writing—original draft preparation, W.G. (Wenxu Gao); writing—review and editing, W.G. (Weichao Gan). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Science and Technology Program of Guangzhou, China under Grant 68000-42050001 and the Natural Science Foundation of China under Grant 61773022.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors are grateful to Ting Gao for academic exchange and Sun Yat-sen University for the information supportive platform.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Differential Manifolds and Tangent Spaces

The so-called topological manifold [44] is a topological space which is locally homeomorphic to a Euclidean space  $R^n$ . If  $M$  is an  $n$ -dimensional topological manifold, there must be a local coordinate system  $\{(U_i, \varphi_i) | i \in I\}$ , where  $U_i$  is an open set of  $M$ ,  $\varphi_i$  is a homeomorphic mapping from  $U_i$  to an open set of  $R^n$  and  $M = \bigcup_{i \in I} U_i$ .  $(U_i, \varphi_i)$  is then called a local coordinate of  $M$ .

With regard to any two local coordinates of the topological manifold  $M$ ,  $(U, \varphi)$  and  $(V, \xi)$ , which satisfy  $U \cap V \neq \emptyset$ ,  $\xi \circ \varphi^{-1} : \varphi(U \cap V) \rightarrow \xi(U \cap V)$  is a mapping between two open sets of  $R^n$ . If  $\xi \circ \varphi^{-1}$  is infinitely differentiable,  $M$  will be called a differential manifold [44]. Differentiable functions, smooth curves and tangent vectors can be defined on differential manifolds.

Let  $f : U_f \rightarrow R$  ( $U_f$  is an open set of  $M$ ) and  $(U, \varphi)$  be a local coordinate of  $M$  which satisfies  $U \cap U_f \neq \emptyset$ .  $f \circ \varphi^{-1} : \varphi(U \cap U_f) \rightarrow R$  is a function which is called the pull-



back function of  $f$  defined in  $R^n$ . If  $f \circ \varphi^{-1}$  is infinitely differentiable, then we call  $f$  a differentiable function.

Similarity, let  $\rho : (-\varepsilon, \varepsilon) \rightarrow M$  be a curve on  $M$ ,  $t_0 \in (-\varepsilon, \varepsilon)$ . It follows that  $\rho(t_0) \in M$ . If  $\rho(t_0) \in U$  for any local coordinate  $(U, \varphi)$ , the function  $\varphi \circ \rho : \rho^{-1}(\rho(-\varepsilon, \varepsilon) \cap U) \rightarrow \varphi(\rho(-\varepsilon, \varepsilon) \cap U)$  is infinitely differentiable,  $\rho$  will be infinitely differentiable at  $t_0$ . If  $\rho$  is infinitely differentiable at each point of  $(-\varepsilon, \varepsilon)$ , then  $\rho$  is called a smooth curve on  $M$ .

Suppose  $p \in M$ ,  $G_p(M)$  represents the set of entire differentiable functions defined in the domain containing  $p$  on  $M$ . Let  $\Phi_p : G_p(M) \rightarrow R$ . If  $\Phi_p$  satisfies the following two condition, for any  $f, g \in G_p(M)$  and  $\alpha, \beta \in R$ ,

1. Linear:  $\Phi_p(\partial f + \beta g) = \alpha \Phi_p(f) + \beta \Phi_p(g)$
2. Leibniz Law:  $\Phi_p(fg) = f(p)\Phi_p(f) + g(p)\Phi_p(f)$

then,  $\Phi_p$  is called a tangent vector of  $p$ .

Smooth curves can also be used to represent tangent vectors. Let  $\rho : (-\varepsilon, \varepsilon) \rightarrow M$  be a smooth curve on  $M$ . If  $t_0 \in (-\varepsilon, \varepsilon)$ , then  $\rho(t_0) \in M$ . Define:  $\rho'(t_0) : G_{\rho(t_0)}(M) \rightarrow R$ , for any  $f \in G_{\rho(t_0)}$ ,

$$\rho'(t_0)(f) = \left. \frac{d(f \circ \rho)(t)}{dt} \right|_{t=t_0} \quad (A1)$$

It can be proved that  $\rho'(t_0)$  satisfies linear and Leibniz laws.  $\rho'(t_0)$  is consequently called a tangent vector of  $\rho(t_0)$ . Moreover, we can prove it the other way around, for any tangent vector  $\Phi_p$  of  $p$ , there must be a smooth curve  $\rho$  that goes through  $p$ , such that  $\rho'(t_0) = \Phi_p$ , where  $\rho(t_0) = p$ .

Moreover, let  $T_p(M)$  represent the entire tangent vectors at  $p$ , define

$$(\Phi_p + \Theta_p)(f) = \Phi_p(f) + \Theta_p(f); (\alpha \Phi_p)(f) = \alpha \Phi_p(f) \quad (A2)$$

then  $T_p(M)$  can form a linear space.

## References

1. Zhang, J.; Yu, J.; Tao, D. Local Deep-Feature Alignment for Unsupervised Dimension Reduction. *IEEE Trans. Image Process.* **2018**, *27*, 2420–2432. [[CrossRef](#)] [[PubMed](#)]
2. Tang, J.; Shao, L.; Li, X.; Lu, K. A Local Structural Descriptor for Image Matching via Normalized Graph Laplacian Embedding. *IEEE Trans. Cybern.* **2015**, *46*, 410–420. [[CrossRef](#)] [[PubMed](#)]
3. Yan, S.; Wang, H.; Fu, Y.; Yan, J.; Tang, X.; Huang, T.S. Synchronized Submanifold Embedding for Person-Independent Pose Estimation and Beyond. *IEEE Trans. Image Process.* **2008**, *18*, 202–210. [[CrossRef](#)]
4. Li, X.; Ng, M.K.; Cong, G.; Ye, Y.; Wu, Q. MR-NTD: Manifold Regularization Nonnegative Tucker Decomposition for Tensor Data Dimension Reduction and Representation. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1787–1800. [[CrossRef](#)] [[PubMed](#)]
5. Chen, Y.-L.; Hsu, C.-T.; Liao, H.-Y.M. Simultaneous Tensor Decomposition and Completion Using Factor Priors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 577–591. [[CrossRef](#)] [[PubMed](#)]
6. Du, B.; Zhang, M.; Zhang, L.; Hu, R.; Tao, D. PLTD: Patch-Based Low-Rank Tensor Decomposition for Hyperspectral Images. *IEEE Trans. Multimed.* **2017**, *19*, 67–79. [[CrossRef](#)]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
9. Huang, Z.; Gool, L.V. A riemannian network for SPD matrix learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2036–2042.
10. Huang, Z.; Wang, R.; Shan, S.; Li, X.; Chen, X. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–1 July 2015; Volume 37, pp. 720–729.
11. Hussein, M.E.; Torki, M.; Gowayyed, M.A.; El-Saban, M. Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2466–2472.
12. Faraki, M.; Harandi, M.; Porikli, F. Image set classification by symmetric positive semi-definite matrices. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 1–8.
13. Chen, K.-X.; Wu, X.-J. Component SPD matrices: A low-dimensional discriminative data descriptor for image set classification. *Comput. Vis. Media* **2018**, *4*, 245–252. [[CrossRef](#)]

14. Tuzel, O.; Porikli, F.; Meer, P. Region Covariance: A Fast Descriptor for Detection and Classification. In Proceedings of the Transactions on Petri Nets and Other Models of Concurrency XV, Aachen, Germany, 23–28 June 2019; Springer Science and Business Media LLC.: Berlin, Germany, 2006; pp. 589–600.
15. Tuzel, O.; Porikli, F.; Meer, P. Pedestrian Detection via Classification on Riemannian Manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1713–1727. [[CrossRef](#)]
16. Harandi, M.; Sanderson, C.; Hartley, R.; Lovell, B.C. Sparse Coding and Dictionary Learning for Symmetric Positive Definite Matrices: A Kernel Approach. In *Transactions on Petri Nets and Other Models of Concurrency XV*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 216–229.
17. Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; Harandi, M. Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 73–80.
18. Harandi, M.; Salzmann, M.; Hartley, R. From Manifold to Manifold: Geometry-Aware Dimensionality Reduction for SPD Matrices. In *Proceedings of the Transactions on Petri Nets and Other Models of Concurrency XV*; Springer Science and Business Media LLC.: Berlin, Germany, 2014; pp. 17–32.
19. Arsigny, V.; Fillard, P.; Pennec, X.; Ayache, N. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM J. Matrix Anal. Appl.* **2007**, *29*, 328–347. [[CrossRef](#)]
20. Ilea, I.; Bombrun, L.B.; Said, S.; Berthoumieu, Y. Covariance Matrices Encoding Based on the Log-Euclidean and Affine Invariant Riemannian Metrics. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 506–509.
21. Li, P.; Wang, Q.; Zuo, W.; Zhang, L. Log-Euclidean Kernels for Sparse Representation and Dictionary Learning. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2013; pp. 1601–1608.
22. Yamin, A.; Dayan, M.; Squarcina, L.; Brambilla, P.; Murino, V.; Diwadkar, V.; Sona, D. Comparison Of Brain Connectomes Using Geodesic Distance On Manifold: A Twins Study. In *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 1797–1800.
23. Zhang, J.; Wang, L.; Zhou, L.; Li, W. Learning Discriminative Stein Kernel for SPD Matrices and Its Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1020–1033. [[CrossRef](#)] [[PubMed](#)]
24. Wang, R.; Wu, X.-J.; Chen, K.-X.; Kittler, J. Multiple Manifolds Metric Learning with Application to Image Set Classification. In *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 627–632.
25. Huang, Z.; Wang, R.; Shan, S.; Van Gool, L.; Chen, X. Cross Euclidean-to-Riemannian Metric Learning with Application to Face Recognition from Video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2827–2840. [[CrossRef](#)] [[PubMed](#)]
26. Vemulapalli, R.; Pillai, J.K.; Chellappa, R. Kernel Learning for Extrinsic Classification of Manifold Features. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2013; pp. 1782–1789.
27. Sanin, A.; Sanderson, C.; Harandi, M.T.; Lovell, B.C. Spatio-temporal covariance descriptors for action and gesture recognition. In Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV), Washington, DC, USA, 15–17 January 2013; pp. 103–110.
28. Vemulapalli, R.; Jacobs, D. Riemannian Metric Learning for Symmetric Positive Definite Matrices. *arXiv* **2015**, arXiv:1501.02393.
29. Zhou, L.; Wang, L.; Zhang, J.; Shi, Y.; Gao, Y. Revisiting Metric Learning for SPD Matrix Based Visual Representation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 7111–7119.
30. Dong, Z.; Jia, S.; Zhang, C.; Pei, M.; Wu, Y. Deep manifold learning of symmetric positive definite matrices with application to face recognition. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4009–4015.
31. Wang, H.; Banerjee, A.; Boley, D. Common component analysis for multiple covariance matrices. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '11, San Diego, CA, USA, 21–24 August 2011; pp. 956–964.
32. Harandi, M.; Salzmann, M.; Hartley, R. Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 48–62. [[CrossRef](#)] [[PubMed](#)]
33. Weinberger, K.Q.; Saul, L.K. Unsupervised Learning of Image Manifolds by Semidefinite Programming. *Int. J. Comput. Vis.* **2006**, *70*, 77–90. [[CrossRef](#)]
34. Ren, J.; Wu, X.-J. Probability Distribution-Based Dimensionality Reduction on Riemannian Manifold of SPD Matrices. *IEEE Access* **2020**, *8*, 153881–153890. [[CrossRef](#)]
35. Grey, D.R. Multivariate analysis, by K. V. Mardia, J.T. Kent and J. M. Bibby. Pp 522. £14.60. 1979. ISBN 0 12 471252 5 (Academic Press). *Math. Gaz.* **1981**, *65*, 75–76. [[CrossRef](#)]
36. Gao, Z.; Wu, Y.; Harandi, M.; Jia, Y. A Robust Distance Measure for Similarity-Based Classification on the SPD Manifold. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3230–3244. [[CrossRef](#)]
37. Wang, W.; Wang, R.; Huang, Z.; Shan, S.; Chen, X. Discriminant Analysis on Riemannian Manifold of Gaussian Distributions for Face Recognition with Image Sets. *IEEE Trans. Image Process.* **2017**, *27*, 1. [[CrossRef](#)]

38. Xu, C.; Lu, C.; Gao, J.; Zheng, W.; Wang, T.; Yan, S. Discriminative Analysis for Symmetric Positive Definite Matrices on Lie Groups. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1576–1585. [[CrossRef](#)]
39. Müller, M.; Röder, T.; Clausen, M.; Eberhardt, B.; Krüger, B.; Weber, A. Documentation Mocap Database HDM05. 2007. Available online: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.7245> (accessed on 23 August 2021).
40. Chan, A.B.; Vasconcelos, N. Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2005; Volume 841, pp. 846–851.
41. Harandi, M.; Salzmann, M.; Baktashmotlagh, M. Beyond Gauss: Image-Set Matching on the Riemannian Manifold of PDFs. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV); Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2015; pp. 4112–4120.
42. Li, Y.; Wang, R.; Shan, S.; Chen, X. Hierarchical hybrid statistic based video binary code and its application to face retrieval in TV-series. In Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia, 4–8 May 2015; Volume 1, pp. 1–8.
43. Leibe, B.; Schiele, B. Analyzing appearance and contour based methods for object categorization. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, 18–20 June 2003; pp. 409–415.
44. Munkres, J.R.; Auslander, L.; MacKenzie, R.E. Introduction to Differentiable Manifolds. *Am. Math. Mon.* **1964**, *71*, 1059. [[CrossRef](#)]