Deep Manifold Learning of Symmetric Positive Definite Matrices with Application to Face Recognition

Zhen Dong,[1] Su Jia,[2] Chi Zhang,[1] Mingtao Pei,[1] Yuwei Wu[1]*
1. Beijing Laboratory of Intelligent Information Technology,
School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
2. Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, USA
dongzhen@bit.edu.cn, su.jia@stonybrook.edu, {zhangchi, peimt, wuyuwei}@bit.edu.cn

Abstract

In this paper, we aim to construct a deep neural network which embeds high dimensional symmetric positive definite (SPD) matrices into a more discriminative low dimensional SPD manifold. To this end, we develop two types of basic layers: a 2D fully connected layer which reduces the dimensionality of the SPD matrices, and a symmetrically clean layer which achieves non-linear mapping. Specifically, we extend the classical fully connected layer such that it is suitable for SPD matrices, and we further show that SPD matrices with symmetric pair elements setting zero operations are still symmetric positive definite. Finally, we complete the construction of the deep neural network for SPD manifold learning by stacking the two layers. Experiments on several face datasets demonstrate the effectiveness of the proposed method.

Introduction

Symmetric positive definite (SPD) matrices have shown powerful representation abilities of encoding image and video information. In computer vision community, the SPD matrix representation has been widely employed in many applications, such as face recognition (Pang, Yuan, and Li 2008; Huang et al. 2015; Wu et al. 2015; Li et al. 2015), object recognition (Tuzel, Porikli, and Meer 2006; Jayasumana et al. 2013; Harandi, Salzmann, and Hartley 2014; Yin et al. 2016), action recognition (Harandi et al. 2016), and visual tracking (Wu et al. 2015).

The SPD matrices form a Riemannian manifold, where the Euclidean distance is no longer a suitable metric. Previous works on analyzing the SPD manifold mainly fall into two categories: the local approximation method and the kernel method, as shown in Figure 1(a). The local approximation method (Tuzel, Porikli, and Meer 2006; Sivalingam et al. 2009; Tosato et al. 2010; Carreira et al. 2012; Vemulapalli and Jacobs 2015) locally flattens the manifold and approximates the SPD matrix by a point of the tangent space. The kernel method (Harandi et al. 2012; Wang et al. 2012; Jayasumana et al. 2013; Li et al. 2013; Quang, San Biagio, and Murino 2014; Yin et al. 2016) embeds the manifold into a higher dimensional Reproducing Kernel Hilbert Space (RKHS) via kernel functions. On new spaces, both methods convert the SPD matrix into a vector and learn a corresponding discriminative representation. However, both local approximation and kernel methods face two problems. First, the SPD matrices are high dimensional, which brings the problem of high computational cost. Second, the vectorization operation on SPD matrices might give rise to the distortion of the manifold geometrical structure.

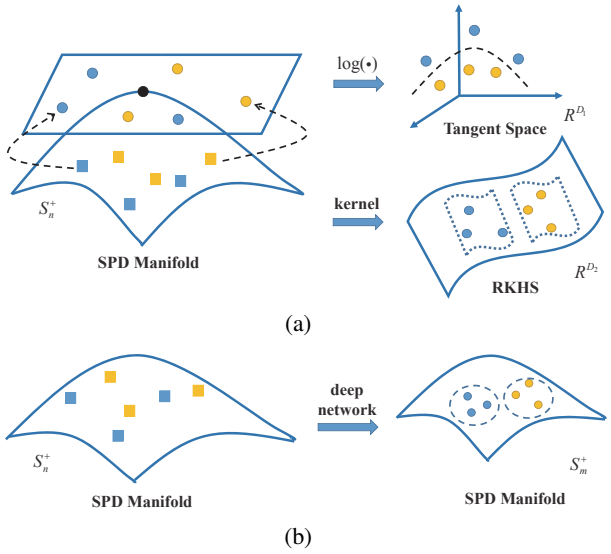


Figure 1: The comparison between our method and previous methods on analyzing the SPD manifold. (a) Previous methods either locally flatten the SPD manifold via tangent space approximation, or embed the manifold into a higher dimensional reproducing kernel Hilbert space. (b) Our method aims to find a non-linear mapping that projects high dimensional SPD matrices into a lower dimensional SPD manifold.

To overcome the two problems mentioned above, we focus on learning a non-linear mapping which projects high dimensional SPD matrices to a low dimensional discriminative SPD manifold, as shown in Figure 1(b). Recently, the deep neural network has shown strong capability of describing complex non-linear maps and been successfully applied on many vision tasks, such as image classification (Krizhevsky,

*corresponding author

Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016) and face recognition (Sun, Wang, and Tang 2015; Taigman et al. 2015; Schroff, Kalenichenko, and Philbin 2015; Parkhi, Vedaldi, and Zisserman 2015). Motivated by these achievements of deep networks, we advocate modeling the non-linear mapping which reduces the dimensionality of high dimensional SPD matrices via a deep neural network.

To achieve this goal, two key issues need to be addressed: dimension reduction and non-linear operation. We introduce two basic layers, *i.e.*, the 2D fully connected layer and the symmetrically clean layer, to realize dimension reduction and non-linear operation, respectively. The 2D fully connected layer reduces the dimensionality of the SPD matrices via a linear mapping, and the symmetrically clean layer sets the symmetric pairs of elements in the SPD matrix as zeros to add non-linearity to the mapping. The two layers should ensure that the output matrices are symmetric positive definite. We thus provide the necessary and sufficient condition for the 2D fully connected layer, and prove that the symmetrically clean layer keeps the symmetry and positive definite properties of SPD matrices. Based on the two layers, the deep neural network for SPD manifold learning is constructed and evaluated on the face recognition tasks. Our network has several advantages compared with the traditional methods on analyzing the SPD manifold. First, learning discriminative representations in new learned low dimensional SPD space brings low computational cost. Second, our method works on the original SPD matrix instead of the vectorization form, which makes full use of the manifold geometrical structure.

This work is, to the best of our knowledge, the first to exploit the deep neural network to analyze the SPD manifold. The contributions of the paper are two-fold: (1) We propose an non-linear operation on the SPD manifold, and prove that SPD matrices with symmetrically clean operation are still symmetric positive definite. (2) The proposed deep neural network is able to project high dimensional SPD matrices to a low dimensional discriminative SPD manifold, and achieves good performances on the face recognition task.

Related Work

In this section, we briefly review several SPD manifold related work including two aspects: SPD manifold metrics and representative work of learning discriminative functions by these metrics.

Let's define the manifold of $n \times n$ SPD matrices as $\mathbb{S}_n^+$. The SPD matrix to the matrix space is similar as positive number to the real number space. A straightforward metric is the Frobenius norm between SPD matrices which is an extension of the Euclidean measure, but several undesirable effects may occur since the Frobenius norm ignores the manifold geometrical structure, such as the swelling of diffusion tensors (Arsigny et al. 2006; Pennec, Fillard, and Ayache 2006). To overcome the problem, several metrics on Riemannian manifold are introduced. The Affine Invariant Metric (AIM) proposed by (Pennec, Fillard, and Ayache 2006) is defined as

$$\begin{aligned}\delta_A(\boldsymbol{A},\boldsymbol{B}) &= \|\log(\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2})\|_F \\ &= \Big(\sum_{i=1}^{n}(\log\lambda_i)^2\Big)^{1/2},\end{aligned} \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, $\log(\cdot)$ is the matrix logarithm operator, $\boldsymbol{A},\boldsymbol{B}\in\mathbb{S}_n^+$ and λ_i is the generalized eigenvalue of $\boldsymbol{A}$ and $\boldsymbol{B}$, *i.e.*, $\det(\lambda_i\boldsymbol{A}-\boldsymbol{B})=0$. Although the AIM is invariant to affine transformations, it is a high computational burden in practice (Arsigny et al. 2007). To reduce the computation cost, the Stein Metric (SM) is studied and introduced by Sra (2012):

$$\delta_S(\boldsymbol{A},\boldsymbol{B}) = \log\det\Big(\frac{\boldsymbol{A}+\boldsymbol{B}}{2}\Big) - \frac{1}{2}\log\det(\boldsymbol{A}\boldsymbol{B}). \tag{2}$$

The δ_S has several similar properties as δ_A and is less expensive to compute (Cherian et al. 2013). Furthermore, Harandi, Salzmann, and Hartley (2014) proved that the length of any curve is the same under δ_S and δ_A up to a scale of $2\sqrt{2}$. Another metric on $\mathbb{S}_n^+$ is the Log-Euclidean Metric (LEM) which is considered by endowing the SPD manifold a Lie group structure (Arsigny et al. 2006; 2007). The LEM is given by

$$\delta_L(\boldsymbol{A},\boldsymbol{B}) = \|\log(\boldsymbol{A}) - \log(\boldsymbol{B})\|_F. \tag{3}$$

Different from δ_A and δ_S, δ_L is a bi-invariance metric, *i.e.*, $\delta_L(\boldsymbol{A},\boldsymbol{B}) = \delta_L(\boldsymbol{B},\boldsymbol{A})$. Since LEM only needs matrix logarithm and Euclidean operations, its computation cost is much less than the AIM and the SM.

Based on these metrics, a few works are proposed to learn discriminative functions on the SPD manifold. One representative work is (Vemulapalli and Jacobs 2015). Their work first flattens the manifold by projecting SPD matrixes to the tangent space at the point of the identity matrix with the matrix logarithm operator $\log(\cdot)$ for local approximation, and then performs the information theoretic metric learning method for the corresponding vectors of the points on the tangent space. They further conduct experiments on face and object datasets, and obtain good performances.

To consider the local manifold structure of manifold data points, the kernel method embeds the SPD manifold into a higher dimensional RKHS via a kernel function and learns discriminative functions on the new space. Based on the LEM, the Covariance Discriminative Learning (CDL) (Wang et al. 2012) employs a new kernel function and conducts partial least squares or linear discriminant analysis in the new space. Besides, many attempts focus on sparse representation and dictionary learning on SPD matrix with appropriate kernels, such as Riemannian Sparse Representation (RSR) (Harandi et al. 2012; 2016) and online dictionary learning (Zhang et al. 2015) based on the SM, and Log-Euclidean Kernel (LEK) (Li et al. 2013) and Manifold Kernel Sparse Representation (MKSR) (Wu et al. 2015) based on the LEM. Yin et al. (2016) further proposed a sparse subspace clustering method for the SPD manifold via an LEM based kernel.

To handle the problems of high computation cost and manifold geometrical structure distortion which methods

mentioned above face, two works are proposed to reduce the dimensionality of the SPD matrix. Harandi, Salzmann, and Hartley (2014) learned a linear mapping which projects the high dimensional SPD manifold into a lower one. The objective function of the learning method encodes the information of intra-class and inter-class distances based on the AIM and the SM. Similarly, Huang et al. (2015) learned a discriminative metric for the SPD manifold. The metric is achieved by reducing the dimensionality of the logarithm of the SPD matrix via a linear mapping. Different from these two works, our method aims at learning a non-linear mapping for SPD matrices via a deep network to facilitate challenging scenarios.

Deep Manifold Learning of SPD Matrices

Although the layers of the neural network for SPD manifold learning is similar to the classical neural network, deep manifold learning of SPD matrices is not straightforward, because each layer of the proposed network takes an SPD matrix as the input and the output must be still an SPD matrix. To achieve this, two types of layers are introduced: one is a 2D fully connected layer for dimensionality reduction of the SPD matrix, and the other is a symmetrically clean layer to ensure the non-linearity of the mapping.

2D Fully Connected Layer

The classical fully connected layer is used for 1D vector, and we extend the layer to make it applicable to 2D matrices and name the new layer 2D Fully Connected Layer. Let $\boldsymbol{X} \in \mathbb{R}^{m_1 \times n_1}$ and $\boldsymbol{Y} \in \mathbb{R}^{m_2 \times n_2}$ be the input and output matrices, respectively. We use $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_{m_1}$ to represent the m_1 rows of $\boldsymbol{X}$, and $\boldsymbol{X}^1, \boldsymbol{X}^2, ..., \boldsymbol{X}^{n_1}$ to represent the n_1 columns of $\boldsymbol{X}$ for simplicity. Figure 2 shows that the 2D fully connected layer is constructed in two steps. First, the neurons in each row $\boldsymbol{X}_i$ are fully connected to n_2 new neurons via the parameter of $\boldsymbol{U}_i \in \mathbb{R}^{n_1 \times n_2}$, which thus generates a new matrix $\boldsymbol{Z} \in \mathbb{R}^{m_1 \times n_2}$:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_{m_1} \end{bmatrix} \rightarrow \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X}_1\boldsymbol{U}_1 \\ \boldsymbol{X}_2\boldsymbol{U}_2 \\ \vdots \\ \boldsymbol{X}_{m_1}\boldsymbol{U}_{m_1} \end{bmatrix}. \tag{4}$$

Second, all the m_1 neurons in each column of $\boldsymbol{Z}$ are fully connected to the m_2 neurons in the corresponding column of $\boldsymbol{Y}$ with the parameter $\boldsymbol{V}_i \in \mathbb{R}^{m_1 \times m_2}$, which is formulated as

$$\begin{gathered} \boldsymbol{Z} = \left[\boldsymbol{Z}^1, \boldsymbol{Z}^2, \ldots, \boldsymbol{Z}^{n_2}\right] \\ \downarrow \\ \boldsymbol{Y} = \left[\boldsymbol{V}_1^\top \boldsymbol{Z}^1, \boldsymbol{V}_2^\top \boldsymbol{Z}^2, \ldots, \boldsymbol{V}_{n_2}^\top \boldsymbol{Z}^{n_2}\right]. \end{gathered} \tag{5}$$

For low complexity of training, the parameters can be shared and reduced in the form of

$$\begin{aligned} \boldsymbol{U} &= \boldsymbol{U}_1 = \boldsymbol{U}_2 = \ldots = \boldsymbol{U}_{m_1}, \\ \boldsymbol{V}^\top &= \boldsymbol{V}_1^\top = \boldsymbol{V}_2^\top = \ldots = \boldsymbol{V}_{n_2}^\top. \end{aligned} \tag{6}$$

Considering the symmetry property of the SPD matrix, *i.e.*, $m_1 = n_1 = n$ and $m_2 = n_2 = m$, the parameters can be further shared: $\boldsymbol{U} = \boldsymbol{V} = \boldsymbol{W}$. The 2D fully connected layer for an SPD matrix is thus formulated as

$$\boldsymbol{Y} = \boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{W}, \tag{7}$$

where $\boldsymbol{X} \in \mathbb{S}_n^+$ is the input, $\boldsymbol{Y} \in \mathbb{S}_m^+ (m < n)$ is the output, and $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ is the parameter. To ensure that $\boldsymbol{Y}$ is also positive definite, $\boldsymbol{W}$ should be column full rank.

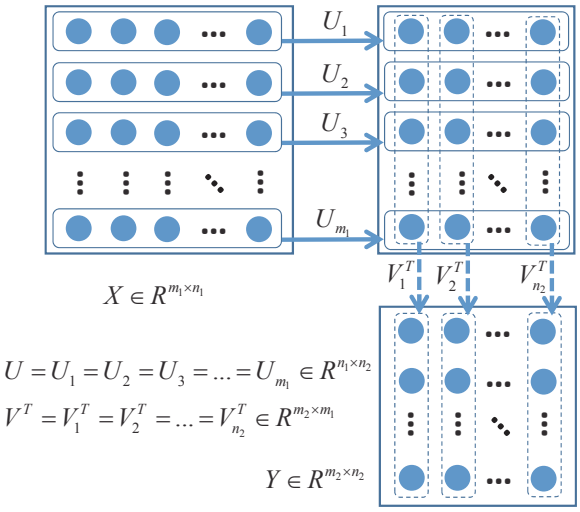


Figure 2: The illustration of the 2D fully connected layer. Neurons in the rectangles connecting to an arrow are fully connected each other. The solid line represents the first step, and the dotted line represents the second step.

Symmetrically Clean Layer

A non-linear operation should be endowed to the SPD matrix to ensure the non-linearity of the mapping. To describe the operation, we first define a function $f(\cdot)$ as

Definition 1. *For a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and a set of ordered pairs $\mathbb{S} \subset \mathbb{T} \times \mathbb{T}$ where $\mathbb{T} = \{1, 2, ..., n\}$, we define $f(\boldsymbol{A}, \mathbb{S})$ as a new $n \times n$ square matrix $\boldsymbol{B}$:*

$$\boldsymbol{B}_{ij} = \begin{cases} 0, & \text{if } (i,j) \in \mathbb{S} \text{ or } (j,i) \in \mathbb{S}, \\ \boldsymbol{A}_{ij}, & \text{otherwise}. \end{cases} \tag{8}$$

The non-linear operation is then defined as

$$\boldsymbol{Y} = f(\boldsymbol{X}, \mathbb{S}), \tag{9}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the input and output of this layer, respectively, and $\mathbb{S} \subset \mathbb{T} \times \mathbb{T} - \{(i,j) | i = j\}, \mathbb{T} = \{1, 2, ..., n\}$. It's easy to verify that $\boldsymbol{Y}$ is symmetric if $\boldsymbol{X}$ is a symmetric matrix. Eq.(9) indicates that the operation is setting some symmetric pairs of elements as zeros, we thus call this layer symmetrically clean layer. Here, we have Proposition 1 to ensure that $\boldsymbol{Y}$ is also an SPD matrix, and the proof of the proposition is in the appendix.

Proposition 1. *Let $\boldsymbol{B} \in \mathbb{S}_n^+$ be an SPD matrix, and $\mathbb{S} \subset \mathbb{T} \times \mathbb{T} - \{(i,j) | i = j\}$ where $\mathbb{T} = \{1, 2, ..., n\}$, then $f(\boldsymbol{B}, \mathbb{S})$ is still an SPD matrix.*

Table 1: Comparison results with other methods on YTC and ICT-TV datasets.

Methods	YTC	ICT-TV-BBT	ICT-TV-PB
AIM (Pennec, Fillard, and Ayache 2006)	30.33 ± 3.72	37.64 ± 3.14	11.29 ± 2.46
SM (Sra 2012)	28.85 ± 3.41	38.07 ± 2.93	11.41 ± 2.85
LEM (Arsigny et al. 2007)	31.34 ± 3.64	40.89 ± 3.08	13.36 ± 2.72
SPDML (Harandi, Salzmann, and Hartley 2014)	40.86 ± 3.24	41.52 ± 2.14	17.94 ± 2.82
RSR (Harandi et al. 2012)	34.01 ± 3.06	47.93 ± 2.72	15.52 ± 2.30
LEK (Li et al. 2013)	33.81 ± 3.83	44.16 ± 2.71	16.74 ± 2.74
CDL (Wang et al. 2012)	31.84 ± 2.54	44.38 ± 2.28	15.26 ± 2.06
ITML-LEM (Vemulapalli and Jacobs 2015)	33.42 ± 3.42	46.62 ± 2.03	14.39 ± 2.52
LEML (Huang et al. 2015)	38.04 ± 2.11	49.60 ± 2.57	18.73 ± 2.20
DCC (Kim, Kittler, and Cipolla 2007)	32.84 ± 3.61	46.68 ± 3.04	15.31 ± 2.83
GDA (Hamm and Lee 2008)	32.09 ± 3.17	46.14 ± 2.98	17.03 ± 2.91
AHISD (Cevikalp and Triggs 2010)	31.16 ± 3.04	41.62 ± 2.72	15.88 ± 2.25
CHISD (Cevikalp and Triggs 2010)	32.08 ± 2.66	45.24 ± 2.58	16.52 ± 2.91
SSDML (Zhu et al. 2013)	34.77 ± 2.59	42.36 ± 2.47	13.71 ± 3.07
Our method	$\mathbf{46.37 \pm 3.07}$	$\mathbf{55.18 \pm 2.94}$	$\mathbf{24.18 \pm 2.05}$

The Proposition 1 shows that cleaning any symmetric pairs (no principal diagonal elements) is capable, and the ReLU operation is used here for simplicity. The ReLU operation assigns all the negative elements in the SPD matrix zeros. The principal diagonal elements of an SPD matrix will be positive forever, so the ReLU operation is a suitable choice.

Stacking the two types of basic layers, we construct a deep neural network to reduce the dimensionality of the SPD manifold. At the end of the network, a triplet loss is utilized to measure the distances between low-dimensional SPD matrices.

Experiments

To verify the effectiveness of the proposed network, we conduct experiments on two face datasets: YouTube Celebrities (YTC) (Kim et al. 2008) and ICT-TV (Li et al. 2015).

Datasets

The YouTube Celebrities (YTC) dataset (Kim et al. 2008) contains $1,190$ videos clips of 47 individuals (actors, actresses, and politicians) collected from the YouTube website. Each individual has about 41 clips segmented from 3 unique long videos, and the frame number of theses video clips varies from 8 to 400. The face video clips in this dataset exhibit larger variations in pose, illumination, and expressions. What's more, most of the videos are low resolution and recorded at high compression rates, which leads to noisy and low-quality image frames. The YTC dataset is thus a more challenging face video dataset.

The ICT-TV dataset (Li et al. 2015) contains two large scale video collections from two hit American shows, *i.e.*, the Big Bang Theory (BBT) and Prison Break (PB). These two TV series are quite different in their filming styles. The BBT is a sitcom with 5 main characters, and most scenes are taken indoors during each episode of about 20 minutes long. Differently, many shots of the PB are taken outside during the episodes with the length of about 42 minutes, which results in a large range of different illumination. All the face video shots are collected from the whole first season of both TV series, *i.e.*, 17 episodes of BBT, and 22 episodes of PB, and the number of shots of the two sets are $4,667$ and $9,435$, respectively. The collected video shots are stored in the form of images with size of 150×150 frame by frame.

Experimental Setting

On the YTC dataset, we pre-process the face frames as follows for face recognition. All the faces are detected and resized to 48×60 pixels, followed by the histogram equalization for reducing lighting effects. The face frame is flattened into a vector of size $2,880$, and the PCA is executed to reduce the dimension to 100. For each face video clip, the kernel representation which shows better performance than covariance matrix (Wang et al. 2015) is extracted by using these vectors, and the kernel representation is an SPD matrix size of 100×100. For each individual, the gallery set is composed of 3 randomly selected video clips which come from 3 unique videos, and the probe set consists 6 randomly selected video clips where each unique video provides 2 video clips. The recognition accuracy is used as the evaluation criterion.

We preprocess face frames in the ICT-TV dataset in the same way as the YTC dataset. The face videos of 5 and 11 main characters of BBT and PB are used in the experiments, respectively. For each character, the videos are randomly split into training and testing set with the ration of 1:1.

Results

The comparison results are shown in Table 1. Among the comparison methods, the AIM, SM, and LEM are basic metrics and used as the baseline, SPDML and RSR are AIM and SM based supervised methods, LEK, CDL, ITML-LEM, and LEML are LEM based supervised methods, and others are set model methods which are not based on SPD manifold. The RSR and LEM are sparse representation based methods, so they use sparse representation based classifier for the face recognition task, and other methods use the

nearest neighbor classifier. As shown in Table 1, our methods achieves the best performance on the face recognition task, the primal reason is that the proposed deep network describes a good non-linear mapping which projects high dimensional SPD matrices to a low dimensional discriminative SPD manifold, and the end-to-end training manner helps to improve the discriminative power of the low dimensional manifold.

Conclusion

In this paper, we have constructed a deep neural network which projects high dimensional SPD matrices to a more discriminative low dimensional SPD manifold. Two basic layers are introduced to implement the network. The 2D fully connected layer reduces the dimensionality of the SPD matrices, and the symmetrically clean layer aims to add non-linearity to the mapping. Experiments on several face datasets showed the effectiveness of the proposed network.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No.61472038 and No.61375044, and in part by the Double First-rate Program of BIT under Grant 3070012331601.

Appendix

We prove Proposition 1 in the appendix. To prove the proposition, we need two lemmas.

Lemma 1. *Let* $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^\top & b \end{bmatrix}$ *be a square matrix where* $\boldsymbol{D} \in \mathbb{R}^{(n-1)\times(n-1)}, \boldsymbol{\beta} \in \mathbb{R}^{(n-1)\times 1}$*, and* $a \in \mathbb{R}$*. If the determinant* $\det(\boldsymbol{D}) \neq 0$*, then* $\det(\boldsymbol{E}) = \det(\boldsymbol{D})(b - \boldsymbol{\beta}^\top \boldsymbol{D}^{-1}\boldsymbol{\beta})$.

Proof. We use $\boldsymbol{D}^{-1}$ to represent the inverse matrix of $\boldsymbol{D}$ since $\det(\boldsymbol{D}) \neq 0$. Let

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{D}^{-1}\boldsymbol{\alpha} \\ \boldsymbol{0}^\top & 1 \end{bmatrix} \tag{10}$$

where $\boldsymbol{I} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the identity matrix, and $\boldsymbol{0} \in \mathbb{R}^{(n-1)\times 1}$ is a vector whose elements are all zeros, then $\det(\boldsymbol{F}) = 1$.

We have that

$$\boldsymbol{E}\boldsymbol{F} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{\beta}^\top & b - \boldsymbol{\beta}^\top \boldsymbol{D}^{-1}\boldsymbol{\beta} \end{bmatrix}. \tag{11}$$

Therefore, $\det(\boldsymbol{E}) = \det(\boldsymbol{E})\det(\boldsymbol{F}) = \det(\boldsymbol{E}\boldsymbol{F}) = \det(\boldsymbol{D})(b - \boldsymbol{\beta}^\top \boldsymbol{D}^{-1}\boldsymbol{\beta})$. □

Lemma 2. *For any* $\boldsymbol{G} \in \mathbb{S}_n^+$*, the inverse matrix* $\boldsymbol{G}^{-1} \in \mathbb{S}_n^+$.

Proof. We do the following case analysis:

(1) $\boldsymbol{G}^{-1}$ is symmetric.

$$\left(\boldsymbol{G}^{-1}\right)^\top = \left(\boldsymbol{G}^\top\right)^{-1} = \boldsymbol{G}^{-1}. \tag{12}$$

(2) $\boldsymbol{G}^{-1}$ is positive definite.
Since $\boldsymbol{G}$ is a real symmetric matrix and thus a normal matrix, we have that $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\top$ where $\boldsymbol{W}$ is an unitary matrix, *i.e.*, $\boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}$, and $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is a real diagonal matrix whose principal diagonal elements are eigenvalues $\lambda_i > 0$. Then, $\boldsymbol{G}^{-1} = (\boldsymbol{W}^\top)^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{W}^{-1} = \boldsymbol{W}\boldsymbol{\Lambda}^{-1}\boldsymbol{W}^\top$. Note that $\boldsymbol{\Lambda}^{-1} = diag(\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_n^{-1})$ is also a real diagonal matrix whose principal diagonal elements are eigenvalues of $\boldsymbol{G}^{-1}$. Since all the eigenvalues λ_i^{-1} are positive, $\boldsymbol{G}^{-1}$ is a positive definite matrix .

Therefore, $\boldsymbol{G}^{-1}$ is an SPD matrix. □

The proof of Proposition 1 is as follow:

Proof. The proof includes three procedures.

(1) $\forall \boldsymbol{B} \in \mathbb{S}_n^+, f\big(\boldsymbol{B}, \{(n-1, n)\}\big) \in \mathbb{S}_n^+$:
We rewrite $\boldsymbol{B}$ as

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^\top & a \end{bmatrix} \tag{13}$$

where $\boldsymbol{\alpha}^\top = \begin{bmatrix} \boldsymbol{\beta}^\top & b \end{bmatrix}, \boldsymbol{\beta} \in \mathbb{R}^{(n-2)\times 1}$, and $b \in \mathbb{R}$, then

$$\boldsymbol{C} = f\big(\boldsymbol{B}, \{(n-1, n)\}\big) = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{\gamma} \\ \boldsymbol{\gamma}^\top & a \end{bmatrix} \tag{14}$$

where $\boldsymbol{\gamma}^\top = \begin{bmatrix} \boldsymbol{\beta}^\top & 0 \end{bmatrix}$.
Since $\boldsymbol{B}$ is a positive definite matrix, the determinants of all the leading principal submatrices of $\boldsymbol{B}$ are positive, including $\det(\boldsymbol{B}) > 0$. $\boldsymbol{A}$ is one of the leading principal submatrix of $\boldsymbol{B}$, so all the leading principal submatrices of $\boldsymbol{A}$ are positive, and thus $\boldsymbol{A}$ is also an SPD matrix. It's no wonder that $\det(\boldsymbol{A}) > 0$, and according to Lemma 2, the inverse matrix $\boldsymbol{A}^{-1}$ is an SPD matrix.
We prove the hypothesis by contradiction. Assume to the contrary that $\boldsymbol{C}$ is not an SPD matrix. Because the determinants of all leading principal submatrices of $\boldsymbol{A}$ are all positive, the determinant of $\boldsymbol{C}$ is not positive, *i.e.* $\det(\boldsymbol{C}) \leq 0$. According to Lemma 1, we have that $\det(\boldsymbol{C}) = \det(\boldsymbol{A})(a - \boldsymbol{\gamma}^\top \boldsymbol{A}^{-1}\boldsymbol{\gamma})$, and since $\det(\boldsymbol{A}) > 0$, then

$$a \leq \boldsymbol{\gamma}^\top \boldsymbol{A}^{-1}\boldsymbol{\gamma}. \tag{15}$$

Similarly, $\det(\boldsymbol{B}) = \det(\boldsymbol{A})(a - \boldsymbol{\alpha}^\top \boldsymbol{A}^{-1}\boldsymbol{\alpha}) > 0$ and $\det(\boldsymbol{A}) > 0$, we have that

$$a > \boldsymbol{\alpha}^\top \boldsymbol{A}^{-1}\boldsymbol{\alpha}. \tag{16}$$

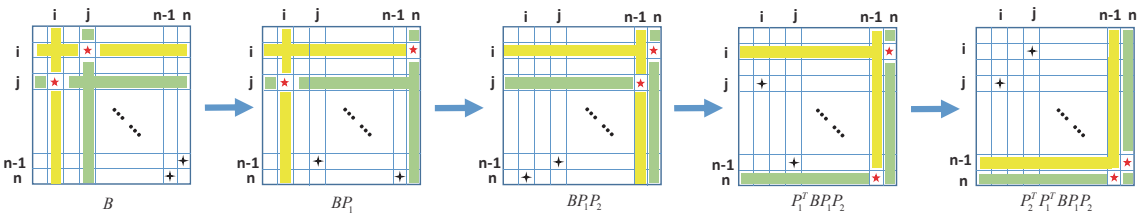
Considering Eq.(15) and Eq.(16) jointly, it's easy to see that

$$\boldsymbol{\alpha}^\top \boldsymbol{A}^{-1}\boldsymbol{\alpha} < \boldsymbol{\gamma}^\top \boldsymbol{A}^{-1}\boldsymbol{\gamma}. \tag{17}$$

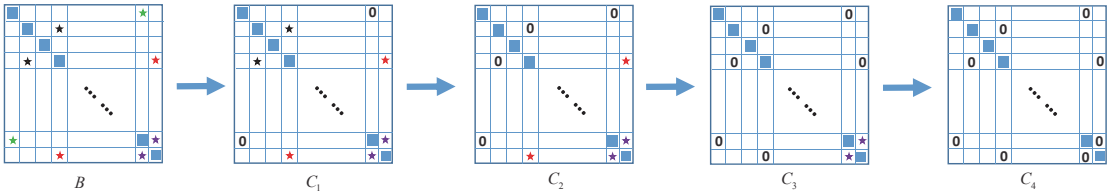
Let $\boldsymbol{\delta}^\top = \begin{bmatrix} \boldsymbol{0}^\top & b \end{bmatrix}$ where $\boldsymbol{0} \in \mathbb{R}^{(n-2)\times 1}$ is a vector of zeros, then

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta} \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ b \end{bmatrix} = \boldsymbol{\gamma} + \boldsymbol{\delta}. \tag{18}$$

Substitute Eq.(18) into Eq.(17), and note that $\boldsymbol{\gamma}^\top \boldsymbol{A}^{-1}\boldsymbol{\delta} = \boldsymbol{\delta}^\top \boldsymbol{A}^{-1}\boldsymbol{\gamma} = 0$, we have $\boldsymbol{\delta}^\top \boldsymbol{A}^{-1}\boldsymbol{\delta} < 0$, which is conflict with that $\boldsymbol{A}^{-1}$ is positive definite and concludes the proof of procedure (1).



(a)



(b)

Figure 3: The basic ideas of 2-nd and 3-rd procedure in the proof of Proposition 1. (a) In the 2-nd step, the matrix $\boldsymbol{C}$ is generated by performing following four interchanging procedures on $\boldsymbol{B}$. The transformation from $\boldsymbol{B}$ to $\boldsymbol{C}$ is invertible, and the inverse of the transformation is itself. (b) An illustration of 3-rd procedure. Let's take $m = 4$ as an example. Colorful stars represent elements to be set to zero, and they are set one by one, *i.e.*, the $f(\boldsymbol{B}, \mathbb{S})$ is performed in 4 steps.

(2) $\forall \boldsymbol{B} \in \mathbb{S}_n^+, f\big(\boldsymbol{B}, \{(i,j)\}\big) \in \mathbb{S}_n^+$ where $(i,j) \in \mathbb{S}$.
Let's consider two $n \times n$ elementary matrices: $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. The $\boldsymbol{P}_1$ is created by interchanging the j-th and the n-th columns of an identity matrix of size $n \times n$, and the $\boldsymbol{P}_2$ is created by interchanging the i-th and the $(n-1)$-th columns of an identity matrix of size $n \times n$. It's easy to prove that

$$\boldsymbol{P}_1 = \boldsymbol{P}_1^\top = \boldsymbol{P}_1^{-1}, \boldsymbol{P}_2 = \boldsymbol{P}_2^\top = \boldsymbol{P}_2^{-1}. \tag{19}$$

We define a matrix $\boldsymbol{P} = \boldsymbol{P}_1\boldsymbol{P}_2$. Please note that $\boldsymbol{P}_1\boldsymbol{P}_2 = \boldsymbol{P}_2\boldsymbol{P}_1$, then $\boldsymbol{P}$ has the property that

$$\boldsymbol{P} = \boldsymbol{P}^\top = \boldsymbol{P}^{-1}. \tag{20}$$

Figure 3(a) shows that a matrix $\boldsymbol{C}$ is generated by performing following four procedures on $\boldsymbol{B}$: interchanging the j-th and the n-th columns, interchanging the i-th and the $(n-1)$-th columns, interchanging the j-th and the n-th rows, and interchanging the i-th and the $(n-1)$-th rows, which can be formulated as

$$\boldsymbol{C} = \boldsymbol{P}^\top \boldsymbol{B} \boldsymbol{P}. \tag{21}$$

In this way, the elements of (i,j) pair and $(n-1,n)$ pair are interchanged. Since $\boldsymbol{B}$ is symmetric and positive definite and $\boldsymbol{P}$ is invertible, $\boldsymbol{C}$ is thus also an SPD matrix. We then perform $f\big(\cdot, \{(n-1,n)\}\big)$ on $\boldsymbol{C}$ and obtain $\boldsymbol{D}$:

$$\boldsymbol{D} = f\big(\boldsymbol{C}, \{(n-1,n)\}\big). \tag{22}$$

As proved in the procedure (1), $\boldsymbol{D}$ is also an SPD matrix. The transformation from $\boldsymbol{B}$ to $\boldsymbol{C}$ is invertible, and the inverse of the transformation is itself, as shown in Eq.(20) and Figure 3(a), so $\boldsymbol{B} = \boldsymbol{P}^\top \boldsymbol{C} \boldsymbol{P}$. Since $\boldsymbol{C}$ and $\boldsymbol{D}$ are only different at the elements of $(n-1,n)$ and $(n,n-1)$,

$$f\big(\boldsymbol{B}, \{(i,j)\}\big) = \boldsymbol{P}^\top \boldsymbol{D} \boldsymbol{P}. \tag{23}$$

We thus have that $f\big(\boldsymbol{B}, \{(i,j)\}\big) \in \mathbb{S}_n^+$ because $\boldsymbol{D} \in \mathbb{S}_n^+$ and $\boldsymbol{P}$ is invertible.

(3) $f(\boldsymbol{B}, \mathbb{S})$ is an SPD matrix.
We rewrite $\mathbb{S} = \{(i_k, j_k) | k = 1, 2, ..., m\}$ and $m \leq n(n-1)/2$. Figure 3(b) shows that the $f(\boldsymbol{B}, \mathbb{S})$ can be performed in m steps:

$$\begin{aligned} \boldsymbol{B} \xrightarrow{f(\cdot,\{(i_1,j_1)\})} \boldsymbol{C}_1 \xrightarrow{f(\cdot,\{(i_2,j_2)\})} \boldsymbol{C}_2 \rightarrow ... \\ ... \rightarrow \boldsymbol{C}_{m-1} \xrightarrow{f(\cdot,\{(i_m,j_m)\})} \boldsymbol{C}_m. \end{aligned} \tag{24}$$

It's easy to verify that $\boldsymbol{C}_m = f(\boldsymbol{B}, \mathbb{S})$. By using the proof in the procedure (2), we have that $\boldsymbol{C}_1, \boldsymbol{C}_2, ...,$ and $\boldsymbol{C}_m$ are all SPD matrices. Therefore, $f(\boldsymbol{B}, \mathbb{S}) \in \mathbb{S}_n^+$.

To sum up, $f(\boldsymbol{B}, \mathbb{S})$ is still an SPD matrix. □

References

Arsigny, V.; Fillard, P.; Pennec, X.; and Ayache, N. 2006. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *MRM* 56(2):411–421.

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

Carreira, J.; Caseiro, R.; Batista, J.; and Sminchisescu, C. 2012. Semantic segmentation with second-order pooling. In *ECCV*, 430–443. Springer.

Cevikalp, H., and Triggs, B. 2010. Face recognition based on image sets. In *CVPR*, 2567–2573. IEEE.

Cherian, A.; Sra, S.; Banerjee, A.; and Papanikolopoulos, N. 2013. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *TPAMI* 35(9):2161–2174.

Hamm, J., and Lee, D. D. 2008. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, 376–383. ACM.

Harandi, M. T.; Sanderson, C.; Hartley, R.; and Lovell, B. C. 2012. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *ECCV*. Springer. 216–229.

Harandi, M. T.; Hartley, R.; Lovell, B.; and Sanderson, C. 2016. Sparse coding on symmetric positive definite manifolds using bregman divergences. *TNNLS* 27(6):1294–1306.

Harandi, M. T.; Salzmann, M.; and Hartley, R. 2014. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *ECCV*, 17–32. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*. IEEE.

Huang, Z.; Wang, R.; Shan, S.; Li, X.; and Chen, X. 2015. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 720–729.

Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; and Harandi, M. 2013. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, 73–80. IEEE.

Kim, M.; Kumar, S.; Pavlovic, V.; and Rowley, H. 2008. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 1–8. IEEE.

Kim, T.-K.; Kittler, J.; and Cipolla, R. 2007. Discriminative learning and recognition of image set classes using canonical correlations. *TPAMI* 29(6):1005–1018.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Li, P.; Wang, Q.; Zuo, W.; and Zhang, L. 2013. Log-euclidean kernels for sparse representation and dictionary learning. In *ICCV*, 1601–1608. IEEE.

Li, Y.; Wang, R.; Shan, S.; and Chen, X. 2015. Hierarchical hybrid statistic based video binary code and its application to face retrieval in tv-series. In *FG*, 1–8. IEEE.

Pang, Y.; Yuan, Y.; and Li, X. 2008. Gabor-based region covariance matrices for face recognition. *TCSVT* 18(7):989–993.

Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition. In *BMVC*, volume 1, 6.

Pennec, X.; Fillard, P.; and Ayache, N. 2006. A riemannian framework for tensor computing. *IJCV* 66(1):41–66.

Quang, M. H.; San Biagio, M.; and Murino, V. 2014. Log-hilbert-schmidt metric between positive definite operators on hilbert spaces. In *NIPS*, 388–396.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 815–823. IEEE.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sivalingam, R.; Morellas, V.; Boley, D.; and Papanikolopoulos, N. 2009. Metric learning for semi-supervised clustering of region covariance descriptors. In *International Conference on Distributed Smart Cameras*, 1–8. IEEE.

Sra, S. 2012. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *NIPS*, 144–152.

Sun, Y.; Wang, X.; and Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2892–2900. IEEE.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9. IEEE.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2015. Web-scale training for face identification. In *CVPR*, 2476–2574. IEEE.

Tosato, D.; Farenzena, M.; Spera, M.; Murino, V.; and Cristani, M. 2010. Multi-class classification on riemannian manifolds for video surveillance. In *ECCV*, 378–391. Springer.

Tuzel, O.; Porikli, F.; and Meer, P. 2006. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 589–600. Springer.

Vemulapalli, R., and Jacobs, D. W. 2015. Riemannian metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1501.02393*.

Wang, R.; Guo, H.; Davis, L. S.; and Dai, Q. 2012. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, 2496–2503. IEEE.

Wang, L.; Zhang, J.; Zhou, L.; Tang, C.; and Li, W. 2015. Beyond covariance: Feature representation with nonlinear kernel matrices. In *ICCV*, 4570–4578. IEEE.

Wu, Y.; Jia, Y.; Li, P.; Zhang, J.; and Yuan, J. 2015. Manifold kernel sparse representation of symmetric positive-definite matrices and its applications. *TIP* 24(11):3729–3741.

Yin, M.; Guo, Y.; Gao, J.; He, Z.; and Xie, S. 2016. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *CVPR*. IEEE.

Zhang, S.; Kasiviswanathan, S.; Yuen, P. C.; and Harandi, M. 2015. Online dictionary learning on symmetric positive definite manifolds with vision applications. In *AAAI*, 3165–3173.

Zhu, P.; Zhang, L.; Zuo, W.; and Zhang, D. 2013. From point to set: Extend the learning of distance metrics. In *ICCV*, 2664–2671. IEEE.