Set-to-Set Distance Metric Learning on SPD Manifolds

Zhi Gao, Yuwei Wu(✉), and Yunde Jia

Beijing Laboratory of Intelligent Information Technology,
School of Computer Science, Beijing Institute of Technology (BIT),
Beijing 100081, People's Republic of China
{gaozhi_2017,wuyuwei,jiayunde}@bit.edu.cn

Abstract. The Symmetric Positive Definite (SPD) matrix on the Riemannian manifold has become a prevalent representation in many computer vision tasks. However, learning a proper distance metric between two SPD matrices is still a challenging problem. Existing metric learning methods of SPD matrices only regard an SPD matrix as a global representation and thus ignore different roles of intrinsic properties in the SPD matrix. In this paper, we propose a novel SPD matrix metric learning method of discovering SPD matrix intrinsic properties and measuring the distance considering different roles of intrinsic properties. In particular, the intrinsic properties of an SPD matrix are discovered by projecting the SPD matrix to multiple low-dimensional SPD manifolds, and the obtained low-dimensional SPD matrices constitute a set. Accordingly, the metric between two original SPD matrices is transformed into a set-to-set metric on multiple low-dimensional SPD manifolds. Based on the learnable alpha-beta divergence, the set-to-set metric is computed by summarizing multiple alpha-beta divergences assigned on low-dimensional SPD manifolds, which models different roles of intrinsic properties. The experimental results on four visual tasks demonstrate that our method achieves the state-of-the art performance.

Keywords: SPD manifold · Metric learning · Set-to-set metric
Multiple manifolds

1 Introduction

The Symmetric Positive Definite (SPD) matrix has become a prevalent representation in many visual tasks, such as face recognition [12], action recognition [30], and object detection [25]. It utilizes the second-order or higher-order statistics information to capture the desirable feature distribution. There are several works try to model a more discriminative SPD matrix [16,27,28] from local features. Meanwhile, calculating the distance metric in the SPD manifold is a crucial problem coming along with the SPD matrix representation. Due to the no-Euclidean structure of SPD manifolds, the Euclidean metric can't be applied

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 452–464, 2018.
https://doi.org/10.1007/978-3-030-03338-5_38

directly on it. In this paper, we focus on a robust metric learning method on SPD manifolds.

Many efforts have been devoted to the SPD matrix metric, such as the Affine Invariant Metric (AIM) [19], Log-Euclidean Metric (LEM) [2], Bregman divergence [14], Stein divergence [21], and alpha-beta divergence [3,4,22]. Given a concrete metric, metric learning aims at learning proper metric parameters that keep similar pairs close and separate dissimilar pairs. Most of the existing metric learning methods on the SPD manifold learn a discriminative metric on the tangent Euclidean space [11,23,31].

However, how to learn a proper SPD matrix metric is still a challenging problem. The SPD matrix is aggregated from local features, and contains different essential intrinsic properties. Existing SPD matrix metric learning methods [11,23,31] just regard an SPD matrix as a global representation and exploit a direct metric on the complex manifold, ignoring the different roles of intrinsic properties in the SPD matrix. It is unsuitable to treat intrinsic properties equally when they have different roles, *e.g.,* different distribution or significance. Therefore, we argue that an SPD matrix metric modeling different roles of intrinsic properties will achieve a better performance.

In this paper, a novel metric learning method on SPD manifolds is proposed to solve the issues mentioned above. Firstly we discover intrinsic properties of an SPD matrix, and then calculate the SPD matrix metric considering different roles of them. In particular, our method aims to jointly learn multiple low-dimensional projections and a set-to-set metric. As the property discovery can be seen as the feature extraction, we apply multiple low-dimensional manifold projections on the SPD matrix to discover discriminative intrinsic properties. Thus, the distance metric between two original SPD matrices is transformed into the distance metric between the two sets which contain several corresponding projected low-dimensional SPD matrices. The alpha-beta divergences is a learnable SPD matrix metric, so it is applied in our set-to-set metric to be adaptive to the intrinsic property. We assign multiple alpha-beta divergences on different low-dimensional manifolds as the sub-metrics and summarize these sub-metrics discriminatively as the SPD matrix metric. Through this, the different roles of intrinsic properties are involved in the SPD matrix metric. Evaluated by experiments, the proposed learnable metric is extremely helpful to capture meaningful nearest neighbors of different original SPD matrices.

In summary, our contributions are three-fold.

(1) We propose a robust SPD matrix metric learning method of discovering discriminative intrinsic properties and modeling their different roles in metric computation.
(2) We formulate the metric learning as the two-component joint optimization problem, *i.e.,* multiple low-dimensional manifold projections and a set-to-set metric are learned jointly.
(3) We introduce the manifold optimization method which can learn metric parameters to guarantee the robustness of the proposed metric.

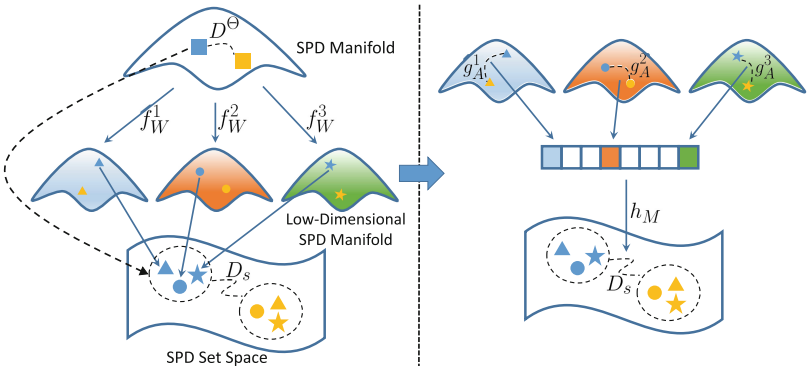

Fig. 1. The flowchart of our SPD matrix metric learning method. Left: multiple projections f_W^1, f_W^2, and f_W^3 used to discover intrinsic properties; Right: the computation of the set-to-set distance D_s which considers different roles of intrinsic properties.

2 The Proposed Method

Throughout this paper, scalars are denoted by the lower-case letters; the vectors are represented by the bold lower-case letters; the matrices are denoted by the upper-case letters; the sets are represented by the bold upper-case letters.

2.1 Problem Definition

This work aims to discover discriminative intrinsic properties in an SPD matrix and compute the distance of SPD matrices considering different roles of discovered properties. The property discovery can be regarded as a feature extraction process that projects an original SPD matrix to multiple low-dimensional SPD manifolds to form a set of the low-dimensional SPD matrices. We propose a set-to-set metric to consider different roles of intrinsic properties. Individual sub-metrics are assigned on low-dimensional manifolds and summarized discriminatively. Consequently, our metric learning method is composed of two components, multiple low-dimensional manifold projections and a set-to-set metric. Given two SPD matrices X_i and X_j, the distance $D^{\Theta}(X_i, X_j)$ is

$$\begin{aligned} D^{\Theta}(X_i, X_j) &= D_s(\mathbf{X}_i, \mathbf{X}_j) \\ &= D_s\Big(\{f_W^1(X_i), \cdots, f_W^m(X_i)\}, \{f_W^1(X_j), \cdots, f_W^m(X_j)\}\Big) \\ &= h_M\Big(g_A^1\big(f_W^1(X_i), f_W^1(X_j)\big), \cdots, g_A^m\big(f_W^m(X_i), f_W^m(X_j)\big)\Big), \end{aligned} \tag{1}$$

where $f_W^k(\cdot)$ is the low-dimensional manifold projection, and $\mathbf{X}_i = \{f_W^k(X_i)\}_{k=1}^m$ is the set containing low-dimensional SPD matrices. The distance $D^{\Theta}(X_i, X_j)$ between original SPD matrices X_i and X_j is transformed into a set-to-set distance $D_s(\mathbf{X}_i, \mathbf{X}_j)$, where the sub-metric on the k-th low-dimensional manifold

is calculated by $g_A^k(\cdot,\cdot)$ and all sub-metrics of properties are summarized by $h_M(\cdot)$. W, A, M are the projection parameter, the sub-metric parameter, and the summarization parameter, respectively. We exploit a learnable parameter set $\Theta = \{W, A, M\}$ to represent the parameters. The framework of our metric learning method for the SPD matrix is shown in Fig. 1.

The goal of metric learning is to learn the metric parameter Θ from an SPD matrix similar pair set $\mathcal{S}$, a dissimilar pair set $\mathcal{D}$, and their labels Y, where $y_{ij} = 1$ means X_i and X_j are similar, otherwise $y_{ij} = 0$. The metric parameter Θ can be learned by optimizing the loss function $\mathcal{L}(\Theta, \mathcal{S}, \mathcal{D}, Y)$ which is the punishment of both far similar sample pairs and close dissimilar sample pairs. We define $\mathcal{L}(\Theta, \mathcal{S}, \mathcal{D}, Y)$ in the following subsection. Moreover, we impose the manifold constraints on W and M to obtain a more robust metric.

2.2 Multiple Low-Dimensional Manifold Projections

For an SPD matrix sample $X_i \in \mathbb{R}^{n\times n}$, we project X_i to m low-dimensional manifolds to discover the intrinsic properties,

$$\begin{aligned} X_i^1 &= f_W^1(X_i) = W_1^\top X_i W_1 \\ &\cdots \\ X_i^m &= f_W^m(X_i) = W_m^\top X_i W_m, \end{aligned} \tag{2}$$

where $X_i^k \in \mathbb{R}^{p\times p}$ is the k-th low-dimensional SPD matrix, $k \in \{1, 2, \cdots, m\}$. An SPD matrix X_i is projected to a set $\mathbf{X}_i = \{X_i^k\}_{k=1}^m$, which contains several low-dimensional SPD matrices.

We expect that each low-dimensional matrix X_i^k is guaranteed to be still an SPD matrix having the ability of capturing desirable feature distribution, and any two low-dimensional SPD manifolds are unrelated to preserve as much information as possible in the low-dimensional SPD matrix set. The learnable parameter W_k needs to be a column full rank matrix to make X_i^k be an SPD matrix as well. Based on the affine invariance [3,7] of the alpha-beta divergence, we relax the column full rank constraint of W_k to the semi-orthogonal constraint, *i.e.,* $W_k^\top W_k = I_p$. In order to preserve more information in the $\mathbf{X}_i = \{X_i^k\}_{k=1}^m$ set, we expect that any two low-dimensional manifolds have a low relevance. For any $k \neq l$, we set $W_k^\top W_l = \mathbf{0}$, where $\mathbf{0} \in \mathbb{R}^{p\times p}$ is a matrix whose elements are all "0"s, to reduce relevance between X_i^k and X_i^l. These low-dimensional SPD manifolds can be seen as analogies of different PCA subspaces. A total projection matrix W is composed of all W_k, $W = [W_1, W_2, \cdots, W_m] \in \mathbb{R}^{n\times mp}$, in which W_k is a partitioned matrix of W containing p columns. Note that, W is a semi-orthogonal matrix, *i.e.,* $W^\top W = I_{mp}$, which is on the non-Euclidean Stiefel manifold [1].

2.3 The Set-to-Set Metric

Based on multiple manifold projections, the distance $D^\Theta(X_i, X_j)$ of two SPD matrices is transformed into the set-to-set distance $D_s(\mathbf{X}_i, \mathbf{X}_j)$. Firstly

$\{g_A^k(\cdot,\cdot)\}_{k=1}^m$ is exploited to compute sub-metrics on m low-dimensional SPD manifolds, and then $h_M(\cdot)$ is utilized to summarize the m sub-metrics, where A and M are learnable parameters. We use the flexible alpha-beta divergence [3,4,22] as the sub-metric $g_A^k(\cdot,\cdot)$. For two SPD sets $\mathbf{X}_i = \{X_i^k\}_{k=1}^m$, $\mathbf{X}_j = \{X_j^k\}_{k=1}^m$, the distance d_{ij}^k between X_i^k and X_j^k is computed by the k-th alpha-beta divergence,

$$d_{ij}^k = g_A^k(X_i^k, X_j^k) = D^{(\alpha_k,\beta_k)}\left(X_i^k \| X_j^k\right) = \frac{1}{\alpha_k\beta_k}\sum_{u=1}^{p} log\left(\frac{\alpha_k(\lambda_{iju}^k)^{\beta_k} + \beta_k(\lambda_{iju}^k)^{-\alpha_k}}{\alpha_k+\beta_k}\right), \tag{3}$$

where λ_{iju}^k is the u-th eigenvalue of $X_i^k(X_j^k)^{-1}$, and (α_k, β_k) is the individual parameter of the k-th alpha-beta divergence. We denote all alpha-beta divergence parameters as a matrix $A = [(\alpha_1,\beta_1),(\alpha_2,\beta_2),...,(\alpha_m,\beta_m)] \in \mathbb{R}^{m\times 2}$, and a distance vector between $\mathbf{X}_i$ and $\mathbf{X}_j$ as $\mathbf{d}_{ij} = [d_{ij}^1, d_{ij}^2, ..., d_{ij}^m] \in \mathbb{R}^{m\times 1}$. Since (α_k, β_k) needs to be adaptive to the k-th low-dimensional manifold, we exploit a learnable strategy to update (α_k, β_k), which is detailed in the next subsection. After computing all sub-metrics, the distance metric $D^{\Theta}(X_i, X_j)$ between two original SPD matrices X_i and X_j is formulated as

$$\begin{aligned} &D^{\Theta}(X_i,X_j) = D_s(\mathbf{X}_i,\mathbf{X}_j) = h_M(d_{ij}^1, d_{ij}^2, ..., d_{ij}^m) = \mathbf{d}_{ij}^{\top} M \mathbf{d}_{ij} \\ &= \sum_{k=1}^{m}\sum_{l=1}^{m}\Big(D^{(\alpha_k,\beta_k)}\big(W_k^{\top}X_iW_k \| W_k^{\top}X_jW_k\big)\cdot M_{kl}\cdot D^{(\alpha_l,\beta_l)}\big(W_l^{\top}X_iW_l \| W_l^{\top}X_jW_l\big)\Big), \end{aligned} \tag{4}$$

where $M \in \mathbb{R}^{m\times m}$ is the metric parameter, and M_{kl} is an element of M in the k-th row and l-th column, reflecting the significance and relationship of properties. If $X_i = X_j$, then $\mathbf{d}_{ij}$ is a zero vector, and $D^{\Theta}(X_i, X_j) = 0$. If $X_i \neq X_j$, then $\mathbf{d}_{ij}$ is a non-zero vector, and $D^{\Theta}(X_i, X_j)$ should be larger than 0. The nonnegativity of the metric forces M to be an SPD matrix and $M \in Sym_m^+$.

To learn the parameter Θ, we formulate loss function $\mathcal{L}(\Theta, \mathcal{S}, \mathcal{D}, Y)$ as

$$\begin{aligned} \min_{\Theta}\mathcal{L}(\Theta,\mathcal{S},\mathcal{D},Y) = &\frac{1}{|\mathcal{S}|}\sum_{i,j\in\mathcal{S}} y_{ij}\cdot max\big(D^{\Theta}(X_i,X_j)-\zeta_s, 0\big)^2 \\ &+ \frac{1}{|\mathcal{D}|}\sum_{i,j\in\mathcal{D}}(1-y_{ij})\cdot max\big(\zeta_d - D^{\Theta}(X_i,X_j), 0\big)^2 \\ &+ \xi\cdot\gamma(M, M_0). \end{aligned} \tag{5}$$

We expect that the distance between similar samples is smaller than a threshold ζ_s, and the distance between dissimilar samples is larger than a threshold ζ_d. We add two coefficients $\frac{1}{|\mathcal{S}|}$ and $\frac{1}{|\mathcal{D}|}$ to solve the imbalance issue of similar and dissimilar sample pairs, where $|\mathcal{S}|$ and $|\mathcal{D}|$ are the pair numbers of sets $\mathcal{S}$ and $\mathcal{D}$. In addition, we add a regularization term $\xi\cdot\gamma(M, M_0)$ on M in Eq. (5). $\gamma(M, M_0) = Tr(MM_0^{-1}) - logdet(MM_0^{-1}) - m$ is the burgman divergence [5,8,10], where $Tr(\cdot)$ is the trace of a matrix, M_0 is the prior information, and ξ is the trade-off coefficient.

2.4 Optimization

$\mathcal{L}(\Theta, \mathcal{S}, \mathcal{D}, Y)$ in Eq. (5) is not a convex function with respect to W, A, and M. Accordingly, we apply the gradient descent to learn Θ. The gradients are computed as follows.

(1) The gradient of $\mathcal{L}$ with respect to M
The gradient of $\mathcal{L}$ with respect to M can be computed by

$$\nabla_M(\mathcal{L}) = \frac{1}{|\mathcal{S}|}\sum_{i,j\in\mathcal{S}} \mathbf{d}_{ij}\nabla_{D_{ij}^{\Theta}}(\mathcal{L})\mathbf{d}_{ij}^{\top} + \frac{1}{|\mathcal{D}|}\sum_{i,j\in\mathcal{D}} \mathbf{d}_{ij}\nabla_{D_{ij}^{\Theta}}(\mathcal{L})\mathbf{d}_{ij}^{\top} + \xi\cdot\nabla_M(\gamma(M, M_0)), \tag{6}$$

where $\nabla_{D_{ij}^{\Theta}}(\mathcal{L})$ is the gradient of $\mathcal{L}$ with respect to $D^{\Theta}(X_i, X_j)$,

$$\nabla_{D_{ij}^{\Theta}}(\mathcal{L}) = 2\cdot y_{ij}\cdot max(D_{ij}^{\Theta} - \zeta_s, 0) + 2\cdot(y_{ij}-1)\cdot max(\zeta_d - D_{ij}^{\Theta}, 0), \tag{7}$$

and $\nabla_M(\gamma(M, M_0))$ is the gradient of $\gamma(M, M_0)$ with respect to M,

$$\nabla_M(\gamma(M, M_0)) = M_0^{-1} - M^{-1}. \tag{8}$$

(2) The gradient of $\mathcal{L}$ with respect to A
The gradients of $\mathcal{L}$ with respect to α_k and β_k in A are

$$\nabla_{\alpha_k}(\mathcal{L}) = \frac{1}{|\mathcal{S}|}\sum_{i,j\in\mathcal{S}} \nabla_{d_{ij}^k}(\mathcal{L})\cdot\nabla_{\alpha_k}(d_{ij}^k) + \frac{1}{|\mathcal{D}|}\sum_{i,j\in\mathcal{D}} \nabla_{d_{ij}^k}(\mathcal{L})\cdot\nabla_{\alpha_k}(d_{ij}^k), \tag{9}$$

$$\nabla_{\beta_k}(\mathcal{L}) = \frac{1}{|\mathcal{S}|}\sum_{i,j\in\mathcal{S}} \nabla_{d_{ij}^k}(\mathcal{L})\cdot\nabla_{\beta_k}(d_{ij}^k) + \frac{1}{|\mathcal{D}|}\sum_{i,j\in\mathcal{D}} \nabla_{d_{ij}^k}(\mathcal{L})\cdot\nabla_{\beta_k}(d_{ij}^k). \tag{10}$$

$\nabla_{d_{ij}^k}(\mathcal{L})$ is the k-th element of $\nabla_{\mathbf{d}_{ij}}(\mathcal{L})$ which is the gradient of $\mathcal{L}$ with respect to $\mathbf{d}_{ij}$,

$$\nabla_{\mathbf{d}_{ij}}(\mathcal{L}) = \nabla_{D_{ij}^{\Theta}}(\mathcal{L})\cdot\nabla_{\mathbf{d}_{ij}}(D_{ij}^{\Theta}) = \nabla_{D_{ij}^{\Theta}}(\mathcal{L})\mathbf{d}_{ij}^{\top}(M^{\top} + M). \tag{11}$$

$\nabla_{\alpha_k}(d_{ij}^k)$ and $\nabla_{\beta_k}(d_{ij}^k)$ are the gradients of d_{ij}^k with respect to α_k and β_k, respectively,

$$\begin{aligned}\nabla_{\alpha_k}(d_{ij}^k) = &\frac{1}{\alpha_k^2\beta_k}\sum_{u=1}^{p}\Bigg(\frac{\alpha_k(\lambda_{iju}^k)^{\beta_k} - \alpha_k\beta_k(\lambda_{iju}^k)^{-\alpha_k}log\lambda_{iju}^k}{\alpha_k(\lambda_{iju}^k)^{\beta_k} + \beta_k(\lambda_{iju}^k)^{-\alpha_k}} \\ &- \frac{\alpha_k}{\alpha_k+\beta_k} - log\frac{\alpha_k(\lambda_{iju}^k)^{\beta_k} + \beta_k(\lambda_{iju}^k)^{-\alpha_k}}{\alpha_k+\beta_k}\Bigg),\end{aligned} \tag{12}$$

$$\nabla_{\beta_k}(d_{ij}^k) = \frac{1}{\alpha_k \beta_k^2} \sum_{u=1}^{p} \left(\frac{\beta_k (\lambda_{iju}^k)^{-\alpha_k} - \alpha_k \beta_k (\lambda_{iju}^k)^{\beta_k} log \lambda_{iju}^k}{\alpha_k (\lambda_{iju}^k)^{\beta_k} + \beta_k (\lambda_{iju}^k)^{-\alpha_k}} - \frac{\beta_k}{\alpha_k + \beta_k} - log \frac{\alpha_k (\lambda_{iju}^k)^{\beta_k} + \beta_k (\lambda_{iju}^k)^{-\alpha_k}}{\alpha_k + \beta_k} \right). \tag{13}$$

(3) The gradient of $\mathcal{L}$ with respect to W
The gradient of $\mathcal{L}$ with respect to each W_k is

$$\nabla_{W_k}(\mathcal{L}) = \sum_{i}^{N} \left((X_i)^\top W_k \nabla_{X_i^k}(\mathcal{L}) + X_i W_k \nabla_{X_i^k}(\mathcal{L})^\top \right), \tag{14}$$

where N is the number of training samples, and $N = 2 \times (|\mathcal{S}| + |\mathcal{D}|)$. $\nabla_{X_i^k}(\mathcal{L})$ is the gradient of $\mathcal{L}$ with respect to the low-dimensional SPD matrix X_i^k. The eigenvalue decomposition of $X_i^k (X_j^k)^{-1}$ is $X_i^k (X_j^k)^{-1} = U_{ij}^k \Sigma_{ij}^k (U_{ij}^k)^\top$. Σ_{ij}^k is the diagonal matrix eigenvalues, and λ_{iju}^k is the u-th eigenvalue. The gradients $\nabla_{X_i^k}(\mathcal{L})$ and $\nabla_{X_j^k}(\mathcal{L})$ are

$$\nabla_{X_i^k}(\mathcal{L}) = U_{ij}^k \nabla_{\Sigma_{ij}^k}(\mathcal{L}) (U_{ij}^k)^\top (X_i^k)^{-\top}, \tag{15}$$

$$\nabla_{X_j^k}(\mathcal{L}) = (-1) \cdot (X_j^k)^{-\top} (X_i^k)^\top U_{ij}^k \nabla_{\Sigma_{ij}^k}(\mathcal{L}) (U_{ij}^k)^\top (X_j^k)^{-\top}, \tag{16}$$

where $\nabla_{\Sigma_{ij}^k}(\mathcal{L})$ is the gradient of Σ_{ij}^k with respect to $\mathcal{L}$. $\nabla_{\Sigma_{ij}^k}(\mathcal{L})$ is a diagonal matrix, and the u-th element is

$$\begin{aligned} \nabla_{\lambda_{iju}^k}(\mathcal{L}) &= \nabla_{d_{ij}^k}(\mathcal{L}) \cdot \nabla_{\lambda_{iju}^k}(d_{ij}^k) \\ &= \nabla_{d_{ij}^k}(\mathcal{L}) \cdot \frac{1}{\alpha_k \beta_k} \frac{\alpha_k \beta_k (\lambda_{iju}^k)^{\beta_k - 1} - \alpha_k \beta_k (\lambda_{iju}^k)^{-\alpha_k - 1}}{\alpha_k (\lambda_{iju}^k)^{\beta_k} + \beta_k (\lambda_{iju}^k)^{-\alpha_k}}. \end{aligned} \tag{17}$$

Since the gradients $\nabla_W(\mathcal{L})$, $\nabla_M(\mathcal{L})$, and $\nabla_A(\mathcal{L})$ are obtained, the metric parameter set Θ can be updated. A is optimized by the standard gradient descent, $A := A - \eta \nabla_A(\mathcal{L})$, where η is the learning rate. W and M are updated by the Riemannian optimization algorithm [1,6,20]. The computation details are presented below,

$$\begin{cases} \nabla_{W_R}(\mathcal{L}) = \nabla_W(\mathcal{L}) - W \frac{1}{2} (W^\top \nabla_W(\mathcal{L}) + \nabla_W(\mathcal{L})^\top W) \\ W := q\big(W - \eta \nabla_{W_R}(\mathcal{L})\big) \end{cases}, \tag{18}$$

and

$$\begin{cases} \nabla_{M_R}(\mathcal{L}) = M \frac{1}{2} \big(\nabla_M(\mathcal{L}) + \nabla_M(\mathcal{L})^\top\big) M \\ M := M^{\frac{1}{2}} expm\big(-\eta M^{-\frac{1}{2}} \nabla_{M_R}(\mathcal{L}) M^{-\frac{1}{2}}\big) M^{\frac{1}{2}} \end{cases}, \tag{19}$$

where $\nabla_{W_R}(\mathcal{L})$ and $\nabla_{M_R}(\mathcal{L})$ are the Riemannian gradients with respect to W and M. In Eq. (18), $q(\cdot)$ is the retraction operation mapping the data back to the Stiefel manifold. $q(W)$ denotes the Q matrix of the QR decomposition to a matrix W, *i.e.,* for the matrix $W \in \mathbb{R}^{n\times p}$, $W = QR$, where $Q \in \mathbb{R}^{n\times p}$ is a semi-orthogonal matrix and $R \in \mathbb{R}^{p\times p}$ is a upper triangular matrix. In Eq. (19), $expm(\cdot)$ is the matrix exponential function. We summarize the learning process of our method in Algorithm 1, w.

Algorithm 1. Trainging Process of Our Method

Input: Training SPD sample pairs $\mathcal{S}$ and $\mathcal{D}$, labels Y. The initial projection matrix W. The initial metric matrix M. The initial alpha-beta divergence parameter A. Learning rate η.

Output: The learned W, M, and A.

while not converge **do**
  For each SPD matrix, compute subspaces by Eq.(2).
  For each sample pairs, compute the distance between their sets by Eq.(3) and Eq.(4).
  Compute the loss $\mathcal{L}$ by Eq.(5).
  Compute the gradient $\nabla_M(\mathcal{L})$ by Eq.(7), Eq.(8), and Eq.(6).
  Compute the gradient $\nabla_A(\mathcal{L})$ by Eq.(12), Eq.(13), Eq.(9), and Eq.(10).
  Compute the gradient $\nabla_W(\mathcal{L})$ by Eq.(17), Eq.(15), Eq.(16), and Eq.(14).
  Update the parameter W by Eq.(18).
  Update the parameter A by $A := A - \eta\nabla_A(\mathcal{L})$.
  Update the parameter M by Eq.(19).
end while
return W, M and A

3 Experiments

In order to test the efficiency of our method, we conduct experiments on the object recognition, video-based face recognition, action recognition, and texture classification tasks. Four datasets are utilized: the ETH-80 [15], the MSR-Action3D [17], the YouTube Celebrities (YTC) [13], and the UIUC [18] datasets.

3.1 Datasets and Settings

The ETH-80 is an object image dataset, which contains 80 image sets of eight categories. Each category consists of 10 image sets, and each set includes 41 images captured under different views. In our experiment, all the images of the ETH-80 are resized to 20×20 and denoted by the intensity features. The YTC is a video-based face dataset, collecting 1910 videos of 47 persons. Face regions are detected from each frame by a cascaded face detector and resized to 30×30, followed by the histogram equalized operation, and represented by

the gray values. The MSR-Action3D is a 3D action dataset, containing totally 567 videos of 20 actions. There are 20 skeleton joints in the body of actions. In the experiments, each frame is represented by a 120-dimensional feature, which is the 3D coordinate differences of skeleton joints between this frame and its two neighborhood frames. The UIUC material dataset contains 216 samples of 18 categories. We resize each image to 400×400. Then 128-dimensional dense SIFT features are extracted from each image with 4-pixel space concatenated by 27-dimensional RGB color features from 3×3 patches centered at the locations of dense SIFT features.

On the ETH-80, YTC, and UIUC datasets, we compute a covariance matrix C to represent each sample and add a small ridge δI to avoid the matrix singularity, where $\delta = 0.001 \times Tr(C)$. On the MSR-Action3D dataset, we first compute the covariance matrix C with size of 120×120, then transform it to a 121×121 Gaussian distribution SPD matrix, $C = |C|^{-\frac{1}{121}} \begin{bmatrix} C + \mathbf{m}\mathbf{m}^\top & \mathbf{m} \\ \mathbf{m}^\top & 1 \end{bmatrix}$ as the sample representation, where $\mathbf{m}$ is the mean vector of 120-dimensional features. Following the standard protocols [7,11,24,29], for each category, we randomly select half of the samples for training and the rest for testing on the ETH-80, MSR-Action3D, and UIUC datasets. On the YTC dataset, for each person, three videos are randomly selected as the gallery, and six as the probe. In experiments, we set $\xi = 0.01$, $M_0 = I_m$, $\zeta_s = 5$, and $\zeta_d = 100$.

3.2 Evaluation

We exploit the 1-NN classifier to evaluate the performance of all metric learning methods. The following methods are evaluated in our experiments: AIM [19], Stein Divergence [21], LEM [2], SPD-DR [7], CDL [29], RSR-ML [9], LEML [11], and α-CML [31]. AIM, Stein Divergence, and LEM are the basic SPD matrix metrics, measuring the geodesic distance between SPD matrices. SPD-DR implements the dimensionality reduction on the SPD matrix and then applies the AIM or Stein Divergence between samples. CDL is a Riemannian kernel discriminative learning approach on the SPD manifold. RSR-ML employs sparse coding and dictionary learning scheme on the SPD manifold. LEML and α-CML are two LEM based SPD matrix metric learning methods which project SPD matrices to the tangent space and utilize the LEM to compute the distance between them.

Table 1 shows the comparisons of the four visual tasks. In the object recognition task, we set the dimensionality of the low-dimensional manifolds is 10×10 and the number of them is 20, *i.e.,* $m = 20$. We find that LEM has a better performance than AIM, 93.0 vs 85.0, showing that the point on the tangent space is more discriminative. If the manifold point is projected to a low-dimensional discriminative space, *i.e.,* the SPD-DR method, the performance can be improved to 96.0, 0.5 better than LEML. Compared with SPD-DR, our method achieves 97.5, 1.5 higher than it, which shows the power of discovering discriminative properties and their roles.

In the video-based face recognition task, the dimensionality of projected manifolds is 10×10, and the number of them is 40. We achieve 49.2 in this task, 2.5

higher than SPD-DR and 10 percent higher than the basic SPD matrix metrics approximately. However, due to the large variable faces caused by posture, illumination, scale, and occlusion, the performance of linear metric learning methods is far less than it of the nonlinear kernel method CDL. The reason we think is that the samples in the original space are not separable, a more higher-dimensional RKHS space can relieve this problem.

In the action recognition task, the dimensionality of the low-dimensional manifolds is 8×8 and the number of them is 15. Nonlinear kernel methods CDL and RSR-ML achieve 95.4 and 95.0 respectively and have a better performance than the existing metric methods [7,11,31]. In this case, our linear method obtains the comparable performance with CDL and RSR-ML, achieving 95.8. Besides, Wang *et al.* [26] shows that the nonlinear kernel matrix representation has a better performance than the linear SPD representation, while our accuracy is 3.1 higher than α-CML whose performance is based on the kernel matrix [26] rather than the Gaussian distribution SPD matrix.

In the texture classification task, in our method, we set the dimensionality of the low-dimensional manifolds is 8×8, and there are totally 18 low-dimensional manifolds. We can see that, the three basic SPD matrix metrics *i.e.,* AIM, Stein Divergence, and LEM achieve comparable performance in the UIUC dataset, 35.6, 35.8 and 36.7 respectively. Meanwhile, metric learning methods can bring a remarkable improvement. CDL achieves 54.9, and the accuracy of LEML is 53.9. SPD-DR achieves a better performance 58.3, showing that there are too much noise and information redundancy in the original SPD representation. Our method further improves the result to 60.8 showing that our method can not only remove the noise and information redundancy but also bring the benefits of discovering discriminative intrinsic properties and their different roles.

Table 1. Accuracies (%) on the four visual tasks. Our method is bold in the last line.

Method	Eth-80	YTC	MSR-Action3D	UIUC
AIM [19]	85.0	38.2	84.7	35.6
Stein [21]	-	-	83.5	35.8
LEM [2]	93.0	40.8	84.7	36.7
AIM-DR [7]	96.0	46.7	93.1	58.3
Stein-DR [7]	-	-	94.6	58.1
CDL [29]	94.5	67.5	95.4	54.9
RSR-ML [9]	94.8	-	95.0	-
LEML [11]	95.5	-	92.3	53.9
α-CML [31]	-	-	92.7	-
Ours	**97.5**	**49.2**	**95.8**	**60.8**

4 Conclusions

In this paper, we have proposed a novel metric learning method on the SPD manifold, which can discover discriminative intrinsic properties and computes the metric considering their different roles. We can formulate the SPD manifold metric learning process as the multiple projections and a set-to-set metric joint optimization problem. Moreover, we force the projection matrix and the metric matrix on manifolds, obtaining a robust metric. Extensive experiments have shown that our method outperforms existing metric learning methods on the SPD manifold. As our method is differentiable in the whole process, in the future, we will endow it with deep learning for the desirable nonlinearity.

Acknowledgements. This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 61702037 and No. 61773062, and Beijing Municipal Natural Science Foundation under Grant No. L172027, in part by Beijing Institute of Technology Research Fund Program for Young Scholars.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)
2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magn. Reson. Med. **56**(2), 411 (2006)
3. Cherian, A., Stanitsas, P., Harandi, M., Morellas, V., Papanikolopoulos, N.: Learning discriminative $\alpha\beta$-divergences for positive definite matrices. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4270–4279 (2017)
4. Cichocki, A., Cruces, S., Amari, S.: Log-determinant divergences revisited: alpha-beta and gamma log-det divergences. Entropy **17**(5), 2988–3034 (2015)
5. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 209–216 (2007)
6. Harandi, M., Fernando, B.: Generalized backpropagation, Étude de cas: Orthogonality. arXiv preprint arXiv:1611.05927 (2016)
7. Harandi, M., Salzmann, M., Hartley, R.: Dimensionality reduction on SPD manifolds: the emergence of geometry-aware methods. IEEE Trans. Pattern Anal. Mach. Intell. **40**, 48–62 (2017)
8. Harandi, M., Salzmann, M., Hartley, R.: Joint dimensionality reduction and metric learning: a geometric take. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1404–1413 (2017)
9. Harandi, M.T., Sanderson, C., Hartley, R., Lovell, B.C.: Sparse coding and dictionary learning for symmetric positive definite matrices: a kernel approach. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, pp. 216–229. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_16
10. Hoffman, J., Rodner, E., Donahue, J., Kulis, B., Saenko, K.: Asymmetric and category invariant feature transformations for domain adaptation. Int. J. Comput. Vis. **109**(1–2), 28–41 (2014)

11. 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 International Conference on Machine Learning (ICML), pp. 720–729 (2015)
12. Huang, Z., Wang, R., Van Gool, L., Chen, X., et al.: Cross Euclidean-to-Riemannian metric learning with application to face recognition from video. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2018)
13. Kim, M., Kumar, S., Pavlovic, V., Rowley, H.: Face tracking and recognition with visual constraints in real-world videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
14. Kulis, B., Sustik, M.A., Dhillon, I.S.: Low-rank kernel learning with Bregman matrix divergences. J. Mach. Learn. Res. **10**(1), 341–376 (2009)
15. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. II-409 (2003)
16. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: IEEE International Conference on Computer Vision, pp. 2089–2097 (2017)
17. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14 (2010)
18. Liao, Z., Rock, J., Wang, Y., Forsyth, D.: Non-parametric filtering for geometric detail extraction and material representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 963–970 (2013)
19. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. Int. J. Comput. Vis. **66**(1), 41–66 (2006)
20. Roy, Kumar, S., Mhammedi, Z., Harandi, M.: Geometry aware constrained optimization techniques for deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1 (2018)
21. Sra, S.: A new metric on the manifold of kernel matrices with application to matrix geometric means. In: Advances in Neural Information Processing Systems, pp. 144–152 (2012)
22. Thiyam, D.B., Cruces, S., Olias, J., Cichocki, A.: Optimization of Alpha-Beta Log-Det divergences and their application in the spatial filtering of two class motor imagery movements. Entropy **19**(3), 89 (2017)
23. Vemulapalli, R., Jacobs, D.W.: Riemannian metric learning for symmetric positive definite matrices. arXiv preprint arXiv:1501.02393 (2015)
24. Vemulapalli, R., Pillai, J.K., Chellappa, R.: Kernel learning for extrinsic classification of manifold features. In: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1782–1789 (2013)
25. Wang, H., Wang, Q., Gao, M., Li, P., Zuo, W.: Multi-scale location-aware kernel representation for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1 (2018)
26. Wang, L., Zhang, J., Zhou, L., Tang, C., Li, W.: Beyond covariance: feature representation with nonlinear kernel matrices. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4570–4578 (2015)
27. Wang, Q., Li, P., Zhang, L.: G2DeNet: global Gaussian distribution embedding network and its application to visual recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6507–6516 (2017)

28. Wang, Q., Li, P., Zuo, W., Zhang, L.: RAID-G: robust estimation of approximate infinite dimensional Gaussian with application to material recognition. In: Computer Vision and Pattern Recognition, pp. 4433–4441 (2016)
29. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: a natural and efficient approach to image set classification. In: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2496–2503 (2012)
30. Zhang, T., Zheng, W., Cui, Z., Zong, Y., Li, Y.: Deep manifold-to-manifold transforming network for action recognition. arXiv preprint arXiv:1705.10732 (2017)
31. Zhou, L., Wang, L., Zhang, J., Shi, Y., Gao, Y.: Revisiting metric learning for SPD matrix based visual representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3241–3249 (2017)