Geometry-aware Similarity Learning on SPD Manifolds for Visual Recognition

Zhizhu Huang, Member, IEEE, Ruiping Wang, Member, IEEE, Xianqiu Li, Wenxian Liu, Shiguang Shan, Senior Member, IEEE, Luc Van Gool, Member, IEEE and Xilin Chen, Fellow, IEEE

Abstract—Symmetric Positive Definite (SPD) matrices have been widely used for data representation in many visual recognition tasks. The success mainly attributes to learning discriminative SPD matrices with encoding the Riemannian geometry of the underlying SPD manifold. In this paper, we propose a geometry-aware SPD similarity learning (SPDSL) framework to learn discriminative SPD features by directly pursuing manifold- transformation matrix of column full-rank. Specifically, by exploiting the Riemannian geometry of the manifold of fixed-rank Positive Semidefinite (PSD) matrices, we present a new solution to reduce optimizing over the space of column full-rank transformation matrices to optimizing on the PSD manifold which has a well-established Riemannian structure. Under this solution, we exploit a new supervised SPD similarity learning technique to learn the transformation by regressing the similarities of selected SPD data pairs to their ground-truth learning tasks. The success mainly attributes to learning discriminative SPD features by directly pursuing manifold-geometry-aware SPD similarity learning (SPDSL) framework to the underlying SPD manifold. In this paper, we propose a

Index Terms—discriminative SPD matrices, Riemannian geometry, SPD manifold, geometry-aware SPD similarity learning, PSD manifold.

I. INTRODUCTION

Recently, Symmetric Positive Definite (SPD) matrices of real numbers appear in many branches of computer vision. Examples include region covariance matrices for pedestrian detection [1], [2] and texture categorization [3], [4], [5], joint covariance descriptor for action recognition [6], [5], diffusion tensors for DT image segmentation [7], [8], [4] and image set based covariance matrix for video face recognition [9], [10], [11]. Due to the effectiveness of measuring data variations, such SPD features have been shown to provide powerful representations for images and videos.

However, such advantages of the SPD matrices often accompany with the challenge of their non-Euclidean data structure which underlies a specific Riemannian manifold [7], [8]. Applying the Euclidean geometry directly to SPD matrices often results in poor performances and undesirable effects, such as the swelling of diffusion tensors in the case of SPD matrices [12], [13]. To overcome the drawbacks of the Euclidean representation, recent works [13], [8], [4] have introduced Riemannian metrics, e.g., Affine-Invariant metric [7]. Log-Euclidean metric [8], to encode the Riemannian geometry of SPD manifold properly.

By applying these classical Riemannian metrics, a couple of works attempt to extend Euclidean algorithms to work on manifolds of SPD matrices for learning more discriminative SPD matrices or their vector-forms. To this end, several studies exploit effective methods on one SPD manifold by either flattening it via tangent space approximation [2], [15], [16], [17] (See Fig. 1(a)→(b)) or mapping it into a high dimensional Reproducing Kernel Hilbert Space (RKHS) [3], [9], [4], [18], [19], [20], [21] (See Fig. 1(a)→(c)→(b)). Obviously, both of the two families of methods inevitably distort the geometrical structure of the original SPD manifold due to the procedure of mapping the manifold into a flat Euclidean space or a high dimensional RKHS. Therefore, the two learning schemes would lead to sub-optimal solutions for the problem of discriminative SPD matrix learning.

To more faithfully respect the original Riemannian geometry, another kind of SPD-based discriminant learning methods[4]

Fig. 1. Three different learning schemes for SPD features. The first one (a)→(b) is to firstly flatten the original manifold Sym+n into tangent space and then learn a map g to a discriminative Euclidean space Rn. The second one (a)→(c)→(b) is to firstly embed Sym+n with an implicit map ϕ into an RKHS H and then learn a mapping h to a more discriminative Euclidean space Rm. The last one (a)→(d) aims to learn a map f from the original SPD manifold Sym+n to a more discriminative SPD manifold Sym+n. Here, X ∈ Sym+n and f(X) ∈ Sym+n are the SPD matrices, TX Sym+n and Tf(X)Sym+n are the tangent spaces.
II. BACKGROUND

Let $\text{Sym}_n = \{ H : H^T = H \}$ be a set of real, symmetric matrices of size $n \times n$ and $\text{Sym}_n^+ = \{ X \in \text{Sym}_n : \omega^T X \omega > 0, \forall \omega \in \mathbb{R}^n, \omega \neq 0 \}$ be a set of SPD matrices. The mapping space $\text{Sym}_n$ is endowed with usual Euclidean metric (i.e., inner product) $\langle H_1, H_2 \rangle = Tr(H_1^T H_2)$. As noted in [21, 28], the set of SPD matrices $\text{Sym}_n^+$ is an open convex subset of $\text{Sym}_n$. Thus, the tangent space to $\text{Sym}_n^+$ at any SPD matrix in it can be identified with the set $\text{Sym}_n$. A smoothly-varying family of inner products on each tangent space is known as Riemannian metric, endowing which the space of SPD matrices $\text{Sym}_n^+$ would yield a Riemannian manifold. With such Riemannian metric, the geodesic distance between two elements $X_1, X_2$ on the SPD manifold is generally measured by $\langle \log_{X_1}(X_2), \log_{X_1}(X_2) \rangle_{X_1}$. Several Riemannian metrics and divergences have been proposed to equip SPD manifolds. For example, Affine-Invariant metric [7], Stein divergence [26], Jeffereys divergence [18] are designed to be invariant to affine transformation. That is, for any $M \in GL(n)$ (i.e., the group of real invertible $n \times n$ matrices), the metric function $\delta_A$ has the property $\delta_A^2(X_1, X_2) = \delta_A^2(MX_1M^T, MX_2M^T)$. In contrast, Log-Euclidean metric [8], Cholesky distance [27] and Power-Euclidean metric [27] are not affine invariant. Among these metrics, only Affine-Invariant metric [7] and Log-Euclidean metric [8] define a true geodesic distance on the SPD manifold [4]. In addition, the Stein divergence are also widely used due to its favorable properties and high performances in visual recognition tasks [26]. Therefore, this paper focuses on studying such three representative Riemannian metrics.

Definition 1. By defining the inner product in the tangent space at the SPD point $X_1$ on the SPD manifold as $\langle H_1, H_2 \rangle_{X_1} = \langle X_1^{-1/2}H_1X_1^{-1/2}, X_2^{-1/2}H_2X_2^{-1/2} \rangle$ and the logarithmic maps as $\log_{X_1}(X_2) = X_1^{-1/2}\log(X_1^{-1/2}X_2X_1^{-1/2})X_1^{-1/2}$, the geodesic distance between two SPD matrices $X_1, X_2$ on the SPD manifold is induced by Affine-Invariant metric (AIM) as

$$
\delta_A^2(X_1, X_2) = \left\| \log(X_1^{-1/2}X_2X_1^{-1/2}) \right\|_F^2.
$$

(1)

Definition 2. The approximated geodesic distance between two SPD matrices $X_1, X_2$ on the SPD manifold is defined by using Stein divergence as

$$
\delta_S^2(X_1, X_2) = \ln \det \left( \frac{X_1 + X_2}{2} \right) - \frac{1}{2} \ln \det(X_1X_2).
$$

(2)

Definition 3. By defining the inner product in the tangent space at the SPD point $X_1$ on the SPD manifold as $\langle H_1, H_2 \rangle_{X_1} = \langle D\log(X_1)[H_1], D\log(X_2)[H_2] \rangle$ ($D\log(X_1)[H]$ denotes the directional derivative) and the logarithmic maps as $\log_{X_1}(X_2) = D^{-1}\log(X_1)[\log(X_2) - \log(X_1)]$, the geodesic distance between two SPD matrices $X_1, X_2$ is derived by Log-Euclidean metric (LEM) as

$$
\delta_L^2(X_1, X_2) = \left\| \log(X_1) - \log(X_2) \right\|_F^2.
$$

(3)

III. PROPOSED APPROACH

In this section, we first propose a new solution of Riemannian geometry-aware dimensionality reduction for SPD matrices, and then present our supervised SPD similarity learning method under the solution. Finally, we give a detailed description of our developed optimization algorithm.

A. Riemannian Geometry-aware Dimensionality Reduction on SPD manifolds

Given a set of SPD matrices $X = \{ X_1, \ldots, X_N \}$, where each matrix $X_i \in \text{Sym}_n^+$, and a transformation $W \in \mathbb{R}^{n \times m}$ ($m < n$) is pursued for mapping the original SPD manifold $\text{Sym}_n^+$ to a lower-dimensional SPD manifold $\text{Sym}_m^+$. Formally, this procedure attempts to learn the parameter $W$, of a
mapping in the form \( f : Sym^+_n \times \mathbb{R}^{n \times m} \rightarrow Sym^+_m \), which is defined as:
\[
f(X_i, W) = W^T X_i W.
\] (4)
To ensure the resulting mapping yields a valid SPD manifold \( Sym^+_n \ni W^T X_i W > 0 \), the manifold-manifold transformation \( W \) is basically required to be a column full-rank matrix \( W \in \mathbb{R}^{n \times m} \).

Since the solution space is a non-compact Stiefel manifold \( \mathbb{R}^{n \times m} \) where the distance function has no upper bound, directly optimizing on the manifold is infeasible. Fortunately, the conjugates (taking the form of \( WW^T \)) of column full-rank matrices span a compact manifold \( Sym^+_n(m) \) of Positive Semidefinite (PSD) matrices, which is a quotient space of \( \mathbb{R}^{n \times m} \) and owns a well-established Riemannian structure. In contrast, by additionally assuming the transformation \( W \) to be orthogonal as done in [5], Eqn.4 could be optimized on compact Stiefel manifold, which is a subset of the non-compact Stiefel manifold \( \mathbb{R}^{n \times m} \). Further, for the affine invariant metrics (e.g., AIM), optimizing on Stiefel manifold can be reduced to optimizing over Grassmannian [5]. However, such orthogonal solution space is smaller than the original solution space \( \mathbb{R}^{n \times m} \), making the optimization theoretically yield suboptimal solution of \( W \). Thus, we choose to optimize on the PSD manifold to search the optimal solution of \( W \) in a more faithful way. Now, we need to study the geometry of the PSD manifold \( Sym^+_n(m) \).

For all orthogonal matrices \( O \) of size \( m \times m \), the map \( W \rightarrow WO \) leaves \( WW^T \) unchanged. This property of \( W \) results in the equivalence class of the form \( \{W\} = \{W \mid O \in \mathbb{R}^{n \times m}, OT^m O = I_m\} \), and yields a one-to-one correspondence with the rank-\( m \) PSD matrix \( Q = WW^T \in Sym^+_n(m) \). By quotienting this equivalence relation out, the set of rank-\( m \) PSD matrices \( Sym^+_n(m) \) is reduced to the quotient of the manifold \( \mathbb{R}^{n \times m} \) by the orthogonal group \( O(m) = \{O \in \mathbb{R}^{n \times m}, OT^m O = I_m\} \), i.e., \( Sym^+_n(m) = \mathbb{R}^{n \times m} / O(m) \). With the studied relationship between \( Sym^+_n(m) \) and \( \mathbb{R}^{n \times m} \), the function \( \phi : Sym^+_n(m) \rightarrow \mathbb{R}^q, Q \rightarrow \phi(Q) \) is able to derive the function \( g : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^q, W \rightarrow g(W) \) defined as \( g(W) = \phi(WW^T) \). Here, \( g \) is defined in the total space \( \mathbb{R}^{n \times m} \) and descends as a well-defined function in the quotient manifold \( Sym^+_n(m) \). Therefore, optimizing over the total space \( \mathbb{R}^{n \times m} \) is reduced to optimizing on the PSD manifold \( Sym^+_n(m) \), which is well-studied in several works [28], [29],[30],[31]. Note that, as each element \( Q = WW^T \) on the PSD manifold is simply parameterized by \( W \), optimizing on the manifold actually deals directly with \( W \). To more easily understand this point, one can take the well-known Grassmann manifold as an analogy, where each element can be similarly represented by the equivalence class \( [W] \) or the projection matrix \( WW^T \) (here, \( W^T W = I \)), and the optimization on it directly seeks the solution of \( W \).

It can be further proven that the quotient \( Sym^+_n(m) \) presents the structure of a Riemannian manifold [28]. As a result, endowing the total space \( \mathbb{R}^{n \times m} \) with the usual Riemannian structure of a Euclidean space (i.e., the inner product \( \langle H_1, H_2 \rangle = Tr(H_1^T H_2) \)), a Riemannian structure for the quotient space \( Sym^+_n(m) \) follows. The inner product occurs on the tangent space \( T_W \) of the manifold \( \mathbb{R}^{n \times m} \). In the case of the manifold \( Sym^+_n(m) \), the corresponding tangent space is decomposed into two orthogonal subspaces, the vertical space \( \mathcal{V}_W = \{\Omega \mid \Omega \in \mathbb{R}^{n \times m}, \Omega^T = -\Omega\} \) and the horizontal space \( \mathcal{H}_W = \{H \in T_W \mid HH^T = W^T W\} \), to achieve the inner product \( \langle H_1, H_2 \rangle \). This Riemannian metric facilitates several classical optimization techniques such as Riemannian Conjugate Gradient (RCG) algorithm [28] working on the PSD manifold \( Sym^+_n(m) \). As for more detailed background on the Riemannian geometry of the PSD manifold, please refer to the works [28],[30].

By exploiting the Riemannian geometry of the fixed-rank PSD manifold \( Sym^+_n(m) \), we here open up the possibility of directly pursuing an optimal column full-rank manifold-manifold transformation matrix to solve the problem of dimensionality reduction on SPD features.

### B. Supervised SPD similarity learning

As studied before, under the proposed framework of dimensionality reduction on SPD features, a target SPD manifold \( Sym^+_n(m) \) of lower dimensionality can be derived. On the new SPD manifold \( Sym^+_n(m) \), the geodesic distance between the two original SPD points \( X_i, X_j \) is achieved by:
\[
\delta^2(X_i, X_j) = \delta^2(f(X_i, W), f(X_j, W)),
\] (5)
where \( f(X_i, W) \) is the manifold-manifold transformation computed by Eqn.4. \( \delta \) can be the geodesic distance induced by the commonly-used affine or non-affine invariant Riemannian metrics Eqn.1 and Eqn.2.

In this paper, we are focusing on the problem of supervised SPD similarity learning for more robust visual classification tasks where SPD features have shown great power. Formally, for each SPD matrix \( X_i \in Sym^+_n \), we define its class indicator vector: \( y_i = [0, \ldots, 1, \ldots, 0] \in \mathbb{R}^c \), where the \( k \)-th entry being 1 and other entries being 0 indicates that \( X_i \) belongs to the \( k \)-th class of \( c \) classes in total. As discriminant learning techniques developed in Euclidean space, we assume that prior knowledge is known regarding the distances between pairs of SPD points on the new SPD manifold \( Sym^+_n(m) \). Let’s take the similarity or dissimilarity between pairs of SPD points into account: two SPD points are similar if the similarity based on the geodesic distance between them on the new manifold is larger, while two SPD points are dissimilar if their similarity is smaller.

Given a set of the similarity constraints, our goal is to learn the manifold-manifold transformation matrix \( W \) that parameterizes the similarities of SPD points on the target SPD manifold \( Sym^+_n(m) \). To this end, we exploit the supervised criterion of centered kernel target alignment [32],[33],[34] to learn discriminative features on the SPD manifold by regressing the similarities of selected sample pairs to the target similarities. Formally, our supervised SPD similarity learning (SPDSL) approach is to maximize the following objective function:
\[
J(W) = \frac{(UG \circ k(W)U + (YY^T))_F}{\|UG \circ k(W)U\|_F}, s.t.W \in \mathbb{R}^{n \times m},
\] (6)
where $\langle \cdot \rangle_F$ and $\| \cdot \|_F$ are Frobenius inner product and norm respectively. The elements of matrix $k(\mathbf{W})$ encodes the similarities of SPD data while the elements of $\mathbf{YY}^T$ presents the ground-truth similarities of the involved SPD points. The matrix $\mathbf{G}$ is used to select the pairs of SPD points when the corresponding elements are 1. The matrix $\mathbf{U} = \mathbf{I}_N - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T$ is employed for centering the data similarity matrix $k(\mathbf{W})$ and the similarity matrix $\mathbf{YY}^T$ on labels. $N$ is the number of samples, $\mathbf{I}_N$ is the identity matrix of size $N \times N$, $\mathbf{1}_N$ is the vector of size $N$ with all entries being ones, $\mathbf{Y} = [y_1, \ldots, y_N]^T$ is here supposed to be centered, i.e., $\mathbf{U}(\mathbf{YY}^T) \mathbf{U} \rightarrow \mathbf{YY}^T$, for simplicity. In the following, we will give the formulations of the two matrices $k(\mathbf{W})$ and $\mathbf{G}$ in more details.

More specifically, the employed matrix $k(\mathbf{W})$ in Eqn(6) encodes the similarity between each pair of SPD points ($\mathbf{X}_i, \mathbf{X}_j$) on the SPD manifold $\text{Sym}^+_n$, which takes a form as:

$$k_{ij}(\mathbf{W}) = \exp(-\beta \delta^2(\mathbf{X}_i, \mathbf{X}_j)), \quad (7)$$

where $\delta^2(\mathbf{X}_i, \mathbf{X}_j)$ is computed by Eqn(5), $\beta$ is typically set as $\beta = \frac{1}{\sigma^2}$, $\sigma$ is empirically set to mean of distances of the original training sample pairs. Actually, the function $\delta^2$ takes a form of Gaussian kernel function. However, as the objective function $\text{Eqn}(6)$ can be expressed as sum of the similarity regression results of selected sample pairs, the function $\delta^2$ just serves as a tool to encode the similarities and is thus not necessarily positive definite (PD).

In practical application, the computational burden of handling the full kernel matrix $k(\mathbf{W})$ on the SPD manifold scales quadratically with the size of training SPD data. To address this problem, we exploit the idea of Graph Embedding technique [35] to select a limited number of data pairs to construct a sparse kernel matrix (non PD) with a large number of elements being zero. With this idea in mind, the matrix $\mathbf{G}$ is defined to select the pairs of SPD points for SPD similarity learning. By employing it, $\mathbf{G} \circ k(\mathbf{W})$ can be regarded as the sparse kernel matrix, where the operation $\circ$ denotes Hadamard product and the matrix $\mathbf{G} = \mathbf{G}_w + \mathbf{G}_b$. Here, $\mathbf{G}_w$ and $\mathbf{G}_b$ are defined as:

$$\mathbf{G}_w(i, j) = \begin{cases} 1, & \text{if } \mathbf{X}_i \in N_w(\mathbf{X}_j) \text{ or } \mathbf{X}_j \in N_w(\mathbf{X}_i) \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

$$\mathbf{G}_b(i, j) = \begin{cases} 1, & \text{if } \mathbf{X}_i \in N_b(\mathbf{X}_j) \text{ or } \mathbf{X}_j \in N_b(\mathbf{X}_i) \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where $N_w(\mathbf{X}_i)$ is the set of $v_w$ nearest neighbors of $\mathbf{X}_i$ that share the same class label as $y_i$, and $N_b(\mathbf{X}_i)$ is the set of $v_b$ nearest neighbors of $\mathbf{X}_i$ with different class labels from $y_i$. According to the theory of Graph Embedding [35], the within-class similarity graph $\mathbf{G}_w$ and the between-class dissimilarity graph $\mathbf{G}_b$, respectively defined in Eqn(8) and Eqn(9), can encode the local geometrical structure of the space of the processing data. Thus, in addition to speeding up the discriminant learning on the SPD features, exploiting the Graph Embedding technique can not only learn the discriminative information of SPD data but also characterize the local Riemannian geometry of the underlying SPD manifold. The efficiency and effectiveness of the proposed discriminant learning approach working on SPD manifolds will be further studied in the experimental part.

### C. Riemannian Conjugate Gradient Optimization

As discussed before, optimizing in the solution space $\mathbb{R}^n \times m$ of the column full-rank transformation matrices in our objective function can be reduced to optimizing on the Riemannian manifold of rank-$m$ PSD matrices, $\text{Sym}^+_m(m)$. Therefore, in this section, we exploit the Riemannian Conjugate Gradient (RCG) algorithm [28] to optimize our objective function $\mathcal{J}(\mathbf{W})$ in Eqn(6) by deriving its corresponding gradient on the PSD manifold $\text{Sym}^+_m(m)$.

As the Conjugate Gradient algorithm developed in Euclidean space, the RCG algorithm on Riemannian manifolds is an iterative procedure. As given in Algorithm 1 an outline for the iterative part of the algorithm goes as follows: at the k-th iteration, find $\mathbf{W}_k$ by searching the minimum of $\mathcal{J}$ along the geodesic in the direction $\mathbf{H}_{k-1}$ from $\mathbf{W}_{k-1}$, compute the Riemannian gradient $\nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k)$ at this point, choose the new search direction by $\mathbf{H}_k = -\nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k) + \eta \tau (\mathbf{H}_{k-1}, \mathbf{W}_{k-1}, \mathbf{W}_k)$ and iterate until convergence. In the procedure, the Riemannian gradient $\nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k)$ can be easily approximated from its corresponding Euclidean gradient $D_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k)$ by the computation $\nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k) = D_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k) - \mathbf{W}_k \mathbf{W}_k^T D_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k)$, and the operation $\tau (\mathbf{H}_{k-1}, \mathbf{W}_{k-1}, \mathbf{W}_k)$ is the parallel transport of tangent vector $\mathbf{H}_{k-1}$ from $\mathbf{W}_{k-1}$ to $\mathbf{W}_k$. For more details, we refer readers to [28], [5].

As for now, we just need to compute the Euclidean gradient for our objective function $\mathcal{J}(\mathbf{W})$ in Eqn(6). As the Euclidean gradient $D_{\mathbf{W}} \mathcal{J}(\mathbf{W})$ and its corresponding directional derivatives are related with the following equality:

$$D_{\mathbf{W}} \mathcal{J}(\mathbf{W})[\mathbf{W}] = \langle D_{\mathbf{W}} \mathcal{J}(\mathbf{W}), \mathbf{W} \rangle. \quad (10)$$

By employing the basic rule and standard properties of the directional derivatives, $D_{\mathbf{W}} \mathcal{J}(\mathbf{W})[\mathbf{W}]$ can be derived by:

$$D_{\mathbf{W}} \mathcal{J}(\mathbf{W})[\mathbf{W}] = \langle \mathbf{U} G \circ D_{\mathbf{W}} k(\mathbf{W})[\mathbf{W}] U, \mathbf{G} \circ (\mathbf{YY}^T) \rangle_F \| \mathbf{C} \|_F^2$$

$$= \langle \mathbf{L}, G \circ (\mathbf{YY}^T) \rangle_F \| \mathbf{C} \|_F^2 U G \circ D_{\mathbf{W}} k(\mathbf{W})[\mathbf{W}] U \rangle_F$$

$$= \langle D_{\mathbf{W}} k(\mathbf{W})[\mathbf{W}], U \left( G \circ (\mathbf{YY}^T) - \mathcal{J}(\mathbf{W}) \mathbf{C} \right) \| \mathbf{C} \|_F^2 \rangle_F U, \quad (11)$$

**Algorithm 1 Optimization algorithm**

**Input:** The initial matrix $\mathbf{W}_0$

1. $\mathbf{H}_0 \leftarrow 0, \mathbf{W} \leftarrow \mathbf{W}_0$

2. **Repeat**

3. $\mathbf{H}_k \leftarrow -\nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}_k) + \eta \tau (\mathbf{H}_{k-1}, \mathbf{W}_{k-1}, \mathbf{W}_k)$

4. Line search along the geodesic $\gamma$ with the direction $\mathbf{H}_k$ from $\mathbf{W}_{k-1} = \gamma(k - 1)$ to find $\mathbf{W}_k = \arg \min_{\mathbf{W}} \mathcal{J}(\mathbf{W})$

5. $\mathbf{H}_{k-1} \leftarrow \mathbf{H}_k, \mathbf{W}_{k-1} \leftarrow \mathbf{W}_k$

6. **Until** convergence

**Output:** The optimized matrix $\mathbf{W}$

where \( L = U G \circ k(W)U \), \( \langle \cdot, \cdot \rangle_F \) indicates Frobenius inner product, \( \| \cdot \|_F \) denotes Frobenius norm.

Accordingly, the key issue in Eqn\(^{11}\) is to estimate \( D_W k(W) \), where \( k(W) \) is formulated by Eqn\(^{7}\). When \( \delta \) in Eqn\(^{5}\) is the geodesic distance of AIM defined in Eqn\(^{4}\) the Euclidean gradient of \( k(W) \) can be derived as:

\[
D_W k_{ij}(W) = -4\beta k_{ij}(W)(B_i\hat{X}_i^{-1} - B_j\hat{X}_j^{-1}) \log(\hat{X}_j^{-\frac{1}{2}}\hat{X}_i\hat{X}_j^{-\frac{1}{2}}),
\]

where \( B_i = X_iW, \hat{X}_i = W^T X_iW \in Sym_m^+ \).

For other affine invariant metrics such as Stein divergence\(^{26}\), the corresponding Euclidean gradient of \( k(W) \) with the geodesic distance function \( \delta \) being defined in Eqn\(^{2}\) can be computed by:

\[
D_W k_{ij}(W) = -\beta k_{ij}(W)(B_i + B_j)A_{ij}^{-1} - B_i\hat{X}_i^{-1} - B_j\hat{X}_j^{-1},
\]

where \( A_{ij} = W^TX_i^2X_i^2W \), and hence be able to work in our new proposed framework.

When endowing the SPD manifold with the non-affine invariant metric LEM, it seems not easy to calculate the Euclidean gradient of \( D_W k(W) \) due to the matrix logarithms in it. Thus, we need to study the problem of the computation of the Euclidean gradient for the LEM case in the following.

First, we decompose the derivative of LEM w.r.t. \( W \) into three derivatives with the trace form \( Tr(\cdot) \):

\[
\begin{align*}
D_W(\| \log(W^TX_iW) - \log(W^TX_iW)\|_F) &= D_W(Tr(\log^2(W^TX_iW))) + D_W(Tr(\log^2(W^TX_iW))) \\
&- 2D_W(Tr(\log(W^TX_iW)\log(W^TX_iW))).
\end{align*}
\]

Proposition 1. The derivatives of the three trace forms \( Tr(\cdot) \) in Eqn\(^{12}\) can be respectively computed by (Here, \( B_i = X_iW, \hat{X}_i = W^TX_iW \)):

\[
\begin{align*}
D_W(Tr(\log^2(\hat{X}_i))) &= 4B_iD\log(\hat{X}_i)[\log(\hat{X}_i)]. \\
D_W(Tr(\log^2(\hat{X}_j))) &= 4B_jD\log(\hat{X}_j)[\log(\hat{X}_j)]. \\
D_W(Tr(\log(\hat{X}_i)\log(\hat{X}_j))) &= 2B_iD\log(\hat{X}_i)[\log(\hat{X}_j)] + 2B_jD\log(\hat{X}_j)[\log(\hat{X}_i)].
\end{align*}
\]

Proof. The three formulas for the gradients with the matrix logarithm correspond to the three ones with rotation matrices in \(^{39}\) (section 5.3), where a detailed proof is given.

By using Proposition 1, i.e. Eqn\(^{15}\) Eqn\(^{16}\) Eqn\(^{17}\) and the sum rule of the directional derivatives, we derive \( D_W k(W) \) with \( \delta \) being the geodesic distance of LEM in Eqn\(^{5}\) as:

\[
D_W k_{ij}(W) = -4(B_iD\log(\hat{X}_i)[\log(\hat{X}_i)] - \log(\hat{X}_i)) \\
+ B_jD\log(\hat{X}_j)[\log(\hat{X}_j)] - \log(\hat{X}_i)]\beta k_{ij}(W).
\]

To calculate the formula Eqn\(^{18}\) we then apply a function of block triangular matrix developed in \(^{37}\) to compute the form of \( \log(\hat{X})[H] \), which is the directional (Fréchet) derivative of \( \log \) at \( \hat{X} \in Sym_m^+ \) along \( H \in Sym_m \). The following theorem shows that the directional derivative appears as the \((1,2)\) block of the resulting big matrix when \( f : \hat{X} \mapsto \log(\hat{X}) \) is evaluated at a certain block triangular matrix.

Theorem 1. Let \( f : \hat{X} \mapsto \log(\hat{X}) \) be \( 2n - 1 \) times continuously differentiable on \( G \) and let the spectrum of \( \hat{X} \) lie in \( \mathbb{G} \), where \( \mathbb{G} \) is an open subset of \( \mathbb{R} \). Then

\[
f \left( \begin{bmatrix} \hat{X} & H \\ 0 & \hat{X} \end{bmatrix} \right) = \begin{bmatrix} f(\hat{X}) & D\log(\hat{X})[H] \\ 0 & f(\hat{X}) \end{bmatrix}.
\]

Proof. The result is proved by Naifeld and Havel\(^{38}\) (Theorem 4.11) under the assumption that \( f \) is analytic.

By using Theorem 1, the directional derivative of the matrix logarithm can be easily computed. The pseudo matlab code of computing \( D\log(\hat{X})[H] \) is simply listed as: \( n = \text{size}(X, 1); Z = \text{zeros}(n); A = \log([X, H ; Z, X]); D = A(1:n, (n+1):end), \) where \( D = D\log(\hat{X})[H] \).

With the derived gradient formulas in Eqn\(^{12}\) Eqn\(^{13}\) and Eqn\(^{18}\) the Euclidean gradient Eqn\(^{11}\) of the objective function Eqn\(^{6}\) for these cases can be computed to feed into the exploited RCG algorithm working on the PSD manifold. Since the global convergence of the RCG algorithm has been well-studied in the survey\(^{39}\), we do not investigate it any further here. The main time complexity of the algorithm is computing the gradient Eqn\(^{11}\) being \( O(\ell k_0n^2m + k_1nm^2) \) (\( \ell \) is the iteration number, \( k_0/k_1 \) is the number of selected samples/pairs, \( n/m \) is the dimension of the original/target manifold) in the LEM case. In the experiment, we will also study the running time of each iteration of the algorithm by varying the number of selected between-class pairs for each SPD sample.

IV. EXPERIMENTS

In this section, we study the effectiveness of the proposed geometry-aware SPD similarity learning (SPDSL) approach by conducting experimental evaluations for three visual classification tasks including face recognition, material categorization and action recognition.

In these three tasks, the SPD features have been shown to provide powerful representations for images and videos via set-based covariances\(^{29}\), \(^{10}\), \(^{11}\), region covariances\(^{1}\), \(^{2}\) and joint covariance descriptors\(^{6}\), \(^{5}\). Therefore, they are natural choices to evaluate the proposed SPDSL exploiting Affine-Invariant metric (AIM), Stein divergence and Log-Euclidean metric (LEM).

To evaluate the effectiveness of the proposed SPDSL approach, we compare three categories of SPD-based learning methods, including basic Riemannian metric baseline methods,
1) Basic Riemannian metrics on SPD manifold: Affine-Invariant Metric (AIM) [13], Stein divergence [20], Log-Euclidean Metric (LEM) [8]
2) Kernel learning based SPD matrix learning methods: PLS-based Covariance Discriminative Learning (CDL) [9], Riemannian Sparse Representation (RSR) [3] and Log-Euclidean Kernels (LEK) [40]
3) Dimensionality reduction based SPD matrix learning methods: Log-Euclidean Metric Learning (LEML) [25] and SPD Manifold Learning (SPDML-AIM and SPDML-Stein) [5] with AIM and Stein divergence

Note that, the proposed SPDSL belongs to the last category of SPD discriminant learning methods. As this paper focuses on studying the problem of supervised SPD discriminant learning, we here only report the performances of the original discriminant learning methods such as SPDML rather than those of further coupling them with other classifiers as done in the work [5]. In addition, in order to study the discriminant learning power of our proposed framework, we replace its supervised learning scheme with that of SPDML but still perform optimization on the exploited solution space. The adaption of the proposed SPDSL is denoted with SPDSL^*, SPDSL-Stein^* and SPDSL-LEM^*.

For RSR, the parameter $\beta$ was densely sampled around the order of the mean distance and the parameter $\lambda$ is sampled in the range of $[0.0001, 0.001, 0.01, 0.1]$. For LEK, there are three implementations based on polynomial, exponential and radial basis kernels, which are respectively denoted as LEK-$\kappa_p$, LEK-$\kappa_e$ and LEK-$\kappa_g$. For LEK-$\kappa_p$ and LEK-$\kappa_e$, we densely sampled the $n$ from 1 to 50. The parameters $\beta$ in LEK-$\kappa_g$ and the $\lambda$ in the three LEK versions were all tuned in the same way as RSR. For LEML, the parameter $\eta$ is tuned in the range of $[0.1, 1, 10]$, and $\zeta$ is tuned from 0.1 to 0.5. For SPDML and our method SPDSL, the maximum iteration number of the optimization algorithm is set to 50, the parameters $v_w$ is fixed as the minimum number of samples in one class, the dimensionality of the lower-dimensional SPD manifold and $v_b$ were tuned by cross-validation. The parameter $\beta$ in our method is set to $\beta = \frac{1}{\sigma^2}$, where $\sigma$ is equal to the mean distance of all pairs of training data.

A. Face Recognition

As the first experiment, we use YouTube Celebrities (YTC) video face database [41] to perform the task of video face recognition. The dataset is a quite challenging and widely used in the study of video face recognition. It has 1,910 video clips of 47 subjects collected from YouTube. Most clips contains hundreds of frames, which are often low resolution and highly compressed with noise and low quality.

For the testing protocol, following [9], [10], [25], this dataset is randomly split into the gallery and the probe, which have 3 image sets and 6 image sets respectively for each subject. The process of random testing was repeated 10 times for the evaluation on video face recognition.

In our experiment, each face image in videos is cropped into $20 \times 20$ intensity image and then histogram equalized to eliminate lighting effects. Following the works [9], [25], we extract the set-based covariance matrix for each video sequence of frames on this dataset. To avoid matrix singularity, we add a small ridge $\delta I$ to each covariance matrix $\Sigma$, where $\delta = 10^{-3} \times \text{trace}(\Sigma)$ and $I$ is the identity matrix. In the literature, the mean face in each video has been proved to benefit video face recognition. Therefore, we improve the set-based covariance matrix feature by concatenating it with the mean to yield a $(d + 1)$-dimensional SPD matrix as $\text{trace}(\Sigma + \mu \mu^T)$ $\mu^T 1$, $\mu^T 1$, where $\mu \in \mathbb{R}^d$ and $\Sigma \in S^d_+$ represents the mean and the covariance matrix of one image set. Note that the dimensions of target manifolds for dimensionality reduction methods are all set as 40 for the YTC database.

As can be seen from Table 1, the baseline method LEM outperforms the other two baselines AIM and Stein in most of cases, which demonstrates that the LEM is more effective than the other two Riemannian metrics in the evaluation. The results in Table 1 also show that most of the kernel learning based (Category2) and dimensionality reduction based (Category3) methods boost the accuracies of the baselines AIM, Stein and LEM. This also concludes that learning discriminative SPD features in these methods can help the visual recognition tasks.

Compared with the state-of-the-art kernel learning based methods CDL and RSR, the dimensionality reduction based methods LEM and SPDML perform worse in the task. In contrast, our SPDSL improves LEM and SPDML by around 2% and 7% respectively, and achieve comparable performance with CDL and RSR. In the comparison with SPDML, the performances of the adaption of our new SPD similarity learning framework SPDML-AIM^* and SPDML-Stein^* are close to those of SPDML-AIM and SPDML-Stein. This shows the former solution can be approximated by the latter solution when the involved Riemannian metric is affine invariant. Nevertheless, after using the proposed supervised learning technique, SPDSL-AIM and SPDSL-Stein clearly outperform SPDML method. In addition, our SPDSL can handle the
TABLE I
AVERAGE RANK-1 FACE RECOGNITION RATES (%) WITH STANDARD DEVIATION OF THREE CATEGORIES OF COMPETING METHODS INCLUDING THE PROPOSED SPDSL ON THE YTC DATABASE.

| Category   | AIM   | Stein | LEM   |
|------------|-------|-------|-------|
| Accuracy   | 62.85 ± 3.46 | 61.46 ± 3.52 | 63.91 ± 3.25 |
| Category 2 | CDL [9] | RSR [13] | LEK-w_{ns} [40] | LEK-w_{np} [40] | LEK-w_{sc} [40] |
| Accuracy   | 72.67 ± 2.47 | 72.77 ± 2.69 | 61.85 ± 3.24 | 62.17 ± 3.52 | 56.30 ± 3.62 |
| Category 3 | LEML [25] | SPDML-AIM [15] | SPDML-Stein [15] |
| Accuracy   | 70.53 ± 2.95 | 64.66 ± 2.92 | 61.57 ± 3.43 |
| The proposed | SPDML-AIM* | SPDML-Stein* | SPDML-LEM* |
| Accuracy   | 64.27 ± 2.84 | 62.31 ± 3.48 | 69.32 ± 2.04 |

*SPDML-AIM, SPDML-Stein, SPDML-LEM

Fig. 3. Recognition accuracy of the proposed SPDSL-LEM on the YTC dataset for varying values of $v_b$ (i.e., different sparse degrees of the involved kernel matrix $k(W)$).

Fig. 4. Running time of the proposed SPDSL-LEM on the YTC dataset for varying values of $v_b$ (i.e., different sparse degrees of the involved kernel matrix $k(W)$).

Fig. 5. Convergence behavior of the exploited RCG algorithm for SPDSL-LEM in 10 random testings of the YTC dataset with the parameter $v_b = 2$.

The efficiency of the proposed SPDSL technique is studied as well. As shown in Fig[4], the running time is average training time of each iteration of the optimization algorithm, which typically iterates 50 times. Specifically, we perform the testing on the YTC dataset, and employ an Intel(R) Core(TM) i5-2400 (3.10GHz) PC. As the value of $v_b$ increases, the running time turns to be much higher especially when $k(W)$ is full, whose running time is around 13,975 seconds (i.e., about 30 times of the case of $v_b = 2$ at each iteration, and extremely expensive when the algorithm iterates 50 times) running on YTC. Hence, when huge datasets are involved, the sparse kernel case scales much better than the full (PD) kernel case with very slight gain/loss of accuracy.

In the end, we also investigate the convergence behavior of the exploited RCG algorithm for our SPDSL approach. As seen from the results in Fig[5] the optimization algorithm on the PSD manifold is able to converge to a favorable solution after several tens of iterations.
TABLE II
AVERAGE RECOGNITION ACCURACIES (%) WITH STANDARD DEVIATION OF THREE CATEGORIES OF COMPETING METHODS INCLUDING THE PROPOSED SPDSL ON THE UIUC DATABASE.

| Category1 | AIM | Stein | LEM |
|-----------|-----|-------|-----|
| Accuracy  | 46.30 ± 2.90 | 42.87 ± 2.27 | 46.30 ± 2.86 |
| Category2 | CDL [9] | RSR [13] | LEK-κ e [40] | LEK-κ e [40] | LEK-κ e [40] |
| Accuracy  | 54.91 ± 4.72 | 52.41 ± 4.03 | 48.89 ± 3.79 | 49.54 ± 3.67 | 49.63 ± 3.03 |
| Category3 | LEM [25] | SPDML-AIM [15] | SPDML-Stein [15] |
| Accuracy  | 52.53 ± 2.13 | 48.09 ± 1.82 | 49.17 ± 2.37 |
| The proposed | SPDML-AIM* | SPDML-Stein* | SPDML-LEM* | SPDML-AIM* | SPDML-Stein* | SPDML-LEM* |
| Accuracy  | 50.00 ± 3.60 | 49.35 ± 2.47 | 50.28 ± 3.78 | 52.31 ± 3.55 | 51.57 ± 4.16 | 52.13 ± 3.49 |

Fig. 6. Samples from the UIUC material dataset [42].

B. Material Categorization

For the task of material categorization, we conduct experiments on the UIUC material dataset [42]. This dataset includes 18 subcategories of materials taken in the wild from four general categories: bark, fabric, construction materials, and outer coat of animals. Each subcategory contains 12 images taken at different scales. Several samples from this database are shown in Fig 6.

Both Region Covariance Matrices (RCMs) [1] and SIFT features [43] have been shown to be robust and discriminative for material categorization [42]. As done in [5], we extract RCMs of size 128 × 128 using 128 dimensional SIFT features from gray scale images. Specifically, we resize each image to 400 × 400 and compute the dense SIFT descriptors on a grid with 4 pixels spacing (each patch size is 16x16, the number of angles is 8, the number of Bins is 4). In each grid point, one 128-dimensional SIFT feature is thus yielded. For dimensionality reduction methods, the dimensions of target manifolds are all set as 40 in the evaluation.

Following the work [5], on the UIUC dataset, we randomly select half of the images from each subcategory as training data, and the remaining images as testing data. This process of evaluation is conducted 10 times in our experiment.

In Table I for the competing methods, we report their average accuracies with standard deviations of 10 random testings on the UIUC dataset. As concluded in the last evaluation, the proposed dimensionality reduction technique SPDSL improves the most related method SPML method by 2%-4%, and achieves comparable performances with the state-of-the-art methods.

C. Action Recognition

We then employ the HDM05 database [44] to handle with the problem of human action recognition from motion capture sequences. As shown in Fig 7, this dataset contains 2,337 sequences of 130 motion classes, e.g., ‘clap above head’, ‘lie down floor’, ‘rotate arms’, ‘throw basket ball’, in 10 to 50 realizations executed by various actors.

The 3D locations of 31 joints of the subjects are provided over time acquired at the speed of 120 frames per second. Following the previous works [6], [5], we represent an action of a K joints skeleton observed over m frames by its joint covariance descriptor. This descriptor is an form of SPD matrix of size 3K × 3K, which is computed by the second order statistics of 93-dimensional vectors concatenating the 3D coordinates of the 31 joints in each frame.

As the evaluation protocol on UIUC, on this dataset, we also conduct 10 times random evaluations, in which half of sequences (around 1,100 sequences) are randomly selected for training data, and the rest are used for testing. On the HDM05 database, the work [5] only used 14 motion classes for evaluation while we tested these methods for identifying 130 action classes.

Table III summarizes the performances of the comparative algorithms on the UIUC dataset. In the evaluation, the dimensions of resulting manifolds achieved by dimensionality reduction methods are all set as 30. Different from the last two evaluations, CDL and RSR performance worse than other competing methods. The proposed SPDSL again improves the existing dimensionality reduction based methods LEML and SPDML with 1%-3%, and achieve state-of-the-art performance on the HDM05 database.
TABLE III
AVERAGE RECOGNITION ACCURACIES (%) WITH STANDARD DEVIATION OF THREE CATEGORIES OF COMPETING METHODS INCLUDING THE PROPOSED SPDSL ON THE HDM05 DATABASE.

| Category | AIM ± 1.74 | Stein ± 2.63 | LEM ± 2.13 |
|----------|------------|--------------|------------|
| Accuracy | 42.70 ± 2.13 | 42.12 ± 2.53 | 43.98 ± 2.13 |

| Category2 | CDL [9] | RSR [13] | LEK-κ [40] | LEK-κ [40] | LEK-κ [40] |
|-----------|---------|---------|----------|----------|----------|
| Accuracy  | 41.74 ± 1.92 | 41.12 ± 2.53 | 47.22 ± 1.62 | 46.87 ± 1.72 | 48.72 ± 3.00 |

| Category3 | LEML [25] | SPDSL-AIM [5] | SPDSL-Stein [5] |
|-----------|-----------|---------------|-----------------|
| Accuracy  | 46.87 ± 2.19 | 47.25 ± 2.78 | 46.21 ± 2.65 |

The proposed | SPDSL-AIM* | SPDSL-Stein* | SPDSL-LEM | SPDSL-AIM | SPDSL-Stein | SPDSL-LEM |
|------------|-----------|-------------|----------|----------|------------|----------|
| Accuracy   | 47.93 ± 2.62 | 46.35 ± 2.45 | 48.88 ± 3.18 | 48.09 ± 2.49 | 49.02 ± 2.93 | 49.13 ± 2.74 |

D. Discussion

Since our method SPDSL and the two methods SPDML, LEML adopt the same SPD matrix learning scheme, we here mainly make two pieces of discussions between them.

First, compared with the related manifold learning method SPDML, our SPDSL framework proposes a more general solution and a more favorable objective function. This point has been validated by the three evaluations. As can be seen from Table I, Table II and Table III, there are two key conclusions observed from the three visual recognition tasks:

a) As for the new solution, its main benefits lie in enlarging the search domain and opening up the possibility of using non-affine invariant metrics (e.g. LEM). While SPDML* for affine invariant metrics AIM and Stein improves SPDML mildly (this may depend on the data), the gains of SPDML*-LEM over the AIM and Stein cases are relatively obvious, i.e. 1.65%, 2.15%, 6.21% on average, respectively for the three datasets.

b) The new objective function (for similarity regression) is quite different from that (for graph embedding) used in [5]. While it’s hard to theoretically prove the gains, we have empirically studied its priority. By comparing SPDSL with SPDML*, the improvements for the three datasets are 2.13%, 1.03%, 6.34% on average for the three used databases, respectively.

Second, in contrast to LEML which focuses on metric learning, our SPDSL learns discriminative similarities on SPD manifolds. Besides, while LEML performs metric learning on the tangent space of SPD manifolds, the proposed SPDSL learns similarity directly on the SPD manifolds. Intuitively, our learning scheme would more faithfully respect the Riemannian geometry of the data space, and thus could lead to more favorable SPD features for classification tasks. From the above three evaluations, we can see some improvements of SPDSL over LEML.

V. Conclusions

We have proposed a geometry-aware SPD similarity learning (SPDSL) framework for more robust visual classification tasks. Under this framework, by exploiting the Riemannian geometry of PSD manifolds, we open the possibility of directly learning the manifold-manifold transformation matrix. To achieve the discriminant learning on the SPD features, this work devises a new SPDSL technique working on SPD manifolds. With the objective of the proposed SPDSL, we derive an optimization algorithm on PSD manifolds to pursue the transformation matrix. Extensive evaluations have studied both the effectiveness of efficiency of our SPDSL on three challenging datasets.

For future work, the study on the relationship between the selected Riemannian metrics of PSD manifolds and SPD manifolds would be interesting for the problem of supervised SPD similarity learning. Besides, if neglecting the designed discriminant function on SPD features, learning the transformation on SPD features for object sets is equal to learning the projection on single object features. Thus, this work can be extended to learn hierarchical representations on object feature by leveraging the current powerful deep learning techniques.

ACKNOWLEDGMENT

This work has been carried out mainly at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). It is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61173065, 61222211, and 61379083.

REFERENCES

[1] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in Proc. Euro. Conf. Comput. Vision, 2006.
[2] O. Tuzel, F. Porikli, and P. Meer., “Pedestrian detection via classification on Riemannian manifolds,” IEEE Trans. on Pattern Anal. Mach. Intell., vol. 30, no. 10, pp. 1713–1727, 2008.
[3] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, “Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach,” in Proc. Euro. Conf. Comput. Vision, 2012.
[4] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. T. Harandi, “Kernel methods on the Riemannian manifold of symmetric positive definite matrices,” in Proc. Comput. Vision Pattern Recog., 2013.
[5] M. T. Harandi, M. Salzmann, and R. Hartley, “From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices,” in Proc. Euro. Conf. Comput. Vision, 2014.
