Log-Euclidean Metric Learning on Symmetric Positive Definite Manifold with Application to Image Set Classification

Zhiwu Huang[†‡] ZHIWU.HUANG@VIPL.ICT.AC.CN
Ruiping Wang[†§] WANGRUIPING@ICT.AC.CN
Shiguang Shan[†§] SGSHAN@ICT.AC.CN
Xianqiu Li[†‡] XIANQIU.LI@VIPL.ICT.AC.CN
Xilin Chen[†§] XLCHEN@ICT.AC.CN
[†]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[‡]University of Chinese Academy of Sciences, Beijing, 100049, China
[§]Cooperative Medianet Innovation Center, China

Abstract

The manifold of Symmetric Positive Definite (SPD) matrices has been successfully used for data representation in image set classification. By endowing the SPD manifold with Log-Euclidean Metric, existing methods typically work on vector-forms of SPD matrix logarithms. This however not only inevitably distorts the geometrical structure of the space of SPD matrix logarithms but also brings low efficiency especially when the dimensionality of SPD matrix is high. To overcome this limitation, we propose a novel metric learning approach to work directly on logarithms of SPD matrices. Specifically, our method aims to learn a tangent map that can directly transform the matrix logarithms from the original tangent space to a new tangent space of more discriminability. Under the tangent map framework, the novel metric learning can then be formulated as an optimization problem of seeking a Mahalanobis-like matrix, which can take the advantage of traditional metric learning techniques. Extensive evaluations on several image set classification tasks demonstrate the effectiveness of our proposed metric learning method.

1. Introduction

Over the years, a surge of methods (Wang et al., 2008; Wang & Chen, 2009; Cevikalp & Triggs, 2010; Hu et al., 2011; Harandi et al., 2011; Wang et al., 2012; Yang et al., 2013; Zhu et al., 2013; Lu et al., 2013; 2014; Hayat et al., 2014; Mahmood et al., 2014; Huang et al., 2015b; Wang et al., 2015) have been suggested for the problem of image set classification, where each sample either for training or for testing refers to a set of image instances. In contrast to a single image, a set of images provide more information to describe subjects of interest. However, large intra-set and/or inter-set variations pose great challenges for the image set classification tasks.

In the existing literature, with strong ability to characterize variations, Symmetric Positive Definite (SPD) matrices have been proven to provide powerful representations for image sets. One typical instance of SPD matrix is covariance matrix, which is employed by several works (Wang et al., 2012; Vemulapalli et al., 2013; Lu et al., 2013; Huang et al., 2014; 2015a) to present the second-order statistics of the set of images. Another type of SPD matrix is the resulting matrix of Gaussian distribution, which is exploited by a few works (Shakhnarovich et al., 2002; Arandjelović et al., 2005; Huang et al., 2015b; Wang et al., 2015) to capture the entire probabilistic model of object variations for more robust image set classification. By employing information geometry theory (Amari & Nagaoka, 2000; Lovrić et al., 2000), the two works (Huang et al., 2015b; Wang et al., 2015) transform the Gaussian model to an SPD matrix as a compact and effective representation for image set.

However, such advantages come along with the challenge of representing and handling the SPD matrices appropriately. As studied in (Arsigny et al., 2005; Pennec et al., 2006; Arsigny et al., 2007; Sra, 2012), SPD matrices do not lie in a vector space but instead on a specific Riemannian manifold. Therefore, previous metrics defined or learned based on Euclidean structure are no longer adequate (if not in-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

valid), due to their neglecting the manifold geometry. Furthermore, they even lead to several undesirable effects such as the swelling of diffusion tensors and the asymmetry after inversion in the case of SPD matrices (Arsigny et al., 2006; Pennec et al., 2006).

To circumvent these problems, some Riemannian metrics are proposed on the SPD manifold. For instance, with the Affine-Invariant Metric (AIM) proposed in (Pennec et al., 2006), the swelling effect has disappeared and the symmetry with respect to inversion is respected. However, the price paid for its success is a high computational burden in practice (Arsigny et al., 2007). Later, a new Riemannian metric named Log-Euclidean Metric (LEM) is introduced by giving the space of SPD matrices a Lie group structure (Arsigny et al., 2006; 2007). This Riemannian metric equips the SPD manifold with the bi-invariant metric induced by Lie group to reduce the manifold to a flat Riemannian space. In this space, classical Euclidean computations can be applied to the domain of SPD matrix logarithms (i.e., the tangent space at identity matrix on SPD manifold). Therefore, LEM fully overcomes the computational limitations of the AIM framework, while conserving excellent theoretical properties.

By employing the well-studied LEM, quite a few works (Sivalingam et al., 2009; Tosato et al., 2010; Wang et al., 2012; Carreira et al., 2012; Jayasumana et al., 2013; Vemulapalli et al., 2013; Li et al., 2013; Minh et al., 2014; Vemulapalli & Jacobs, 2015) proposed a series of methods to learn discriminant function for SPD matrices. For example, several works (Sivalingam et al., 2009; Tosato et al., 2010; Carreira et al., 2012; Vemulapalli & Jacobs, 2015) first apply LEM to convert each SPD matrix logarithm of size $d \times d$ into a vector of size $\frac{d(d+1)}{2}$ in the tangent space at the identity matrix, and then learn more discriminant vectors of size l in this tangent space (see Fig. 1 (a)-(b1)-(c)). Another family of works (Wang et al., 2012; Jayasumana et al., 2013; Vemulapalli et al., 2013; Li et al., 2013; Minh et al., 2014) derive LEM based kernel functions in order to embed the Riemannian manifold of SPD matrices into a high-dimensional Hilbert space. Actually, this kind of methods also first converts the matrix logarithms of size $d \times d$ into a vector of size d^2 in the tangent space at the identity matrix and then learns more discriminant vectors of size l (see Fig. 1 (a)-(b2)-(c)). Obviously, these two schemes of handling SPD matrix logarithms are not perfect, because the basic elements are symmetric matrices, but not usual matrices. Therefore, in these methods, the vector operation of the matrix logarithms will inevitably distort the intrinsic geometrical structure of the tangent space.

In this paper, we develop a novel metric learning method on the SPD manifold. Unlike existing methods that generally unfold SPD matrix logarithms to vectors, our method

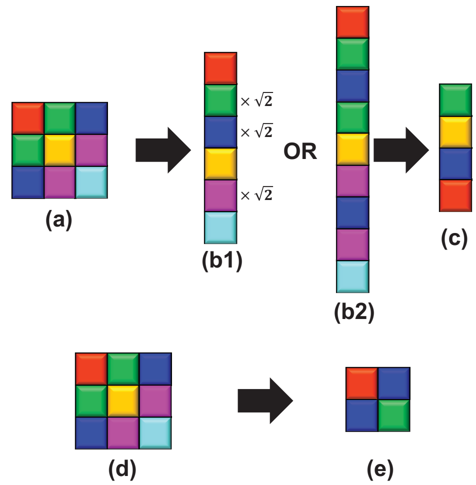

Figure 1. Different schemes to handle the SPD matrix logarithm under Log-Euclidean Metric (LEM) framework. Most of the traditional methods commonly first convert the original $(d \times d)$-dimensional (here $d = 3$) matrix logarithm **(a)** into a $\frac{d(d+1)}{2}$-dimensional vector **(b1)** or d^2-dimensional vector **(b2)** and finally learn an l-dimensional (here $l = 4$) vector representation **(c)**. In contrast, the proposed method works directly on the original $(d \times d)$-dimensional matrix logarithm **(d)** and learns a discriminative $(k \times k)$-dimensional (here $k = 2$) matrix logarithm **(e)**.

works directly on the original $(d \times d)$-dimensional SPD matrix logarithms and finally learns a discriminant $(k \times k)$-dimensional symmetric matrix (which can be thought of SPD matrix logarithms) under the LEM framework (see Fig .1 (d)-(e)). By performing metric learning in this scheme, our proposed method has two important advantages over the traditional ones. **First**, our learned mapping from the original tangent space to a new discriminant space of learned SPD matrix logarithms will more faithfully respect the geometrical structure of the original tangent space and thus benefit from its useful properties. Additionally, keeping the symmetry of the learned square matrix will also obtain the corresponding manifold of SPD matrices via the inverse mapping (i.e. the matrix exponential) of the matrix logarithm which is a smooth diffeomorphism (Arsigny et al., 2007). **Second**, learning discriminant function on square matrices is more efficient than learning on data vectors. To understand this point, one can take PCA and 2DPCA (Yang et al., 2004) as an analogy. In previous methods, the (within and between class) scatter matrix of vectorized SPD matrix logarithm is of size $d^2 \times d^2$ or $\frac{d(d+1)}{2} \times \frac{d(d+1)}{2}$, while ours is much smaller, only $d \times d$. Therefore, it is easier to evaluate the scatter matrix of the logarithms of SPD matrices accurately, and thus more efficient to perform learning on the smaller scatter matrix.

With this idea in mind, we formulate the new metric learning on the SPD manifold as a problem of learning Mahalanobis-like matrix, which inherits the favorable properties of the traditional metric learning methods developed for vector space. Specifically, the main contribution of this work is threefold:

- To overcome the limitation of traditional methods that rely on the intermediate vectorization of SPD matrix logarithm, we develop a novel Log-Euclidean Metric Learning (LEML) framework to directly manipulate the original SPD matrix logarithm.
- To learn a more discriminant metric on SPD manifold, we propose a LogDet divergence based objective function by extending the metric learning machinery (Davis et al., 2007) from the vector-form version to the matrix-form version.
- To optimize this objective function, we exploit cyclic Bregman projections (Kulis et al., 2009) by designing an update rule that is best tailored to preserve the original SPD tangent space structure.

2. Background

In this section, we first introduce the Riemannian manifold of SPD matrices, and then review the Log-Euclidean Metric on the SPD manifold.

2.1. The Manifold of SPD Matrices

As studied in (Arsigny et al., 2005; Pennec et al., 2006; Arsigny et al., 2007), the space of $d \times d$ SPD matrices, when endowed with an appropriate Riemannian metric, forms a specific type of Riemannian manifold, i.e., SPD manifold $\mathbb{S}_+^d$. The SPD manifold is a topological space locally similar to Euclidean space and with globally defined differential structure, which leads the possibility to define the derivatives of the curves on the manifold. By using the logarithm map $\log_{\boldsymbol{S}_1} : \mathbb{S}_+^d \rightarrow T_{\boldsymbol{S}_1}\mathbb{S}_+^d (\boldsymbol{S}_1 \in \mathbb{S}_+^d)$, the derivatives at the point $\boldsymbol{S}_1$ on the manifold lie in a tangent space $T_{\boldsymbol{S}_1}\mathbb{S}_+^d$, which has an inner product $\langle , \rangle_{\boldsymbol{S}_1}$. The family of inner products on all tangent spaces is known as the Riemannian metric of the manifold. With the Riemannian metric, the geodesic distance between two points $\boldsymbol{S}_1, \boldsymbol{S}_2$ on the SPD manifold is generally measured by $\langle \log_{\boldsymbol{S}_1}(\boldsymbol{S}_2), \log_{\boldsymbol{S}_1}(\boldsymbol{S}_2) \rangle_{\boldsymbol{S}_1}$.

Arising from a smoothly varying inner product, the two mostly widely used Riemannian metrics on the SPD manifold, i.e., Affine-Invariant Metric (AIM) (Pennec et al., 2006) and Log-Euclidean Metric (LEM) (Arsigny et al., 2007), are qualified to derive true geodesic on the SPD manifold. Due to the curvature of the SPD manifold, the computational cost of AIM is too expensive to work in practice (Arsigny et al., 2007). In contrast, LEM only needs Euclidean computations in the domain of matrix logarithms and thus results in a drastic reduction in computation time. Therefore, in this paper, we focus on studying LEM on the manifold of SPD matrices.

2.2. Log-Euclidean Metric on SPD Manifold

In the study (Arsigny et al., 2007), Log-Euclidean Metric (LEM) for the SPD manifold $\mathbb{S}_+^d$ is derived by exploiting the Lie group structure under the group operation $\boldsymbol{S}_1 \odot \boldsymbol{S}_2 := \exp(\log(\boldsymbol{S}_1) + \log(\boldsymbol{S}_2))$ for $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathbb{S}_+^d$ where $\exp(\cdot)$ and $\log(\cdot)$ denote the usual matrix exponential and logarithm operators.

LEM on the Lie group of SPD matrices corresponds to a Euclidean metric in the SPD matrix logarithmic domain. With LEM on $\mathbb{S}_+^d$, the scalar product between two elements $\boldsymbol{T}_1, \boldsymbol{T}_2$ in the tangent space at a point $\boldsymbol{S}$ is given by:

$$\langle \boldsymbol{T}_1, \boldsymbol{T}_2 \rangle_{\boldsymbol{S}} = \langle D_{\boldsymbol{S}} \log . \boldsymbol{T}_1, D_{\boldsymbol{S}} \log . \boldsymbol{T}_2 \rangle. \tag{1}$$

where $D_{\boldsymbol{S}} \log . \boldsymbol{T}$ indicates the directional derivative of the matrix logarithm at $\boldsymbol{S}$ along $\boldsymbol{T}$. The logarithmic and exponential maps associated with the metric can be expressed in terms of matrix logarithms and exponential:

$$\begin{aligned} \log_{\boldsymbol{S}_1}(\boldsymbol{S}_2) &= D_{\log(\boldsymbol{S}_1)} \exp . (\log(\boldsymbol{S}_2) - \log(\boldsymbol{S}_1)), \\ \exp_{\boldsymbol{S}_1}(\boldsymbol{T}_2) &= \exp(\log(\boldsymbol{S}_1) + D_{\boldsymbol{S}_1} \log . \boldsymbol{T}_2)). \end{aligned} \tag{2}$$

where $D_{\log(\boldsymbol{S})} \exp . = (D_{\boldsymbol{S}} \log .)^{-1}$ is yielded by the differentiation of the equality $\log \circ \exp = \boldsymbol{I}$, and here $\boldsymbol{I}$ is the identity matrix. For more details on the derivation of Eq.1 and Eq.2, please kindly refer to (Arsigny et al., 2007).

From Eq.1 and Eq.2, the geodesic distance between two SPD matrices is achieved by LEM:

$$\begin{aligned} \mathcal{D}_{\ell e}(\boldsymbol{S}_1, \boldsymbol{S}_2) &= \langle \log_{\boldsymbol{S}_1}(\boldsymbol{S}_2), \log_{\boldsymbol{S}_1}(\boldsymbol{S}_2) \rangle_{\boldsymbol{S}_1} \\ &= \| \log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2) \|_F^2. \end{aligned} \tag{3}$$

which corresponds to a Euclidean distance in the logarithmic domain, i.e. the tangent space at identity matrix. In other words, under the LEM framework, the distance between any two points on the SPD manifold is obtained by propagating by translation the scalar product in the tangent space at identity matrix. Therefore, endowed with this metric, the space of SPD matrices is reduced to a flat Riemannian space (Arsigny et al., 2007).

3. Log-Euclidean Metric Learning on SPD Manifold

In this section, we describe our proposed Log-Euclidean Metric Learning (LEML) framework. Specifically, we first introduce the tangent map from the original tangent space to a more discriminative one, and then present the new metric learning problem on the SPD manifold. To solve this new paradigm, an optimization is finally derived.

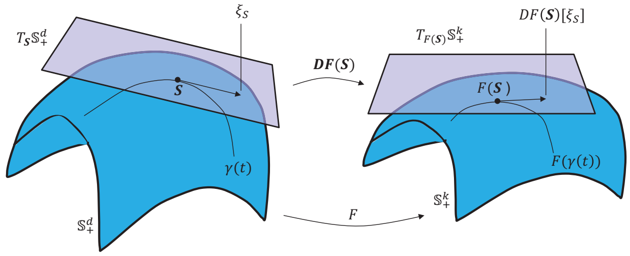

Figure 2. Conceptual illustration of the proposed Log-Euclidean Metric Learning (LEML) on the SPD manifold. To directly manipulate the SPD matrix logarithm, LEML learns a tangent map $DF(\boldsymbol{S}) : T_{\boldsymbol{S}}\mathbb{S}_+^d \rightarrow T_{F(\boldsymbol{S})}\mathbb{S}_+^k$ from the original tangent space $T_{\boldsymbol{S}}\mathbb{S}_+^d$ to a more discriminative tangent space $T_{F(\boldsymbol{S})}\mathbb{S}_+^k$. With this mapping, the corresponding manifold map $F : \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$ from the original SPD manifold $\mathbb{S}_+^d$ to the target SPD manifold $\mathbb{S}_+^k$ can be derived. Here, $\boldsymbol{\xi_S}$ is the tangent vector at a point $\boldsymbol{S}$ of $\mathbb{S}_+^d$, γ is any curve that realizes $\boldsymbol{\xi_S}$.

3.1. Tangent Map

Suppose a group of SPD matrices have been computed from the image set data. We denote them as $\mathcal{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n\}$, where $\boldsymbol{S}_i \in \mathbb{S}_+^d$ with class label l_i (i.e. the category of the corresponding image set). Let $F : \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$ be a smooth mapping from the original SPD manifold $\mathbb{S}_+^d$ to a new one $\mathbb{S}_+^k$ ($k \leq d$), and $\boldsymbol{\xi_S}$ denote the tangent vector at a point $\boldsymbol{S}$ on $\mathbb{S}_+^d$. Then, on the new manifold $\mathbb{S}_+^k$, the tangent vector $DF(\boldsymbol{S})[\boldsymbol{\xi_S}]$ is realized by $F \circ \gamma$, where γ is any curve that realizes $\boldsymbol{\xi_S}$. As shown in Fig.2, the mapping

$$DF(\boldsymbol{S}) : T_S\mathbb{S}_d^+ \rightarrow T_{F(\boldsymbol{S})}\mathbb{S}_k^+ : \boldsymbol{\xi} \mapsto DF(\boldsymbol{S})[\boldsymbol{\xi_S}]. \quad (4)$$

is a linear transformation called tangent map (or differential map) of F at $\boldsymbol{S}$, $T_{\boldsymbol{S}}\mathbb{S}_+^d$ is the tangent space of $\mathbb{S}_+^d$ at point $\boldsymbol{S}$. As indicated in (Absil et al., 2008) , if and only if the tangent map $DF(\boldsymbol{S}) : T_{\boldsymbol{S}}\mathbb{S}_+^d \rightarrow T_{F(\boldsymbol{S})}\mathbb{S}_+^k$ is an injection (respectively, surjection) for every $\boldsymbol{S} \in \mathbb{S}_+^d$, then the mapping $F : \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$ from manifold to manifold is an immersion, i.e., a smooth map whose differential is everywhere injective.

As shown in Eq.3, when equipping the SPD manifold with Log-Euclidean Metric (LEM), the geodesic distance between two SPD matrices $\boldsymbol{S}_i, \boldsymbol{S}_j$ can be reduced to the Euclidean distance between their matrix logarithms in the tangent space at identity matrix. Consequently, the LEM framework is very compatible with the mathematical property of tangent map. With this in mind, in this paper, we focus on learning the tangent map $DF(\boldsymbol{S}) : T_S\mathbb{S}_+^d \rightarrow T_{F(\boldsymbol{S})}\mathbb{S}_+^k$ which is an injection and thus is eligible to derive the original map $F : \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$. Thus, we attempt to define the tangent map by seeking a transformation $\boldsymbol{W} \in \mathbb{R}^{d \times k}$ for the SPD matrix logarithms in the tangent space at identity matrix, which is defined as:

$$f(\log(\boldsymbol{S})) = \boldsymbol{W}^T \log(\boldsymbol{S})\boldsymbol{W}. \quad (5)$$

To ensure that the resulting space is a tangent space on the new manifold $\mathbb{S}_+^k$, we require $\boldsymbol{W}$ to have column full rank (i.e., $rank(\boldsymbol{W}) = k$) such that $\boldsymbol{W}^T \log(\boldsymbol{S})\boldsymbol{W}$ is a real, symmetric matrix, and thus forms a valid tangent space (i.e., the space of SPD matrix logarithms). Obviously, the tangent mapping f is an injection, and can derive the manifold-manifold mapping $F : \mathbb{S}_+^d \rightarrow \mathbb{S}_+^k$.

In contrast to the proposed tangent map framework which directly works on the original SPD matrix logarithm, most traditional LEM-based methods (Sivalingam et al., 2009; Wang et al., 2012; Carreira et al., 2012; Li et al., 2013; Vemulapalli & Jacobs, 2015) first convert the SPD matrix logarithms into vectors of size $\frac{d(d+1)}{2}$, and then learn a map from the vector space to another one for getting l-dimensional vectors. In fact, their learning procedure can also yield a space of symmetric matrices with the finally learned vectors only when satisfying the constraint that the dimensionality l should be the form of $\frac{k(k+1)}{2}$ (e.g., l=3, then k=2). Therefore, these methods do not always result in a valid tangent space, and thus cannot benefit from useful properties of the tangent space to learn a desirable map.

To our knowledge, there are only three similar works (Jung et al., 2012; Harandi et al., 2014; Huang et al., 2015c) seeking to learn the mapping from manifold to manifold. However, the work (Jung et al., 2012) and our previous work (Huang et al., 2015c) learn the manifold-manifold mapping on another class of Riemannian manifold (i.e., sphere or Grassmann manifold). The other work (Harandi et al., 2014) employs two different types of metrics, i.e., Affine-Invariant Metric (AIM) (Pennec et al., 2006) and Stein divergence (Sra, 2012), to learn an embedding of a high-dimensional SPD manifold into another low-dimensional one. In their work, the price paid for the success of employing AIM is a high computational burden in practice, while Stein divergence fails to define true geodesics on SPD manifold (Jayasumana et al., 2013). In addition, different from the two works (Jung et al., 2012; Harandi et al., 2014) aiming to learn a transformation for dimensionality reduction, both our prior work (Huang et al., 2015c) and this current work alternatively learn a Mahalanobis-like matrix, which inherits the appealing properties of the traditional metric learning methods without enforcing explicit constraints for reducing the dimensionality of the original data.

3.2. Metric Learning

As discussed above, with the designed injective tangent map, a new SPD manifold $\mathbb{S}_+^k$ can be derived. Under the LEM framework, the geodesic distance between the transformed data on the new resulting SPD manifold $\mathbb{S}_+^k$

is achieved by (here, we denote $\boldsymbol{T} = \log(\boldsymbol{S})$ for simplicity and use the property that the trace is invariant under cyclic permutations):

$$\begin{aligned}\mathcal{D}_{\ell e}^{W}(\boldsymbol{T}_i, \boldsymbol{T}_j) &= \|\boldsymbol{W}^T\boldsymbol{T}_i\boldsymbol{W} - \boldsymbol{W}^T\boldsymbol{T}_j\boldsymbol{W}\|_F^2 \\ &= tr((\boldsymbol{T}_i - \boldsymbol{T}_j)^T\boldsymbol{P}(\boldsymbol{T}_i - \boldsymbol{T}_j)\boldsymbol{P}).\end{aligned} \tag{6}$$

where $\boldsymbol{P} = \boldsymbol{W}\boldsymbol{W}^T$ is a rank-k symmetric positive semidefinite (PSD) matrix of size $d \times d$. Since both $\boldsymbol{P}$ and $(\boldsymbol{T}_i - \boldsymbol{T}_j)$ are symmetric, any permutation in the trace is allowed and we then rewrite Eq.6 as:

$$\mathcal{D}_{\ell e}^{Q}(\boldsymbol{T}_i, \boldsymbol{T}_j) = tr(\boldsymbol{Q}(\boldsymbol{T}_i - \boldsymbol{T}_j)(\boldsymbol{T}_i - \boldsymbol{T}_j)). \tag{7}$$

where $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{P}$. If $\boldsymbol{P}$ is decomposed by SVD as $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, then $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Lambda}^2\boldsymbol{U}^T$ is thus also a rank-k PSD matrix of size $d \times d$, which takes a similar form as Mahalanobis matrix. Note that, in practical applications, we often set $k = d$ for initializing $\boldsymbol{Q}$ as an identity matrix.

If we regard the form of $(\boldsymbol{T}_i - \boldsymbol{T}_j)(\boldsymbol{T}_i - \boldsymbol{T}_j)$ in Eq.7 as a pair-wise computation of scatter matrix, we see that this matrix is of size $d \times d$. In contrast, the traditional methods (e.g., (Vemulapalli & Jacobs, 2015)) typically transform the SPD matrix logarithms $\boldsymbol{T}_i$ into vectors of size $\frac{d(d+1)}{2}$ or d^2. Consequently, their computed scatter matrix is of size $\frac{d(d+1)}{2} \times \frac{d(d+1)}{2}$ or $d^2 \times d^2$, which is much bigger than ours in Eq.7 leading more difficulties to perform robust metric learning in the higher-dimensional space.

In this paper, we are focusing on the classification tasks on the SPD manifold. Like traditional metric learning methodology, we assume that prior knowledge is known regarding the distances between pairs of points on the new SPD manifold $\mathbb{S}_+^k$. Let's consider the relationships constraining the similarity or dissimilarity between pairs of points: two points are similar if the geodesic distance between them on the new manifold is smaller than a given upper bound, i.e., $\mathcal{D}_{\ell e}^{Q}(\boldsymbol{T}_i, \boldsymbol{T}_j) \leq u$ for a relatively small value of u. Similarly, two points are dissimilar if $\mathcal{D}_{\ell e}^{Q}(\boldsymbol{T}_i, \boldsymbol{T}_j) \geq l$ for a sufficiently large value of l.

Given a set of interpoint distance constraints as described above, our problem is to learn a PSD matrix $\boldsymbol{Q}$ that parameterizes the corresponding Log-Euclidean distance on the new manifold $\mathbb{S}_+^k$. Inspired by the work (Davis et al., 2007) developed for Mahalanobis distance in Euclidean space, we exploit the LogDet divergence $\mathcal{D}_{\ell d}$ to formulate our objective function on the SPD manifold considering its good properties (e.g., fixing the rank) for PSD matrices (Davis et al., 2007; Kulis et al., 2009):

$$\begin{aligned}\min_{\boldsymbol{Q}\succeq 0,\boldsymbol{\xi}} \quad & \mathcal{D}_{\ell d}(\boldsymbol{Q}, \boldsymbol{Q}_0) + \eta\mathcal{D}_{\ell d}(diag(\boldsymbol{\xi}), diag(\boldsymbol{\xi}_0)), \\ s.t. \quad & \boldsymbol{\delta}_{ij}\mathcal{D}_{\ell e}^{Q}(\boldsymbol{T}_i, \boldsymbol{T}_j) \leq \boldsymbol{\xi}_{ij}, \forall c(i,j).\end{aligned} \tag{8}$$

where $\boldsymbol{Q}_0 \in \mathbb{R}^{d\times d}$ is an initialization of $\boldsymbol{Q}$, $\mathcal{D}_{\ell d}(\boldsymbol{Q}, \boldsymbol{Q}_0) = tr(\boldsymbol{Q}\boldsymbol{Q}_0^{-1}) - logdet(\boldsymbol{Q}\boldsymbol{Q}_0^{-1}) - d$, d is the dimensionality of the data sample on the original SPD manifold $\mathbb{S}_+^d$. $\mathcal{D}_{\ell e}^{Q}(\boldsymbol{T}_i, \boldsymbol{T}_j)$ is the distance between two samples in the new tangent space of the generated manifold, which can be computed by Eq.7. $c(i,j)$ denotes the index of the (i,j)-th constraint involving the two samples $\boldsymbol{T}_i, \boldsymbol{T}_j$ in the tangent space. $\boldsymbol{\delta}_{ij} = 1$ if the pair of samples for $c(i,j)$ come from the same class, otherwise $\boldsymbol{\delta}_{ij} = -1$. $\boldsymbol{\xi}_{ij}$ is a vector of slack variables and is initialized to $\boldsymbol{\delta}_{ij}\rho - \zeta\tau$, ρ is the threshold for distance comparison, τ is the margin, ζ is the tuning scale of the margin. In practice, ρ and τ are respectively set as mean and standard deviation of distances of the original training sample pairs. So, there are only two parameters η and ζ that actually need to tune and can be both commonly optimized in a small range.

3.3. Optimization

To solve the optimization problem in Eq.8, we adopt the cyclic Bregman projection algorithm (Bregman, 1967; Censor & Zenios, 1997), which is to choose one constraint per iteration, and perform a projection so that the current solution satisfies the chosen constraint. In the case of inequality constraints, appropriate corrections of $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{Q}$ are also enforced. This process is then repeated by cycling through the constraints. Specifically, by introducing the dual variable $\boldsymbol{\alpha}_{ij}$ to Eq.8 and using Eq.7, we form the Lagrangian $\mathcal{L} = \mathcal{D}_{\ell d}(\boldsymbol{Q}^{t+1}, \boldsymbol{Q}^t) + \eta\mathcal{D}_{\ell d}(diag(\boldsymbol{\xi}^{t+1}), diag(\boldsymbol{\xi}^t)) + \boldsymbol{\alpha}_{ij}(\boldsymbol{\delta}_{ij}tr(\boldsymbol{Q}^{t+1}\boldsymbol{A}) - \boldsymbol{\xi}_{ij}^{t+1})$, where $\boldsymbol{A} = (\boldsymbol{T}_i - \boldsymbol{T}_j)(\boldsymbol{T}_i - \boldsymbol{T}_j)$. To apply the Bregman projection to the LogDet divergence, we set the gradient of the Lagrangian w.r.t. $\boldsymbol{\alpha}_{ij}$, $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{Q}$ to zero and get the following update equations for them:

$$0 = \boldsymbol{\delta}_{ij}tr(\boldsymbol{Q}^{t+1}\boldsymbol{A}) - \boldsymbol{\xi}_{ij}^{t+1}. \tag{9}$$

$$\boldsymbol{\xi}_{ij}^{t+1} = \frac{\eta\boldsymbol{\xi}_{ij}^t}{\eta + \boldsymbol{\delta}_{ij}\boldsymbol{\alpha}_{ij}\boldsymbol{\xi}_{ij}^t}. \tag{10}$$

$$\boldsymbol{Q}^{t+1} = \boldsymbol{V}_t((\boldsymbol{V}_t^T\boldsymbol{Q}^t\boldsymbol{V}_t)^{-1} - \boldsymbol{\delta}_{ij}\boldsymbol{\alpha}_{ij}(\boldsymbol{V}_t^T\boldsymbol{A}\boldsymbol{V}_t))^{-1}\boldsymbol{V}_t^T. \tag{11}$$

where the eigen-decomposition of $\boldsymbol{Q}^t$ is $\boldsymbol{V}_t^T\boldsymbol{\Lambda}_t\boldsymbol{V}_t$, $\boldsymbol{T}_i = \log(\boldsymbol{S}_i), \boldsymbol{T}_j = \log(\boldsymbol{S}_j) \in T_I\mathbb{S}_+^d$ are the constrained data points in the tangent space at identity matrix.

As given in Eq.11, the projection update rule for the learned PSD matrix $\boldsymbol{Q}$ is with a constraint matrix $\boldsymbol{A} = (\boldsymbol{T}_i - \boldsymbol{T}_j)(\boldsymbol{T}_i - \boldsymbol{T}_j)$, where $\boldsymbol{T}_i, \boldsymbol{T}_j$ are symmetric matrices. In contrast, in the works (Davis et al., 2007; Kulis et al., 2009), the projection update rule for the learned PSD matrix is with a rank-one constraint matrix, i.e., $\boldsymbol{A} = (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$, where $\boldsymbol{x}_i, \boldsymbol{x}_j$ are vector-forms. However, following the work (Kulis et al., 2009), we can also apply the Shermann-Morrison inverse formula (Sherman & Morrison, 1950; Golub et al., 2012), i.e., $(B+uv^T)^{-1} = B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1+v^TB^{-1}u}$ to the projection update Eq.11 and finally get a similar ex-

Algorithm 1 Log-Euclidean Metric Learning (LEML)

Input: Training data $\{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots, \boldsymbol{T}_n\}, \boldsymbol{T}_i \in T_I\mathbb{S}_+^d$, constraints $c(i,j)$ with $\boldsymbol{\delta}_{ij} \in \pm 1$, slack parameter η, input PSD matrix $\boldsymbol{Q}_0$, distance threshold ρ, margin parameter τ and tuning scale ζ.
1. $t \leftarrow 1, \boldsymbol{Q}^1 \leftarrow \boldsymbol{Q}_0, \boldsymbol{\xi}_{ij} \leftarrow \boldsymbol{\delta_{ij}}\rho - \zeta\tau, \boldsymbol{\lambda}_{ij} \leftarrow 0, \forall c(i,j)$.
2. **Repeat**
3. Update the dual variable $\boldsymbol{\alpha}_{ij}$ using Eq.13 and Eq.14.
4. Update the vector of slack variables $\boldsymbol{\xi}_{ij}^{t+1}$ using Eq.10.
5. Update the PSD matrix $\boldsymbol{Q}^{t+1}$ using Eq.12.
6. **Until** convergence

Output: PSD matrix $\boldsymbol{Q}$.

pression for $\boldsymbol{Q}^{t+1}$:

$$\boldsymbol{Q}^{t+1} = \boldsymbol{Q}^t + \frac{\boldsymbol{\delta}_{ij}\boldsymbol{\alpha}_{ij}\boldsymbol{Q}^t\boldsymbol{A}\boldsymbol{Q}^t}{1-\boldsymbol{\alpha}_{ij}tr(\boldsymbol{Q}^t\boldsymbol{A})}. \quad (12)$$

By substituting Eq.12 and Eq.10 into Eq.9, we can compute $\boldsymbol{\alpha}_{ij}$ in a closed form as:

$$\boldsymbol{\alpha}_{ij} = \frac{\boldsymbol{\delta_{ij}}\eta}{\eta+1}\left(\frac{1}{tr(\boldsymbol{Q}^t\boldsymbol{A})} - \frac{1}{\boldsymbol{\xi}_{ij}^t}\right). \quad (13)$$

As Eq.8 contains an inequality constraint, we use $\boldsymbol{\lambda}_{ij} \geq 0$ as the corresponding dual variable. To maintain non-negativity of the dual variable (which is necessary for satisfying the KKT conditions), we solve Eq.8 by continually updating α after using Eq.13 as:

$$\boldsymbol{\alpha}_{ij} \leftarrow min(\boldsymbol{\alpha}_{ij}, \boldsymbol{\lambda}_{ij}), \quad \boldsymbol{\lambda}_{ij} \leftarrow \boldsymbol{\lambda}_{ij} - \boldsymbol{\alpha}_{ij}. \quad (14)$$

The resulting optimization procedure is formulated as Algorithm 1. The inputs to the algorithm are the training data with the constrains, the starting PSD matrix $\boldsymbol{Q}_0$, the slack parameter η, distance threshold ρ, margin parameter τ and tuning scale ζ. As noted in (Davis et al., 2007; Kulis et al., 2009), this kind of method of cyclic Bregman projections is guaranteed to converge to the globally optimal solution. The main time cost of this algorithm is to update $\boldsymbol{Q}^{t+1}$ in Eq.12, which is $O(d^2)$ (d is the dimensionality of the original SPD manifold) for each constraint projection. Therefore, the total time cost is $O(Ld^2)$ where L is the total number of the updating $\boldsymbol{Q}^{t+1}$ in Step 5 in Algorithm 1.

4. Experiments

In this section, we study the effectiveness of the proposed approach on three image set classification tasks: set-based object categorization, video-based face recognition and video-based face verification. These tasks are respectively implemented on ETH-80 (Leibe & Schiele, 2003), YouTube Celebrities (Kim et al., 2008) and YouTube Face DB (Wolf et al., 2011) databases.

4.1. Databases and settings

ETH-80 (Leibe & Schiele, 2003) database has 80 image sets of 8 object categories: apples, cows, cups, dogs, horses, pears, tomatoes, and cars. Each category includes 10 different image sets, each of which has 41 images of an object captured under different views. All the images are resized into 20×20 intensity images. Following (Wang et al., 2012; Vemulapalli et al., 2013), we randomly split this dataset into ten different tests, where each category has 5 objects for gallery and the other 5 objects for probes.

YouTube Celebrities (YTC) (Kim et al., 2008) database collects 1910 video sequences of 47 subjects from YouTube. Most of them are highly compressed and low resolution videos. The face region in each image is resized into 20×20 intensity image, and histogram equalized to eliminate lighting effects. Each video clip generates an image set of faces. Following (Wang & Chen, 2009; Wang et al., 2012; Vemulapalli et al., 2013), 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 splitting was repeated ten times.

YouTube Face (YTF) (Wolf et al., 2011) database contains 3425 videos of 1595 different people downloaded from the YouTube. The face frames in each video form an image set and are often of large variations in pose, illumination, and expression. The face images are directly cropped according to the provided data and then resized into 24×40 pixels. We extract the raw intensity feature of resized images. On this dataset, we follow the standard protocol (Wolf et al., 2011) for face verification with 5000 video pairs. These pairs are equally divided into 10 folds, each of which has 250 intra-personal pairs and 250 inter-personal pairs. The experiment is performed in the restricted training setting.

4.2. Evaluation

On the three used databases, we compute a single Gaussian model $\mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$ ($\boldsymbol{m} \in \mathbb{R}^d$ is mean, and $\boldsymbol{S} \in \mathbb{S}_+^d$ is covariance matrix) for each image set, and then construct the corresponding SPD features. Specifically, to avoid matrix singularity, we added a small ridge $\delta\boldsymbol{I}$ to each covariance matrix $\boldsymbol{S}$, where $\delta = 10^{-3} \times tr(\boldsymbol{S})$ and $\boldsymbol{I}$ is the identity matrix. Following our previous works (Huang et al., 2015b; Wang et al., 2015) which employ information geometry theory (Amari & Nagaoka, 2000; Lovrić et al., 2000), we also transform the Gaussian model $\mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$ into a $(d+1)$-dimensional SPD matrix as $|\boldsymbol{S}|^{-\frac{1}{d+1}}\begin{bmatrix}\boldsymbol{S}+\boldsymbol{m}\boldsymbol{m}^T & \boldsymbol{m}\\ \boldsymbol{m}^T & 1\end{bmatrix}$. To more clearly understand the resulting SPD feature, we take the YTC dataset for example. On this dataset, since the image size is 20×20, the calculated sample mean is of size 400, the covariance matrix is of size 400×400, and our finally used SPD matrix is of size 401×401.

Methods	ETH-80	YTC	YTF
AIM (Pennec et al., 2006)	87.50 ± 5.77	62.85 ± 3.46	59.28 ± 2.25
Stein (Sra, 2012)	88.00 ± 5.11	61.46 ± 3.53	58.70 ± 1.97
LEM (Arsigny et al., 2007)	89.25 ± 4.72	63.91 ± 3.25	61.48 ± 2.27
SPDML-AIM (Harandi et al., 2014)	90.75 ± 3.34	64.66 ± 2.92	62.16 ± 2.16
SPDML-Stein (Harandi et al., 2014)	90.50 ± 3.87	61.57 ± 3.43	62.56 ± 2.49
RSR (Harandi et al., 2012)	93.25 ± 3.34	72.77 ± 2.69	N/A
LEK-K_p (Li et al., 2013)	93.26 ± 4.42	61.85 ± 3.24	N/A
LEK-K_e (Li et al., 2013)	93.00 ± 3.69	62.17 ± 3.52	N/A
LEK-K_g (Li et al., 2013)	92.75 ± 3.43	56.30 ± 3.62	N/A
CDL-LDA (Wang et al., 2012)	93.75 ± 3.43	72.70 ± 2.93	66.76 ± 1.89
CDL-PLS (Wang et al., 2012)	93.50 ± 4.44	72.67 ± 2.47	N/A
ITML-LEM (Vemulapalli & Jacobs, 2015)	90.75 ± 4.72	66.51 ± 3.67	60.02 ± 1.84
DCC (Kim et al., 2007)	90.75 ± 4.42	65.48 ± 3.51	68.28 ± 2.21
GDA (Hamm & Lee, 2008)	92.25 ± 4.16	65.02 ± 2.91	66.76 ± 1.72
AHISD (Cevikalp & Triggs, 2010)	77.25 ± 7.50	63.70 ± 2.89	64.80 ± 1.54
CHISD (Cevikalp & Triggs, 2010)	74.25 ± 5.01	66.62 ± 2.79	66.30 ± 1.21
SSDML (Zhu et al., 2013)	80.00 ± 4.23	68.79 ± 2.45	65.38 ± 1.86
LEML	94.75 ± 2.49	70.53 ± 2.95	65.12 ± 1.54
LEML-CDL-LDA	94.25 ± 3.34	72.63 ± 2.49	72.34 ± 2.07
LEML-CDL-PLS	96.00 ± 2.11	73.31 ± 2.49	N/A

Table 1. Image set classification results of the state-of-the-art methods on ETH-80, YTC and YTF databases.

To fully evaluate our proposed approach, we compare three categories of SPD-based methods and other state-of-the-art image set classification methods which employ other types of set models:

1. Basic metrics on SPD manifold:

   Affine-Invariant Metric (AIM) (Pennec et al., 2006), Stein divergence (Sra, 2012), Log-Euclidean Metric (LEM) (Arsigny et al., 2007)

2. AIM and Stein based supervised methods:

   SPD Manifold Leaning (SPDML-AIM, SPDML-Stein) (Harandi et al., 2014), Riemannian Sparse Representation (RSR) (Harandi et al., 2012)

3. LEM based supervised methods:

   Log-Euclidean Kernel (LEK) (Li et al., 2013), Covariance Discriminative Learning (CDL) (Wang et al., 2012), Information-Theoretic Metric Learning with LEM (ITML-LEM) (Vemulapalli & Jacobs, 2015)

4. Other set model based methods:

   Discriminant Canonical Correlations (DCC) (Kim et al., 2007), Grassmann Discriminant Analysis (GDA) (Hamm & Lee, 2008), Affine (Convex) Hull based Image Set Distance (AHISD, CHISD) (Cevikalp & Triggs, 2010), Set-to-Set Distance Metric Learning (SSDML) (Zhu et al., 2013).

The parameters of the comparative methods are empirically tuned according to their original works: For RSR, we tune the parameter β around the order of the mean distance and λ in the range of $[0.0001, 0.001, 0.01, 0.1]$. For SPDML-AIM/SPDML-Stein, following the original work (Harandi et al., 2014), v_w is fixed as the minimum number of samples in one class while the dimensionality of the target SPD manifold and v_b are tuned by cross-validation. For LEK, we compare its three versions, which are polynomial kernel (LEK-K_p), exponential kernel (LEK-K_e) and radial kernel (LEK-K_g) respectively. The parameter n in LEK-K_p, LEK-K_e is tuned from 1 to 50, and β in LEK-K_g is tuned by using the same strategy as RSR. The parameter λ in LEK-K_p, LEK-K_e and LEK-K_g is tuned in the range of $[0.0001, 0.001, 0.01, 0.1]$. For CDL, we implement both LDA-based and PLS-based versions by adopting the same setting as (Wang et al., 2012). For ITML-LEM, we tune the parameters following the setting of the original ITML (Davis et al., 2007). For AHISD, CHISD and DCC, PCA is performed by preserving 95% energy to learn the linear subspace and the first 10 maximum canonical correlations are used. For GDA, the dimensionality of Grassmann manifold is set to 10. For SSDML, we set $\lambda_1 = 0.001, \lambda_2 = 0.5$, the numbers of positive and negative pairs per set are respectively set to 10 and 20. For our proposed method LEML, the parameter η is tuned in the range of [0.1, 1, 10], ζ is tuned from 0.1 to 0.5. To perform the final classification, we employ sparse representation based classifier for two sparse representation methods (i.e., RSR and LEK) and apply the nearest neighbor classifier on the other methods.

Table 1 shows the average classification accuracy and standard deviation across the 10 random splits of each dataset. For verification task on YTF, since RSR/LEK apply sparse representation based classifier, and CDL-PLS adopts a regression based classifier, they fail to work on this dataset and thus we do not report their results on this dataset. For DCC, GDA, CDL-LDA, we modify them by constructing the within-class scatter matrix from intra-class pairs and the between-class scatter matrix from inter-class pairs.

Firstly, we want to find the improvement of our proposed metric learning approach LEML over the basic metrics on SPD manifold. As can be seen in Table 1, on the three datasets, the Riemannian metric LEM mostly outperforms the other two metrics AIM and Stein, which shows AIM and Stein are both inferior to LEM. Under the LEM framework, our LEM-based metric learning method improves the baseline performances by a large margin.

Secondly, compared with the state-of-the-art methods, our proposed method LEML achieves at least comparable performance with them and even outperform them in several cases on the three datasets. Among these results, we especially care about the comparison between the related works SPDML-AIM/SPDML-Stein and LEML. It can be observed that our method significantly outperforms them, which demonstrates the potential superiority of LEM based metric learning framework over AIM and Stein based ones, due to the inherent advantages of the basic metric LEM, as advocated in (Wang et al., 2012; Jayasumana et al., 2013). Furthermore, we would like to compare the ITML-LEM and LEML, both of which employ the metric learning framework using the same basic metric LEM. The comparisons between them show that ITML-LEM working on vector-form of SPD matrix logarithm is clearly outperformed by our LEML directly working on the original matrix-form. Two reasons can be adduced: 1) As discussed in Sec.3.1, since LEML more faithfully respects the geometry of the original tangent space, it can benefit from its useful properties of the tangent space. 2) As mentioned in Sec.3.2, LEML performs learning on much smaller scatter matrix than ITML-LEM, and thus can learn more desirable distance metric on the SPD manifold.

Lastly, since our method actually derives a new manifold by decomposing the learned PSD matrix $\boldsymbol{Q}$ into a transformation for dimensionality reduction, it can be readily fed into other SPD based methods such as CDL. It is interesting to find that LEML-CDL-LDA and LEML-CDL-PLS favorably (especially on YTF) improve the original versions. This demonstrates that our new metric learning method is qualified to boost the classification performance of other methods by employing it as a pre-processing step.

Methods	Train	Test
SPDML-AIM (Harandi et al., 2014)	15072.56	9.35
SPDML-Stein (Harandi et al., 2014)	108.50	0.04
ITML-LEM (Vemulapalli & Jacobs, 2015)	92007.13	0.02
LEML	**56.30**	**0.02**

Table 2. Computation time (seconds) of the related SPD-based metric/manifold learning methods on YTC for training and testing (classification of one video).

4.3. Time comparison

To show the efficiency of our proposed LEML, we finally compare the training and testing time of the related SPD-based manifold/metric learning methods on YTC using an Intel(R) Core(TM) i7-37700M (3.40GHz) PC. The average training time and testing time for each method is tabulated in Table 2. Compared with ITML-LEM, our method has significantly reduced the training burden, which again justifies the superiority of our LEML working on matrices over ITML-LEM working on vectors in terms of time efficiency. While comparing with other methods, it is also observed that our LEML performs much better than the competing SPD manifold learning techniques SPDML-AIM/SPDML-Stein when training.

5. Conclusion and Future Work

In this paper we have proposed a novel metric learning framework on the SPD manifold for image set classification. Different from most existing methods, our approach seeks to learn the tangent map directly transforming the SPD matrix logarithms (rather than a vector) from the tangent space of the original SPD manifold to a new tangent space. The extensive experiments have shown that our proposed method is competitive to the state-of-the-art approaches. To the best of our knowledge, this is the first attempt to learn discriminant SPD matrix logarithms in the tangent space for the problem of discriminative learning on the SPD manifold.

Essentially, the proposed method aims to learn more discriminant SPD matrix based features by pursuing an appropriate transformation. If ignoring the specific discriminant function designed on SPD matrix, learning the transformation of SPD matrix based feature for image set is equal to learning the projection of basic image feature. Accordingly, this work can be extended to learn hierarchical representations on image feature by coupling the existing deep learning techniques with the objective function designed by our proposed metric learning.

Acknowledgments

This work is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61222211, 61379083, and 61390511.

References

Absil, P-A, Mahony, R., and Sepulchre, R. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

Amari, S. and Nagaoka, H. *Methods of information geometry*. Oxford University Press, 2000.

Arandjelović, O., Shakhnarovich, G., Fisher, J.W., Cipolla, R., and Darrell, T. Face recognition with image sets using manifold density divergence. In *CVPR*, 2005.

Arsigny, V., Fillard, P., Pennec, X., and Ayache., N. Fast and simple computations on tensors with Log-Euclidean metrics. *Res. Rep. RR-5584,INRIA*, 2005.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 2006.

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

Bregman, L.M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3): 200–217, 1967.

Carreira, J., Caseiro, R., Caseiro, J., and Sminchisescu, C. Semantic segmentation with second-order pooling. In *ECCV*, 2012.

Censor, Y. and Zenios, S. *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press, 1997.

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

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *ICML*, 2007.

Golub, G.H., Van, L., and Charles, F. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.

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

Harandi, M. T., Sanderson, C., Shirazi, S., and Lovell, B. C. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *CVPR*, 2011.

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

Harandi, M.T., Salzmann, M., and Hartley, R. From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *ECCV*. 2014.

Hayat, M., Bennamoun, M., and An, S. Reverse training: An efficient approach for image set classification. In *ECCV*, 2014.

Hu, Y., Mian, A.S., and Owens, R. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

Huang, Z., Wang, R., Shan, S., and Chen, X. Learning Euclidean-to-Riemannian metric for point-to-set classification. In *CVPR*, 2014.

Huang, Z., Shan, S., Wang, R., Zhang, H., Lao, S., Kuerban, A., and Chen, X. A benchmark and comparative study of video-based face recognition on COX face database. *IEEE T-IP*, 2015a.

Huang, Z., Wang, R., Shan, S., and Chen, X. Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning. *Pattern Recognition*, 2015b.

Huang, Z., Wang, R., Shan, S., and Chen, X. Projection metric learning on Grassmann manifold with application to video based face recognition. In *CVPR*, 2015c.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M.T. Kernel methods on the Riemannian manifold of symmetric positive definite matrices. In *CVPR*, 2013.

Jung, S., Dryden, I.L., and Marron, J.S. Analysis of principal nested spheres. *Biometrika*, 99(3):551–568, 2012.

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

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

Kulis, B., Sustik, M.A., and Dhillon, I. S. Low-rank kernel learning with bregman matrix divergences. *JMLR*, 10: 341–376, 2009.

Leibe, B. and Schiele, B. Analyzing appearance and contour based methods for object categorization. In *CVPR*, 2003.

Li, P., Wang, Q., Zuo, W., and Zhang, L. Log-Euclidean kernels for sparse representation and dictionary learning. In *ICCV*, 2013.

Lovrić, M., Min-Oo, M., and Ruh, E.A. Multivariate normal distributions parametrized as a Riemannian symmetric space. *Journal of Multivariate Analysis*, 74(1):36–48, 2000.

Lu, J., Wang, G., and Moulin, P. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

Lu, J., Wang, G., Deng, W., and Moulin, P. Simultaneous feature and dictionary learning for image set based face recognition. In *ECCV*, 2014.

Mahmood, A., Mian, A., and Owens, R. Semi-supervised spectral clustering for image set classification. In *CVPR*, 2014.

Minh, H.Q., Biagio, M.S., and Murino, V. Log-hilbert-schmidt metric between positive definite operators on Hilbert spaces. In *NIPS*, 2014.

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

Shakhnarovich, G., Fisher, J.W., and Darrell, T. Face recognition from long-term observations. In *ECCV*, 2002.

Sherman, J. and Morrison, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, pp. 124–127, 1950.

Sivalingam, R., Morellas, V., Boley, D., and Papanikolopoulos, N. Metric learning for semi-supervised clustering of region covariance descriptors. In *ICDSC*, 2009.

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

Tosato, D., Farenzena, M., Cristani, M., Spera, M., and Murino, V. Multi-class classification on Riemannian manifolds for video surveillance. In *ECCV*, 2010.

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

Vemulapalli, R., Pillai, J.K., and Chellappa, R. Kernel learning for extrinsic classification of manifold features. In *CVPR*, 2013.

Wang, R. and Chen, X. Manifold discriminant analysis. In *CVPR*, 2009.

Wang, R., Shan, S., Chen, X., and Gao, W. Manifold-Manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

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

Wang, W., Wang, R., Huang, Z., Shan, S., and Chen, X. Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets. In *CVPR*, 2015.

Wolf, L., Hassner, T., and Maoz, I. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

Yang, J., Zhang, D., Frangi, A.F., and Yang, J. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE T-PAMI*, 26 (1):131–137, 2004.

Yang, M., Zhu, P., VanGool, L., and Zhang, L. Face recognition based on regularized nearest points between image sets. In *FG*, 2013.

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