Log-Euclidean Kernels for Sparse Representation and Dictionary Learning

Peihua Li[1], Qilong Wang[2], Wangmeng Zuo[3,4], Lei Zhang[4]
[1]Dalian University of Technology, [2]Heilongjiang University, [3]Harbin Institute of Technology
[4]The Hong Kong Polytechnic University

peihuali@dlut.edu.cn, wangqilong.415@163.com, cswmzuo@gmail.com, cslzhang@comp.polyu.edu.hk

Abstract

The symmetric positive definite (SPD) matrices have been widely used in image and vision problems. Recently there are growing interests in studying sparse representation (SR) of SPD matrices, motivated by the great success of SR for vector data. Though the space of SPD matrices is well-known to form a Lie group that is a Riemannian manifold, existing work fails to take full advantage of its geometric structure. This paper attempts to tackle this problem by proposing a kernel based method for SR and dictionary learning (DL) of SPD matrices. We disclose that the space of SPD matrices, with the operations of logarithmic multiplication and scalar logarithmic multiplication defined in the Log-Euclidean framework, is a complete inner product space. We can thus develop a broad family of kernels that satisfies Mercer's condition. These kernels characterize the geodesic distance and can be computed efficiently. We also consider the geometric structure in the DL process by updating atom matrices in the Riemannian space instead of in the Euclidean space. The proposed method is evaluated with various vision problems and shows notable performance gains over state-of-the-arts.

1. Introduction

The symmetric positive definite (SPD) matrices can be introduced in the imaging, pre-processing, feature extraction and representation processes, and have been widely adopted in many computer vision applications [4, 29, 27]. For example, each voxel of the tensor-valued images produced by Diffusion Tensor Imaging (DTI) [4] is a three-dimensional SPD matrix, and the DTI images are very promising in neuroscience study since they capture the tissue microstrucutral characteristics in an non-invasive way through diffusion of water molecules. In pre-processing, the structure tensor [3] obtained by computing the second (or higher) order moments of image features in a neighboring region can be represented by SPD matrices; it has been used for orientation estimation and local structure analysis, finding applications in optical flow, texture, corner and edge detection [29], and recently in foreground segmentation [6]. As to feature extraction, the covariance matrices that model the second-order statistics of image features also result in SPD matrices, which have been successfully applied to detection [27], recognition [18], and classification [5], etc. Due to their wide applications, the investigations on learning methods for SPD matrices have recently received considerable research interests.

One key problem of SPD matrix-based learning methods is the model and computation of SPD matrices. It is known that the space of $n \times n$ SPD matrices, denoted by $\mathcal{S}_n^+$, is not a linear space [1] but forms a Lie group that is a Riemannian manifold [1]. Hence, the mathematical modeling in this space is different from what is commonly done in the Euclidean space [6, 31], and new operators for SPD matrices should be introduced for SPD matrix-based learning. In this work, we take sparse representation (SR) and dictionary learning (DL) [8] as examples, and focus on extending SR and DL to SPD matrix-based learning. Since conventional SR and DL methods are proposed for vector data in the Euclidean space rather than the SPD matrices in the Riemannian manifold, in order to use SR and DL for SPD matrix-based learning, we should consider the following issues in developing new operators in $\mathcal{S}_n^+$. (1) In the Euclidean space the linear combination of atom vectors can be naturally obtained using the conventional matrix operators; but it would be challenging to represent an SPD matrix as a linear combination of atom matrices since $\mathcal{S}_n^+$ is not a linear space. (2) To evaluate the reconstruction error, ℓ_2-norm is commonly used in the Euclidean space; however, in $\mathcal{S}_n^+$ the Riemannian metrics would be more appropriate as they can measure the intrinsic distance between SPD matrices. (3) The updating of dictionary atoms involves solving a constrained optimization problem in $\mathcal{S}_n^+$ and it is more appro-

[1] $\mathcal{S}_n^+$ is not a linear space with the operations of conventional matrix addition and scalar-matrix multiplication. However, with the operations of logarithmic multiplication and scalar logarithmic multiplication $\mathcal{S}_n^+$ is not only a linear space [1] but also a complete inner product space as shown in section 2.2.

Table 1. Comparison of various methods on sparse representation and dictionary learning in $\mathcal{S}_n^+$

Method	Representation given atoms	Riemannian Metric?	Riemannian atom update?	Mercer's condition?
TSC [22, 23]	Linear in Euclidean space	No-LogDet divergence	No-Euclidean	N/A
GDL[25]	Linear in Euclidean space	No-Frobenius norm	No-Euclidean	N/A
LogE-SR [11, 34]	Linear in Log-domain	Yes	No-Euclidean	N/A
RSR [12]	Linear in RKHS	Approximation-Stein divergence	No-Euclidean	Satisfy-conditionally
Proposed method	Linear in RKHS	Yes	Yes-Riemannian	Satisfy

priate to consider the geometry of $\mathcal{S}_n^+$. It has been shown [19, 1] that the methods failing to consider the geometric structure of $\mathcal{S}_n^+$ may result in unsatisfactory performance or even break down.

Generally, there are two strategies to address the three issues mentioned above. First, one can extend SR and DL by introducing proper linear decomposition and reconstruction error measures for SPD matrices. In Tensor Spare Coding (TSC) [22], an SPD matrix is linearly decomposed as a set of of atom matrices and LogDet (or Bregman matrix) divergence [15] was adopted to measure the reconstruction error. Based on this framework, dictionary learning methods are further proposed to learn atom matrices [23]. In the generalized dictionary learning (GDL) algorithm [25], each SPD matrix is represented as a linear combination of rank-1 atom matrices; the error between one SPD matrix and its linear combination is evaluated by matrix Frobenius norm. The GDL algorithm is fast and scalable to large datasets.

Second, we can explicitly or implicitly map SPD matrices to some Reproducing Kernel Hilbert Space (RKHS), and use the kernel SR or DL framework for SPD matrix-based learning. Harandi *et al*. [12] first studied kernel-based method for SR and DL in $\mathcal{S}_n^+$. They adopted Stein kernel to map the SPD matrices to higher dimensional RKHS. This method is in contrast with those methods which directly embed SPD matrices into Euclidean space (LogE-SR) [11, 34] and achieves state-of-the-art performance compared to its counterparts.

Based on the previous analysis, the pros and cons of various methods are summarized in Table 1. We argue that the previous work fails to take full advantage of the geometry of $\mathcal{S}_n^+$. Linear decomposition of SPD matrices as in the Euclidean space [22, 23, 25] is not natural and may induce errors as $\mathcal{S}_n^+$ is not a linear space; the method of embedding SPD matrices into Euclidean space by matrix logarithm [11, 34] has improved performance but the gains are limited [12]. Furthermore, these work uses directly either the Euclidean norm [25] or the Bregman matrix divergence [22, 23] to evaluate the reconstruction error. The linear decomposition makes sense in high- or infinite-dimensional RKHS [12]; however, the Stein divergence is only an approximation of Riemannian metric and is a positive definite (p.d.) kernel [2] only under some restricted conditions [24].

Our work is inspired by [12]. We also embed $\mathcal{S}_n^+$ into RKHS as linear representation is a natural and reasonable consequence in the Hilbert space. The main difference is that we develop a novel family of kernel functions based on the Log-Euclidean framework [1]. The proposed kernels characterize the geodesic distance and thus can accurately measure the reconstruction error; they also satisfy the Mercer's condition under broad conditions. These are in contrast to the Stein kernel which is only an approximation of the geodesic distance and satisfies the Mercer's condition only under some restricted conditions. In addition, we explicitly consider in DL the geometric structure of $\mathcal{S}_n^+$. Note that the Gaussian kernel based on the Log-Euclidean metric is simultaneously presented in [14, 28]. Our work differs from them in that we disclose the inner product structure of $\mathcal{S}_n^+$, by which we can develop a broad variety of kernel functions and the Gaussian kernel is a special case of ours.

2. Log-Euclidean Kernel

This section starts with a brief introduction of Log-Euclidean framework [1]; subsequently, we show that $\mathcal{S}_n^+$ forms an inner product space; based on this, we design a family of kernel functions.

2.1. $\mathcal{S}_n^+$ as a Lie Group and Geodesic Distance

Let $\mathcal{S}_n$ be the space of $n \times n$ symmetric matrices. The matrix exponential exp: $\mathcal{S}_n \mapsto \mathcal{S}_n^+$ is bijective and smooth and its inverse map, denoted by log, is smooth as well. In the Log-Euclidean framework, an operation of logarithmic multiplication $\odot$: $\mathcal{S}_n^+ \times \mathcal{S}_n^+ \mapsto \mathcal{S}_n^+$ is defined as [1]

$$\mathbf{S}_1 \odot \mathbf{S}_2 = \exp(\log(\mathbf{S}_1) + \log(\mathbf{S}_2)) \qquad (1)$$

It can be verified that $\mathcal{S}_n^+$ is a group with the identity element being the identity matrix and with the inverse operation the regular matrix inverse. $\mathcal{S}_n^+$ is a Lie group because the group operation and inverse operation are both smooth and because it is a Riemannian manifold (half convex cone in Euclidean space). Note that $\odot$ is commutative and $\mathcal{S}_n^+$ is therefore a commutative Lie group (Abelian group).

The commutative Lie group $\mathcal{S}_n^+$ admits a bi-invariant metric. The geodesics equipped with a bi-invariant metric are the left translates of the geodesics through the identity element, given by one-parameter subgroup $\exp(t\mathbf{V})$, where $t \in \mathbb{R}$ and $\mathbf{V} \in \mathcal{S}_n$. After some derivations and manipulations, one can obtain the geodesics, the Riemannian exponential and Riemannian logarithm, and finally the geodesic distance between two SPD matrices $\mathbf{S}$ and $\mathbf{T}$ as follows:

$$\rho_{\text{geo}}(\mathbf{S},\mathbf{T}) = \|\log(\mathbf{S}) - \log(\mathbf{T})\|_F \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Interested readers may refer to [1] for details on the corresponding theory.

2.2. $\mathcal{S}_n^+$ as a Complete Inner Product Space

It is known that $\mathcal{S}_n^+$ is not a linear space with the operations of the conventional matrix addition and scalar-matrix multiplication but forms a Riemannian manifold [1, 31]. However, as shown in [1], in the Log-Euclidean framework it is endowed with a linear space structure with the logarithmic multiplication (1) and the following scalar logarithmic multiplication [1]:

$$\lambda \otimes \mathbf{S} = \exp(\lambda \log(\mathbf{S})) = \mathbf{S}^{\lambda} \tag{3}$$

where λ is a real number. It is straightforward to show that two operations $\odot$ and $\otimes$ satisfy the conditions of a linear space, with the identity matrix being the identity element and regular matrix inversion operation as inverse mapping.

Indeed, not only a linear space, $\mathcal{S}_n^+$ is also an inner product space as described by the following corollary which is not disclosed previously:

Corollary 1 *With two operations $\odot$ and $\otimes$, the function from the product space of $\mathcal{S}_n^+$ to the space $\mathbb{R}$ of real number*

$$\begin{aligned} \langle\cdot,\cdot\rangle_{\log} : \mathcal{S}_n^+ \times \mathcal{S}_n^+ &\mapsto \mathbb{R} \\ \langle\mathbf{S},\mathbf{T}\rangle_{\log} &= \operatorname{tr}(\log(\mathbf{S})\log(\mathbf{T})) \end{aligned} \tag{4}$$

is an inner product, where tr *denotes the matrix trace, and $\mathcal{S}_n^+$ is a complete inner product space (Hilbert space). The induced norm can be used to define the distance that equals to the geodesic distance.*

Let $\mathbf{S},\mathbf{T},\mathbf{R} \in \mathcal{S}_n^+, \lambda \in \mathbb{R}$ be arbitrary, below we verify the axioms of inner product in terms of $\odot$ and $\otimes$:

Symmetry $\langle\mathbf{S},\mathbf{T}\rangle_{\log} = \operatorname{tr}(\log(\mathbf{S})\log(\mathbf{T})) = \operatorname{tr}(\log(\mathbf{T})\log(\mathbf{S})) = \langle\mathbf{T},\mathbf{S}\rangle_{\log}$. Here we use the property that the trace of a matrix equals to that of its transpose.

Linearity $\langle\mathbf{S} \odot \mathbf{R},\mathbf{T}\rangle_{\log} = \operatorname{tr}((\log(\mathbf{S}) + \log(\mathbf{R}))\log(\mathbf{T})) = \langle\mathbf{S},\mathbf{T}\rangle_{\log} + \langle\mathbf{R},\mathbf{T}\rangle_{\log}$, $\langle\lambda \otimes \mathbf{S},\mathbf{T}\rangle_{\log} = \operatorname{tr}((\lambda\log(\mathbf{S}))\log(\mathbf{T})) = \lambda\langle\mathbf{S},\mathbf{T}\rangle_{\log}$.

Non-negativity $\langle\mathbf{S},\mathbf{S}\rangle_{\log} = \operatorname{tr}(\log(\mathbf{S})\log(\mathbf{S})) = \|\log(\mathbf{S})\|_F^2 \geq 0$, and it is obvious that $\langle\mathbf{S},\mathbf{S}\rangle_{\log} = 0$ if and only if $\mathbf{S}$ is the identity matrix $\mathbf{I}$.

Obviously $\mathcal{S}_n^+$ is of finite dimension and therefore it is a complete inner product space (Hilbert space) [26]. The norm induced by the inner product is expressed as $\|\mathbf{S}\|_{\log} = \langle\mathbf{S},\mathbf{S}\rangle_{\log}^{\frac{1}{2}}$, and hence, $\mathcal{S}_n^+$ is a normed linear space. The distance (metric) between $\mathbf{S}$ and $\mathbf{T}$ is

$$\begin{aligned} \rho_{\log}(\mathbf{S},\mathbf{T}) &= \|\mathbf{S}\odot\mathbf{T}^{-1}\|_{\log} = \langle\mathbf{S}\odot\mathbf{T}^{-1},\mathbf{S}\odot\mathbf{T}^{-1}\rangle_{\log}^{\frac{1}{2}} \\ &= \|\log(\mathbf{S}) - \log(\mathbf{T})\|_F = \rho_{\text{geo}}(\mathbf{S},\mathbf{T}) \end{aligned}$$

where $\mathbf{T}^{-1}$ denotes the inverse of matrix $\mathbf{T}$, which equals to the geodesic distance when $\mathcal{S}_n^+$ is viewed as a Lie group.

Arsigny et al. [1] established the linear space structure of $\mathcal{S}_n^+$ and pointed out that a similarity invariant metric can be defined [1, Proposition 3.11]. However, a linear space or a normed liner space is not necessarily an inner product space unless a function that satisfies the axioms of the inner product is defined. In the above, we have shown that $\mathcal{S}_n^+$ is a complete inner product space (Hilbert space). We argue that Corollary 1 might change our philosophy of data processing on $\mathcal{S}_n^+$: since it is a Hilbert space, we can perform data processing directly on $\mathcal{S}_n^+$ with logarithmic multiplication $\odot$ and scalar logarithmic multiplication $\otimes$, unlike [1] which involves mapping SPD matrices to logarithmic domain, performing data processing therein and then mapping back to $\mathcal{S}_n^+$ again.

We can define other inner products, for example $\langle\mathbf{S},\mathbf{T}\rangle_{\log,\mathbf{A}}$ as described by the following corollary.

Corollary 2 *Let $\mathbf{A} \in \mathcal{S}_n^+$ be arbitrary, with two operations $\odot$ and $\otimes$, the function*

$$\langle\mathbf{S},\mathbf{T}\rangle_{\log,\mathbf{A}} = \operatorname{tr}(\log(\mathbf{S})\mathbf{A}\log(\mathbf{T})) \tag{5}$$

is an inner product, and the induced norm is $\|\mathbf{S}\|_{\log,\mathbf{A}} = \langle\mathbf{S},\mathbf{S}\rangle_{\log,\mathbf{A}}^{\frac{1}{2}}$.

It reduces to (4) if $\mathbf{A}$ is the identity matrix. In practice we can learn $\mathbf{A}$ from training data on $\mathcal{S}_n^+$ by designing distance metric learning methods, and the learned distance may be more descriminative and suitable for specific vision tasks.

2.3. Log-Euclidean Kernels

Based on Corollary 2, $\langle\cdot,\cdot\rangle_{\log,\mathbf{A}}$ is a p.d. kernel as $\sum_{i,j} c_i c_j \langle\mathbf{S}_i,\mathbf{S}_j\rangle_{\log,\mathbf{A}} = \langle\sum_{\odot,i} c_i \otimes \mathbf{S}_i, \sum_{\odot,i} c_i \otimes \mathbf{S}_i\rangle_{\log,\mathbf{A}} \geq 0$. In a similar way, we can show that its normalized version $\langle\mathbf{S},\mathbf{T}\rangle_{\log,\mathbf{A}}/(\|\mathbf{S}\|_{\log,\mathbf{A}}\|\mathbf{S}\|_{\log,\mathbf{A}})$ is also a kernel. Here $\sum_{\odot,i}$ denotes the logarithmic multiplication of terms indexed by i. The following statements are based on [2, pp. 69∼70] (see proof therein):

Proposition 1 *Let $\mathbb{X}$ be a non-empty set, and $\phi_1,\phi_2 : \mathbb{X}\times\mathbb{X} \mapsto \mathbb{R}$ be arbitrary p.d. kernels. We have: (1) The pointwise product $\phi_1\phi_2 : \mathbb{X}\times\mathbb{X} \mapsto \mathbb{R}$ is a p.d. kernel; (2) The tensor product $\phi_1 * \phi_2 : (\mathbb{X}\times\mathbb{X}) \times (\mathbb{X}\times\mathbb{X})$ is a p.d. kernel; and (3) If $f(z) = \sum_{n=0}^{\infty} a_n z^n$ is holomorphic (analytic) in its domain and $a_n \geq 0$ for all $n \geq 0$, the composed function $f \circ \phi$ is a p.d. kernel.*

We can develop a broad variety of p.d. kernels, such as polynomial, exponential, radial basis, B-Spline kernels, or Fourier kernel etc. [13]. Below we give some commonly used kernels.

Corollary 3 *Let* $\mathbf{A} \in \mathcal{S}_n^+$ *and* p_n *be a polynomial of degree* $n \geq 1$ *with positive coefficients, we have p.d. kernels*

Log-E poly. kernel $\quad \kappa_{p_n}(\mathbf{S},\mathbf{T}) = p_n\left(\langle \mathbf{S},\mathbf{T}\rangle_{\log}\right),$
Log-E exp. kernel $\quad \kappa_{e_n}(\mathbf{S},\mathbf{T}) = \exp\left(p_n(\langle \mathbf{S},\mathbf{T}\rangle_{\log})\right),$
Log-E Gaus. kernel $\quad \kappa_g(\mathbf{S},\mathbf{T}) = \exp(-\|\mathbf{S} \odot \mathbf{T}^{-1}\|^2_{\log,\mathbf{A}})$

Through Corollaries 1, 2 and Proposition 1, we can easily see that $\kappa_{p_n}(\mathbf{S},\mathbf{T})$ is a p.d. kernel. From the series expansion of exp, we know $\kappa_{e_n}(\mathbf{S},\mathbf{T})$ is a p.d. kernel as well. Since $\exp(-\|\mathbf{S}\|_{\log,\mathbf{A}})\exp(-\|\mathbf{T}\|_{\log,\mathbf{A}})$ is a p.d. kernel which can be proved straightforwardly by the definition, $\exp(-\|\mathbf{S}\|_{\log,\mathbf{A}})\exp(-\|\mathbf{T}\|_{\log,\mathbf{A}})\exp(2\langle \mathbf{S},\mathbf{T}\rangle_{\log,\mathbf{A}}) = \kappa_g(\mathbf{S},\mathbf{T})$ (after some manipuliation) is also a kernel. $\kappa_g(\mathbf{S},\mathbf{T})$ is an anisotropic Gaussian kernel; if $\mathbf{A}$ is a diagonal matrix $\mathbf{A} = \mathrm{diag}\{\beta\}$ with $\beta > 0$, $\kappa_g(\mathbf{S},\mathbf{T})$ reduces to a special form $\exp(-\beta\|\log(\mathbf{S}) - \log(\mathbf{T})\|_F^2)$, which is identical to the Gaussian kernel in [14, 28].

Here we compare the proposed kernels with Stein kernel [24]. Let $\mathbf{S},\mathbf{T} \in \mathcal{S}_n^+$ and $\boldsymbol{\gamma} = [\log(\lambda_1) \ \ldots \ \log(\lambda_n)]^T$, where λ_i are the generalized eigenvalues between $\mathbf{S}$ and $\mathbf{T}$. The Affine-Riemannian distance between the two matrices is $d_A(\mathbf{S},\mathbf{T}) = \|\boldsymbol{\gamma}\|_2$ [19]. The symmetric Stein divergence $d_S(\mathbf{S},\mathbf{T}) = \log\left(\det\left(\frac{\mathbf{S}+\mathbf{T}}{2}\right)\right) - \frac{1}{2}\log(\det(\mathbf{S}\mathbf{T}))$, derived from Bregman matrix divergence [15], satisfies the the sandwiching inequality [24] $\frac{d_A^2(\mathbf{S},\mathbf{T})}{2d_T(\mathbf{S},\mathbf{T})} - n\log(2) \leq d_S(\mathbf{S},\mathbf{T}) \leq \frac{1}{8}d_A^2(\mathbf{S},\mathbf{T})$, where $d_T(\mathbf{S},\mathbf{T}) = \|\boldsymbol{\gamma}\|_\infty$. From our experience, most of the values of the leftmost hand side term in the sandwiching inequality are negatives. Fig. 1 shows the histogram computed from the SPD matrices used in texture classification (described in Section 4.1). Hence, $d_S(\mathbf{S},\mathbf{T})$ might be only upper-bounded by the geodesic distance $\frac{1}{8}d_A^2(\mathbf{S},\mathbf{T})$. It is unclear that to what extent the Stein divergence approximates the Riemannian metric. Above all, $\kappa_S(\mathbf{S},\mathbf{T}) = \exp(-\beta d_S(\mathbf{S},\mathbf{T}))$ is a p.d. kernel under restricted condition, that is, $\beta = \frac{1}{2},\ldots,\frac{n-2}{2}$ or $\beta \geq \frac{n-1}{2}$[24]. In contrast, the proposed family of kernels κ_{p_n}, κ_{e_n} and κ_g characterize the true rather than the approximation of Riemannian metric. So they can evaluate the reconstruction error accurately. In addition, $\kappa_{p_n}, \kappa_{e_n}$ are kernels for any order of polynomials with positive coefficients and κ_g is a kernel for any $\beta > 0$. This produces flexibility for adjusting the parameters to obtain better performance for various problems.

The logarithm of SPD matrices can be computed through the eigen-decomposition. Let $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ be the eigen-decomposition of $\mathbf{S} \in \mathcal{S}_n^+$, where $\boldsymbol{\Lambda}$ is a diagonal matrix consisting of eigen-values $\lambda_i, i = 1,\ldots,n$, of $\mathbf{S}$, i.e., $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_i\}$, and $\mathbf{U}$ is an orthonormal matrix consisting of the corresponding eigen-vectors. As we have $\log(\mathbf{S}) = \mathbf{U}\mathrm{diag}\{\log(\lambda_i)\}\mathbf{U}^T$, the computational complexity of the proposed kernels is $O(10n^3)$ which is higher than that of the Stein kernel. The logarithms of the involved SPD matrices can generally be computed beforehand because of their "decoupling" property either in the inner product or distance; in these cases, the complexity of the proposed kernels becomes $O(n^3)$ and is the same as that of the Stein kernel.

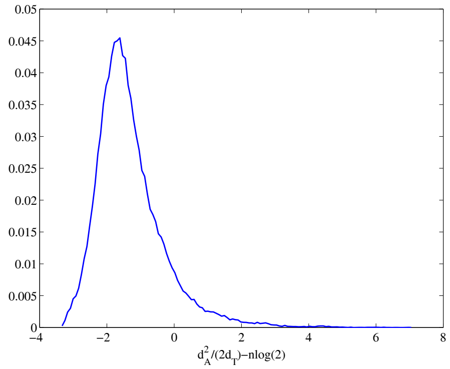

Figure 1. Histogram of the values of $d_A^2(\mathbf{S},\mathbf{T})/(2d_T(\mathbf{S},\mathbf{T})) - n\log(2)$

3. Sparse Representation and Dictionary Learning

Kernel-based SR and DL have been studied in the literature [10] for vector data in the Euclidean space $\mathbb{R}^n$, which have shown notable performance improvement over the non-kernel based methods. Harandi *et al.* [12] first presented kernel-based SR and DL for SPD matrices. While this method outperforms state-of-the-arts, the symmetric Stein divergence only approximates the Riemannian metric and the Stein kernel only satisfies Mercer's condition under restricted conditions. In this section, we develop SR and DL methods based on the Log-Euclidean Kernels, which address the shortcomings of the Stein kernel.

3.1. Sparse Representation

Let $\mathbf{Y} \in \mathcal{S}_n^+$ and $\mathbf{S}_i \in \mathcal{S}_n^+, i = 1,\ldots,N$ be a set of atom matrices. Let ϕ be the function that maps SPD matrices to RKHS, SR of $\mathbf{Y}$ can be formulated as the following kernelised LASSO problem [12]:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \left\|\phi(\mathbf{Y}) - \sum_{i=1}^{N} x_i\phi(\mathbf{S}_i)\right\|_2^2 + \lambda\left\|\mathbf{x}\right\|_1 \qquad (6)$$

subject to $\|\phi(\mathbf{S}_i)\|_2 \leqslant 1, \forall i$, where $\mathbf{x} = [x_1 \ \ldots \ x_N]^T$ is the sparse vector, $\lambda > 0$ is the regularization parameter, and $\|\cdot\|_2$ and $\|\cdot\|_1$ denote ℓ_2-norm and ℓ_1-norm, respectively. In the kernel methods since $\|\phi(\mathbf{S}_i)\|_2 = 1$ the constraints are satisfied naturally and can thus be neglected. After some manipulations, the SR (6) can be expressed in the form of kernels as

$$\min_{\mathbf{x}\in\mathbb{R}^N} -2\sum_{i=1}^{N} x_i\kappa(\mathbf{Y},\mathbf{S}_i)+\sum_{i=1}^{N}\sum_{i'=1}^{N} x_i x_{i'}\kappa(\mathbf{S}_i,\mathbf{S}_{i'})+\lambda\|\mathbf{x}\|_1$$

Minimaization of the above equation is similar to regular sparse coding in Euclidean space [10], and we use the method introduced in [12] for its solution.

3.2. Dictionary Learning

Given a set of training data $\mathbf{Y}_j, j = 1,\ldots,M$, the atom matrices can be obtained by learning method so that they have more powerful representation capability. The learning problem may be expressed as minimization of the function

$$f(\mathbf{S}_1,\ldots,\mathbf{S}_N,\mathbf{x}_j,\ldots,\mathbf{x}_M) = \sum_{j=1}^{M}\Big\|\phi(\mathbf{Y}_j)-\sum_{i=1}^{N} x_{j,i}\phi(\mathbf{S}_i)\Big\|_2^2+\lambda\big\|\mathbf{x}_j\big\|_1 \quad (7)$$

w.r.t $\mathbf{S}_i, i = 1,\ldots,N$, and $\mathbf{x}_j, j = 1,\ldots,M$, where $\mathbf{x}_j = [x_{j,1},\ldots,x_{j,N}]$ denotes the sparse vector of $\mathbf{Y}_j$. The problem (7) is commonly solved by iterating two procedures [10, 12]. First, suppose that the atom matrices $\mathbf{S}_i \in \mathcal{S}_n^+, i = 1,\ldots,N$, are fixed, the problem (7) reduces to kernel-based SR problems: for each $\mathbf{Y}_j, j = 1,\ldots,M$, we compute its sparse vector $\mathbf{x}_j$ as described in the previous section; then, let $\mathbf{x}_j$ be fixed, we update dictionary atom matrices $\mathbf{S}_i, i = 1,\ldots,N$.

In the following, we illustrate the atom matrices update scheme using Gaussian kernel κ_g. As in [12], we also adopt a method that is similar to K-SVD [8, Chap. 12] and update an atom matrix at one time. Re-writing (7) in kernel function κ, we have the partial derivative of $f(\cdot)$ w.r.t $\mathbf{S}_r$

$$\frac{\partial f}{\partial \mathbf{S}_r} = -2\beta\mathbf{S}_r^{-1}\Big(\sum_{j=1}^{M} x_{j,r}\kappa(\mathbf{S}_r,\mathbf{Y}_j)(\log(\mathbf{S}_r)-\log(\mathbf{Y}_j)) - \sum_{j=1}^{M}\sum_{i=1}^{N} x_{j,r}x_{j,i}(\log(\mathbf{S}_r)-\log(\mathbf{S}_i))\Big) \quad (8)$$

One may update $\log\mathbf{S}_r$ instead of $\mathbf{S}_r$, which is equivalent to transforming by logarithm the SPD matrices to Euclidean space in which atoms are updated. However, in practice we find this update scheme is unstable. We thus instead update the atom matrices in the Lie group as follows:

$$\mathbf{S}_r = \exp\big(\log(\mathbf{S}_r) + d_{\mathbf{S}_r}\log(-\epsilon\partial f/\partial\mathbf{S}_r)\big) \quad (9)$$

where $d_{\mathbf{S}_r}(\mathbf{U})$ denotes the differential of matrix logarithm at $\mathbf{S}_r$ with the displacement of the tangent matrix $\mathbf{U}$. Hence, the marching now is along the geodesics and the algorithm becomes more stable.

4. Experiments

In this section, we first evaluate the performance of the proposed family of kernels on sparse representation without dictionary learning. As in [30], the training samples are adopted as atom matrices and the reconstruction errors are used for classification. Then we learn the atom matrices from the training data and the sparse codes obtained from the learned atom matrices are used for classification with the nearest neighbor classifier or support machine vector (SVM).

4.1. Sparse Representation

In the papers that focus on SR in $\mathcal{S}_n^+$, the FERET dataset [20] and the Brodatz database [21] are commonly used for classification performance evaluation [22, 23, 12]. Hence, to facilitate comparison with state-of-the-arts we also adopt them here.

Face Recognition As in [33, 22, 23, 12], we select the “b” subset of FERET database for evaluation of classification performance. The subset consists of 198 subjects, each of which has 7 images. The training examples are composed of frontal face images with neutral expression “b”, smiling expression “bj”, and illumination changes “bk”, while the test examples involve face images of varying pose angle: “bd”–$+25^\circ$, “be”–$+15^\circ$, “bf”–-15°, and “bg” –-25°. As in [18], the image features to compute covariance descriptors consist of intensity value, x and y coordinates, and intensity values of the filtered image via Gabor filters along 5 orientations and 8 angles. Thus each image is represented by a 43×43 covariance matrix. We adopt the classification method in [30], and the preprocessing method in [12].

Fig. 2 shows the recognition accuracy of the proposed kernels with the regularization parameter $\lambda = 10^{-3}$, where the recognition rates of RSR using Stein kernel [12] are also shown as baseline (red dash-dotted). The top row shows the classification rates of $\kappa_{p_n}, \kappa_{e_n}(p_n(x) = x^n)$ versus n. Note that the recognition rates of κ_{e_n} are less sensitive to the polynomial order than κ_{p_n}. The bottom row shows those of κ_g versus β, from which we see that the recognition rate increases as β gets larger, reaching peak at about 2×10^{-2} and decreasing afterwards. It can be seen that the proposed kernels are clearly better than the Stein kernel on all datasets. In particular, for two difficult datasets that have large pose variations, the performance gains are substantial.

Table 2 lists the comparison results on FERET dataset with sate-of-the-arts: Sparse Representation Classification (SRC) [30], the Gabor feature-based sparse representation in Euclidean space (GSRC)[33], Log-Euclidean sparse representation (logE-SR) which performs sparse representation in the logarithm domain [11, 34], Tensor Sparse Coding (TSC) [22], Riemannian Sparse Representation (RSR) based on Stein kernel [12]. The results are reproduced from

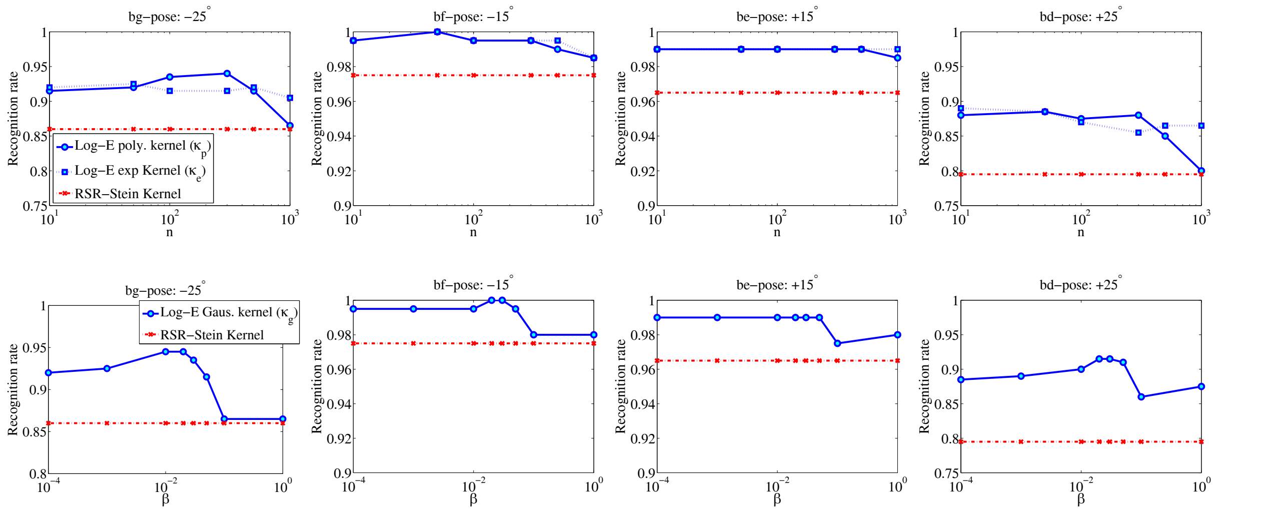

Figure 2. Classification rates on the FERET dataset. Top row: $\kappa_{p_n}, \kappa_{e_n}$ vs. n; bottom row: κ_g vs. β. From left to right are results on bg, bf, be, and bd, respectively. The classification rates of RSR that uses the Stein kernel [12] are shown as baseline (red dash-dotted line).

the respective papers. TSC has unsatisfactory performance and we owe it to the linear representation of SPD matrices in the Euclidean space without use of the Riemannian metric. By using the Riemannian metric, LogE-SR has improved recognition rates but the sparse decomposition is performed in the logarithm domain rather than in the original Riemannian manifold. The kernel-based method, RSR, compared with TSR and LogE-SR, achieves larger performance gains. Two reasons may account for this: (1) linear representation in RKHS naturally makes sense; and (2) the Stein divergence is an approximation to the Riemannian metric. The results of the proposed methods are obtained with $\kappa_{p_n}, \kappa_{e_n}$ ($p_n(x) = x^{50}$), and κ_g ($\beta = 2 \times 10^{-2}$). The proposed three kernels have comparable performance while κ_g is a little better. We see that our methods achieve largely notable performance increase and we attribute this to full use of data geometry.

Table 2. Comparison with state-of-the-arts on the FERET database

	SRC [30]	GSRC [33]	LogE-SR [11, 34]	TSC [22]	RSR [12]	Log-E kernel		
						κ_{p_n}	κ_{e_n}	κ_g
bg	26.0	79.0	46.5	44.5	86.0	92.0	91.5	**94.5**
bf	61.0	97.0	91.0	73.5	97.5	100	99.5	**100**
be	55.5	93.5	81.0	73.0	96.5	99.0	99.0	**99.0**
bd	27.5	77.0	34.5	36.0	79.5	88.5	88.0	**91.5**
ave.	42.5	86.6	63.3	56.8	89.9	94.9	94.5	**96.3**

Texture Classification We employ the Brodatz dataset and follow the experimental setting in [22, 23, 12] for fair comparison. Note that our purpose here is not to develop competing texture classification algorithm [17] but to testify the proposed method with closely related work. In the Brodatz dataset each class contains only one image and we use the mosaics of 5-texture (‘5c’, ‘5m’, ‘5v’, ‘5v2’, ‘5v3’), 10-texture (‘10’,‘10v’), and 16-texture (‘16c’, ‘16v’). Every image is resized to 256×256 which is then uniformly divided into 8×8 subimages. Each subimage is represented by a covariance matrix computed from the feature vectors of intensity and the absolute values of the 1st- and 2nd-order partial derivatives with respect to spatial coordinates. Among the 64 covariance matrices per class, 5 are randomly selected for training and the remaining ones are for testing. The final classification rate is averaged over 20 trials. Figure 3 presents the comparison of classification rates on nine mosaics from the Brodatz dataset. We see that for all mosaics but 5v, the proposed method (κ_g with $\lambda = 10^{-3}, \beta = 2 \times 10^{-2}$) has higher classification rates than RSR. For the 5v mosaics, the classification rate of RSR is a little higher (0.012). The average classification rates on all the nine mosaics are 0.66, 0.81, 0.87, and 0.92 for LogE-SR, TSC, RSR, and Log-E Kernel, respectively.

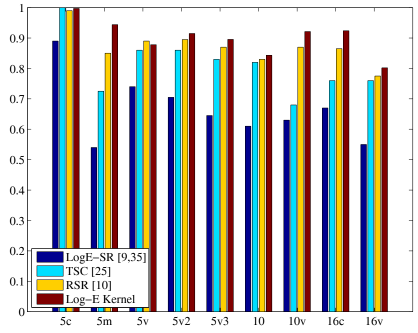

Figure 3. Classification rates on nine mosaics from the Brodatz dataset. Average rates on all nine mosaics are 0.66, 0.81, 0.87, and 0.92 for LogE-SR, TSC, RSR, and Log-E Kernel, respectively.

4.2. Dictionary Learning

To testify the effectiveness of the proposed dictionary learning method, we compare three methods: random sampling, K-Means clustering and dictionary learning. In all the methods, the sparse vectors are obtained via Log-E Kernel κ_g. The K-Means clustering is performed in the Log-Euclidean framework [1]: the covariance matrices are first mapped to the linear space $\mathcal{S}_n$ by matrix logarithm, in which the clustering is performed and the results are then mapped back to $\mathcal{S}_n^+$.

Texture Classification We use the Brodatz dataset and follow the experimental setting in [12]. All the 111 texture classes are used. In each image we randomly select 50 image patches of 32×32 pixels, from which a 5×5 covariance matrix is computed. The 5-dimensional feature vectors to compute the covariance matrix comprise grayscale intensity, and the 1st and 2nd partial derivatives with respect to spatial coordinates. In each class, 20 samples are randomly selected for training; in the remaining ones, 20 are used as probe samples and 10 as gallery ones. We thus have 2200 covariance matrices in total for dictionary learning.

We use the k-nearest neighbour classifier ($k = 3$) for classification. Figure 4 shows the curves of classification accuracy vs. the number of atom matrices. It can be seen that the dictionary learning method is consistently superior to random dictionary and Log-E K-Means, particularly when the number of atom matrices are small. It is interesting to notice that the random dictionary is better than the learned dictionary via Log-E K-Means if the number of atom matrices are less than 80. This may be because that in this texture dataset, on the whole the textures tend to be regular, and generally any patch may be representative of the texture while K-Means brings bias. We also observe that the performance of both random sampling and Log-E K-Means improves with the increase of atom matrix number.

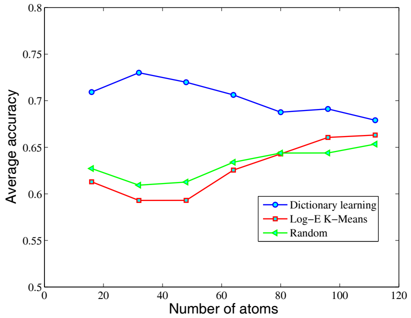

Figure 4. Classification accuracy on the Brodatz dataset

Scene Categorization We use the popular benchmark database *Scene15* [16] for classification performance evaluation. It consists of 15 categories each of which includes about 200∼400 images of average size of 300×250 pixels, and there are 4,485 images in total. In each image, we extract 8×8 covariance matrices at dense grids with a stride of 8 pixels. The patch size to compute the covariance matrix is 16×16 and the raw features are orientation histogram of 8 bins [9]. Each image is represented by a histogram computed from the sparse vectors of sampled patches via the max pooling strategy [32]. The SVM is trained using the LIBSVM package [7].

We adopt the methodology in [16] (BoW+SPM+SVM) for training and classification. First, among covariance matrices of all images, 50,000 ones are randomly chosen which are used to obtain atom matrices. For each class, 100 images are randomly selected as training data and the rest as testing data. The experiments are repeated 20 times and the results are averaged. Table 3 presents the classification rates of different methods vs. the number of atom matrices. It can be seen that in all cases the classification rates of the proposed method are over 18 percent higher than the random dictionary. We can also observe that the proposed method has over 8 percent, 4 percent, and 2 percent advantages over Log-E K-Means Clustering for 32, 64, and 128 atom matrices, respectively.

From both of the above experiments, we observe that as atom matrix number grows, the performance gains of the dictionary learning over the other two methods gets smaller. As the current dictionary is generative without discriminative information, more powerful representational capability does not necessarily mean better discriminability. This may explain the above findings and we think that the performance difference between the three methods will get smaller or even negligible as the atom matrix number becomes much larger.

Table 3. Classification accuracy on the Scene15 database

Num. of atoms	32	64	128
Random dictionary	44.80±0.90	57.64±0.59	62.25±0.65
LogE K-Means	67.69±0.56	76.25±0.48	78.80±0.53
Dictionary learning	**75.84±0.64**	**79.27±0.65**	**80.92±0.44**

5. Conclusion

This paper presented a novel Riemannian metric based kernel method for SR and DL in $\mathcal{S}_n^+$. It embeds the SPD matrices into RKHS so that the linear decomposition makes sense. The proposed kernels are based on the Log-Euclidean framework. They not only characterize the geodesic distance between SPD matrices, but also satisfy the Mercer's condition in general conditions. Our method overcomes the disadvantages of existing work which fails to make full use of the Riemannian manifold structure of $\mathcal{S}_n^+$. Experiments have shown the superiority of our method to state-of-the-arts.

We disclosed that the space of SPD matrices is a complete inner product space, and developed a broad family of p.d. kernels. These kernels are readily suitable for kernel-based regression, function estimation, and classification on the space of SPD matrices. It is also interesting, by using the proposed kernels, to explore kernel-based distance metric learning methods to adapt to various tasks of image retrieval and classification based on the covariance descriptors.

Acknowledgments: The work was supported by NSFC 60973080, 61170149, Program for New Century Excellent Talents in University (NCET-10-0151), the Fundamental Research Funds for the Central Universities (DUT13RC(3)02), Key Project by Chinese Ministry of Education (210063). We thank Dr. Harandi for providing us with the code of RSR and pre-processed face images.

References

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. on Matrix Analysis and Applications*, 2006.

[2] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups : Theory of Positive Definite and Related Functions*. Springer, 1984.

[3] J. Bigun and G. Granlund. Optimal orientation detection of linear symmetry. In *ICCV*, pages 433–438, 1987.

[4] D. L. Bihan, J. Mangin, C. Poupon, C. Clark, S. Pappata, and N. Molko. Diffusion tensor imaging: Concepts and applications. *J. Magn. Reson. Imaging*, 13(4):534–546, 2001.

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

[6] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista. A nonparametric Riemannian framework on tensor field with application to foreground segmentation. In *ICCV*, pages 1–8, 2011.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

[8] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[9] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1994.

[10] S. Gao, I. Tsang, and L.-T. Chia. Sparse representation with kernels. *TIP*, 22(2):423–434, 2013.

[11] K. Guo, P. Ishwar, and J. Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. In *AVSS*, pages 188–195, 2010.

[12] 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 *ECCV(2)*, 2012.

[13] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

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

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

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[17] L. Liu and P. Fieguth. Texture classification from random features. *PAMI*, 34:574 – 586, 2012.

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

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

[20] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *PAMI*, 22(10):1090–1104, 2000.

[21] T. Randen and J. Husoy. Filtering for texture classification: a comparative study. *PAMI*, 21(4):291–310, 1999.

[22] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Tensor sparse coding for region covariances. In *ECCV (4)*, pages 722–735, 2010.

[23] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Positive definite dictionary learning for region covariances. In *ICCV*, pages 1013–1019, 2011.

[24] S. Sra. Positive definite matrices and the symmetric stein divergence. *arXiv:1110.1773*, 2012.

[25] S. Sra and A. Cherian. Generalized dictionary learning for symmetric positive definite matrices with application to nearest neighbor retrieval. In *ECML PKDD(3)*, pages 318–332, 2011.

[26] A. E. Taylor and D. C. Lay. *Introduction to functional analysis, 2nd ed.* Krieger Publishing Co., Inc., 1986.

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

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

[29] J. Weichert and H. Hagen, editors. *Visualization and image processing of tensor fields*. Springer, 2006.

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[31] Y. Xie, B. C. Vemuri, and J. Ho. Statistical analysis of tensor fields. In *MICCAI'10*, pages 682–689, 2010.

[32] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[33] M. Yang and L. Zhang. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. In *ECCV*, pages 448–461, 2010.

[34] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo. Action recognition using sparse representation on covariance manifolds of optical flow. In *ACCV*, pages 343–353, 2009.