Sparse Coding on Symmetric Positive Definite Manifolds Using Bregman Divergences

Mehrtash T. Harandi, *Member, IEEE*, Richard Hartley, *Fellow, IEEE*, Brian Lovell, *Senior Member, IEEE*, and Conrad Sanderson, *Member, IEEE*

***Abstract*— This paper introduces sparse coding and dictionary learning for symmetric positive definite (SPD) matrices, which are often used in machine learning, computer vision, and related areas. Unlike traditional sparse coding schemes that work in vector spaces, in this paper, we discuss how SPD matrices can be described by sparse combination of dictionary atoms, where the atoms are also SPD matrices. We propose to seek sparse coding by embedding the space of SPD matrices into the Hilbert spaces through two types of the Bregman matrix divergences. This not only leads to an efficient way of performing sparse coding but also an online and iterative scheme for dictionary learning. We apply the proposed methods to several computer vision tasks where images are represented by region covariance matrices. Our proposed algorithms outperform state-of-the-art methods on a wide range of classification tasks, including face recognition, action recognition, material classification, and texture categorization.**

***Index Terms*— Bregman's divergences, dictionary learning, kernel methods, Riemannian's geometry, sparse coding.**

I. Introduction

SPARSITY is a popular concept in signal processing [1]–[3] and stipulates that natural signals like images can be efficiently described using only a few nonzero coefficients of a suitable basis (i.e., dictionary) [1]. This paper introduces techniques to perform sparse coding on Symmetric Positive Definite (SPD) matrices. More specifically, unlike traditional sparse coding schemes that work on vectors, in this paper, we discuss how SPD matrices can be described by sparse combination of dictionary atoms, where the atoms are also SPD matrices.

Our motivation stems from pervasive role of SPD matrices in machine learning, computer vision, and related areas. For example, SPD matrices have been used in medical imaging, texture classification [4], [5], action recognition and gesture categorization [6], as well as face recognition [5], [7].

Extending sparse coding methods to SPD matrices is not trivial, since such matrices form the interior of the positive semidefinite cone. In other words, simply vectorizing SPD matrices and employing Euclidean geometry (e.g., Euclidean norms) do not lead to accurate representations [8]–[10]. To overcome the drawbacks of Euclidean structure, SPD matrices are usually analyzed using a Riemannian structure, known as SPD or tensor manifold [8]. This is where the difficulties arise. On one hand, considering the Riemannian geometry is important, as discussed in various recent studies [5], [8]–[10]. On the other hand, the nonlinearity of the Riemannian structure is a hindrance and demands specialized machineries.

Generally speaking, two approaches to handle the nonlinearity of Riemannian manifolds are: 1) locally flattening them via tangent spaces [6], [9] and 2) embedding them in higher dimensional Hilbert spaces [5], [10], [11]. The latter has recently received a surge of attention, since embedding into a reproducing kernel Hilbert space (RKHS) through kernel methods [12] is a well-established and principled approach in machine learning. However, embedding SPD manifolds into RKHS requires nontrivial kernel functions defined on such manifolds, which, according to Mercer's theorem [12], must be positive definite (pd).

The contributions in this paper[1] are four fold.

1) We propose sparse coding and dictionary learning algorithms for data points (matrices) on SPD manifolds, by embedding the manifolds into RKHS. This is advantageous, as linear geometry applies in RKHS.
2) For the embedding we propose kernels derived from two Bregman matrix divergences, namely, the Stein and Jeffrey divergences. While the kernel property of the Jeffrey divergence was discovered in 2005 [13], to the best of authors' knowledge, this is one of the first attempts to benefit from this kernel for analyzing SPD matrices.
3) For both kernels, we devise a closed-form solution for updating an SPD dictionary atom by atom.
4) We apply the proposed methods to several computer vision tasks where images are represented by region covariance matrices. Our proposed algorithms

Manuscript received October 1, 2013; revised December 18, 2014; accepted December 27, 2014. This work was supported by the ARC Discovery Grant DP150104645. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council (ARC) through the ICT Centre of Excellence Program.

M. Harandi and R. Hartley are with NICTA, Canberra Research Laboratory, Canberra, ACT 2601, Australia, and also with the College of Engineering and Computer Science, Australian National University, Canberra ACT 0200, Australia (email: mehrtash.harandi@nicta.com.au; richard.hartley@nicta.com.au).

B. Lovell is with the University of Queensland, Brisbane, QLD 4072, Australia (e-mail: lovell@itee.uq.edu.au).

C. Sanderson is with NICTA, Queensland Research Laboratory, Brisbane, QLD 4000, Australia, and also with the University of Queensland, Brisbane, QLD 4072, Australia (e-mail: conradsand@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2387383

[1]This paper is a thoroughly extended and revised version of our earlier work [5]. In addition to providing more insights on the proposed methods, we extend our primary ideas by studying and devising coding and dictionary learning methods in the RKHS induced by the Jeffrey kernel. We also devise an efficient algorithm to obtain sparse codes in our RKHS-based formulation.

outperform state-of-the-art methods on several classification tasks, including face recognition, texture classification, and action recognition.

II. Related Work

In computer vision, SPD matrices are used in various applications, including object tracking [4], texture classification [4], [5], face recognition [5], [7], action recognition [6], [14], pedestrian detection [9], and object recognition [10]. This is mainly because region covariance descriptors (RCMs) [4], which encode second order statistics, are straightforward and relatively robust descriptors for images and videos. Moreover, structure tensors, which are by nature SPD matrices, encode important image features (e.g., texture and motion in optical flow estimation and motion segmentation). Finally, diffusion tensors that naturally arise in medical imaging are described by 3×3 SPD matrices [8].

Our interest in this paper is to perform sparse coding and dictionary learning on SPD matrices, since modern systems in various applications benefit from the notion of sparse coding. However, while significant steps have been taken to develop the theory of the sparse coding and dictionary learning in Euclidean spaces, only a handful of studies tackle similar problems for SPD matrices [14]–[16].

Sra and Cherian [15] proposed to measure the similarity between SPD matrices using the Frobenius norm and formulated the sparse coding and dictionary learning problems accordingly. While solving the problems using purely Euclidean structure of SPD matrices is computationally attractive, it neglects the Riemannian structure of SPD manifolds.

A somehow similar and straightforward idea is to flatten an SPD manifold using a fixed tangent space. Sparse coding by embedding manifolds into their identity tangent spaces, which identifies the Lie algebra of SPD manifolds, is considered in [14], [17], [18]. Though such embedding considerably simplifies the sparse coding formulation, the pair-wise distances are no longer adequate, which can affect discrimination performance. This is exacerbated for manifolds with negative curvature (e.g., SPD manifolds), since pair-wise distances are not even directly bounded.[2]

A more involved approach to learn a Riemannian dictionary was proposed very recently by Ho *et al*. [16]. The underlying idea is to exploit the tangent bundle of the manifold. To avoid a trivial solution in this approach, an affine constraint has to be added to the general formulation [16]. While this results in independency to the origin, it no longer addresses the original problem. Furthermore, switching back and forth to tangent spaces of SPD manifolds (as required by this formulation) can be computationally very demanding for high-dimensional manifolds.

Sivalingam *et al*. [20], [21] proposed Tensor Sparse Coding (TSC) which utilizes the Burg divergence (an asymmetric type of Bregman divergence) to perform sparse coding and dictionary learning on SPD manifolds. To this end, they show that when the Burg divergence is used as the proximity measure, the problem of sparse coding becomes a determinant maximization (MAXDET) problem that is convex and hence can be solved by interior point algorithms [20]. As for dictionary learning, two methods were proposed in [20] and [21]. In the first method, a gradient descent approach was utilized to update dictionary atoms one by one. Inspired by the K-SVD algorithm [22], the second method updates dictionary atoms by minimizing a form of residual error over training data, which speeds up the process of dictionary learning. Besides the asymmetric nature of the Burg divergence, we note that the computational complexity of the TSC algorithm is high, especially for high-dimensional SPD manifolds.

[2]For manifolds with positive curvature, pair-wise distances on tangent spaces are greater or equal to true geodesic distances on the manifold according to the Toponogov theorem [19]. Such property does not hold for manifolds with negative curvature.

III. Preliminaries

This section provides an overview on the Riemannian geometry of SPD manifolds, the Bregman divergences, and their properties. It provides the groundwork for techniques described in following sections. Throughout this paper, bold capital letters denote matrices (e.g., $\boldsymbol{X}$) and bold lower-case letters denote column vectors (e.g., $\boldsymbol{x}$). Notation x_i is used to indicate element at position i of vector $\boldsymbol{x}$. $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\|\boldsymbol{x}\|_2 = (\boldsymbol{x}^T\boldsymbol{x})^{1/2}$ and $\|\boldsymbol{x}\|_1 = \sum_i |x_i|$ denote the ℓ_2 and ℓ_1 norms, respectively, with T indicating the matrix transpose. $\|\boldsymbol{X}\|_F = (\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X}))^{1/2}$ designates the Frobenius norm. $\mathrm{GL}(n)$ denotes the general linear group, the group of real invertible $n \times n$ matrices. $\mathrm{Sym}(n)$ is the space of real $n \times n$ symmetric matrices.

A. Riemannian Geometry of SPD Manifolds

An $n \times n$, real SPD matrix $\boldsymbol{X}$ has the property that $\boldsymbol{v}^T\boldsymbol{X}\boldsymbol{v} > 0$ for all nonzero $\boldsymbol{v} \in \mathbb{R}^n$. The space of $n \times n$ SPD matrices, denoted by $\mathcal{S}_{++}^n$, is not a vector space since multiplying an SPD matrix by a negative scalar results in a matrix that does not belong to $\mathcal{S}_{++}^n$. Instead, $\mathcal{S}_{++}^n$ forms the interior of a convex cone in the $n(n+1)/2$-D Euclidean space. The $\mathcal{S}_{++}^n$ space is mostly studied when endowed with a Riemannian metric and thus forms a Riemannian manifold [8].

On a Riemannian manifold, a natural way to measure nearness is through the notion of geodesics, which are curves analogous to straight lines in $\mathbb{R}^n$. The geodesic distance is thus defined as the length of the shortest curve connecting the two points. The tangent space at a point $\boldsymbol{P}$ on the manifold $T_{\boldsymbol{P}}\mathcal{M}$ is a vector space that consists of the tangent (i.e., velocity) vectors of all possible curves passing through $\boldsymbol{P}$.

Two operators, namely, the exponential map $\exp_{\boldsymbol{P}}(\cdot) : T_{\boldsymbol{P}}\mathcal{M} \to \mathcal{M}$ and the logarithm map $\log_{\boldsymbol{P}}(\cdot) = \exp_{\boldsymbol{P}}^{-1}(\cdot) : \mathcal{M} \to T_{\boldsymbol{P}}\mathcal{M}$, are defined over the Riemannian manifolds to switch between the manifold and tangent space at $\boldsymbol{P}$. The exponential operator maps a tangent vector Δ to a point $\boldsymbol{X}$ on the manifold. The property of the exponential map ensures that the length of Δ becomes equal to the geodesic distance between $\boldsymbol{X}$ and $\boldsymbol{P}$. The logarithm map is the inverse of the exponential map, and maps a point on the manifold to

the tangent space $T_{\boldsymbol{P}}$. The exponential and logarithm maps vary as point $\boldsymbol{P}$ moves along the manifold.

On the SPD manifold, the affine invariant Riemannian metric (AIRM) [8], defined as

$$\begin{aligned}\langle \boldsymbol{V}, \boldsymbol{W}\rangle_{\boldsymbol{P}} &\triangleq \langle \boldsymbol{P}^{-1/2}\boldsymbol{V}\boldsymbol{P}^{-1/2}, \boldsymbol{P}^{-1/2}\boldsymbol{W}\boldsymbol{P}^{-1/2}\rangle \\ &= \mathrm{Tr}(\boldsymbol{P}^{-1}\boldsymbol{V}\boldsymbol{P}^{-1}\boldsymbol{W}) \end{aligned} \tag{1}$$

for $\boldsymbol{P} \in \mathcal{S}_{++}^n$ and $\boldsymbol{V}, \boldsymbol{W} \in T_{\boldsymbol{P}}\mathcal{M}$, induces the following geodesic distance between points $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\delta_R(\boldsymbol{X}, \boldsymbol{Y}) = \|\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\|_F \tag{2}$$

with $\log(\cdot)$ being the principal matrix logarithm.

B. Bregman Divergences

In this section, we introduce two divergences derived from the Bregman matrix divergence, namely, the Jeffrey and Stein divergences. We discuss their properties and establish their relations to AIRM. This provides motivation and grounding for our formulation of sparse coding and dictionary learning using the aforementioned divergences.

Definition 1: Let $\zeta : \mathcal{S}_{++}^n \to \mathbb{R}$ be a strictly convex and differentiable function defined on the symmetric positive cone $\mathcal{S}_{++}^n$. The Bregman matrix divergence $d_\zeta : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \to [0, \infty)$ is defined as

$$d_\zeta(\boldsymbol{X}, \boldsymbol{Y}) = \zeta(\boldsymbol{X}) - \zeta(\boldsymbol{Y}) - \langle \nabla_\zeta(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y}\rangle \tag{3}$$

where $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = \mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{Y})$ and $\nabla_\zeta(\boldsymbol{Y})$ represents the gradient of ζ evaluated at $\boldsymbol{Y}$.

Loosely speaking, the Bregman divergence between $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be understood as the distance between the function $\zeta(\boldsymbol{X})$ and its first-order Taylor approximation constructed at $\boldsymbol{Y}$. The Bregman divergence is asymmetric, nonnegative, and definite [i.e., $d_\zeta(\boldsymbol{X}, \boldsymbol{Y}) = 0$, iff $\boldsymbol{X} = \boldsymbol{Y}$]. While the Bregman divergence enjoys a variety of useful properties [23], its asymmetric behavior can be a hindrance [e.g., in support vector machines (SVMs), the kernels need to be symmetric, hence asymmetric divergences cannot be used to devise kernels]. In this paper, we are interested in two types of symmetrized Bregman divergences, namely, the *Jeffrey* and the *Stein* divergences.

Definition 2: The J divergence (also known as Jeffrey or symmetric KL divergence) is obtained from the Bregman divergence of (3) using $\zeta(\boldsymbol{X}) = -\log|\boldsymbol{X}|$ as the seed function where $|\cdot|$ denotes a determinant

$$\begin{aligned} J(\boldsymbol{X}, \boldsymbol{Y}) &\triangleq \frac{1}{2}d_\zeta(\boldsymbol{X}, \boldsymbol{Y}) + \frac{1}{2}d_\zeta(\boldsymbol{Y}, \boldsymbol{X}) \\ &= \frac{1}{2}\mathrm{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) - \frac{1}{2}\log|\boldsymbol{X}^{-1}\boldsymbol{Y}| \\ &\quad + \frac{1}{2}\mathrm{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - \frac{1}{2}\log|\boldsymbol{Y}^{-1}\boldsymbol{X}| - n \\ &= \frac{1}{2}\mathrm{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - n. \end{aligned} \tag{4}$$

Definition 3: The Stein or S divergence (also known as Jensen–Bregman log-determinant divergence [24]) is obtained from the Bregman divergence of (3) by again using $\zeta(\boldsymbol{X}) = -\log|\boldsymbol{X}|$ as the seed function but through Jensen–Shannon symmetrization

$$\begin{aligned} S(\boldsymbol{X}, \boldsymbol{Y}) &\triangleq \frac{1}{2}d_\zeta\left(\boldsymbol{X}, \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) + \frac{1}{2}d_\zeta\left(\boldsymbol{Y}, \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) \\ &= \log\left|\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right| - \frac{1}{2}\log|\boldsymbol{X}\boldsymbol{Y}|. \end{aligned} \tag{5}$$

C. Properties of J and S Divergences

The J and S divergences have a variety of properties that are akin to those of AIRM. The pertinent properties which inspired us to seek sparse coding on $\mathcal{S}_{++}^n$ using such divergences are as follows.

1) Both the J and S divergences as well as AIRM (as its name implies) are invariant to affine transformation [8], [25], [26].
2) The length of curves under AIRM and S divergence is equal up to a scale [27].
3) The geometric mean of two tensors under AIRM coincides with the geometric mean under J and S divergences (see [25] for the S divergence and Appendix A for the proof on the J divergence).

The key message worth noting is the Hilbert space embedding property of the J and S divergences, which does not hold for AIRM [5], [10].

1) Hilbert Space Embedding (SPD Kernels): Both J and S divergences admit a Hilbert space embedding in the form of a radial basis function (RBF) kernel [12]. More specifically, for the J-divergence it has been shown that the kernel

$$k_J(\boldsymbol{X}, \boldsymbol{Y}) = \exp\{-\beta J(\boldsymbol{X}, \boldsymbol{Y})\} \tag{6}$$

is conditionally pd (cpd) [13]. Formally, it is explained in the following definition.

Definition 4 (Conditionally Positive Definite Kernels): Let $\mathcal{X}$ be a nonempty set. A symmetric function $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a cpd kernel on $\mathcal{X}$ if and only if $\sum_{i,j=1}^n c_i c_j k(x_i, x_j) \geq 0$ for any $n \in \mathbb{N}$, $x_i \in \mathcal{X}$ and $c_i \in \mathbb{R}$ with $\sum_{i=1}^n c_i = 0$.

The relations between pd and cpd kernels are studied by Berg *et al*. [28] and Schölkopf [29]. An important property of cpd kernels is presented as follows.

Proposition 1: For a kernel algorithm that is translation invariant, cpd kernels can be used instead of pd kernels [29].

This property relaxes the requirement of having pd kernels for certain types of kernel algorithms. For example, in SVMs, a cpd kernel can be seamlessly used instead of a pd kernel. We note that in [30], the kernel $k_J(\cdot, \cdot)$ was claimed to be pd. However, a formal proof is not available according to the best of authors' knowledge. For the Stein divergence, the kernel

$$k_S(\boldsymbol{X}, \boldsymbol{Y}) = \exp\{-\beta S(\boldsymbol{X}, \boldsymbol{Y})\} \tag{7}$$

is guaranteed to be pd for

$$\beta \in \left\{\frac{1}{2}, \frac{2}{2}, \ldots, \frac{n-1}{2}\right\} \cup \left\{\tau \in \mathbb{R} : \tau > \frac{1}{2}(n-1)\right\}. \tag{8}$$

Interested reader is referred to [25] for further details. For values of β outside of the above set, it is possible to convert a pseudokernel into a true kernel, as discussed in [31].

IV. Sparse Coding

Given a query $\boldsymbol{x} \in \mathbb{R}^n$, sparse coding in vector spaces optimizes the objective function

$$l_E(\boldsymbol{x}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \boldsymbol{x} - \sum_{j=1}^{N} y_j \boldsymbol{d}_j \right\|_2^2 + \mathrm{Sp}(\boldsymbol{y}) \tag{9}$$

with $\mathbb{D}_{n\times N} = [\ \boldsymbol{d}_1 \mid \boldsymbol{d}_2 \mid \cdots \mid \boldsymbol{d}_N\ ]$, $\boldsymbol{d}_i \in \mathbb{R}^n$, $N > n$ being a dictionary of size N. The function $\mathrm{Sp}(\boldsymbol{y})$ penalizes the solution if it is not sparse. The most common form of $l_E(\boldsymbol{x}, \mathbb{D})$ in the literature is obtained via ℓ_1-norm regularization

$$l_E(\boldsymbol{x}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \boldsymbol{x} - \sum_{j=1}^{N} y_j \boldsymbol{d}_j \right\|_2^2 + \lambda \|\boldsymbol{y}\|_1. \tag{10}$$

As elaborated in [32], directly translating the sparse coding problem to a nonflat Riemannian manifold $\mathcal{M}$ with a metric $\|\cdot\|_{\mathcal{M}}$ (such as geodesic distance) leads to rewriting (10) as

$$l_{\mathcal{M}}(\boldsymbol{X}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \boldsymbol{X} \ominus \biguplus_{j=1}^{N} y_j \odot \boldsymbol{D}_j \right\|_{\mathcal{M}}^2 + \lambda \|\boldsymbol{y}\|_1 \tag{11}$$

where $\mathbb{D} = \{\boldsymbol{D}_i\}_{i=1}^N$, $\boldsymbol{D}_i \in \mathcal{M}$ is a Riemannian dictionary and $\boldsymbol{X} \in \mathcal{M}$ is a query point. The operators $\ominus$, $\biguplus$ and $\odot$ are the Riemannian replacements for subtraction, summation, and scalar multiplication, respectively. We note that the operators $\ominus$ and $\biguplus$ should be commutative and associative.

There are several difficulties in solving (11). For example, metrics on the Riemannian manifolds do not generally result in (11) being convex [32]. As such, instead of solving (11), here we propose to side step the difficulties by embedding the manifold $\mathcal{M}$ into a Hilbert space $\mathcal{H}$ and replacing the idea of combination on manifolds with the general concept of linear combination in the Hilbert spaces.

For the SPD manifold $\mathcal{S}_{++}^n$, our idea is implemented as follows. Let $\mathbb{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_N\}$; $\boldsymbol{D}_i \in \mathcal{S}_{++}^n$ and $\phi : \mathcal{S}_{++}^n \to \mathcal{H}$ be a Riemannian dictionary and an embedding function on $\mathcal{S}_{++}^n$, respectively. Given a Riemannian point $\boldsymbol{X}$, we seek a sparse vector $\boldsymbol{y} \in \mathbb{R}^N$ such that $\phi(\boldsymbol{X})$ admits the sparse representation $\boldsymbol{y}$ over $\{\phi(\boldsymbol{D}_1), \phi(\boldsymbol{D}_2), \ldots, \phi(\boldsymbol{D}_N)\}$. In other words, we are interested in solving the following:

$$l_\phi(\boldsymbol{X}, \mathbb{D}) \triangleq \min_{\boldsymbol{y}} \left\| \phi(\boldsymbol{X}) - \sum_{j=1}^{N} y_j \phi(\boldsymbol{D}_j) \right\|_2^2 + \lambda \|\boldsymbol{y}\|_1. \tag{12}$$

For both J and S divergences, an embedding ϕ with a reproducing kernel property [12] exists, as explained in Section III. This enables us to use the kernel property $k(\boldsymbol{X}, \boldsymbol{Y}) = \phi(\boldsymbol{X})^T \phi(\boldsymbol{Y})$ to expand the ℓ_2 term in (12) as

$$\begin{aligned} &\left\| \phi(\boldsymbol{X}) - \sum_{j=1}^{N} y_j \phi(\boldsymbol{D}_j) \right\|_2^2 \\ &\quad = \phi(\boldsymbol{X})^T \phi(\boldsymbol{X}) - 2 \sum_{j=1}^{N} y_j \phi(\boldsymbol{D}_j)^T \phi(\boldsymbol{X}) \\ &\qquad + \sum_{i,j=1}^{N} y_i y_j \phi(\boldsymbol{D}_i)^T \phi(\boldsymbol{D}_j) \\ &\quad = k(\boldsymbol{X}, \boldsymbol{X}) - 2\boldsymbol{y}^T \mathcal{K}(\boldsymbol{X}, \mathbb{D}) + \boldsymbol{y}^T \mathbb{K}(\mathbb{D}, \mathbb{D}) \boldsymbol{y} \end{aligned} \tag{13}$$

where $\mathcal{K}(\boldsymbol{X}, \mathbb{D}) = [a_i]_{N\times 1}$; in which $a_i = k(\boldsymbol{X}, \boldsymbol{D}_i)$ and $\mathbb{K}(\mathbb{D}, \mathbb{D}) = [a_{ij}]_{N\times N}$ with $a_{ij} = k(\boldsymbol{D}_i, \boldsymbol{D}_j)$. Since $k(\cdot, \cdot)$ is a reproducing kernel, $\mathbb{K}$ is pd. This reveals that the optimization problem in (12) is convex and similar to its counterpart in Euclidean space, except for the definition of $\mathcal{K}$ and $\mathbb{K}$. Consequently, greedy or relaxation solutions can be adapted to obtain the sparse codes [1]. To solve (12) efficiently, we have extended the feature-sign search algorithm (FSSA) [33] to its kernel version FSSA in Appendix B.

We note that kernel sparse coding and dictionary learning in traditional Euclidean spaces are studied recently in [34] and [35]. In contrast, our aim is to obtain sparse coding of points on SPD manifolds using SPD matrices as dictionary atoms. In our proposed solution, this requires dedicated SPD kernels. Moreover, as will be discussed in Section V, dedicated algorithms for dictionary learning should be devised.

A. Classification Based on Sparse Representation

If the atoms in the sparse dictionary are not labeled (for example, if $\mathbb{D}$ is a generic dictionary not tied to any particular class), the generated sparse codes (vectors) for both training and query data can be fed to Euclidean-based classifiers like support vector machines [36] for classification. In a supervised classification scenario, i.e., if the atoms in sparse dictionary $\mathbb{D}$ are labeled, the generated sparse codes of the query sample can be directly used for classification. Let $\boldsymbol{y}_i = [y_0 \delta(l(0) - i), y_1 \delta(l(1) - i), \ldots, y_N \delta(l(N) - i)]^T$ be the class-specific sparse codes, where $l(j)$ is the class label of atom $\boldsymbol{D}_j$ and $\delta(x)$ is the discrete Dirac function [36]. An efficient way of utilizing class-specific dictionary is through computing residual errors [3]. In this case, the residual error of query sample $\boldsymbol{X}$ for class i is defined as

$$\varepsilon_i(\boldsymbol{X}) = \left\| \phi(\boldsymbol{X}) - \sum_{j=1}^{N} y_j \phi(\boldsymbol{D}_j) \delta(l(j) - i) \right\|^2. \tag{14}$$

Expanding (14) and noting that $k(\boldsymbol{X}, \boldsymbol{X})$ is not class dependent, the following can be obtained:

$$\varepsilon_i(\boldsymbol{X}) = -2\boldsymbol{y}_i^T \mathcal{K}(\boldsymbol{X}, \mathbb{D}) + \boldsymbol{y}_i^T \mathbb{K}(\mathbb{D}, \mathbb{D}) \boldsymbol{y}_i. \tag{15}$$

Alternatively, the similarity between query sample $\boldsymbol{X}$ to class i can be defined as $S_i(\boldsymbol{X}) = h(\boldsymbol{y}_i)$. The function $h(\cdot)$ could be a linear function like $h(\boldsymbol{y}_i) = \boldsymbol{y}_i^T \boldsymbol{1}_{N\times 1}$ or even a nonlinear one like $h(\boldsymbol{y}_i) = \max(\boldsymbol{y}_i)$. Preliminary experiments suggest that (15) leads to higher classification accuracies compared with the aforementioned alternatives.

B. Computational Complexity

In terms of computational complexity, we note that the complexity of computing the determinant of an $n \times n$ matrix through the Cholesky decomposition is $O(1/3n^3)$. Therefore, computing $S(\boldsymbol{X}, \boldsymbol{D}_i)$ by storing the determinant of dictionary atoms during learning costs $O(2/3n^3)$.

For the J divergence, we note that the inverse of an $n \times n$ SPD matrix can be computed through the Cholesky decomposition with $1/2n^3$ flops. Therefore, $J(\boldsymbol{X}, \boldsymbol{D}_i)$ can be computed in $2n^{2.3} + 1/2n^3$ flops if matrix multiplication is

done efficiently. As a result, computing the J divergence is cheaper than computing S divergence for SPD matrices of size less than 35.

The complexity of sparse coding is dictated by $\mathcal{K}(\boldsymbol{X}, \mathbb{D})$ in (15). Neglecting the complexity of the exponential in kernel functions, the complexity of generating (15) is $O(N(2n^{2.3} + 1/2n^3))$ for J divergence and $O(2N/3n^3)$ for S divergence.

Note that while the computational complexity is cubic in n, it is linear in N, i.e., number of dictionary atoms. To give the reader an idea on the speed of the proposed methods, it is worth mentioning that performing sparse coding on 93×93 covariance descriptors used in Section VI-A1 took less than 10 and 7 seconds with Jeffrey and Stein divergences, respectively (on an Intel i7 machine using MATLAB). Performing a simple nearest neighbor search using AIRM required more than 75 s on the same dataset.

V. Dictionary Learning

Given a finite set of observations $\mathbb{X} = \{\boldsymbol{X}_i\}_{i=1}^m,\ \boldsymbol{X}_i \in \mathcal{S}_{++}^n$, learning a dictionary $\mathbb{D} = \{\boldsymbol{D}_i\}_{i=1}^N,\ \boldsymbol{D}_i \in \mathcal{S}_{++}^n$ by embedding SPD manifolds into the Hilbert space can be formulated as minimizing the following energy function with respect to $\mathbb{D}$:

$$f(\mathbb{X}, \mathbb{D}) \triangleq \sum_{i=1}^{m} l_\phi(\boldsymbol{X}_i, \mathbb{D}). \tag{16}$$

Here, $l_\phi(\boldsymbol{X}, \mathbb{D})$ is the loss function defined in (12). $f(\mathbb{X}, \mathbb{D})$ should be small if $\mathbb{D}$ is good at representing the signals $\boldsymbol{X}_i$. Among the various solutions to the problem of dictionary learning in Euclidean spaces, iterative methods like K-SVD have received much attention [1]. Borrowing the idea from Euclidean spaces, we propose to minimize the energy in (16) iteratively.

To this end, we first initialize the dictionary $\mathbb{D}$ randomly. It is also possible to use intrinsic k-means clustering using the Karcher mean [8] to initialize the dictionary. Each iteration of dictionary learning then constitutes of two parts, namely, a sparse coding step and a dictionary update step. In the sparse coding step, the dictionary $\mathbb{D}$ is fixed and sparse codes, $\{\boldsymbol{y}_i\}_{i=1}^m$ are computed, as discussed in Section IV. In the dictionary update step, $\{\boldsymbol{y}_i\}_{i=1}^m$ are held fixed while $\mathbb{D}$ is updated, with each dictionary atom updated independently. This resembles the expectation maximization algorithm [37] in nature. In Sections V-A–V-C, we discuss how dictionary atoms can be updated for both J and S divergences.

A. Dictionary Updates for J Divergence

As mentioned above, to update $\boldsymbol{D}_r$, we keep $\boldsymbol{D}_j,\ j \neq r$ and the sparse codes $\{\boldsymbol{y}_i\}_{i=1}^m$ in (16) fixed. Generally speaking, one can update $\boldsymbol{D}_r$ using gradient descend algorithms on SPD manifolds. This can be done at iteration t by exploiting the tangent space at $\boldsymbol{D}_r^{(t)}$ and moving along the direction of steepest descent and utilizing the exponential map to obtain $\boldsymbol{D}_r^{(t+1)}$ as a point on $\mathcal{S}_{++}^n$.

In this paper, we propose to learn the dictionary in an online manner. Our proposal results in an analytical and closed-form solution for updating dictionary atoms one by one. In contrast to [16], our formulation does not exploit the tangent bundle and exponential maps, and is hence faster and more scalable.

By fixing $\boldsymbol{D}_j,\ j \neq r$ and $\{\boldsymbol{y}_i\}_{i=1}^m$, the derivative of (16) with respect to $\boldsymbol{D}_r$ can be computed as

$$\begin{aligned}\frac{\partial f(\mathbb{X}, \mathbb{D})}{\partial \boldsymbol{D}_r} &= \sum_{i=1}^{m} \frac{\partial l_\phi(\boldsymbol{X}_i, \mathbb{D})}{\partial \boldsymbol{D}_r} \\ &= \sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(\sum_{j=1}^{N} \boldsymbol{y}_{i,j} \frac{\partial k(\boldsymbol{D}_j, \boldsymbol{D}_r)}{\partial \boldsymbol{D}_r} - 2 \frac{\partial k(\boldsymbol{X}_i, \boldsymbol{D}_r)}{\partial \boldsymbol{D}_r} \right).\end{aligned} \tag{17}$$

For the J divergence, we note that

$$\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2}(\boldsymbol{Y}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{Y}\boldsymbol{X}^{-1}). \tag{18}$$

Therefore

$$\frac{\partial k_J(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}} = -\frac{1}{2}\beta k_J(\boldsymbol{X}, \boldsymbol{Y})(\boldsymbol{Y}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{Y}\boldsymbol{X}^{-1}). \tag{19}$$

Plugging (19) into (17) and defining

$$\begin{aligned}\boldsymbol{P} &= \sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(\sum_{j=1}^{N} \boldsymbol{y}_{i,j} k_J(\boldsymbol{D}_j, \boldsymbol{D}_r) \boldsymbol{D}_j^{-1} - 2k_J(\boldsymbol{X}_i, \boldsymbol{D}_r) \boldsymbol{X}_i^{-1} \right) \\ \boldsymbol{Q} &= \sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(\sum_{j=1}^{N} \boldsymbol{y}_{i,j} k_J(\boldsymbol{D}_j, \boldsymbol{D}_r) \boldsymbol{D}_j - 2k_J(\boldsymbol{X}_i, \boldsymbol{D}_r) \boldsymbol{X}_i \right)\end{aligned} \tag{20}$$

the root of (17), i.e., $\partial f(\mathbb{X}, \mathbb{D})/\partial \boldsymbol{D}_r = 0$ can be written as

$$\boldsymbol{D}_r^{-1} \boldsymbol{Q} \boldsymbol{D}_r^{-1} = \boldsymbol{P}. \tag{21}$$

This equation is identified as a *Riccati* equation [38]. Its solution is pd and given as

$$\boldsymbol{D}_r = \boldsymbol{Q}^{\frac{1}{2}} \left(\boldsymbol{Q}^{-\frac{1}{2}} \boldsymbol{P}^{-1} \boldsymbol{Q}^{-\frac{1}{2}} \right)^{\frac{1}{2}} \boldsymbol{Q}^{\frac{1}{2}} \tag{22}$$

provided that both $\boldsymbol{P}$ and $\boldsymbol{Q}$ are pd. We note that in deriving the solution, we have assumed that $k_J(\boldsymbol{D}_r, \cdot)$ at iteration t can be replaced by $k_J(\boldsymbol{D}_r^{t-1}, \cdot)$ and hence $k_J(\boldsymbol{D}_r, \cdot)$ are treated as scalars.

B. Dictionary Updates for S Divergence

Similar to Section V-A, we need to compute the gradient of (16) with respect to $\boldsymbol{D}_r$, while $\{\boldsymbol{y}_i\}_{i=1}^m$ and other atoms are fixed. Noting that

$$\nabla_{\boldsymbol{X}} S(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X} + \boldsymbol{Y})^{-1} - \frac{1}{2}\boldsymbol{X}^{-1} \tag{23}$$

the solution of $\partial f(\mathbb{X}, \mathbb{D})/\partial \boldsymbol{D}_r = 0$ with $k_S(\cdot, \cdot)$ can be written as

$$\begin{aligned}&\sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(2k_S(\boldsymbol{X}_i, \boldsymbol{D}_r) \left((\boldsymbol{X}_i + \boldsymbol{D}_r)^{-1} - \frac{1}{2}\boldsymbol{D}_r^{-1} \right) \right) \\ &= \sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(\sum_{j=1}^{N} \boldsymbol{y}_{i,j} k_S(\boldsymbol{D}_j, \boldsymbol{D}_r) \left((\boldsymbol{D}_j + \boldsymbol{D}_r)^{-1} - \frac{1}{2}\boldsymbol{D}_r^{-1} \right) \right).\end{aligned} \tag{24}$$

Since (24) contains inverses and kernel values, a closed-form solution for computing $\boldsymbol{D}_r$ cannot be sought. As such, we propose an alternative solution by exploiting previous values of $(\boldsymbol{D}_i + \boldsymbol{D}_r)^{-1}$ in the update step. More specifically, rearranging (24) and replacing $k(\cdot, \boldsymbol{D}_r)$ as well as $(\boldsymbol{D}_i + \boldsymbol{D}_r)^{-1}$ by their previous values, atom $\boldsymbol{D}_r$ at iteration $t+1$ is updated according to

$$\boldsymbol{D}_r^{(t+1)} = \frac{2\boldsymbol{P}^{-1}}{\sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(2k_S(\boldsymbol{X}_i, \boldsymbol{D}_r) - \sum_{j=1}^{N} \boldsymbol{y}_{i,j} k_S(\boldsymbol{D}_j, \boldsymbol{D}_r) \right)} \tag{25}$$

where

$$\boldsymbol{P} = \sum_{i=1}^{m} \boldsymbol{y}_{i,r} \left(2k_S(\boldsymbol{X}_i, \boldsymbol{D}_r) \left(\boldsymbol{X}_i + \boldsymbol{D}_r^{(t)} \right)^{-1} - \sum_{j=1}^{N} \boldsymbol{y}_{i,j} k_S(\boldsymbol{D}_j, \boldsymbol{D}_r) \left(\boldsymbol{D}_j + \boldsymbol{D}_r^{(t)} \right)^{-1} \right). \tag{26}$$

C. Practical Considerations

The dictionary update in (22) results in an SPD matrix provided that matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are SPD. In practice, this might not be the case and as such projection to the pd cone is required. The same argument holds for (25). Given an arbitrary square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, the problem of finding the closest SPD matrix to $\boldsymbol{A}$ has received considerable attention in the literature [39]. While projecting onto pd cone can be achieved by thresholding (i.e., replacing negative eigenvalues by a small positive number), a more principal approach can be used as follows. If square matrix $\boldsymbol{X}$ is pd, then $\boldsymbol{X} + \boldsymbol{X}^T$ is also pd. As such, the following convex problem can be solved[3] to obtain the closest SPD matrix $\boldsymbol{X}$ to the square matrix $\boldsymbol{A}$:

$$\min \|\boldsymbol{A} - \boldsymbol{X}\|_F \quad \text{s.t.} \quad \boldsymbol{X} + \boldsymbol{X}^T \succ 0. \tag{27}$$

We note that the formulation provided here works for nonsymmetric matrix $\boldsymbol{A}$ as well. This is again useful in practice as numerical issues might create nonsymmetric matrices [e.g., $\boldsymbol{P}$ and $\boldsymbol{Q}$ in (22) might not become symmetric due to the limited numerical accuracy in a computational implementation].

VI. Experiments

Two sets of experiments[4] are presented in this section. In the first set, we evaluate the performance of the proposed sparse coding methods (as described in Section IV) without dictionary learning. This is to contrast sparse coding to previous state-of-the-art methods on several popular closed-set classification tasks. To this end, each point in the training set is considered as an atom in the dictionary. Since the atoms in the dictionary are labeled in this case, the residual error approach for classification (as described in Section IV-A) will be used to determine the label of a query point. In the second set of experiments, the performance of the sparse coding methods is evaluated in conjunction with the proposed dictionary learning algorithms described in Section V. For brevity, we denote the Riemannian sparse representation with J divergence by RSR-J, and the S divergence counterpart by RSR-S.

TABLE I
Recognition Accuracy (in %) for the HDM05–MOCAP Dataset [42]

Method	Recognition Accuracy
logEuc-SC [14]	89.9%
Cov3DJ [43]	95.4%
RSR-J	**98.2%**
RSR-S	97.3%

The first priority of the experiments is to contrast the proposed methods against recent techniques designed to work on SPD manifolds. That is, the tasks and consequently the datasets were chosen to enable fair comparisons against state-of-the-art SPD methods. While exploring other visual tasks such as face verification [41] is beyond the scope of this paper, it is an interesting path to pursue in future work.

A. Sparse Coding

Below, we compare and contrast the performance of RSR-J and RSR-S methods against state-of-the-art techniques in five classification tasks, namely, action recognition from 3-D skeleton data, face recognition, material classification, person reidentification, and texture categorization.

1) Action Recognition From 3-D Skeleton Sequences: We used the motion capture HDM05 dataset [42] for the task of action recognition from skeleton data (see examples in Fig. 1). Each action is encoded by the locations of 31 joints over time, with the speed of 120 frames/s. Given an action by K joints over m frames, we extracted the joint covariance descriptor [43], which is an SPD matrix of size 3k $\times$ 3k as follows. Let $x_i(t)$, $y_i(t)$, and $z_i(t)$ be the x, y, and z coordinates of the ith joint at frame t. Let $\boldsymbol{f}(t)$ be the vector of all joint locations at time t, i.e., $\boldsymbol{f}(t) = [x_1(t), \ldots, x_K(t), y_1(t), \ldots, y_K(t), z_1(t), \ldots, z_K(t)]^T$, which has 3k elements. The action represented over m frames is then described by the covariance of vectors $\boldsymbol{f}(t)$.

We used three subjects (140 action instances) for training, and the remaining two subjects (109 action instances) for testing. The set of actions used in this experiment is: clap above head, deposit floor, elbow to knee, grab high, hop both legs, jog, kick forward, lie down floor, rotate both arms backward, sit down chair, sneak, squat, stand up lie, and throw basketball.

In Table I, we compare the performances of RSR-J and RSR-S against log-Euclidean sparse coding (logEuc-SC) [14] and Covariance descriptor on 3D Joint locations (Cov3DJ) [43]. The TSC algorithm [21] does not scale well to large SPD matrices and thus is not considered here. Cov3DJ encodes the relationship between joint movement and time by deploying multiple covariance matrices over subsequences in a hierarchical fashion. The results show that in this case,

[3] A solver like CVX [40] can be used.

[4] The corresponding MATLAB/Octave source code is available at http://nicta.com.au/people/mharandi.

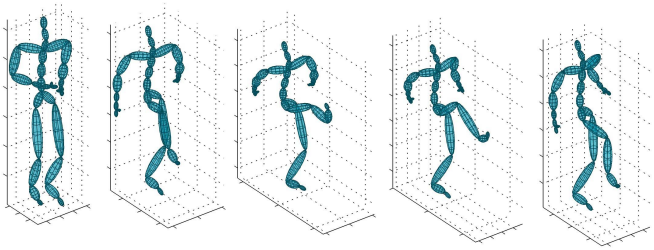


Fig. 1. Example of a kicking action from the HDM05 action dataset [42].

(a) (b) (c) (d) (e) (f) (g)

Fig. 2. Examples from the FERET face dataset [44]. (a) ba. (b) bj. (c) bk. (d) bd. (e) be. (f) bf. (g) bg.

RSR-J is better than RSR-S. Furthermore, both RSR-J and RSR-S outperform logEuc-SC and Cov3DJ.

2) Face Recognition: We used the *b* subset of the FERET dataset [44], which includes 1800 images from 200 subjects. The images were closely cropped around the face and downsampled to 64×64. Examples are shown in Fig. 2.

We performed four tests with various pose angles. Training data was composed of images marked ba, bj, and bk (i.e., frontal faces with expression and illumination variations). Images with bd, be, bf, and bg labels (i.e., nonfrontal faces) were used as test data.

Each face image is described by a 43×43 SPD matrix using the following features:

$$\boldsymbol{f}_{x,y} = [I(x,y), x, y, |G_{0,0}(x,y)|, \ldots, |G_{4,7}(x,y)|]^T$$

where $I(x,y)$ is the intensity value at position (x,y), $|\cdot|$ denotes the magnitude of a complex value, and $G_{u,v}(x,y)$ is the response of a 2-D Gabor wavelet centered at (x,y) with orientation u and scale v. In this paper, we followed [7] and generated 40 Gabor filters in eight orientations and five scales.

The proposed methods are compared against sparse-representation-based classification (SRC) [3], logEuc-SC [14], TSC [21], and its Gabor-based extension (GSRC) [45]. For SRC, Principal Component Analysis (PCA) was used to reduce the dimensionality of data. We evaluated the performance of SRC for various dimensions of PCA space and the maximum performance is reported. For the GSRC algorithm, we followed the recommendations of Yang and Zhang [45] for the downsampling factor in the Gabor filtering. As for the logEuc-SC, we consider a kernel extension of the original algorithm. In other words, instead of directly using $\log(\cdot)$ representations in a sparse coding framework, as done in [14], we consider a kernel extension on log representations using an RBF kernel. The kernel extension of sparse coding is discussed in depth in [34], [35]. This enhances the results in all cases and makes the logEuc-SC and RSR methods more comparable.

Table II shows the performance of all the studied methods for the task of face recognition. Both RSR-J and RSR-S outperform other methods, with RSR-S being marginally better than RSR-J.

TABLE II

RECOGNITION ACCURACY (IN %) FOR THE FERET FACE DATASET [44]

Method	bd	be	bf	bg	average
SRC [3]	27.5%	55.5%	61.0%	26.0%	42.5%
GSRC [45]	77.0%	93.5%	97.0%	79.0%	86.6%
logEuc-SC [14]	74.0%	94.0%	97.5%	80.5%	86.5%
TSC [21]	36.0%	73.0%	73.5%	44.5%	56.8%
RSR-J	**82.5%**	94.5%	**98.0%**	83.5%	89.6%
RSR-S	79.5%	**96.5%**	97.5%	**86.0%**	**89.9%**

Fig. 3. Examples from the Flickr dataset [46].

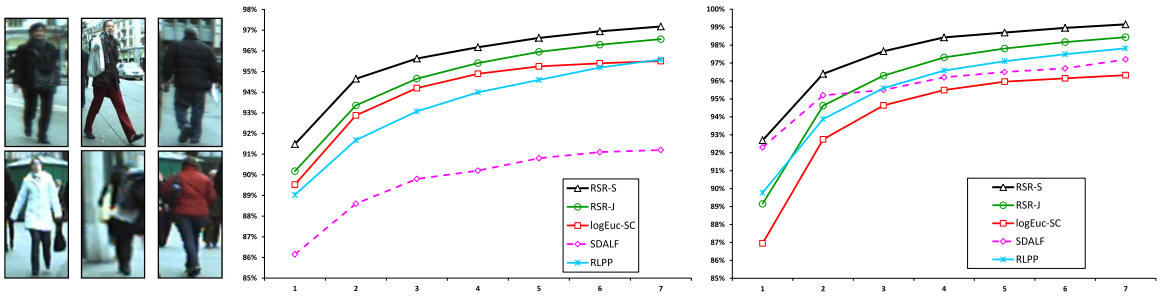
TABLE III

RECOGNITION ACCURACY (IN %) ALONG ITS STANDARD DEVIATION FOR THE FLICKR DATASET [46]

Method	Recognition Acc.
VZ [50]	$23.8\% \pm N/A$
VZ-augmented [49]	$37.4\% \pm N/A$
SD [48]	$29.9\% \pm N/A$
aLDA [49]	$44.6\% \pm N/A$
RSR-J	$44.0\% \pm 3.0$
RSR-S	$\mathbf{51.4\% \pm 1.9}$

3) Material Categorization: We used the Flickr dataset [46] for the task of material categorization. The dataset contains ten categories of materials: *fabric*, *foliage*, *glass*, *leather*, *metal*, *paper*, *plastic*, *stone*, *water*, and *wood*. Each category has 100 images, 50 of which are close-up views and the remaining 50 are views at object-scale (see Fig. 3 for examples). A binary human-labeled mask is provided for each image in the dataset, describing the location of the object in the image. We only consider pixels inside this binary mask for material recognition and disregard all background pixels. Scale-Invariant Feature Transform (SIFT) [47] features have recently been shown to be robust and discriminative for material classification [48]. We therefore constructed RCMs of size 155×155 using 128-D SIFT features (only from gray-scaled images) and 27-D color descriptors. To this end, SIFT descriptors were computed at points on a regular grid with five-pixel spacing. The color descriptor was obtained by simply stacking colors from 3×3 patches centered at grid points.

Table III compares the performance of the proposed methods against the state-of-the-art nonparametric geometric detail extraction method (SD) [48], augmented latent Dirichlet allocation (aLDA) [49], and the texton-based representation introduced in [50]. The results indicate that RSR-S considerably outperforms previous state-of-the-art approaches. We also note that RSR-J outperforms methods proposed in [48] and [50] by a large margin and is only slightly worse than the aLDA algorithm [49].



(a) Sample images (b) Seq. #1 (c) Seq. #2

Fig. 4. Person reidentification using the ETHZ dataset [52]. (a) Examples of pedestrians in the ETHZ dataset. (b) Results on sequence #1. (c) Results on sequence #2. The proposed RSR-J and RSR-S methods are compared with SDALF [53], RLPP [54], and logEuc-SC [14].

4) Person Reidentification: We used the modified ETHZ dataset [51]. The original ETHZ dataset was captured using a moving camera [52], providing a range of variations in the appearance of people. The dataset is structured into three sequences. Sequence 1 contains 83 pedestrians (4857 images), Sequence 2 contains 35 pedestrians (1936 images), and Sequence 3 contains 28 pedestrians (1762 images). See Fig. 4(a) for examples.

We downsampled all images to 64×32 pixels. For each subject we randomly selected 10 images for training and used the rest for testing. Random selection of training and testing data was repeated 20 times to obtain reliable statistics. To describe each image, the covariance descriptor was computed using the following features:

$$\boldsymbol{f}_{\boldsymbol{u}} = \left[\boldsymbol{u}, R_{\boldsymbol{u}}, G_{\boldsymbol{u}}, B_{\boldsymbol{u}}, \dot{R}_{\boldsymbol{u}}, \dot{G}_{\boldsymbol{u}}, \dot{B}_{\boldsymbol{u}}, \ddot{R}_{\boldsymbol{u}}, \ddot{G}_{\boldsymbol{u}}, \ddot{B}_{\boldsymbol{u}}\right]^T$$

where $\boldsymbol{u} = [x, y]$ is the position of a pixel, while $R_{\boldsymbol{u}}$, $G_{\boldsymbol{u}}$ and $B_{\boldsymbol{u}}$ represent the corresponding color information. The gradient and Laplacian for color C are represented by $\dot{C}_{\boldsymbol{u}} = [|\partial C/\partial x|, |\partial C/\partial y|]$ and $\ddot{C}_{\boldsymbol{u}} = [|\partial^2 C/\partial x^2|, |\partial^2 C/\partial y^2|]$, respectively.

We compared the proposed RSR methods with several techniques previously used for pedestrian detection: symmetry-driven accumulation of local features (SDALFs) [53], the Riemannian locality preserving projection (RLPP) [54], and logEuc-SC [14]. The results for TSC [21] could not be generated in a timely manner due to the heavy computational load of the algorithm.

Results for the first two sequences are shown in Fig. 4, in terms of cumulative matching characteristic (CMC) curves. The CMC curve represents the expectation of finding the correct match in the top n matches. The proposed RSR-S method obtains the highest accuracy on both sequences. RSR-J outperforms SDALF, RLPP, and logEuc-SC on sequence one. For the second sequence, RLPP and SDALF perform better than RSR-J for low ranks while RSR-J outperforms them for a rank higher than two.

For Sequence 3 (not shown), very similar performances are obtained by SDALF, RLPP, and the proposed methods. For this sequence, RSR-J and RSR-S achieve rank 1 accuracy of 98.3% and 98.7%, respectively. The CMC curves are almost saturated at perfect recognition at rank 3 for both RSR-J and RSR-S methods.

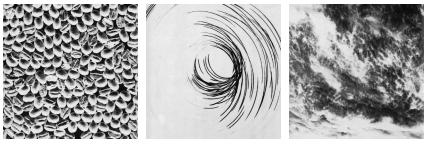


Fig. 5. Examples from the Brodatz texture dataset [55].

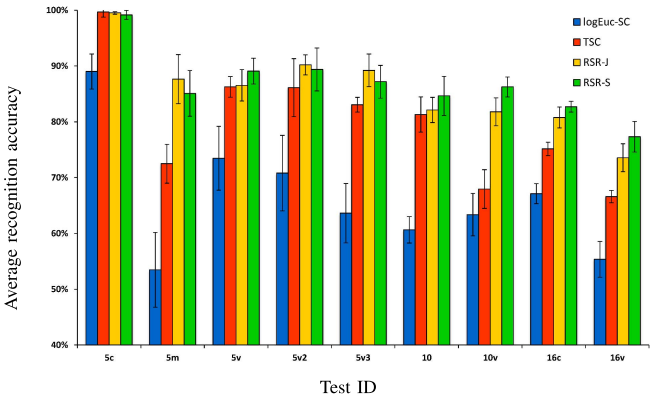


Fig. 6. Average recognition accuracy on the Brodatz texture dataset [55] using logEuc-SC representation [14], TSC [21], and the proposed RSR-J and RSR-S approaches. The black bars indicate standard deviations.

5) Texture Classification: We performed a classification task using the Brodatz texture dataset [55]. Examples are shown in Fig. 5. We followed the test protocol devised in [21] and generated nine test scenarios with various number of classes. This includes five-texture (5c, 5m, 5v, 5v2, and 5v3), 10-texture (10 and 10v), and 16-texture (16c and 16v) mosaics. To create a Riemannian manifold, each image was first downsampled to 256×256 and then split into 64 regions of size 32×32. The feature vector for any pixel $I(x, y)$ is $\boldsymbol{f}(x, y) = [I(x, y), |\partial I/\partial x|, |\partial I/\partial y|, |\partial^2 I/\partial x^2|, |\partial^2 I/\partial y^2|]^T$. Each region is described by a 5×5 covariance descriptor of these features. For each test scenario, five covariance matrices per class were randomly selected as training data and the rest was used for testing. The random selection of training/testing data was repeated 20 times.

Fig. 6 compares the proposed RSR methods against logEuc-SC [14] and TSC [21]. In general, the proposed methods obtain the highest recognition accuracy on all test scenarios except for the 5c test, where both methods have slightly

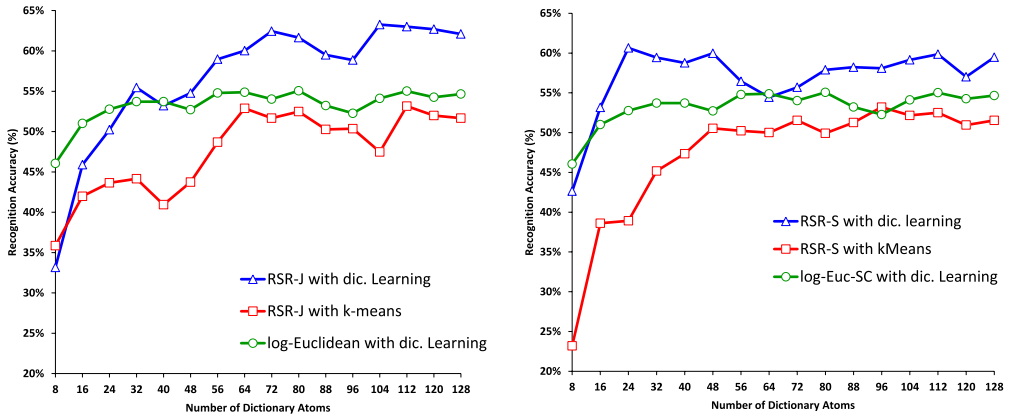


Fig. 7. Comparison of recognition accuracy versus size of dictionary for RSR-J and RSR-S. The red curve shows the accuracy for dictionaries learned by intrinsic k-means algorithm. The green curve shows the accuracy for dictionaries learned by the log-Euclidean method that is dictionary learning (K-SVD) along sparse coding on the identity tangent space. The blue curve shows the accuracy for the proposed learning approach.

worse performance than TSC. We note that in some cases such as 5m and 5v2, RSR-J performs better than RSR-S. However, RSR-S is overall a slightly superior method for this task.

B. Dictionary Learning

Here, we analyze the performance of the proposed dictionary learning techniques as described in Section V on two classification tasks: texture classification and action recognition.

1) Texture Classification: Here, we consider a multiclass classification problem, using 111 texture images of the Brodatz texture dataset [55]. From each image we randomly extracted 50 blocks of size 32×32. To train the dictionary, 20 blocks from each image were randomly selected, resulting in a dictionary learning problem with 2200 samples. From the remaining blocks, 20 per image were used as probe data and 10 as gallery samples. The process of random block creation and dictionary generation was repeated twenty times. The average recognition accuracies over probe data are reported here. In the same manner as in Section VI-A.5, we used the feature vector $\boldsymbol{f}(x, y) = [I(x, y), |\partial I/\partial x|, |\partial I/\partial y|, |\partial^2 I/\partial x^2|, |\partial^2 I/\partial y^2|]^T$ to create the covariance, where the first dimension is the grayscale intensity and the remaining dimensions capture first and second order gradients.

We used the proposed methods to obtain the sparse codes, coupled with a dictionary generated via two separate methods: intrinsic k-means, and the proposed learning algorithm (Section V). The sparse codes were then classified using a nearest neighbor classifier.

Fig. 7 shows the performances of RSR-J and RSR-S for various dictionary sizes. The red curves show the performance when the intrinsic k-means algorithm was utilized for dictionary learning. The blue curves demonstrate the recognition accuracies when the methods proposed in Section V were used for training, and finally, the green curves show the performance of logEuc-SC equipped with K-SVD [22] algorithm for dictionary learning. The figures show that the proposed dictionary learning approach consistently outperforms k-means bar one case (RSR-J for dictionary size 8). Using the proposed dictionary learning approach, RSR-J achieves the maximum recognition accuracy of 63.3% with 104 atoms, while RSR-S obtains the maximum accuracy of 60.6% with 24 atoms. In contrast, when intrinsic k-means is used for dictionary learning, the maximum recognition accuracies for RSR-J and RSR-S are 52.9% and 53.2%, respectively. Furthermore, in all cases, RSR-S is superior to the log-Euclidean solution, while RSR-J performs better than the log-Euclidean approach only for dictionaries with size larger than 24 atoms.

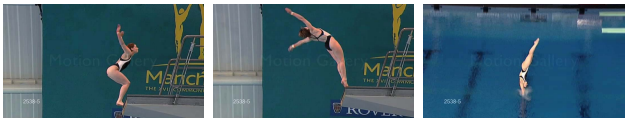


Fig. 8. Examples from the UCF sport action dataset [56].

2) Action Recognition: The UCF sport action dataset [56] consists of ten categories of human actions, including swinging on the pommel horse, driving, kicking, lifting weights, running, skateboarding, swinging at the high bar, swinging golf clubs, and walking (examples of a diving action are shown in Fig. 8). The number of videos for each action varies from 6 to 22 and there are 150 video sequences in total. Furthermore, the videos presented in this dataset have nonuniform backgrounds and both the camera and the subject are moving in some actions. Frames in all video sequences are cropped according to the region of interest provided with the dataset and then resized to 64×64. The standard protocol in this dataset is the leave-one-out (LOO) cross validation [56]–[58].

From each video, we extracted several RCMs by splitting video data into 3-D volumes. The volumes had the size of $32 \times 32 \times 15$ in $x - y - t$ domains with 8 pixels shift in each direction. From each volume, a 12×12 RCM was extracted using kinematic features described in [14]. From training RCMs, we learned separate dictionaries for J and S divergences using the methods described in Section V with 256 atoms each. The dictionaries were then used to determine the video descriptor. To this end, each video was described by simply pooling the sparse codes of its $32 \times 32 \times 15$ volumes

TABLE IV
RECOGNITION ACCURACY (IN %) FOR THE UCF ACTION RECOGNITION DATASET USING HOG3D, HDN [57], AFMKL [58], AND THE PROPOSED RSR-J AND RSR-S APPROACHES

Method	Recognition Accuracy
HOG3D [59]	85.6
HDN [57]	87.3
AFMKL [58]	91.3
logEuc-SC with dic. learning	89.3
RSR-J	90.7
RSR-S	**94.0**

TABLE V
CONFUSION MATRIX (IN %) FOR THE RSR-J METHOD ON THE UCF SPORT ACTION DATASET USING LOO PROTOCOL

	D	GS	K	RH	R	S	PH	HS	W	L
D	**100**	0	0	0	0	0	0	0	0	0
GS	0	**100**	0	0	0	0	0	0	0	0
K	0	0	**100**	0	0	0	0	0	0	0
RH	0	0	0	**100**	0	0	0	0	0	0
R	0	0	0	0	**91.7**	0	0	0	0	8.3
S	0	0	23.1	0	0	**69.2**	0	7.7	0	0
PH	0	16.65	16.65	0	0	0	**25.0**	0	0	41.7
HS	0	0	0	0	0	0	0	**100**	0	0
W	0	0	0	0	0	0	0	0	**100**	0
L	0	0	0	0	0	0	0	0	0	**100**

using maximum operator. Having training and testing descriptors at our disposal, a linear SVM [36] was used as classifier.

In Table IV, the overall performance of the RSR-J and RSR-S methods is compared against three state-of-the-art Euclidean approaches: Histogram of 3D Gradient orientations (HOG3D) [59], hierarchy of discriminative space-time neighborhood (HDN) features [57], and Augmented Features in conjunction with Multiple Kernel Learning (AFMKL) [58]. HOG3D is an extension of histogram of oriented gradient descriptor [60] to spatiotemporal spaces. HDN learns shapes of space-time feature neighborhoods that are most discriminative for a given action category. The idea is to form new features composed of the neighborhoods around the interest points in a video. AFMKL exploits appearance distribution features and spatiotemporal context features in a learning scheme for action recognition. As shown in Table IV, RSR-J outperforms the log-Euclidean approach and is marginally worse than AMFKL. RSR-S achieves the highest overall accuracy.

The confusion matrices for RSR-J and RSR-S divergences are shown in Tables V and VI, respectively. RSR-J perfectly classifies the actions of diving, golf swinging, kicking, riding horse, high bar swinging, and walking and lifting, while RSR-S achieves perfect classification on golf swinging, riding horse, running, high bar swinging, and lifting. Nevertheless, the overall performance of RSR-S surpasses that of RSR-J since RSR-J performs poorly in classifying the pommel-horse action.

VII. MAIN FINDINGS AND FUTURE WORK

With the aim of addressing sparse representation on SPD manifolds, we proposed to seek the solution through embedding the manifolds into RKHS with the aid of two Bregman divergences, namely, Stein and Jeffrey divergences. This led to a relaxed and extended version of the Lasso problem [1] on SPD manifolds.

TABLE VI
CONFUSION MATRIX (IN %) FOR THE RSR-S METHOD ON THE UCF SPORT ACTION DATASET USING LOO PROTOCOL

	D	GS	K	RH	R	S	PH	HS	W	L
D	**92.9**	0	0	0	0	0	0	7.1	0	0
GS	0	**100**	0	0	0	0	0	0	0	0
K	0	0	**95.0**	0	15	0	0	5.0	0	0
RH	0	0	0	**100**	0	0	0	0	0	0
R	0	0	0	0	**100**	0	0	0	0	0
S	7.7	0	15.4	0	0	**69.2**	7.7	0	0	0
P	0	0	0	0	0	0	**83.3**	0	0	16.7
H	0	0	0	0	0	0	0	**100**	0	0
W	7.7	0	0	0	0	0	0	0	**92.3**	0
L	0	0	0	0	0	0	0	0	0	**100**

In Euclidean spaces, the success of many learning algorithms arises from their use of kernel methods [12]. Therefore, one could expect embedding a Riemannian manifold into higher dimensional RKHS, where linear geometry applies, facilitates inference. Such an embedding, however, requires a nontrivial kernel function defined on the manifold, which, according to Mercer's theorem [12], must be pd. The approach introduced here attains its merit from the following facts.

1) By recasting the sparse coding from $\mathcal{S}_{++}^n$ into RKHS, a convex problem is obtained that can be solved quite efficiently. The sparse coding problem is in effect linearized, which is far easier than solving the Riemannian version of sparse coding, as shown in (11).
2) Recasting the sparse coding from $\mathcal{S}_{++}^n$ into RKHS exploits the advantages of higher dimensional Hilbert spaces, such as easier separability of classes.
3) The J and S divergences used in this paper are closely related to the AIRM [8], and have several useful properties such as invariance to inversion and affine transforms. However, unlike AIRM, the J and S divergences admit a Hilbert space embedding (i.e., can be converted to kernel functions).

Experiments on several classification tasks show that the proposed approaches achieve notable improvements in discrimination accuracy, in comparison with state-of-the-art methods such as TSC [21]. We conjuncture that this stems from better exploitation of Riemannian geometry, as both divergences enjoy several properties similar to AIRM on SPD manifolds.

We have furthermore proposed algorithms for learning a dictionary, closely tied to the Stein and Jeffrey divergences. The experiments show that in many cases, better performance is achieved with RSR-S as compared with RSR-J. However, we note that Jeffrey divergence enjoys several unique properties [e.g., closed form solution for averaging and the Hilbert space embedding for all values of β in (6)], which makes it attractive for analyzing SPD matrices. Future venues of exploration include devising other types of inference and machineries based on J and S divergences, such as structured learning.

APPENDIX A
GEOMETRIC MEAN OF J-DIVERGENCE

Theorem 1: For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{S}_{++}^n$, the geometric means of J divergence $\boldsymbol{A} \sharp_J \boldsymbol{B}$ and AIRM $\boldsymbol{A} \sharp_R \boldsymbol{B}$ are the same.

Proof: For the J divergence, we note that

$$\frac{\partial\{J(\boldsymbol{X},\boldsymbol{A})+J(\boldsymbol{X},\boldsymbol{B})\}}{\partial \boldsymbol{X}} = \frac{1}{2}(\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1}-\boldsymbol{X}^{-1}(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{X}^{-1}).$$

Therefore, $\boldsymbol{A} \sharp_J \boldsymbol{B}$ is the solution of

$$\boldsymbol{X}(\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1})\boldsymbol{X} = \boldsymbol{A}+\boldsymbol{B} \tag{28}$$

which is a *Riccati* equation with only one pd solution [38]. We note that

$$\begin{aligned}\boldsymbol{A} \sharp_R \boldsymbol{B} &= \exp_{\boldsymbol{B}}\left(\frac{1}{2}\log_{\boldsymbol{B}}(\boldsymbol{A})\right) = \exp_{\boldsymbol{A}}\left(\frac{1}{2}\log_{\boldsymbol{A}}(\boldsymbol{B})\right) \\ &= \boldsymbol{A}^{\frac{1}{2}}\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{A}^{\frac{1}{2}}. \end{aligned} \tag{29}$$

It can be readily shown that $\boldsymbol{A} \sharp_R \boldsymbol{B}$ satisfies (28), which concludes the proof. □

APPENDIX B
KERNELIZED FEATURE SIGN ALGORITHM

The efficiency of the FSSA [33] for finding sparse codes in vector spaces has been analyzed in [61]. The algorithm was shown to outperform (in terms of speed and accuracy) sparse solvers such as the generic QP solver [40]. The gain is even higher for large and high-dimensional datasets (which are common in computer vision tasks). As such, we have elected to recast the algorithm into its RKHS version, to find the sparse codes on SPD manifolds. We summarize the new version below, with the pseudocode shown in Algorithm 1.

Given the objective function defined in (12), if the signs (positive, zero, or negative) of the y_j are known at the optimal value, each term of $\|\boldsymbol{y}\|_1 = \sum_{j=1}^N |y_i|$ can be replaced by either y_i, $-y_i$, or 0. Considering only nonzero coefficients, this reduces (12) to a standard, unconstrained quadratic optimization problem (QP) [62], which has an analytic solution.

The feature-sign algorithm comprises of four basic steps. The first two steps can be considered as a greedy search. The search is directed toward selecting a new feature that maximizes the rate of decrements in (12). This is accomplished by computing the first-order derivative of (12) with respect to features

$$\frac{\partial}{\partial y_j} l_\phi(\boldsymbol{X}, \mathbb{D}) = \sum_{q=1}^{N} y_q k(\boldsymbol{D}_q, \boldsymbol{D}_j) - 2k(\boldsymbol{X}, \boldsymbol{D}_j) + \lambda. \tag{30}$$

Since an estimate of the feature signs is available, in the third step (feature-sign step), it is possible to find the solution of the unconstrained QP of $\min_{\widehat{\boldsymbol{y}}} \widetilde{f}(\widehat{\boldsymbol{y}})$, where

$$\begin{aligned}\widetilde{f}(\widehat{\boldsymbol{y}}) &= \Big\|\phi(\boldsymbol{X}) - \sum\nolimits_{i\in \text{active\_set}} y_i \phi(\boldsymbol{D}_i)\Big\|^2 + \lambda \widehat{\boldsymbol{\theta}}^T \widehat{\boldsymbol{y}} \\ &= k(\boldsymbol{X},\boldsymbol{X}) - 2\widehat{\boldsymbol{y}}^T \widehat{\boldsymbol{\mathcal{K}}} + \widehat{\boldsymbol{y}}^T \widehat{\mathbb{K}} \widehat{\boldsymbol{y}} + \lambda \widehat{\boldsymbol{\theta}}^T \widehat{\boldsymbol{y}}. \end{aligned} \tag{31}$$

In (31), $\widehat{\boldsymbol{y}} = \boldsymbol{y}(\text{active\_set})$, $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}(\text{active\_set})$ and $\widehat{\boldsymbol{\mathcal{K}}} = \boldsymbol{\mathcal{K}}(\text{active\_set})$ are subvectors corresponding to active_set. Similarly, $\widehat{\mathbb{K}} = \mathbb{K}(\text{active\_set}, \text{active\_set})$ is a submatrix corresponding to active_set with active_set being the subset of selected features. The closed form solution, i.e., $\widehat{\mathbb{K}}^{-1}(\widehat{\boldsymbol{\mathcal{K}}} - \lambda\widehat{\boldsymbol{\theta}}/2)$, can be derived by computing the root of first order derivative of (31) with respect to $\widehat{\boldsymbol{y}}$. The final step of the algorithm is an optimality check to verify that the features and the corresponding signs are truly consistent with the objective function in (12).

The convergence of the feature sign algorithm has been discussed for Euclidean spaces in [33] and can be readily extended to the kernelized case.

Algorithm 1: Kernel Feature-Sign Search, for Finding Sparse Codes in Reproducing Kernel Hilbert Spaces

Input: query $\boldsymbol{X} \in \mathcal{S}_{++}^n$; dictionary $\{\boldsymbol{D}_i\}_{i=1}^N$, $\boldsymbol{D}_i \in \mathcal{S}_{++}^n$; kernel function $k : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \to \mathbb{R}$.
Output: sparse code $\boldsymbol{y}$.

Initialisation.
- *active_set* $\leftarrow \{\}$, $\boldsymbol{y}_{N\times 1} \leftarrow \boldsymbol{0}$, $\boldsymbol{\theta}_{N\times 1} \leftarrow \boldsymbol{0}$

Processing.

1. Feature selection.
From the zero coefficients of $\boldsymbol{y}$, select $i = \arg\max_i \left|\sum_{q=1}^N y_q k(\boldsymbol{D}_q, \boldsymbol{D}_i) - 2k(\boldsymbol{X}, \boldsymbol{D}_i)\right|$.

2. Feature evaluation.
if $\left|\sum_{q=1}^N y_q k(\boldsymbol{D}_q, \boldsymbol{D}_i) - 2k(\boldsymbol{X}, \boldsymbol{D}_i)\right| > \lambda$ **then**
- Add i to *active_set*, i.e., , *active_set* $\leftarrow$ *active_set* $\cup \{i\}$
- Update the sign vector, i.e., ,
- $\theta_i := -\operatorname{sign}\left(\sum_{q=1}^N y_q k(\boldsymbol{D}_q, \boldsymbol{D}_i) - 2k(\boldsymbol{X}, \boldsymbol{D}_i) - \lambda\right)$

3. Feature-sign step.
- $\widehat{\boldsymbol{y}} \leftarrow \boldsymbol{y}(\text{active\_set})$, $\widehat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}(\text{active\_set})$, $\widehat{\boldsymbol{\mathcal{K}}} \leftarrow \boldsymbol{\mathcal{K}}(\text{active\_set})$
- $\widehat{\mathbb{K}} \leftarrow \mathbb{K}(\text{active\_set}, \text{active\_set})$
- $\widehat{\boldsymbol{y}}_{new} \leftarrow \widehat{\mathbb{K}}^{-1}(\widehat{\boldsymbol{\mathcal{K}}} - \frac{\lambda}{2}\widehat{\boldsymbol{\theta}})$, ,i.e., , the closed form solution to $\min_{\boldsymbol{y}} \|\phi(\boldsymbol{X}) - \sum_{i\in \text{active\_set}} y_i \phi(\boldsymbol{D}_i)\|^2 + \lambda \widehat{\boldsymbol{Y}}\widehat{\boldsymbol{\theta}}$.
- **if** $\widehat{\boldsymbol{y}}_{new}$ *is consistent with the current active_set* **then**
  - $\widehat{\boldsymbol{y}} = \widehat{\boldsymbol{y}}_{new}$.
- **else** perform a discrete line search on the closed line segment from $\widehat{\boldsymbol{y}}$ to $\widehat{\boldsymbol{y}}_{new}$:
  - Check the objective value at all points where any coefficient changes sign
  - Update $\widehat{\boldsymbol{y}}$ (and the corresponding entries in $\boldsymbol{y}$) to the point with the lowest objective value.
- Remove zero coefficients of $\widehat{\boldsymbol{y}}$ from *active_set*
- $\boldsymbol{\theta} \leftarrow \operatorname{sign}(\boldsymbol{y})$

4. Check optimality conditions.
- **if** $\exists y_i \neq 0$ s.t. $\sum_{q=1}^N y_q k(\boldsymbol{D}_q, \boldsymbol{D}_i) - 2k(\boldsymbol{X}, \boldsymbol{D}_i) + \lambda \operatorname{sign}(y_i) \neq 0$ **then**
  - Jump to **Feature selection.**
- **else if** $\exists y_i = 0$ s.t. $\left|\sum_{q=1}^N y_q k(\boldsymbol{D}_q, \boldsymbol{D}_i) - 2k(\boldsymbol{X}, \boldsymbol{D}_i)\right| > \lambda$ **then**
  - Jump to **Feature-sign step.**

REFERENCES

[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Heidelberg, Germany: Springer-Verlag, 2010.

[2] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[4] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, Graz, Austria, May 2006, pp. 589–600.

[5] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy, Oct. 2012, pp. 216–229.

[6] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell, "Spatio-temporal covariance descriptors for action and gesture recognition," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Clearwater Beach, FL, USA, Jan. 2013, pp. 103–110.

[7] Y. Pang, Y. Yuan, and X. Li, "Gabor-based region covariance matrices for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 989–993, Jul. 2008.

[8] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, 2006.

[9] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[10] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on the Riemannian manifold of symmetric positive definite matrices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 73–80.

[11] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista, "Semi-intrinsic mean shift on Riemannian manifolds," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy, Oct. 2012, pp. 342–355.

[12] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge Univ. Press, 2004.

[13] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability measures," in *Proc. Int. Conf. Artif. Intell. Statist.*, Barbados, Island, Jan. 2005, pp. 136–143.

[14] K. Guo, P. Ishwar, and J. Konrad, "Action recognition from video using feature covariance matrices," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2479–2494, Jun. 2013.

[15] S. Sra and A. Cherian, "Generalized dictionary learning for symmetric positive definite matrices with application to nearest neighbor retrieval," in *Proc. Eur. Conf. Mach. Learn.*, Athens, Greece, Sep. 2011, pp. 318–332.

[16] J. Ho, Y. Xie, and B. Vemuri, "On a nonlinear generalization of sparse coding and dictionary learning," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, Jun. 2013, pp. 1480–1488.

[17] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo, "Human action recognition under log-Euclidean Riemannian metric," in *Proc. 9th Asian Conf. Comput. Vis. (ACCV)*, Xi'an, China, Sep. 2009, pp. 343–353.

[18] M. Faraki, M. Palhang, and C. Sanderson, "Log-Euclidean bag of words for human action recognition," *IET Comput. Vis.*, to be published. [Online]. Available: http://dx.doi.org/10.1049/iet-cvi.2014.0018

[19] J. M. Lee, *Introduction to Smooth Manifolds* (Graduate Texts in Mathematics). Heidelberg, Germany: Springer-Verlag, 2012.

[20] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Positive definite dictionary learning for region covariances," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 1013–1019.

[21] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Tensor sparse coding for positive definite matrices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 592–605, Mar. 2014.

[22] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[23] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *J. Mach. Learn. Res.*, vol. 10, pp. 341–376, Feb. 2009.

[24] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Jensen–Bregman LogDet divergence with application to efficient similarity search for covariance matrices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2161–2174, Sep. 2013.

[25] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric means," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, Ed. Red Hook, NY, USA: Curran Associates, 2013, pp. 144–152.

[26] Z. Wang and B. C. Vemuri, "An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Washington, DC, USA, Jun./Jul. 2004, pp. I-228–I-233.

[27] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 17–32.

[28] C. van den Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups*. New York, NY, USA: Springer-Verlag, 1984.

[29] B. Schölkopf, "The kernel trick for distances," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001, pp. 301–307.

[30] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback–Leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA, USA: MIT Press, 2004, pp. 1385–1392.

[31] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *J. Mach. Learn. Res.*, vol. 10, pp. 747–776, Mar. 2009.

[32] M. Harandi, C. Sanderson, C. Shen, and B. Lovell, "Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 3120–3127.

[33] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA, USA: MIT Press, 2007, pp. 801–808.

[34] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, vol. 6314. Heraklion, Greece, Sep. 2010, pp. 1–14.

[35] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.

[36] C. M. Bishop, *Pattern Recognition and Machine Learning*. Heidelberg, Germany: Springer-Verlag, 2006.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[38] R. Bhatia, *Positive Definite Matrices*. Princeton, NJ, USA: Princeton Univ. Press, 2007.

[39] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear Algebra Appl.*, vol. 103, pp. 103–118, May 1988.

[40] M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, Version 2.0 Beta*. [Online]. Available: http://cvxr.com/cvx, accessed Dec. 18, 2014.

[41] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3025–3032.

[42] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation: Mocap database HDM05," Dept. Comput. Graph., Univ. Bonn, Bonn, Germany, Tech. Rep. CG-2007-2, 2007.

[43] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, Beijing, China, Aug. 2013, pp. 2466–2472.

[44] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[45] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, vol. 6316. Heraklion, Greece, Sep. 2010, pp. 448–461.

[46] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *J. Vis.*, vol. 9, no. 8, 2009, Art. ID 784.

[47] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[48] Z. Liao, J. Rock, Y. Wang, and D. Forsyth, "Non-parametric filtering for geometric detail extraction and material representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 963–970.

[49] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a Bayesian framework for material recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 239–246.

[50] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2032–2047, Nov. 2009.

[51] W. R. Schwartz and L. S. Davis, "Learning discriminative appearance-based models using partial least squares," in *Proc. 22nd Brazilian Symp. Comput. Graph. Image Process.*, Rio de Janeiro, Brazil, Oct. 2009, pp. 322–329.
[52] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.
[53] L. Bazzani, M. Cristani, and V. Murino, "Symmetry-driven accumulation of local features for human characterization and re-identification," *Comput. Vis. Image Understand.*, vol. 117, no. 2, pp. 130–144, 2013.
[54] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell, "Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Breckenridge, CO, USA, Jan. 2012, pp. 433–439.
[55] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.
[56] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.
[57] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2046–2053.
[58] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, Jun. 2011, pp. 489–496.
[59] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., Sep. 2009, pp. 124.1–124.11.
[60] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.
[61] H. Lee, "Unsupervised feature learning via sparse hierarchical representations," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2010.
[62] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

Mehrtash T. Harandi (M'10) received the B.Sc. degree in electronics from the Sharif University of Technology, Tehran, Iran, and the M.Sc. and Ph.D. degrees in computer science from the University of Tehran, Tehran.

He is currently a Senior Researcher with the Computer Vision Research Group, NICTA, Canberra Research Laboratory, Canberra, Australia. His current research interests include theoretical and computational methods in computer vision and machine learning with a focus on Riemannian geometry.

Richard Hartley (F'10) is a member of the computer vision group in the Research School of Engineering, Australian National University, where he has been since January, 2001. He is also a member of the computer vision research group in NICTA, Canberra Research Laboratory, Canberra, Australia. Dr. Hartley worked at the General Electric (GE) Research and Development Center from 1985 to 2001, working first in VLSI design, and later in computer vision. He became involved with Image Understanding and Scene Reconstruction working with GE's Simulation and Control Systems Division. He has authored a book entitled *Multiple View Geometry in Computer Vision* (with A. Zisserman).

Brian Lovell (SM'05) received the B.Eng. degree in electrical engineering, the B.Sc. degree in computer science, and the Ph.D. degree in signal processing from the University of Queensland (UQ), Brisbane, QLD, Australia, in 1982, 1983, and 1991, respectively.

He was a Research Leader with NICTA, Canberra Research Laboratory, Canberra, Australia, and the Research Director of the Surveillance Research Group, UQ.

Prof. Lovell serves on the Editorial Board of *Pattern Recognition Letters* and as a reviewer for many of the major journals in Computer Vision.

Conrad Sanderson (M'01) received the Ph.D. degree from Griffith University, Nathan, QLD, Australia, in 2003, and the M.B.A. degree from the University of Queensland, Brisbane, QLD, Australia, in 2012.

He has been involved in speech recognition and language translation with NICTA, Queensland Research Laboratory, QLD, Australia, in audio-visual biometrics with the IDIAP Research Institute, Martigny, Switzerland, in computer vision for military applications with the University of Adelaide, Adelaide, SA, Australia, and in natural language processing, bioinformatics, and automated surveillance with NICTA, Queensland Research Laboratory, QLD, Australia. His work on innovative surveillance technologies has led to several industry awards. He is currently a Research Leader with National ICT Australia Ltd. His current research interests include adaptation and commercialization of research outcomes toward industrial use, and machine learning, pattern recognition, and predictive analytics.

Dr. Sanderson has served as a reviewer for major international conferences and scientific journals.