Riemannian joint dimensionality reduction and dictionary learning on symmetric positive definite manifolds

Hiroyuki Kasai
Graduate School of Informatics and Engineering
The University of Electro-Communications
Tokyo, Japan
kasai@is.uec.ac.jp

Bamdev Mishra
Microsoft
Hyderabad, India
bamdevm@microsoft.com

***Abstract*—Dictionary leaning (DL) and dimensionality reduction (DR) are powerful tools to analyze high-dimensional noisy signals. This paper presents a proposal of a novel Riemannian joint dimensionality reduction and dictionary learning (R-JDRDL) on symmetric positive definite (SPD) manifolds for classification tasks. The joint learning considers the interaction between dimensionality reduction and dictionary learning procedures by connecting them into a unified framework. We exploit a Riemannian optimization framework for solving DL and DR problems jointly. Finally, we demonstrate that the proposed R-JDRDL outperforms existing state-of-the-arts algorithms when used for image classification tasks.**

***Index Terms*—dictionary leaning, dimensionality reduction, SPD matrix, Riemannian manifold**

I. Introduction

Dictionary leaning (DL) combined with sparse representation (SR) has become popular for many computer vision tasks. Many DL algorithms, e.g., K-SVD [1], were applied originally for unsupervised learning tasks. Recently, some supervised DL algorithms have been proposed for classification tasks which exploit *class label* information in the training samples. They include D-KSVD [2] and LC-KSVD [3], to name a few. However, DL for high-dimensional data is computationally expensive. To circumvent this issue, dimensionality reduction (DR) techniques are used which reduce the computational cost and highlight the low-dimensional discriminative feature of the data.

In general, DR is applied first to the data samples, and then the dimensionality-reduced data are used for DL. The separately pre-learned DR projection matrix, however, does not fully promote the latent structure of data or preserve the best feature for DL [4]. To address this issue, Feng et al. [5] have proposed integration of DL and DR for improvement of the discriminative classification performance, in which a specific constraint similar to the Fisher linear discriminative analysis is imposed on the coefficient matrix. Similarly, Yang et al. [6] propose learning of the projection matrix and class-specific dictionary jointly. Li et al. [7] report an integrated learning method of the non-negative projection matrix. Foroughi et al. [8] discuss specific constraints on the coefficient matrix and on the projection matrix.

In many computer vision tasks, data of interest often reside on a *manifold*, which is a generalization of the Euclidean space. A particular manifold of interest is the manifold of *symmetric positive definite* (SPD) matrices that has been widely used in many applications. For example, region covariance matrices (RCM), which are symmetric positive definite, give good performance in texture classification and face recognition tasks [9], [10]. The diagonal elements of a RCM represent the variances of coponent features, and the off-diagonal elements indicate the respective correlations among them. Therefore, the RCM can represent multiple features in a natural way. It should be noted that the SPD matrices form a *Riemannian manifold*, which allows to understand the geometry of the space [11]. Cherian and Sra [12] exploit the manifold structure to propose a Riemannian DL and sparse coding (SC) algorithm. Separately, the Riemannian DR techniques have been proposed in several works [13]–[16].

In this paper, our main contribution is to learn DL and DR jointly in the Riemannian framework. We propose R-JDRDL, an algorithm for jointly learning the projection matrix for DR and the discriminative dictionary on the SPD matrices for classification tasks. The joint learning considers the interaction between DR and DL procedures by connecting them into a unified framework. The model is formulated as an objective function over a sparse coefficient matrix and a *Cartesian product manifold* that consists of the Stiefel manifold and multiple SPD manifolds. Optimization on the Cartesian product manifold is cast as an optimization problem on Riemannian manifolds [17]. Optimization on the sparse coefficient matrix, on the other hand, is a convex program.

This paper is organized as follows. Section II briefly introduces the SPD manifold and the Riemannian DL. Section III details the proposed R-JDRDL algorithm. Our initial results on the MNIST image classification task in Section IV show that R-JDRDL outperforms state-of-the-art algorithms in the domain.

ISBN 978-90-827970-1-5

II. SPD MANIFOLD AND RIEMANNIAN DL

This section briefly explains the geometry of SPD manifold and then introduces the Riemannian DL. Hereinafter, we denote the scalars with lower-case letters $(a, b, \ldots)$, vectors with bold lower-case letters $(\boldsymbol{a}, \boldsymbol{b}, \ldots)$, and matrices with boldface capitals $(\mathbf{A}, \mathbf{B}, \ldots)$. We denote a multidimensional or multi-*order* array as a *tensor*, which is denoted by $(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \ldots)$.

A. Geometry of SPD manifold [11]

A manifold $\mathcal{M}$ of dimensional d is a topological space that locally resembles the Euclidean space $\mathbb{R}^d$ in a neighborhood of each point $\mathbf{X} \in \mathcal{M}$. All the tangent vectors at $\mathbf{X}$ form a vector space called the tangent space of $\mathcal{M}$ at $\mathbf{X}$ and denoted as $T_{\mathbf{X}}\mathcal{M}$. When endowed with a smoothly defined metric, i.e., inner product $\langle \cdot, \cdot \rangle_{\mathbf{X}}$ between vectors in the tangent space at $\mathbf{X} \in \mathcal{M}$, the manifold $\mathcal{M}$ is called a Riemannian manifold. The space of $d \times d$ SPD matrices, denoted as $\mathcal{S}_{++}^d$, is a Riemannian manifold, called *SPD manifold*, when endowed with an appropriate *Riemannian metric*. The tangent space at any point on $\mathcal{S}_{++}^d$ is identifiable with the set symmetric matrices $\mathcal{S}^d$.

One particular choice of the Riemannian metric on the SPD manifold is the affine-invariant Riemannian metric (AIRM) [11], [18]. If $\mathbf{P}$ is an element on $\mathcal{S}_{++}^d$, the AIRM is defined as

$$\langle \mathbf{V}, \mathbf{W} \rangle_{\mathbf{P}} := \langle \mathbf{P}^{-1/2}\mathbf{V}\mathbf{P}^{-1/2}, \mathbf{P}^{-1/2}\mathbf{W}\mathbf{P}^{-1/2} \rangle,$$

where $\mathbf{V}, \mathbf{W} \in T_{\mathbf{P}}\mathcal{S}_{++}^d$. The choice of metric does not change with affine action by $\mathrm{GL}(d)$, which means that $[\mathbf{X} \rightarrow \mathbf{M}\mathbf{X}\mathbf{M}^T, \mathbf{X} \in \mathcal{S}_{++}^d, {}^\forall\mathbf{M} \in \mathrm{GL}(d)]$ on $\mathbf{V}, \mathbf{W}$ and $\mathbf{P}$. The Riemannian metric provides a way to compute the distance between two points on the manifold. Because the SPD manifold with the AIRM metric has a unique shortest path, which is called *geodesic*, between every two points [11, Section 6], the geodesic distance $d : \mathcal{S}_{++}^n \times \mathcal{S}_{++}^n \rightarrow [0, \infty]$ is given as

$$d^2(\mathbf{A}, \mathbf{B}) := \mathrm{Log}\|\mathbf{A}^{-1/2}\mathbf{B}\mathbf{A}^{-1/2}\|_F^2,$$

where $\mathbf{A}, \mathbf{B} \in \mathcal{S}_{++}^n$, $\|\cdot\|_F$ denotes the Frobenius norm, and Log denotes the matrix logarithm.

B. Riemannian DL (R-DL)

Let $\boldsymbol{\mathcal{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\} \in \mathbb{R}^{d \times d \times N}$ be the input training sample set, where $\mathbf{X}_n$ denotes n-th sample that forms a SPD matrix $\mathbf{X}_n \in \mathcal{S}_{++}^d$. The dictionary to be learned is denoted as $\boldsymbol{\mathcal{D}} = \{\mathbf{D}_1, \ldots, \mathbf{D}_H\} \in \prod^H \mathcal{S}_{++}^d$, where $\mathbf{D}_h \in \mathcal{S}_{++}^d$ is an atom of the dictionary. It should be noted that $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{D}}$ are third-order tensors. We also denote a sparse coefficient vector as $\boldsymbol{a}_n \in \mathbb{R}_+^H$, which forms a coefficient matrix $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N] \in \mathbb{R}_+^{H \times N}$, to represent a query SPD matrix $\mathbf{X}_n$ using the dictionary $\boldsymbol{\mathcal{D}}$. It should also be emphasized that $\boldsymbol{a}_n$ is required to be *non-negative* to ensure that the resultant combination with the dictionary is positive definite. Therefore, we specifically represent a sparse *conic combination* of the dictionary and the coefficient vector as $\boldsymbol{\mathcal{D}} \otimes \boldsymbol{a}_n := \sum_{h=1}^H \boldsymbol{a}_{n,h}\mathbf{D}_h$ for $\boldsymbol{a}_{n,h} * \in \mathbb{R}_+^H$. Finally, the problem formulation is defined as

$$\min_{\boldsymbol{\mathcal{D}} \in \prod^H \mathcal{S}_{++}^d, \mathbf{A} \in \mathbb{R}_+^{H \times N}} \frac{1}{2} \sum_{n=1}^N d^2(\mathbf{X}_n, \boldsymbol{\mathcal{D}} \otimes \boldsymbol{a}_n) + R_a(\boldsymbol{a}_n) + R_D(\boldsymbol{\mathcal{D}}),$$

where $R_a(\boldsymbol{a}_n)$ and $R_D(\boldsymbol{\mathcal{D}})$ respectively represent the regularizers on the coefficient vector and the dictionary [12]. To optimize this non-convex problem, an alternative minimization algorithm is used for the DL and the SC sub-problems.

III. R-JDRDL ON SPD MANIFOLDS

A. Problem formulation of R-JDRDL

Let $\boldsymbol{\mathcal{X}}$ be the set of N SPD matrices of size $m \times m$ accompanied with K class labels, i.e., $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_k, \ldots, \boldsymbol{\mathcal{X}}_K\} \in \mathbb{R}^{m \times m \times N}$, where $\boldsymbol{\mathcal{X}}_k$ denotes the k-th class training samples. $\boldsymbol{\mathcal{X}}_k$ is further composed of individual samples as $\boldsymbol{\mathcal{X}}_k = \{\mathbf{X}_{k,1}, \ldots, \mathbf{X}_{k,n}, \ldots, \mathbf{X}_{k,N_k}\}$, where $\mathbf{X}_{k,n} \in \mathcal{S}_{++}^m$ and N_k is the number of samples of the k-th class in the training set, i.e., $\sum_{k=1}^K N_k = N$. Both $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}_k$ are third-order tensors. The dictionary is denoted as $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_k, \ldots, \boldsymbol{\mathcal{D}}_K\}$, where $\boldsymbol{\mathcal{D}}_k$ is the class-specific sub-dictionary associated with the k-th class. $\boldsymbol{\mathcal{D}}_k$ is also composed as $\boldsymbol{\mathcal{D}}_k = \{\mathbf{D}_{k,1}, \ldots, \mathbf{D}_{k,h}, \ldots, \mathbf{D}_{k,H_k}\}$, where H_k is the number of atoms of the k-th class sub-dictionary, and $\sum_{k=1}^K H_k = H$.

As described earlier, the proposed R-JDRDL algorithm learns not only the dictionary $\boldsymbol{\mathcal{D}}$, but also the projection matrix $\mathbf{U} \in \mathbb{R}^{m \times d} (d < m)$, which projects m-dimensional data onto d-dimensional data space. More specifically, $\mathbf{X}_{k,n} \in \mathcal{S}_{++}^m$ is mapped into $\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U} \in \mathcal{S}_{++}^d$. Here, we need only *full-rankness* of $\mathbf{U}$ to guarantee that $\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}$ is a SPD matrix. Equivalently, we could enforce a *unitary constraint* on $\mathbf{U}$, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The space of unitary matrices is called the *Stiefel manifold* $\mathrm{St}(d, m) := \{\mathbf{U} \in \mathbb{R}^{m \times d} : \mathbf{U}^T\mathbf{U} = \mathbf{I}\}$.

Considering that model parameters are $(\mathbf{U}, \boldsymbol{\mathcal{D}}) \in \mathcal{N}$ and $\mathbf{A} \in \mathbb{R}_+^{H \times N}$, where $\mathcal{N}$ denotes the space of the product manifold $\{\mathrm{St}(d, m) \times \prod^H \mathcal{S}_{++}^d\}$, our proposed formulation is

$$\begin{aligned} \{\hat{\mathbf{U}}, \hat{\boldsymbol{\mathcal{D}}}, \hat{\mathbf{A}}\} = \underset{(\mathbf{U}, \boldsymbol{\mathcal{D}}) \in \mathcal{N}, \mathbf{A} \in \mathbb{R}_+^{H \times N}}{\operatorname{argmin}} & J_d(\mathbf{U}, \boldsymbol{\mathcal{D}}, \mathbf{A}) \\ & + \lambda_a J_a(\mathbf{A}) + \lambda_u J_u(\mathbf{U}) \\ & + \lambda_1 R_s(\mathbf{A}) + \lambda_2 R_r(\mathbf{A}) + \lambda_d R_d(\boldsymbol{\mathcal{D}}), \end{aligned} \quad (1)$$

where $J_d(\mathbf{U}, \boldsymbol{\mathcal{D}}, \mathbf{A})$ is the discriminative reconstruction error and where $J_a(\mathbf{A})$ and $J_u(\mathbf{U})$ represent the graph-based constraints on the coefficient and the projection matrices, respectively. $R_s(\mathbf{A}) = \mathbf{1}_H^T|\mathbf{A}|\mathbf{1}_N$ $(:= \sum_{k=1}^K \sum_{n=1}^{N_k} \|\boldsymbol{a}_{k,n}\|_1)$, which imposes sparsity on $\mathbf{A}$. $R_r(\mathbf{A}) = \|\mathbf{A}\|_F^2$. λs are non-negative regularization parameters. J_d, J_u, and J_a are described below.

Discriminative reconstruction error term J_d: The dictionary $\boldsymbol{\mathcal{D}}$ is expected to approximate the dimensionality-reduced samples from all classes, of which error is represented as $d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}} \otimes \boldsymbol{a}_{k,n})$, where d is the Riemannian geodesic distance on the SPD manifold. In addition, to impose a more

ISBN 978-90-827970-1-5 © EURASIP 2018

discriminative power on $\boldsymbol{\mathcal{D}}$, the k-th sub-dictionary $\boldsymbol{\mathcal{D}}_k$ is expected to approximate the dimensionality-reduced training samples associated with the k-th class. Here, let $\boldsymbol{a}_{k,n}^k$ be the sub-vector that corresponds to the k-th sub-dictionary as $\boldsymbol{a}_{k,n} = [\boldsymbol{a}_{k,n}^1; \ldots; \boldsymbol{a}_{k,n}^k; \ldots; \boldsymbol{a}_{k,n}^K]$, where $\boldsymbol{a}_{k,n}^k \in \mathbb{R}^{H_k}$. The error is equivalent to $d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}}_k \otimes \boldsymbol{a}_{k,n}^k)$. It should be small. The sub-vector $\boldsymbol{a}_{k,n}^j (j \neq k)$ corresponding to other classes should be nearly zero, such that $\|\boldsymbol{\mathcal{D}}_j \otimes \boldsymbol{a}_{k,n}^j\|_F^2$ is small. Consequently, we obtain the cost function for J_d as

$$
\begin{aligned}
&J_d(\mathbf{U}, \boldsymbol{\mathcal{D}}, \mathbf{A}) \\
&\coloneqq \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N_k}(d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}} \otimes \boldsymbol{a}_{k,n}) \\
&+ d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}}_k \otimes \boldsymbol{a}_{k,n}^k)) \\
&+ \lambda_d \sum_{j=1, j\neq k}^{K}\sum_{n=1}^{N_k} \|\boldsymbol{\mathcal{D}}_j \otimes \boldsymbol{a}_{k,n}^j\|_2^2,
\end{aligned}
$$

$\lambda_d > 0$ is the regularization parameter.

Graph-based coefficient term J_a: We enforce $\mathbf{A}$ to be more discriminative, and therefore, we seek to constrain the intra-class coefficients to be mutually similar and the inter-class ones to be highly dissimilar. To this end, we first construct an geometry-aware intrinsic graph of intra-class and a penalty graph for inter-class discrimination for two points $\mathbf{X}_p, \mathbf{X}_q \in \mathcal{S}_{++}^m$ as

$$
\mathbf{G}_{bin}^w(p,q) = \begin{cases} 1 & \text{if } \mathbf{X}_p \in N_w(\mathbf{X}_q) \text{ or } \mathbf{X}_p \in N_w(\mathbf{X}_q) \\ 0 & \text{otherwise,} \end{cases}
$$

$$
\mathbf{G}_{bin}^b(p,q) = \begin{cases} 1 & \text{if } \mathbf{X}_p \in N_b(\mathbf{X}_q) \text{ or } \mathbf{X}_p \in N_b(\mathbf{X}_q) \\ 0 & \text{otherwise,} \end{cases}
$$

where $N_w(\mathbf{X})$ is the set of v_w nearest intra-class neighbors of $\mathbf{X}$ in terms of geodesic distance. Similarly, $N_b(\mathbf{X})$ is the set of v_b nearest inter-class neighbors of $\mathbf{X}$. Considering the distance of pairs of coding coefficient vectors $\boldsymbol{a}_p$ and $\boldsymbol{a}_q$ as an indicator of discrimination capability, the final graph-based coefficient term $J_a(\mathbf{A})$ is defined as

$$
J_a(\mathbf{A}) \;\coloneqq\; \sum_{p=1}^{N}\sum_{q=1}^{N}\frac{1}{2}\|\boldsymbol{a}_p - \boldsymbol{a}_q\|_2^2 \; \mathbf{G}_{bin}(p,q),
$$

where $\mathbf{G}_{bin}(p,q) = \mathbf{G}_{bin}^w(p,q) - \mathbf{G}_{bin}^b(p,q)$ [13]. This term enforces minimization of the difference of the two coding coefficients if they are the same class, although the difference of the code is maximized if they are from different classes.

Graph-based projection term J_u: We also learn a projection matrix $\mathbf{U} \in \mathrm{St}(d,m)$ that can preserve class information and which can map the training samples to a low-dimensional discriminative space. Consequently, $J_u(\mathbf{U})$ is defined as

$$
J_u(\mathbf{U}) \;\coloneqq\; \sum_{p=1}^{N}\sum_{q=1}^{N}\frac{1}{2}d^2(\mathbf{U}^T\mathbf{X}_p\mathbf{U}, \mathbf{U}^T\mathbf{X}_q\mathbf{U}) \; \mathbf{G}_{rd}(p,q),
$$

where the affinity matrix $\mathbf{G}_{rd}$ allows to assign different weights to the Riemannian distance between different points, e.g., the distance $d(\mathbf{X}_p, \mathbf{X}_q)$ is assigned the weight $\mathbf{G}_{rd}(p,q)$.

B. Optimization of R-JDRDL

The objective function of (1) is divided into two sub-problems, which are solved in alternating fashion. We discuss both the sub-problems below.

DL sub-problem on the product manifold: We consider the DL sub-problem of (1) by optimizing the projection matrix $\mathbf{U}$ and the tensor-formed dictionary $\boldsymbol{\mathcal{D}}$, keeping $\mathbf{A}$ fixed to $\hat{\mathbf{A}} = (\hat{\boldsymbol{a}}_{k,n})$. Consequently, the problem is can be re-formulated as

$$
\begin{aligned}
&\min_{(\mathbf{U},\boldsymbol{\mathcal{D}})\in\mathcal{N}} f(\mathbf{U}, \boldsymbol{\mathcal{D}}) \coloneqq J_d(\mathbf{U}, \boldsymbol{\mathcal{D}}, \hat{\mathbf{A}}) + \lambda_u J_u(\mathbf{U}) + \lambda_d R_d(\boldsymbol{\mathcal{D}}) \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N_k}(d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}} \otimes \hat{\boldsymbol{a}}_{k,n}) \\
&+ d^2(\mathbf{U}^T\mathbf{X}_{k,n}\mathbf{U}, \boldsymbol{\mathcal{D}}_k \otimes \hat{\boldsymbol{a}}_{k,n}^k)) \\
&+ \lambda_{da} \sum_{j=1, j\neq k}^{K}\sum_{n=1}^{N_k} \|\boldsymbol{\mathcal{D}}_j \otimes \hat{\boldsymbol{a}}_{k,n}^j\|_2^2 \\
&+ \lambda_u \sum_{p=1}^{N}\sum_{q=1}^{N}\frac{1}{2}d^2(\mathbf{U}^T\mathbf{X}_p\mathbf{U}, \mathbf{U}^T\mathbf{X}_q\mathbf{U})\mathbf{G}_{dr}(p,q) \\
&+ \lambda_d R_d(\boldsymbol{\mathcal{D}}).
\end{aligned}
$$

We exploit the Riemannian optimization framework on the Cartesian product manifold $\mathcal{N}$ (consisting of the Stiefel manifold and multiple SPD manifolds). In particular, we use the Riemannian conjugate gradient (RCG) method for solving the DL sub-problem. Theoretical convergence of the Riemannian algorithms is to a stationary point. The convergence analysis follows from [19], [20]. To this end, we require the expression for the Riemannian gradient. According to [12], the Riemannian gradient is obtained as $\mathrm{grad} f(\mathbf{U}, \boldsymbol{\mathcal{D}}) = \mathbf{D}_{k,h}\mathrm{egrad} f(\mathbf{U}, \boldsymbol{\mathcal{D}})\mathbf{D}_{k,h}$ with respect to $\mathbf{D}_{k,h}$ from the definition of AIRM where $\mathrm{egrad} f(\mathbf{U}, \boldsymbol{\mathcal{D}})$ is the Euclidean gradient of $f(\mathbf{U}, \boldsymbol{\mathcal{D}})$ with respect to $\mathbf{D}_{k,h}$.

SC sub-problem: We consider the SC sub-problem of (1) for solving $\mathbf{A}$, keeping $\mathbf{U}$ and $\boldsymbol{\mathcal{D}}$ fixed to $\hat{\mathbf{U}}$ and $\hat{\boldsymbol{\mathcal{D}}}$, respectively. The problem, therefore, can be re-formulated as

$$
\begin{aligned}
&\min_{\mathbf{A}\in\mathbb{R}_+^{H\times N}} \Psi(\mathbf{A}) \coloneqq \\
&J_d(\hat{\mathbf{U}}, \hat{\boldsymbol{\mathcal{D}}}, \mathbf{A}) + \lambda_a J_a(\mathbf{A}) + \lambda_1 R_s(\mathbf{A}) + \lambda_2 R_r(\mathbf{A}) \\
&= \frac{1}{2}\sum_{k=1}^{K}(\sum_{n=1}^{N_k} d^2(\hat{\mathbf{U}}^T\mathbf{X}_{k,n}\hat{\mathbf{U}}, \hat{\boldsymbol{\mathcal{D}}} \otimes \boldsymbol{a}_{k,n}) \\
&+ d^2(\hat{\mathbf{U}}^T\mathbf{X}_{k,n}\hat{\mathbf{U}}, \hat{\boldsymbol{\mathcal{D}}}_k \otimes \boldsymbol{a}_{k,n}^k)) \\
&+ \lambda_d \sum_{j=1, j\neq k}^{K}\sum_{n=1}^{N_k} \|\hat{\boldsymbol{\mathcal{D}}}_j \otimes \boldsymbol{a}_{k,n}^j\|_2^2 \\
&+ \sum_{p=1}^{N}\sum_{q=1}^{N}\frac{1}{2}\|\boldsymbol{a}_p - \boldsymbol{a}_q\|_2^2 \; \mathbf{G}_{bin}(p,q) \\
&+ \lambda_1 R_s(\mathbf{A}) + \lambda_2 R_r(\mathbf{A}),
\end{aligned}
$$

where $\boldsymbol{a}_{k,n}$ is denoted as $\boldsymbol{a}_p$ for simplicity. Here, we calculate each column of $\mathbf{A}$, i.e., $\boldsymbol{a}_{k,n}$ sequentially by fixing the other coefficients.

ISBN 978-90-827970-1-5 © EURASIP 2018

It should be emphasized that the above problem is a convex problem and is solved with a gradient projection algorithm. Specifically, we use the spectral projected gradient (SPG) solver [12], [21].

Classification scheme: We apply the learned projection matrix $\mathbf{U}$ and the dictionary $\boldsymbol{\mathcal{D}}$ on the query test sample $\mathbf{X}_{test}$ to estimate its class label. For this purpose, the test sample is first projected into the low-dimensional space by $\mathbf{U}$. Subsequently, it is coded over $\boldsymbol{\mathcal{D}}$ by solving the following equation:

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}\in\mathbb{R}^n_+} \frac{1}{2} d^2(\mathbf{U}^T\mathbf{X}_{test}\mathbf{U}\,\boldsymbol{\mathcal{D}} \otimes \boldsymbol{a}) + \lambda_1\|\boldsymbol{a}\|_1,$$

where $\hat{\boldsymbol{a}} = [\hat{\boldsymbol{a}}^1, \ldots, \hat{\boldsymbol{a}}^k, \ldots, \hat{\boldsymbol{a}}^K]^T$. $\hat{\boldsymbol{a}}_k$ is the sub-vector corresponding to the sub-directory $\boldsymbol{\mathcal{D}}_k$. The residual for the k-th class is calculated as

$$e_k = d^2(\mathbf{U}^T\mathbf{X}_{test}\mathbf{U}\,\boldsymbol{\mathcal{D}}_k \otimes \hat{\boldsymbol{a}}^k) + \sigma\|\hat{\boldsymbol{a}} - \boldsymbol{m}_k\|_2^2,$$

where σ is a weight to balance these two terms. $\boldsymbol{m}_k$ is the mean vector of the learned coding coefficient matrix of the k-th class, i.e., $\mathbf{A}_k$. We adopt the distance between $\hat{\boldsymbol{a}}$ and the mean vector of the learned coding coefficient of the corresponding k-th class because it gives better classification results as shown in [22]. Finally, the identity of the testing sample is determined by selecting the class label with the minimum e_k.

IV. Numerical experiments

In this section, we show the effectiveness of the proposed R-JDRDL algorithm against state-of-the-art classification algorithms on SPD matrices.

The comparison methods are the following: NN-AIRM is the AIRM-based nearest neighbor (NN) classifier; NN-Stein is the Stein metric-based NN classifier. The Stein metric $d_S : \mathcal{S}^n_{++} \times \mathcal{S}^n_{++} \to [0, \infty]$ is a symmetric type of Bregman divergence and is defined as $d^2_S(\mathbf{A}, \mathbf{B}) := \ln\det((\mathbf{A}+\mathbf{B})/2) + 0.5\ln\det(\mathbf{AB})$, where $\mathbf{A}$ and $\mathbf{B} \in \mathcal{S}^d_{++}$ [23]. DR-NN-AIRM is the AIRM-based NN classifier with the dimensionality-reduced training samples, which are obtained by R-DR [13]. DR-NN-AIRM is the same algorithm, but the distance metric is the Stein metric. R-SRC-AIRM and R-SRC-Stein are the sparse representation classifiers (SRCs) based on the AIRM and Stein metrics, respectively. R-KSRC stands for kernel-based SRC with the Stein metric. R-DL is the DL with the SRC classifier [12]. R-DR-DL-AIRM and R-DR-DL-Stein are the DL with the SRC classifier after the R-DR algorithm.

We implement our proposed algorithm in Matlab. The DL sub-problem on the product manifold makes use of the Matlab toolbox Manopt [24]. The Matlab codes R-DL, R-DR, and R-KSRC are downloaded from the respective authors' homepages.

We use the MNIST dataset[1], which are handwritten digits of 0–9. It has 60,000 images for training and 10,000 images for testing. For this dataset, we generate 8×8 RCMs [9], which is computed at (x, y) from the feature vector

$$\boldsymbol{f}_{x,y} = [x, y, I(x, y), |I_x|, |I_y|, |I_{xx}|, |I_{yy}|, \theta(x, y)],$$

[1]http://yann.lecun.com/exdb/mnist/.

TABLE I: Accuracy results

Algorithm Dictionary size (H)	Accuracy (Average $\pm$ Standard deviation) 5	 10
NN-AIRM	0.464 ± 0.0433	0.551 ± 0.0400
NN-Stein	0.469 ± 0.0418	0.552 ± 0.0426
DR-NN-AIRM	0.598 ± 0.0643	0.619 ± 0.0547
DR-NN-Stein	0.591 ± 0.0713	0.618 ± 0.0531
RSRC-AIRM	0.543 ± 0.0464	0.610 ± 0.0267
RSRC-Stein	0.546 ± 0.0460	0.612 ± 0.0290
R-KSRC	0.583 ± 0.0392	0.646 ± 0.0331
R-DL	0.506 ± 0.0310	0.598 ± 0.0336
R-DR-DL-AIRM	0.434 ± 0.0455	0.445 ± 0.0687
R-DR-DL-Stein	0.435 ± 0.0481	0.435 ± 0.0610
R-JDRDL (Proposed)	$\mathbf{0.617 \pm 0.0280}$	$\mathbf{0.673 \pm 0.0514}$

where $I(x, y)$ is the pixel value at (x, y), $I_x := \frac{\partial I(x,y)}{\partial x}$, $I_{xx} := \frac{\partial^2 I(x,y)}{\partial x^2}$, and $\theta(x, y) := \arctan\left(\frac{|I_y|}{|I_x|}\right)$. Then, three RCMs, one from the entire image, one from the left half and one from the right, are concatenated diagonally, which produce RCM of 24×24 size for each image. We execute 10 runs under randomly selected 10 test samples (N) with 5 and 10 training samples. The dictionary size H is equal to that of the training sample. Therefore, the case of $H = 5$ represents an extreme situation. We set the parameters of the proposed algorithm, based on cross-validation, to $\lambda_1 = 0.0001$, $\lambda_1 = 0.001$, and $\lambda_a = 0.0001$. λ_u are 0.01 and 0.001 in $H = 5$ and $H = 10$, respectively. We also set $v_w = v_b = H - 1$. The original and reduced dimensions are $m = 24$ and $d = 16$, respectively. We initialize $\mathbf{U}$ from the DR method [13] using single sample per class.

The results of the classification accuracy are presented in Table I. The table presents superior performances of the proposed R-JDRDL against state-of-the-art algorithms. It should be noted that R-DR-DL (both with Stein and AIRM metrics) give poor performance, implying that the separately pre-learned DR projection matrix might not be optimal for the subsequent DL.

V. Conclusions

We have presented a Riemannian joint framework, R-JDRDL, of performing dimensionality reduction along with discriminative dictionary learning on the set of SPD matrices for classification tasks. We formulate the joint learning as an objective function with the reconstruction error term and with the constraints on the projection matrix, the dictionary, and the sparse coefficient codes. Our numerical experiments demonstrate the good performance of jointly performing DL and DR. In particular, R-JDRDL outperforms existing state-of-the-arts algorithms for the MNIST image classification task.

Extending the framework to learning with other metrics on the SPD manifold (e.g., the Stein metric or the log-Euclidean metric) will be a topic of future research, as well as having a competitive numerical implementation with extensive evaluations on other real-world datasets.

ISBN 978-90-827970-1-5 © EURASIP 2018

ACKNOWLEDGEMENTS

H. Kasai was partially supported by JSPS KAKENHI Grant Numbers JP16K00031 and JP17H01732.

REFERENCES

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

[2] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *CVPR*, 2010.

[3] Z. Jiang, Z. Lin, and L. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 2013.

[4] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *ECCV*, 2012, pp. 414–427.

[5] Z. Feng, L. Yang, M. Zhang, Y. Liu, and D. Zhang, "Joint discriminative dimensionality reduction and dictionary learning for face recognition," *Pattern Recognition*, vol. 46, no. 8, pp. 2134–2143, 2013.

[6] B. Q. Yang, C.-C. Gu, K.-J. Wu, T. Zhang, and X.-P. Guan, "Simultaneous dimensionality reduction and dictionary learning for sparse representation based classification," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. pp 8969–8990, 2016.

[7] W. Liu, Z. Yu, Y. Wen, R. Lin, and M. Yang, "Jointly learning non-negative projection and dictionary with discriminative graph constraints for classification," in *BMVC*, 2016.

[8] H. Foroughi, N. Ray, and H. Zhang, "Object classification with joint projection and low-rank dictionary learning," *IEEE Trans. on Image Process.*, vol. 27, no. 2, pp. 806–821, 2018.

[9] 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, 2008.

[10] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *ECCV*, 2006.

[11] R. Bhatia, *Positive definite matrices*, ser. Princeton series in applied mathematics. Princeton University Press, 2007.

[12] A. Cherian and S. Sra, "Riemannian dictionary learning and sparse coding for positive definite matrices," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016.

[13] M. Harandi, M. Salzmann, and H. Richard, "Dimensionality reduction on spd manifolds: The emergence of geometry-aware methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[14] Z. Huang and L. V. Gool, "A riemannian network for spd matrix learning," in *AAAI*, 2017.

[15] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *ICML*, 2015.

[16] Z. Huang, R. Wang, X. Li, W. Liu, S. Shan, L. V. Gool, and X. Chen, "Geometry-aware similarity learning on spd manifolds for visual recognition," *IEEE Trans. Circuits Syst. Video Technol.*, 2017.

[17] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[18] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *Int. Jornal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[19] H. Sato and T. Iwai, "A new, globally convergent Riemannian conjugate gradient method," *Optimization*, vol. 64, no. 4, pp. 1011–1031, 2015.

[20] W. Ring and B. Wirth, "Optimization methods on Riemannian manifolds and their application to shape space," *SIAM J. Optim.*, vol. 22, no. 2, pp. 596–627, 2012.

[21] E. G. Birgin, J. M. Martínez, and M. Raydan, "Spg - software for convex-constrained optimization," *ACM Trans. on Math. Softw.*, vol. 27, no. 3, pp. 340–349, 2001.

[22] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *ICCV*, 2011.

[23] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric means," in *NIPS*, 2012.

[24] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt: a Matlab toolbox for optimization on manifolds," *JMLR*, vol. 15, no. 1, pp. 1455–1459, 2014.

ISBN 978-90-827970-1-5 © EURASIP 2018