Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification

Ruiping Wang[†‡], Huimin Guo[†], Larry S. Davis[†], Qionghai Dai[‡]

[†]Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 20742
[‡]TNLIST, Department of Automation, Tsinghua University, Beijing, 100084, China
{rpwang, qionghaidai}@tsinghua.edu.cn, {hmguo, lsd}@umiacs.umd.edu

Abstract

We propose a novel discriminative learning approach to image set classification by modeling the image set with its natural second-order statistic, i.e. covariance matrix. Since nonsingular covariance matrices, a.k.a. symmetric positive definite (SPD) matrices, lie on a Riemannian manifold, classical learning algorithms cannot be directly utilized to classify points on the manifold. By exploring an efficient metric for the SPD matrices, i.e., Log-Euclidean Distance (LED), we derive a kernel function that explicitly maps the covariance matrix from the Riemannian manifold to a Euclidean space. With this explicit mapping, any learning method devoted to vector space can be exploited in either its linear or kernel formulation. Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS) are considered in this paper for their feasibility for our specific problem. We further investigate the conventional linear subspace based set modeling technique and cast it in a unified framework with our covariance matrix based modeling. The proposed method is evaluated on two tasks: face recognition and object categorization. Extensive experimental results show not only the superiority of our method over state-of-the-art ones in both accuracy and efficiency, but also its stability to two real challenges: noisy set data and varying set size.

1. Introduction

Classification based on image sets has recently attracted increasing interest in the computer vision and pattern recognition community [1], [5], [10], [13], [26], [30]. This problem naturally arises in a wide range of applications, such as video surveillance, classification with images from multi-view cameras or photo albums, and classification based on long term observations. In the task of image set classification, each set generally contains a large number of images that belong to the same class and cover large variations in the object's appearance due to camera pose changes, non-rigid deformations or different lighting conditions. While traditional recognition methods based on single-shot images have achieved a certain level of success under restricted conditions, more robust object recognition can be expected by using sets as input rather than single images. This is mainly because the image set incorporates useful data variability information, which can be efficiently exploited under more realistic conditions with significantly larger variations [1], [5], [12], [13].

Among previous work, there is a category of video-based classification methods [11], [14], which focus on utilizing the temporal dynamic information between consecutive video frames. However, in the general scenario of image set classification, images in a set are collected not necessarily from video sequences but possibly from multiple unordered observations [10], [13], [26], so that the images suitable for recognition are widely spaced in time and location.

1.1. Previous work

For image set classification, existing methods mainly focus on the key issues of how to model the image sets and how to measure their similarity [5], [10], [27]. In most cases, the similarity function is defined specifically for certain image set modeling or representation methods.

From the view of set modeling, relevant approaches to image set classification broadly fall into two categories [13]: parametric and nonparametric representations. Parametric modeling methods seek to represent each image set with some parametric distribution function, e.g., single Gaussian [22] or Gaussian mixture models (GMM) [1], and then measure the similarity between two distributions in terms of the Kullback-Leibler Divergence (KLD). The main limitation of these methods is that they need to solve a difficult parameter estimation problem and may have large performance fluctuations in cases where the training and novel test data sets have weak statistical correlations [13].

In comparison, nonparametric methods typically relax the assumptions on distributions of the set data, and try to model the image set in a more flexible manner. One class of prevalent methods is to use subspace learning techniques to account for the set data variability globally, following the pioneering work of [30]. Such methods attempt to represent the image set either by a single linear subspace [8], [13], or by a more sophisticated manifold in the form of a mixture of linear subspaces [12], [26], [27]. To measure the subspace distance, the method of principal angles [9] is

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

mainly exploited to capture the common variation modes of two subspaces. Since they impose a uniform prior over data variations in different image sets, nonparametric methods have been shown to have many favorable properties [13], [26]. However, for appropriate manifold modeling, they usually require a large data set with dense sampling, while the linear subspace modeling cannot well accommodate the case when the set is of small size but has large and complex data variations. As also indicated in [5], linear subspace based modeling has the limitation that it incorporates only relatively weak information (subspace angles) about the location and boundary of the samples in the input space.

More recently, a new type of nonparametric methods [5], [10] based on matching the closest pair of points from two image sets has been introduced. In [20], a straightforward strategy is adopted to find the nearest actual sample images from the two sets without considering data variations across the set. In contrast, [5], [10] approximate the image set with a more theoretically principled affine subspace model and match the closest virtual points via a convex optimization. While intra-class variations can be effectively handled, such methods are still susceptible to the presence of outliers and have relatively high computational cost [13], [20], due to their inherent single sample-based matching mechanism.

1.2. Overview of our approach

This paper proposes a novel Covariance Discriminative Learning (CDL) approach to image set classification. By representing each image set with its natural second-order statistic - covariance matrix - we formulate the problem as classifying points lying on a Riemannian manifold spanned by SPD matrices, i.e., nonsingular covariance matrices. Since classical learning algorithms cannot take points on the manifold as their direct input, we explore an efficient metric for the SPD matrices, i.e., Log-Euclidean distance (LED), and further derive a kernel function that explicitly maps the covariance matrix from the Riemannian manifold to a Euclidean space. Benefiting from this explicit kernel feature mapping, any learning method originally developed for vector spaces can be used, by taking either the Log-mapped covariance matrices as input to its linear formulation or the derived kernel function as input to its kernel formulation. A conceptual illustration of our approach is shown in Fig. 1.

Here we exploit two representative methods - Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS), for their feasibility for our specific case where the number of samples (i.e., the number of image sets) is considerably smaller than the number of feature dimensions (i.e., the number of covariance matrix entries). Moreover, we investigate the conventional linear subspace based set modeling technique and cast it in a unified framework with our covariance matrix based set modeling.

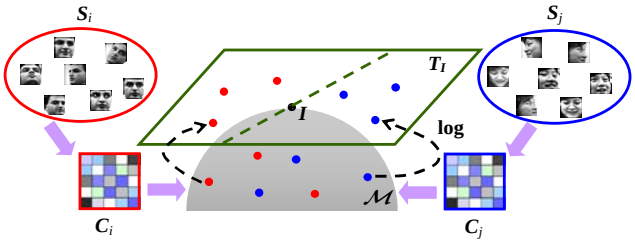

Figure 1: Conceptual illustration of the proposed CDL method. We model the image set $\boldsymbol{S}$ by its sample covariance matrix $\boldsymbol{C}$, and formulate the problem as classifying points on the Riemannian manifold $\boldsymbol{M}$. With the log map, traditional learning methods can be utilized in the tangent space $\boldsymbol{T_I}$ (which is a vector space) at the point of the identity matrix $\boldsymbol{I}$ on the manifold.

2. Set modeling by covariance matrix

Let $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n]$ be the data matrix of an image set with n samples, where $\boldsymbol{s}_i \in \mathbb{R}^d$ denotes the i-th image sample with d-dimensional feature description. Here, the image intensity is used as the raw feature. We represent the image set with the $d \times d$ sample covariance matrix:

$$\boldsymbol{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{s}_i - \bar{\boldsymbol{s}})(\boldsymbol{s}_i - \bar{\boldsymbol{s}})^T , \qquad (1)$$

where $\bar{\boldsymbol{s}}$ is the mean of the image samples. The diagonal entries of the covariance matrix represent the variance of each individual feature, and the non-diagonal entries are their respective correlations.

While it is rather simple to derive and compute, there are several advantages to model the image set with its covariance matrix. As the raw second-order statistic of a set of samples, the covariance matrix makes no assumption about the set data distribution, thus providing a natural representation for an image set with any number of samples and any type of features. This representation leads to an effective way to discriminate image sets of different classes by encoding the feature correlation information specific to each object class. Compared with previous single or mixture of linear subspaces based methods [8], [12], [13], [26], [27], [30], the covariance matrix characterizes the set structure more faithfully. In fact, linear subspace models usually originate from an eigen-decomposition of the covariance matrix while discarding the non-leading eigenvectors and all eigenvalues. This makes the resulting subspace too loose to reflect the underlying set distribution boundary. Compared to previous closest sample pair based methods [5], [10], [20], the covariance matrix representation shows stronger resistance to outliers, since it is a statistic of all samples and noise-corrupting samples are largely filtered out with an average filter during covariance computation.

Prior to our study here, covariance matrices have been used to characterize local regions within an image, named region covariance [23], and applied to several visual

processing tasks, e.g., object detection/recognition, object tracking and texture classification [16], [24]. It is worth noting that as a region descriptor, the covariance matrix in these works differs from the image set descriptor of this paper in several aspects. For region covariance, each pixel inside the image region serves as a sample, and sample features include pixel coordinate, intensity, and higher order derivatives, etc. Since the number of pixels in the region is usually larger than the feature dimension, the region covariance matrix can generally be guaranteed to be nonsingular. However, in image set classification it is often the case that the number of images is less than the feature dimension, i.e. $n < d$, thus leading to the singularity of the set covariance matrix. To tackle this singularity, a simple method is to add a small perturbation to the covariance. Furthermore, when the covariance matrix is utilized as an individual sample for learning algorithms, the number of samples (number of covariance matrices) is considerably smaller than the number of feature dimensions (number of covariance matrix entries) for our set covariance case, which is opposite to the case of region covariance learning [24]. This will be the topic of the next section.

3. Covariance Discriminative Learning

It is well known that the $d \times d$ SPD matrices (i.e., nonsingular covariance matrices) Sym_d^+ do not lie in a Euclidean space but on a Riemannian manifold. We naturally formulate the problem of image set classification as classifying points lying on the Riemannian manifold spanned by SPD matrices. However, it is not trivial to learn a classifier on the manifold since classical learning methods are devoted to operating in vector spaces associated with Euclidean metrics and thus cannot optimally take points on the manifold as their direct input. We next explore Riemannian metrics for covariance matrices by utilizing the Log-Euclidean distance (LED) and develop efficient learning algorithms associated with this metric.

3.1. Riemannian metrics for covariance matrix

Here we consider two different formulations of distance metric for Sym_d^+ that have been well established in the field of Riemannian geometry. The first metric, affine-invariant distance (AID) [6], [17], is defined in terms of generalized eigenvalues of two covariance matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$:

$$d_{AID}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \sqrt{\sum_{i=1}^{d} \ln^2 \lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)} , \quad (2)$$

where the eigenvalues $\lambda_i(\boldsymbol{C}_1, \boldsymbol{C}_2)$ ($i = 1, ..., d$) are obtained from $\left|\lambda \boldsymbol{C}_1 - \boldsymbol{C}_2\right| = 0$. This metric is invariant under affine transformations and inversion, and has been mainly used as the distance measure for region covariance [23], [24]. However, the price paid for its success is a high computational burden in practice [2].

Another distance metric for Sym_d^+ is the Log-Euclidean distance (LED) [2] that results in classical Euclidean computations in the domain of matrix logarithms as:

$$d_{LED}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \left\| \log(\boldsymbol{C}_1) - \log(\boldsymbol{C}_2) \right\|_F , \quad (3)$$

where log is the ordinary matrix logarithm operator and $\|\bullet\|_F$ denotes the matrix Frobenius norm. Let $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T$ be the eigen-decomposition of SPD matrix $\boldsymbol{C}$, its log is a symmetric matrix and can be computed easily by

$$\log(\boldsymbol{C}) = \boldsymbol{U} \log(\boldsymbol{\Sigma}) \boldsymbol{U}^T , \quad (4)$$

where $\log(\boldsymbol{\Sigma})$ is the diagonal matrix of the eigenvalue logarithms. The LED metric is particularly simple to use and avoids the computational expense of the AID metric, while conserving excellent theoretical properties. Please refer to [2] for more detailed discussion on the similarities and differences between the two metrics.

The LED metric can be understood as projecting a point $\boldsymbol{C}$ on the Riemannian manifold $\mathcal{M}$ to a Euclidean space via the logarithm map:

$$\Psi_{log} : \mathcal{M} \mapsto T_{\boldsymbol{I}}, \quad \boldsymbol{C} \to \log(\boldsymbol{C}) . \quad (5)$$

The image $\Psi_{log}(\mathcal{M})$ is the tangent space $T_{\boldsymbol{I}}$ at the point of the identity matrix $\boldsymbol{I}$, which is a vector space spanned by $d \times d$ symmetric matrices. The LED metric thus simply reduces to a Euclidean distance in $\mathbb{R}^{d \times d}$. By computing the inner product in the Euclidean space $T_{\boldsymbol{I}}$, we actually derive a *Riemannian kernel function* on the manifold $\mathcal{M}$:

$$k_{log}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \mathrm{tr}[\log(\boldsymbol{C}_1) \bullet \log(\boldsymbol{C}_2)] . \quad (6)$$

It is easy to check that k_{log} is a symmetric real-valued function: $k_{log}(\boldsymbol{C}_i, \boldsymbol{C}_j) = k_{log}(\boldsymbol{C}_j, \boldsymbol{C}_i)$ for all $\boldsymbol{C}_i, \boldsymbol{C}_j \in \mathcal{M}$. The positive definiteness of this function follows from the properties of the Frobenius norm. For all $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n$ ($\boldsymbol{C}_i \in \mathcal{M}$) and $b_1, \ldots, b_n$ ($b_i \in \mathbb{R}$) for any $n \in \mathbb{N}$, we have

$$\begin{aligned} &\sum_{i,j} b_i b_j k_{log}(\boldsymbol{C}_i, \boldsymbol{C}_j) = \sum_{i,j} b_i b_j \mathrm{tr}[\log(\boldsymbol{C}_i) \bullet \log(\boldsymbol{C}_j)] \\ &= \mathrm{tr}\left[\left(\sum_i b_i \log(\boldsymbol{C}_i) \right)^2 \right] = \left\| \sum_i b_i \log(\boldsymbol{C}_i) \right\|_F^2 \geq 0 \end{aligned} . \quad (7)$$

With these properties, the proposed Riemannian kernel Eq. (6) is shown to satisfy the conditions of Mercer's theorem [21]. It is interesting to note that traditional kernel functions (e.g., Gaussian kernel, polynomial kernel) are usually defined on a Euclidean space, and *implicitly* map the points from this Euclidean space to another higher dimensional Euclidean space, i.e., the so-called RKHS (reproducing kernel Hilbert space) feature space [21]. In contrast, our kernel function is defined on an unconventional Riemannian manifold and *explicitly* maps the points from the manifold to a Euclidean space through Eq. (5).

The explicit kernel feature mapping allows us to utilize any standard vector space learning algorithm. We can

either apply the linear formulation of the method to the Euclidean space T_I by taking the Log-mapped covariance matrices $\log(\boldsymbol{C})$ as input, or apply its kernel formulation to the manifold $\mathcal{M}$ by taking $\boldsymbol{C}$ and the derived kernel function k_{log} as input. Let $D = d^2$, the $d \times d$ matrices are represented as D-dimensional sample vectors in these algorithms. As discussed in Sec. 2, for set covariance learning, the number of samples is much smaller than the number of feature dimensions, thus making the kernel formulation especially well suited and efficient for our special case. In the following, we explore two typical learning methods - LDA and PLS - for their feasibility, by focusing on their kernel formulations. The former learns a discriminant subspace and maps the samples to this subspace followed by Nearest Neighbor (NN) classification, while the latter directly learns a regression model between the observed samples and their corresponding class labels.

3.2. Learning with LDA and its kernel variant

Linear Discriminant Analysis (LDA) is well known for its effectiveness for classification problems. Here we provide a brief description. Suppose we have a set of m samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\} \in \mathbb{R}^D$ belonging to c classes in the input data space, where the number of samples in the k-th class is m_k, thus $\sum_{k=1}^{c} m_k = m$. The kernel variant of LDA (KLDA) [4] is formulated using the kernel trick as follows. Letting $\phi: \mathbb{R}^D \mapsto \mathcal{F}$ be the feature map, an inner product can be defined on the feature space $\mathcal{F}$ with the kernel function as: $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. KLDA seeks to solve the following optimization [4]:

$$\boldsymbol{\alpha}_{opt} = \arg\max \frac{\boldsymbol{\alpha}^T \boldsymbol{KWK\alpha}}{\boldsymbol{\alpha}^T \boldsymbol{KK\alpha}}, \tag{8}$$

where $\boldsymbol{\alpha} = [a_1, \ldots, a_m]^T$, $\boldsymbol{K}$ is the kernel Gram matrix: $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{W}$ is defined as:

$$\boldsymbol{W}_{ij} = \begin{cases} 1/m_k, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \text{ are both in the } k\text{-th class} \\ 0, & \text{otherwise.} \end{cases}. \tag{9}$$

The optimal $\boldsymbol{\alpha}$ are given by the largest eigenvectors of the eigen-problem: $\boldsymbol{KWK\alpha} = \lambda \boldsymbol{KK\alpha}$. By the representer theorem, each eigenvector $\boldsymbol{\alpha}$ gives a discriminant direction vector $\boldsymbol{w}$ in the feature space $\mathcal{F}$ as: $\boldsymbol{w} = \sum_{i=1}^{m} a_i \phi(\boldsymbol{x}_i)$. Grouping the maximum number $(c-1)$ of eigenvectors, we obtain $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{c-1}]$. Given a data example $\boldsymbol{x}_t \in \mathbb{R}^D$ in the input space, its $c-1$-dimensional projection $\boldsymbol{z}_t$ in the discriminant subspace is obtained by

$$\boldsymbol{z}_t = \boldsymbol{A}^T \boldsymbol{K}_t, \text{ where } \boldsymbol{K}_t = [k(\boldsymbol{x}_1, \boldsymbol{x}_t), \ldots, k(\boldsymbol{x}_m, \boldsymbol{x}_t)]^T. \tag{10}$$

For our set covariance learning, suppose we are given m gallery image sets $\boldsymbol{S}_i^g$ $(i = 1, ..., m)$ which are from c classes with known labels for training, and l probe image sets $\boldsymbol{S}_j^p$ $(j = 1, ..., l)$ for testing. We first compute their corresponding covariance matrices $\boldsymbol{C}_i^g$, $\boldsymbol{C}_j^p$, and represent them as D-dimensional sample vectors. The training samples $\boldsymbol{C}_i^g$ and the proposed Riemannian kernel in Eq. (6) are then fed into KLDA to solve the optimization in Eq. (8). In the testing phase, both $\boldsymbol{C}_i^g$ and $\boldsymbol{C}_j^p$ are projected to the discriminant subspace through Eq. (10). Nearest Neighbor (NN) classification in this $c-1$-dimensional subspace is then conducted based on Euclidean distance.

3.3. Learning with PLS and its kernel variant

Partial Least Squares (PLS) is a method for modeling relations between sets of observed variables by means of latent variables. In its general form, PLS creates score/ latent vectors by using the existing correlations between different sets of variables while also keeping most of the variance of both sets. Please refer to [18] for more details.

Let $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^D$ denote a D-dimensional vector of predictor variables in the first set of data and similarly $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^c$ denote a c-dimensional vector of response variables from the second set. Observing m data samples from each set of variables, PLS decomposes matrix $\boldsymbol{X}_{m \times D}$ and $\boldsymbol{Y}_{m \times c}$ (each row contains a sample) into the form

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{TP}^T + \boldsymbol{E} \\ \boldsymbol{Y} &= \boldsymbol{UQ}^T + \boldsymbol{F} \end{aligned} \tag{11}$$

where $\boldsymbol{T}$ and $\boldsymbol{U}$ are $m \times p$ matrices containing the extracted p latent vectors, the $D \times p$ matrix $\boldsymbol{P}$ and the $c \times p$ matrix $\boldsymbol{Q}$ represent loadings, and the $m \times D$ matrix $\boldsymbol{E}$ and the $m \times c$ matrix $\boldsymbol{F}$ are the residuals. Based on the nonlinear iterative partial least squares (NIPALS) algorithm [28], PLS finds weight vectors $\boldsymbol{w}$, $\boldsymbol{v}$ such that

$$\max_{|\boldsymbol{w}|=|\boldsymbol{v}|=1} [cov(\boldsymbol{Xw}, \boldsymbol{Yv})]^2 = [cov(\boldsymbol{t}, \boldsymbol{u})]^2, \tag{12}$$

where $\boldsymbol{t}$ and $\boldsymbol{u}$ are the column vectors of $\boldsymbol{T}$ and $\boldsymbol{U}$ respectively, $cov(\boldsymbol{t}, \boldsymbol{u})$ is the sample covariance. With the obtained latent vectors, the regression coefficients between the two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be estimated by:

$$\boldsymbol{B} = \boldsymbol{W}(\boldsymbol{P}^T \boldsymbol{W})^{-1} \boldsymbol{T}^T \boldsymbol{Y} = \boldsymbol{X}^T \boldsymbol{U}(\boldsymbol{T}^T \boldsymbol{XX}^T \boldsymbol{U})^{-1} \boldsymbol{T}^T \boldsymbol{Y}, \tag{13}$$

which results in $\hat{\boldsymbol{Y}} = \boldsymbol{XB}$ [18].

In the kernel formulation of PLS (KPLS) [19], we use the same notation as in KLDA for consistency and simplicity. The basic idea of KPLS is to map the original $\mathcal{X}$-space data into a RKHS feature space $\mathcal{F}$ with $\phi: \mathbb{R}^D \mapsto \mathcal{F}$, and perform the kernel form of the NIPALS algorithm [19]. Let $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_m)]^T$ be the feature matrix of the training points, the kernel Gram matrix can thus be written as $\boldsymbol{K} = \boldsymbol{\Phi\Phi}^T$. Then the regression coefficients $\boldsymbol{B}_\phi$ in the feature space will have the form

$$\boldsymbol{B}_\phi = \boldsymbol{\Phi}^T \boldsymbol{U}(\boldsymbol{T}^T \boldsymbol{KU})^{-1} \boldsymbol{T}^T \boldsymbol{Y}, \tag{14}$$

Given a testing data example $\boldsymbol{x}_t \in \mathbb{R}^D$ in the $\mathcal{X}$-space, its KPLS prediction $\boldsymbol{y}_t$ in the $\mathcal{Y}$-space can be obtained by

$$\boldsymbol{y}_t^T = [\phi(\boldsymbol{x}_t)]^T \boldsymbol{B}_\phi = \boldsymbol{K}_t{}^T \boldsymbol{U}(\boldsymbol{T}^T \boldsymbol{K}\boldsymbol{U})^{-1}\boldsymbol{T}^T\boldsymbol{Y}\,, \qquad (15)$$

where $\boldsymbol{K}_t$ has the same meaning as in Eq. (10).

When applied to set covariance learning, we use the gallery image sets $\boldsymbol{S}_i^g$ ($i = 1,...,m$) and their associated class labels to learn the KPLS latent model. Specifically, all training covariance matrices $\boldsymbol{C}_i^g$, represented as D-dim sample vectors as in KLDA, are gathered to build the *predictor* matrix $\boldsymbol{X}_{m\times D}$. For each $\boldsymbol{C}_i^g$ we define its class membership indicator vector: $\boldsymbol{y}_i = [0,...,1,...,0]^T \in \mathbb{R}^c$, where the k-th entry being 1 and all other entries being 0 indicates that $\boldsymbol{C}_i^g$ belongs to the k-th class. The *response* matrix $\boldsymbol{Y}_{m\times c}$ can then be easily constructed with $\boldsymbol{y}_i^T$ as its row vector. Taking our Riemannian kernel, KPLS is used to learn the regression model in Eq. (14). In the testing phase, given a probe image set $\boldsymbol{S}_j^p$ and the corresponding covariance matrix $\boldsymbol{C}_j^p$, its class membership indicator vector $\boldsymbol{y}_t$ can be computed from Eq. (15) by treating $\boldsymbol{C}_j^p$ as a testing example $\boldsymbol{x}_t$. The entry index with the largest response in $\boldsymbol{y}_t$ then determines the class label of $\boldsymbol{S}_j^p$.

While PLS operates in a very different manner from LDA for classification, there in fact exists close theoretical connection between the two methods, as has been well studied in [3], [18]. In comparison to LDA, PLS has proven to be useful in situations where the number of observed variables (i.e., D) is much larger than the number of observations (i.e., m). This is just the case for our set covariance learning where $D = 160,000$ and $m < 150$. In addition, PLS is not limited by the $c-1$ discrimination dimensions and may be more suitable in the situation of non-Gaussian class distributions in the feature space [18].

4. Alternative subspace modeling and learning

As stated in Sec. 1.1, one class of conventional set classification methods is based on linear subspace modeling and learning [8], [12], [13], [26], [27], [30]. Given a set $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n]$, $\boldsymbol{s}_i \in \mathbb{R}^d$, its subspace representation is usually obtained by principal component analysis (PCA), which reduces to the eigen-decomposion of the covariance matrix $\boldsymbol{C}$, i.e., $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T$ as in Sec. 3.1. Here, the $d\times q$ orthonormal matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_q]$ contains the q (usually $q < d$) largest eigenvectors of $\boldsymbol{C}$, and serves as the basis of a q-dimensional linear subspace of $\mathbb{R}^d$.

It is well known that the set of q-dimensional linear subspaces of $\mathbb{R}^d$ spans a Grassmann manifold $\mathcal{G}(q,d)$. For two points $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ on the manifold, their distance is typically measured in terms of principal angles [9]. While a number of distances have been exploited in the literature, among them one appealing formulation is the Projection metric [8] that uses all principal angles and finally reduces to the classical Euclidean computation as follows:

$$d_{Proj}(\boldsymbol{U}_1, \boldsymbol{U}_2) = 2^{-1/2}\left\|\boldsymbol{U}_1\boldsymbol{U}_1^T - \boldsymbol{U}_2\boldsymbol{U}_2^T\right\|_F\,, \qquad (16)$$

where $\boldsymbol{U}_i\boldsymbol{U}_i^T$ is the rank-q, $d\times d$ orthogonal projection matrix. In analogy to the LED metric, the Projection metric similarly indicates an explicit mapping of a point $\boldsymbol{U}$ from the Grassmann manifold $\mathcal{G}(q,d)$ to a Euclidean space:

$$\Psi_{Proj}:\ \mathcal{G}(q,d) \mapsto \mathbb{R}^{d\times d},\quad \boldsymbol{U} \to \boldsymbol{U}\boldsymbol{U}^T\,. \qquad (17)$$

The inner product in the Euclidean space $\mathbb{R}^{d\times d}$ spanned by $d\times d$ projection matrices thus induces a Grassmann kernel function on the manifold $\mathcal{G}(q,d)$, as discussed in [8],[29]:

$$k_{Proj}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \mathrm{tr}[(\boldsymbol{U}_1\boldsymbol{U}_1^T)(\boldsymbol{U}_2\boldsymbol{U}_2^T)] = \left\|\boldsymbol{U}_1^T\boldsymbol{U}_2\right\|_F^2\,. \qquad (18)$$

It is also easy to check that k_{Proj} is a valid Mercer kernel.

Similar to the framework for covariance learning, using the explicit Grassmann kernel feature mapping, traditional learning methods can be applied to subspace-based learning. Specifically, we can either use the linear formulation of the method by taking the projection matrix $\boldsymbol{U}\boldsymbol{U}^T$ as input, or utilize its kernel formulation by taking the basis matrix $\boldsymbol{U}$ and the Grassmann kernel k_{Proj} as input. For set modeling, while the subspace representation $\boldsymbol{U}$ originates from the covariance matrix $\boldsymbol{C}$, it extracts only the leading eigenvectors but simply discards those non-leading ones and all eigenvalues. This will inevitably result in a loss of information and consequently inferior set classification accuracy, which will be verified next.

5. Experimental evaluations

We evaluate the proposed method on two visual classification tasks: face recognition with image sets and object categorization. Both tasks are handled as covariance learning and classification problem.

5.1. Databases and settings

For the face recognition task, we consider three widely studied datasets with different characteristics, including two benchmark datasets: Honda/UCSD [14], CMU MoBo [7], and one much challenging dataset: YouTube Celebrities [11], to ensure extensive evaluations. The Honda/UCSD consists of 59 video sequences involving 20 different persons. Each video contains approximately 300~500 frames covering large variations in head pose and facial expression. The CMU MoBo contains 96 sequences of 24 different subjects. Each subject has 4 sequences captured in different walking situations and each sequence has about 300 frames. The YouTube Celebrities has 1,910 video clips of 47 subjects collected from YouTube. Each clip contains hundreds of frames, which are mostly low

resolution and highly compressed. For all three databases, we used a cascaded face detector [25] to collect faces in each video, and then resized each face to a 20×20 intensity image. Histogram equalization was the only pre-processing used to eliminate lighting effects as in [5], [10], [27]. Each video generated an image set of faces.

For the object categorization task, we use the benchmark database ETH-80 [15]. It contains images of 8 categories with each category including 10 objects. Each object has 41 images of different views which form an image set. The task is to classify an image set of an object into a known category. 20×20 intensity images were also used.

To allow comparison with the literature we followed the same protocol as [5], [10], [26], [27]. On all of four datasets, we conducted ten-fold cross validation experiments, i.e., 10 randomly selected gallery/probe combinations, to report average recognition rates of different methods. Specifically, for both Honda and MoBo, each person had one image set as the gallery and the rest sets for probes. For YouTube, in each fold, one person had 3 randomly chosen image sets for the gallery and 6 for probes. For ETH-80, each category had 5 objects for gallery and the other 5 objects for probes.

5.2. Comparative methods and settings

We compared our approach with three categories of set modeling methods, as discussed in Sec. 1.1.

- Linear subspace based:
  1. Mutual Subspace Method (MSM) [30];
  2. Discriminant Canonical Correlations (DCC) [13];
- Nonlinear manifold based:
  3. Manifold-Manifold Distance (MMD) [26];
  4. Manifold Discriminant Analysis (MDA) [27];
- Affine subspace based:
  5. Affine Hull based Image Set Distance (AHISD) [5];
  6. Convex Hull based Image Set Distance(CHISD) [5];
  7. Sparse Approximated Nearest Point (SANP) [10].

For fair comparison, the important parameters of each method were empirically tuned according to the recommendations in the original references as well as the source codes provided by the original authors. In MSM/ DCC/MMD, PCA was performed to learn the single or mixture of linear subspaces by preserving 95% of data energy. In MDA, the number of between-class NN local models and the subspace dimension were specified as [27]. For both AHISD and CHISD, we used their linear version and retained 95% energy by PCA. The error penalty in CHISD was set to $C = 100$ as [5]. For SANP, we adopted the same weight parameters as [10] for the convex optimization. Note that for the DCC learning on Honda and MoBo, the single training image set from each class was randomly divided into two subsets to construct the within-class sets, following the setting of [13], [27].

Table 1. Average recognition rates of different methods on the four datasets by ten-fold experiments

Methods	Honda/ UCSD	CMU MoBo	YouTube	ETH-80
MSM [30]	0.925	0.852	0.611	0.878
DCC [13]	0.980	0.881	0.648	*0.905*
MMD [26]	0.971	0.902	0.629	0.863
MDA [27]	*1.000*	0.943	0.653	0.890
AHISD [5]	0.885	0.951	0.637	0.773
CHISD [5]	0.905	0.940	0.663	0.735
SANP [10]	0.936	*0.963*	*0.684*	0.755
Proj.+LDA	0.964	0.821	0.657	0.928
Proj.+PLS	0.997	0.884	0.677	0.953
COV+LDA	0.980	0.867	0.675	0.945
COV+PLS	**1.000**	**0.941**	**0.701**	**0.965**

For our proposed framework, we tested four different combinations that include set modeling with covariance matrix (referred to as "COV") or alternative linear subspace (i.e., the projection matrix, referred to as "Proj."), followed by discriminative learning with LDA or PLS (both using the kernel formulation). It is worth noting that the scheme of "Proj.+LDA" is indeed the method of [8]. For covariance modeling, as stated in Sec. 2, to avoid the matrix singularity, regularization was applied to the original covariance matrix as: $\boldsymbol{C}^* = \boldsymbol{C} + \lambda \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix and λ was set to $10^{-3} \times trace(\boldsymbol{C})$. For subspace modeling, the PCA dimension was set to $q = 10$ with about 95% data energy. LDA/PLS utilized $c-1$ discriminant/latent dimensions. Since image sizes in all datasets are 20×20, the intensity feature dimension is $d = 400$, and thus $D = 160,000$. The number of gallery (training) image sets, m , is 20, 24, 141, 40 respectively for the four datasets. For the single sample per class learning with LDA on Honda and MoBo, the same strategy as that for DCC above was also adopted.

5.3. Results and analysis

We summarize the recognition results of all methods on the four datasets in Tab. 1. Each reported rate is an average over the ten-fold trials. Comparing the four combinations under our framework, we have the following two consistent observations: (1) COV is better than Proj., which confirms the superiority of covariance modeling over conventional linear subspace modeling; (2) PLS is better than LDA, which indicates PLS is more suitable for our covariance learning and again verifies our analysis in previous sections. While comparing our best performing "COV+PLS" with

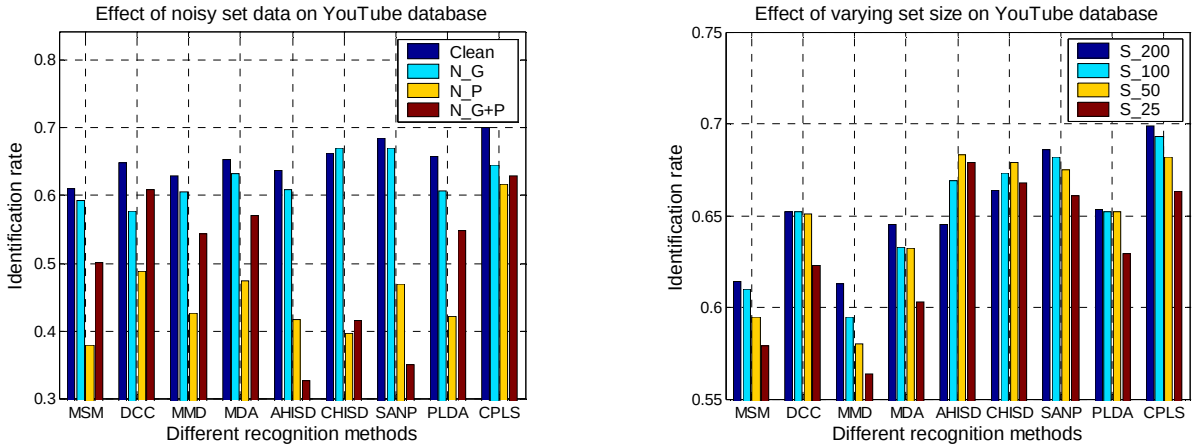

Figure 2: The mean recognition rates of different methods under two practical challenges: (*left*) noisy set data and (*right*) varying set size. In the figure, PLDA and CPLS are shorted for "Proj.+LDA" and "COV+PLS" respectively.

other competing methods, we find that our approach delivers the overall best performance with the highest rate in three out of four datasets. It is interesting to observe that the most recently developed affine subspace based methods [5], [10] exhibit inferior accuracy to other ones by a remarkably large margin in the object categorization task. This is mainly because the similarity of two different objects from the same category lies in their common appearance variation modes, which cannot be accommodated adequately by the closest single points based matching mechanism in [5], [10].

In real-world applications, it is often the case that image sets contain noisy data (i.e., images outside the category) and are of varying size. We next experimentally study these two challenges and evaluate the performance of different methods on the challenging face dataset YouTube. For the noisy set data problem, we followed [5] and conducted three experiments in which the gallery and/or the probe sets were systematically corrupted by adding one image from each of the other classes. The original clean data and the three noisy cases are referred to as "Clean", "N_G" (only gallery has noise), "N_P" (only probe has noise), and "N_G+P" (both) respectively. For the varying set size problem, we uniformly down-sampled each image set (both gallery and probe) and used the obtained subsets for classification. We tested four cases by extracting 200/100/50/25 samples, referred to as "S_200"~"S_25" respectively. If a set contained fewer images than the specified number, the original set was used.

From the comparison results in Fig. 2, it can be seen that our proposed "COV+PLS" shows high robustness against both challenges, with some slight performance drop. This can be mainly attributed to the advantages of using the covariance matrix as the set representation. For the noisy data case, the MSM/DCC/MMD/MDA seem more stable than the AHISD/CHISD/SANP since the former ones taking the set samples as a whole for subspace modeling and matching can alleviate the influence of noise samples to some extent. In contrast, based on matching the closest set points, the latter methods rely highly on the location of each individual sample and their model fitting can be heavily deteriorated by outliers. For the varying size case, we find that all other methods except AHISD/CHISD encounter problems to appropriately fit their models with decreased set size as expected. The unpredictable rate rise of AHISD/CHISD may also be explained by the fact that their model fitting is much sensitive to sample distribution.

Lastly, we compared the computational complexity of different methods with the benchmark Honda/UCSD dataset ($m = 20$) on a Pentium IV, 2.93 GHz PC. The time cost for each method is tabulated in Table 2. Training time is only needed by discriminant methods. Since kernel LDA/PLS learning in our method mainly involves the eigen-decomposition of $m \times m$ kernel Gram matrix, they are very efficient. For testing, we report the classification time for matching one probe image set with the 20 gallery image sets. The superiority of our method can be clearly observed, especially over the three affine subspace based methods. As discussed in Sec.1.1, the single sample-based matching mechanism and complex optimization procedure make these methods less appealing in terms of efficiency.

Table 2. Computation time (seconds) of different methods on Honda/UCSD for training and testing (classification of one image set)

	MSM	DCC	MMD	MDA	AHISD	CHISD	SANP	Proj.+LDA	COV+PLS
Training	N/A	12.397	N/A	8.846	N/A	N/A	N/A	4.001	2.322
Testing	5.114	5.120	6.462	4.288	20.516	8.424	52.976	3.980	2.033

6. Conclusion

We have proposed an efficient image set classification method called Covariance Discriminative Learning (CDL). The method represents each image set with its covariance matrix and models the problem as classifying points on the Riemannian manifold spanned by nonsingular covariance matrices. We derived a novel Riemannian kernel function which successfully bridges the gap between traditional learning methods operating in vector spaces and the learning task on an unconventional manifold. A similar derivation was extended to the classic linear subspace based set modeling technique and it was cast in a unified learning framework with covariance matrix modeling. We explored two typical methods, LDA and PLS, for learning, and demonstrated the advantages of PLS for our specific problem. The promising experimental results show the superiority of our method over the state-of-the-art in terms of accuracy and efficiency, as well as its robustness to the practical challenges of noisy set data and varying set size.

For future work, we are exploring the incorporation of set mean information into covariance matrix modeling. We will also study more robust estimator for the covariance matrix for more challenging problems with heavy noise.

Acknowledgements

This work was done when R. Wang worked as a research associate at UMD. We are grateful to Dr. Tae-Kyun Kim, Dr. Hakan Cevikalp and Dr. Yiqun Hu for sharing their source codes. We also thank Dr. David Harwood, Dr. Xiaopeng Hong and Dr. Shiguang Shan for many helpful discussions. This research was partially supported by the ONR MURI grant N000141010934. R. Wang and Q. Dai were also supported in part by the Project of NSFC (No. 61035002, 60932007).

References

[1] O. Arandjelović, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face Recognition with Image Sets Using Manifold Density Divergence. *CVPR*, pp. 581–588, 2005.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric Means in A Novel Vector Space Structure on Symmetric Positive-definite Matrices. *SIAM J. Matrix Analysis and Applications*, vol. 29, no. 1, pp. 328-347, 2007.

[3] M. Barker and W. S. Rayens. Partial Least Squares for Discrimination. *J. of Chemometrics*, 17, pp. 166-173, 2003.

[4] G. Baudat and F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, vol. 12, pp. 2385–2404, 2000.

[5] H. Cevikalp and B. Triggs. Face Recognition Based on Image Sets. *CVPR*, pp. 2567-2573, 2010.

[6] W. Förstner and B. Moonen. A Metric for Covariance Matrices. Technical Report, Stuttgart Univ., 1999.

[7] R. Gross and J. Shi. The CMU Motion of Body (MoBo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, June 2001.

[8] J. Hamm and D. D. Lee. Grassmann Discriminant Analysis: a Unifying View on Subspace-Based Learning. *ICML*, 2008.

[9] H. Hotelling. Relations between Two Sets of Variates. *Biometrika*, vol. 28, no. 34, pp. 321-372, 1936.

[10] Y. Hu, A.S. Mian, and R. Owens. Sparse Approximated Nearest Points for Image Set Classification. *CVPR*, 2011.

[11] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face Tracking and Recognition with Visual Constraints in Real-World Videos. *CVPR*, 2008.

[12] T.K. Kim, O. Arandjelović, and R. Cipolla. Boosted Manifold Principal Angles for Image Set-based Recognition. *Pattern Recognition*. vol. 40, no. 9, pp. 2475-2484, 2007.

[13] T.K. Kim, J. Kittler, and R. Cipolla. Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations. *PAMI*, vol.29, no.6, pp.1005-1018, 2007.

[14] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman. Video-Based Face Recognition Using Probabilistic Appearance Manifolds. *CVPR*, pp. 313-320, June 2003.

[15] B. Leibe and, B.Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. *CVPR*, vol.2, pp. 409-415, 2003.

[16] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, J. Cheng, and G. Luo. Visual Tracking Via Incremental Log-Euclidean Riemannian Subspace Learning. *CVPR*, 2008.

[17] X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. *Int'l J. Computer Vision*, vol. 66, no. 1, pp. 41-66, 2006.

[18] R. Rosipal and N. Krämer. Overview and Recent Advances in Partial Least Squares. *SLSFS: Lecture Notes in Computer Science*, pp. 34-51, Springer, 2006.

[19] R. Rosipal and L. J. Trejo. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *J. Machine Learning Research*, vol.2, no.2, pp. 97-123, 2001.

[20] S. Satoh. Comparative Evaluation of Face Sequence Matching for Content-Based Video Access. *FG*, 2000.

[21] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[22] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face Recognition from Long-term Observations. *ECCV*, 2002.

[23] O. Tuzel, F. Porikli, and P. Meer. Region Covariance: A Fast Descriptor for Detection and Classification. *ECCV*, 2006.

[24] O. Tuzel, F. Porikli, and P. Meer. Pedestrian Detection via Classification on Riemannian Manifolds. *PAMI*, vol. 30, no. 10, pp. 1713-1727, October 2008.

[25] P. Viola and M. Jones. Robust Real-Time Face Detection. *Int'l J. Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.

[26] R. Wang, S. Shan, X. Chen, W. Gao. Manifold-Manifold Distance with Application to Face Recognition based on Image Set. *CVPR*, pp. 2940-2947, 2008.

[27] R. Wang and X. Chen. Manifold Discriminant Analysis. *CVPR*, pp. 429-436, 2009.

[28] H. Wold. Partial Least Squares. *Encyclopedia of Statistical Sciences*, vol. 6, pp. 581-591, Wiley, New York, 1985.

[29] L. Wolf and A. Shashua. Learning over Sets Using Kernel Principal Angles. *J. Machine Learning Research*, vol. 4, no. 10, pp. 913-931, 2003.

[30] O. Yamaguchi, K. Fukui, and K. Maeda. Face Recognition Using Temporal Image Sequence. *FG*, pp. 318-323, 1998.