2013 IEEE Conference on Computer Vision and Pattern Recognition

Kernel Learning for Extrinsic Classification of Manifold Features

Raviteja Vemulapalli, Jaishanker K. Pillai and Rama Chellappa
Department of Electrical and Computer Engineering
Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

Abstract

In computer vision applications, features often lie on Riemannian manifolds with known geometry. Popular learning algorithms such as discriminant analysis, partial least squares, support vector machines, etc., are not directly applicable to such features due to the non-Euclidean nature of the underlying spaces. Hence, classification is often performed in an extrinsic manner by mapping the manifolds to Euclidean spaces using kernels. However, for kernel based approaches, poor choice of kernel often results in reduced performance. In this paper, we address the issue of kernel-selection for the classification of features that lie on Riemannian manifolds using the kernel learning approach. We propose two criteria for jointly learning the kernel and the classifier using a single optimization problem. Specifically, for the SVM classifier, we formulate the problem of learning a good kernel-classifier combination as a convex optimization problem and solve it efficiently following the multiple kernel learning approach. Experimental results on image set-based classification and activity recognition clearly demonstrate the superiority of the proposed approach over existing methods for classification of manifold features.

1. Introduction

Many applications involving images and videos require classification of data that obey specific constraints. Such data often lie in non-Euclidean spaces. For instance, popular features and models in computer vision like shapes [10], histograms, covariance features [22] , linear dynamical systems (LDS) [6], etc., are known to lie on Riemannian manifolds. In such cases, one needs good classification techniques that make use of the underlying manifold structure.

For features that lie in Euclidean spaces, classifiers based on discriminative approaches such as linear discriminant analysis (LDA), partial least squares (PLS) and support vector machines (SVM) have been successfully used in various applications. However, these approaches are not directly applicable to features that lie on Riemannian manifolds. Hence, classification is often performed in an extrinsic manner by first mapping the manifold to an Euclidean space, and then learning classifiers in the new space. One such popularly used Euclidean space is the tangent space at the mean sample [22, 23]. However, tangent spaces preserves only the local structure of the manifold and can often lead to sub-optimal performance. An alternative approach is to map the manifold to a reproducing kernel Hilbert space (RKHS) [8, 9, 5] by using kernels. Though kernel-based methods have been successfully used in many computer vision applications, poor choice of kernel can often result in reduced classification performance. This is illustrated in figure 1. This gives rise to an important question: *How to find good kernels for Riemannian manifolds ?*.

In this paper, we answer this question using the kernel learning approach [14, 17], in which appropriate kernels are learned directly from the data. Since we are interested in learning good kernels for the purpose of classification, we learn the kernel and the classifier jointly by solving a single optimization problem. To learn a good kernel-classifier combination for features that lie on Riemannian manifolds, we propose the following two criteria: (i) Risk functional associated with the classifier in the mapped space should be minimized for good classification performance, (ii) The mapping should preserve the underlying manifold structure. The second criterion acts as a regularizer in learning the kernel. Our general framework for learning a good kernel-classifier combination can be represented as the following optimization problem

$$\min_{\mathcal{W},\mathcal{K}} \quad \lambda\, \Gamma_s(\mathcal{K}) + \Gamma_c(\mathcal{W}, \mathcal{K}),$$

where $\Gamma_s(\mathcal{K})$ and $\Gamma_c(\mathcal{W}, \mathcal{K})$ are respectively the manifold-structure and the classifier costs expressed as functions of the classifier parameters $\mathcal{W}$ and the kernel $\mathcal{K}$. Here, λ is the regularization parameter used to balance the two criteria.

Due to its superior generalization properties, we focus on using the SVM classifier in this paper. In order to preserve the manifold structure, we constrain the distances in the mapped space to be close to the manifold distances. Under this setting, we formulate the problem of learning a good kernel-classifier combination as a convex optimization problem. While the resulting formulation is an instance of semidefinite programming (SDP) and can be solved using

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.233

IEEE computer society

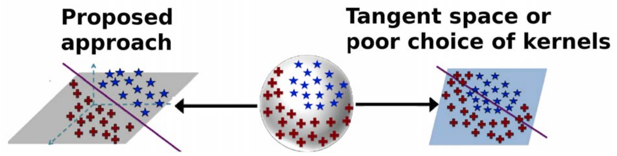

Figure 1: Tangent-space mapping or poorly-chosen kernel can often result in a bad classifier (right), whereas the proposed method (left) learns a mapping that is good for classification by using the classifier cost in the optimization.

standard solvers such as SeDuMi [19], it is transductive in nature: both training and test data need to be present while learning the kernel matrix. Solving SDPs is also computationally expensive for large datasets. To solve both these issues, we follow the multiple kernel learning (MKL) approach [14, 17] and parametrize the kernel as a linear combination of known base kernels. This formulation results in a much simpler convex optimization problem, which can be efficiently solved using gradient-based methods.

We performed experiments using two different manifold features: linear subspaces and covariance features, and three different applications: face recognition using image sets, object recognition using image sets and human activity recognition. The superior performance of the proposed approach clearly shows that it can be successfully used in classification applications that use manifold features.

Contributions: 1) We introduce a general framework for developing extrinsic classifiers for features that lie on Riemannian manifolds using the kernel learning approach. To the best of our knowledge, the proposed approach is the first one to use kernel learning techniques for classification of features that lie on Riemannian manifolds with known geometry. 2) We propose to use a geodesic distance-based regularizer for learning appropriate kernels directly from the data. 3) Focusing on the SVM classifier, we show that the problem of learning a good kernel-classifier combination can be formulated as a convex optimization problem.

Organization: We provide a brief review of the existing literature in section 2 and present the proposed approach in section 3. Section 4 briefly discusses the Riemannian geometry of two popularly used features, namely linear subspaces and covariance features. We present our experimental results in section 5 and conclude the paper in section 6.

2. Previous Work

Existing classification methods for Riemannian manifolds (with known geometry) can be broadly grouped into three main categories: nearest-neighbor methods, Bayesian methods, and Euclidean-mapping methods.

Nearest neighbor: The simplest classifier on a manifold is the nearest-neighbor classifier based on some appropriately defined distance or similarity measure. In [3], the trajectories of human joint positions were represented as subspaces using LDS models, and then classified using Martin and Finsler distances. In [23], LDS models were used to get subspace representations for shape deformations and the Frobenius distance was used for classification. In [27, 13, 12], image sets were modeled using linear subspaces and then compared using the largest canonical correlation in [27], the direct sum of canonical correlations in [13] and a weighted sum canonical correlations in [12].

Bayesian framework: Another possible approach for classification is to use the Bayesian framework by defining probability density functions (pdfs) on manifolds. In [21] parametric pdfs like Gaussian were defined on the tangent space and then wrapped back on to the manifold to define intrinsic pdfs for the Grassmann manifold. Alternatively, Parzen-window based non-parametric density estimation was used in [20] for the Stiefel manifold. Both these approaches along with Bayes classifier were used for human activity recognition and video-based face recognition. In general, parametric approaches are sensitive to the model order, whereas the model-free non-parametric approaches are very sensitive to the choice of window size.

Euclidean mapping: Discriminative approaches like LDA, PLS, SVM, Boosting, etc., can be extended to manifolds by mapping the manifolds to Euclidean spaces. One such Euclidean space is the tangent-space. In [22], a LogitBoost classifier was developed using weak classifiers learned on tangent spaces, and then used for pedestrian detection with covariance features. Tangent spaces only preserves the local structure of the manifold and can often lead to sub-optimal performance. Alternatively, one can map manifolds to Euclidean spaces by defining Mercer kernels on them. In [8, 9], discriminant analysis was used for image set-based recognition tasks using Grassmann kernels. In [24], a kernel defined for the manifold of symmetric positive definite matrices was used with PLS for image set-based recognition tasks. In [5], the Binet-Cauchy kernels defined on non-linear dynamical systems were used for human activity recognition. In general, the success of kernel-based methods is often determined by the choice of kernel. Hence, in this paper we address the issue of kernel-selection for the classification of manifold features.

The idea of using manifold structure as a regularizer was previously explored in the context of data manifolds [4, 18], where the given high dimensional data samples were simply assumed to lie on a lower dimensional manifold. Since the structure of the underlying manifold was unknown, a graph Laplacian-based empirical estimate of the data distribution was used in [4, 18]. Contrary to this, in this paper, we are interested in analytical manifolds, like Grassmann manifold and manifold of symmetric positive definite matrices, whose underlying geometry is known. Hence, the problem addressed in this paper is different from the one in [4, 18].

3. Extrinsic Support Vector Machines

Notations: The standard ℓ_2 norm of a vector $\vec{w}$ is denoted by $\|\vec{w}\|_2$. We use $\vec{1}$ and $\vec{0}$ to denote the column vectors of appropriate lengths with all ones and zeros respectively. We use $\vec{a} \leq \vec{b}$ to represent a set of element wise inequalities. $\mathcal{A}^T$ denotes the transpose of a matrix $\mathcal{A}$ and $\mathcal{A}\, o\, \mathcal{B}$ denotes the Hadamard product between $\mathcal{A}$ and $\mathcal{B}$. $\mathcal{K} \succeq 0$ ($\mathcal{K} \succ 0$) means $\mathcal{K}$ is symmetric and positive semi-definite (definite).

Let $\mathcal{M}$ denote the Riemannian manifold on which the features lie. Let $D_{tr} = \{(x_i, y_i),\ i = 1, \ldots, N_{tr}\}$ be the set of training samples where $y_i \in \{+1, -1\}$, $x_i \in \mathcal{M}$, and $D_{te} = \{x_i,\ i = N_{tr}+1, \ldots, N\}$ be the set of test samples. Let Φ be the mapping to be learned from the manifold $\mathcal{M}$ to some inner product space $\mathcal{H}$. Let $k(\cdot, \cdot)$ be the associated kernel function, and $\mathcal{K}$ be the associated kernel matrix.

$$\mathcal{K} = \begin{pmatrix} \mathcal{K}_{tr,tr} & \mathcal{K}_{tr,te} \\ \mathcal{K}_{te,tr} & \mathcal{K}_{te,te} \end{pmatrix} \in \mathcal{R}^{N \times N}. \tag{1}$$

Then, $\mathcal{K}_{ij} = k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j), \forall x_i, x_j \in \mathcal{M}$.

Since we are interested in performing classification in the mapped space, we jointly learn the kernel and the classifier using a single optimization problem based on the following two criteria:

(i) Risk minimization: For better classification performance, the risk functional associated with the classifier in the mapped space should be minimized .

(ii) Structure preservation: Since the features lie on a Riemannian manifold with a well defined structure, the mapping should be structure-preserving. This criterion can be seen as playing the role of a regularizer in kernel-learning.

Combining the above two criteria we formulate the problem of learning a good kernel-classifier combination as

$$\min_{\mathcal{W}, \mathcal{K}} \quad \lambda\, \Gamma_s(\mathcal{K}) + \Gamma_c(\mathcal{W}, \mathcal{K}), \tag{2}$$

where $\Gamma_s(\mathcal{K})$ and $\Gamma_c(\mathcal{W}, \mathcal{K})$ are the manifold-structure cost and the classifier cost expressed as functions of classifier parameters $\mathcal{W}$ and kernel matrix $\mathcal{K}$. Here, λ is the regularization parameter used to balance the two criteria. Since the mapped space is an inner product space, one can use standard machine learning techniques to perform classification. Due to its superior generalization properties, we focus on the SVM classifier in this paper. However, it is important to note that the framework introduced here is general and can be applied to other classifiers as well.

SVM classifier in the mapped space: The SVM classifier in the mapped space is given by $f(x) = \vec{w}^{*\top} \Phi(x) + b^*$, where the weight vector $\vec{w}^*$ and the bias b^* are given by

$$\vec{w}^*, b^* = \underset{\vec{w}, b, \eta}{\arg\min}\ \frac{1}{2}\|\vec{w}\|_2^2 + C \sum_{i=1}^{N_{tr}} \eta_i, \tag{3}$$

subject to $y_i(\vec{w}^\top \Phi(x_i) + b) \geq 1 - \eta_i,\ \eta_i \geq 0,\ i = 1, \ldots, N_{tr}$. This problem is usually solved in its dual form

$$\max_{\vec{\alpha} \in \Omega} \left(\vec{\alpha}^\top \vec{1} - \frac{1}{2} \vec{\alpha}^\top \left(\vec{y}\vec{y}^\top o\, \mathcal{K}_{tr,tr} \right) \vec{\alpha} \right), \tag{4}$$

where $\Omega = \{\vec{\alpha} \in \mathcal{R}^{N_{tr}} \mid \vec{0} \leq \vec{\alpha} \leq C\vec{1},\ \vec{\alpha}^\top \vec{y} = 0\}$, and $\vec{y}^\top = [y_1, \ldots, y_{N_{tr}}]$.

Preserving the manifold structure: To preserve the manifold structure, we constrain the distances in the mapped space to be close to the manifold distances. The squared Euclidean distance between two points x_i and x_j in the mapped space can be expressed in terms of kernel values as $\|\phi(x_i) - \phi(x_j)\|_2^2 = \mathcal{K}_{ii} + \mathcal{K}_{jj} - \mathcal{K}_{ij} - \mathcal{K}_{ji}$. Hence, we wish to minimize $\sum_{i=1}^N \sum_{j=1}^N \zeta_{ij}^2$, where $\zeta_{ij} = \mathcal{K}_{ii} + \mathcal{K}_{jj} - \mathcal{K}_{ij} - \mathcal{K}_{ji} - d_{ij}^2, \quad 1 \leq i < j \leq N$, and d_{ij} is the manifold distance between the points x_i and x_j. Since ζ_{ij} can be positive or negative, we use ζ_{ij}^2 in the cost.

Combined formulation: Combining both the classifier and the structure costs, the joint optimization problem for learning a good kernel-classifier combination is given by

$$\min_{\mathcal{K} \succeq 0,\ \vec{\zeta}} \ \max_{\vec{\alpha} \in \Omega} \ \lambda \|\vec{\zeta}\|_2^2 + \left(\vec{\alpha}^\top \vec{1} - \frac{1}{2} \vec{\alpha}^\top \left(\vec{y}\vec{y}^\top o\, \mathcal{K}_{tr,tr} \right) \vec{\alpha} \right),$$

$$\text{subject to} \quad \sum_{i=1}^N \sum_{j=1}^N \mathcal{K}_{ij} = 0,$$

$$\mathcal{K}_{ii} + \mathcal{K}_{jj} - \mathcal{K}_{ij} - \mathcal{K}_{ji} - d_{ij}^2 = \zeta_{ij} \ \text{ for } \ 1 \leq i < j \leq N, \tag{5}$$

where $\Omega = \{\vec{\alpha} \in \mathcal{R}^{N_{tr}} \mid \vec{0} \leq \vec{\alpha} \leq C\vec{1},\ \vec{\alpha}^\top \vec{y} = 0\}$, $\vec{\zeta}$ is the column vector of variables ζ_{ij} and $\vec{y} \in \mathcal{R}^{N_{tr}}$ is the column vector of class labels. In the above optimization problem, the centering constraint $\sum_{ij} \mathcal{K}_{ij} = 0$ is added simply to remove the ambiguity associated with the origin in the mapped space [25]. Note that in (5) we are learning the entire kernel matrix $\mathcal{K}$ directly in a non-parametric fashion, and the classifier term has only $\mathcal{K}_{tr,tr}$. Therefore, to ensure meaningful values for $\mathcal{K}_{tr,te}$ and $\mathcal{K}_{te,te}$, we need additional constraints between the training and test samples [14]. Hence, we use both the training and test samples in structure-preserving constraints.

Theorem 1: The optimal $\mathcal{K}$ for problem (5) can be found by solving a semidefinite programming problem.

Proof: This can be easily proved by following [14]. Due to space limitation, we omit the proof here.

SDPs are convex in nature and can be solved using standard solvers such as SeDuMi [19]. Once the kernel matrix $\mathcal{K}$ is obtained, the SVM classifier in the mapped space can be obtained by solving the SVM dual (4). Note that the above formulation is transductive in nature: both training and test data need to be present while learning the kernel matrix. Also in general, solving SDPs can be computationally expensive for large datasets. Both these issues can be addressed by using the MKL approach.

3.1. Extrinsic SVM Using MKL Framework

Instead of learning a non-parametric kernel matrix $\mathcal{K}$, following [14, 17], we parametrize the kernel as a linear combination of fixed base kernels $\mathcal{K}^1, \mathcal{K}^2, ..., \mathcal{K}^M$: $\mathcal{K} = \sum_{m=1}^{M} \mu_m \mathcal{K}^m$, where $\vec{\mu}^\top = [\mu_1, \ldots, \mu_m]$ are positive weights to be learned. Since we use the same linear model for both training and test data, the weights $\vec{\mu}$ can be learned using only the training data, and the kernel values for test data can be computed using the known base kernels and the learned weights. Hence, the formulation becomes inductive. Under the linear combination, the optimization problem (5) now becomes

$$\min_{\vec{\zeta},\, \vec{\mu}} \max_{\vec{\alpha} \in \Omega} \lambda \|\vec{\zeta}\|_2^2 + \left(\vec{\alpha}^\top \vec{1} - \frac{1}{2} \vec{\alpha}^\top (\vec{y}\vec{y}^\top o \sum_{m=1}^{M} \mu_m \mathcal{K}^m_{tr,tr}) \vec{\alpha} \right),$$

$$\text{subject to} \sum_{m=1}^{M} \mu_m (\mathcal{K}^m_{ii} + \mathcal{K}^m_{jj} - \mathcal{K}^m_{ij} - \mathcal{K}^m_{ji}) - d^2_{ij} = \zeta_{ij},$$

$$\text{for } 1 \le i < j \le N_{tr} \text{ and } \vec{\mu} \ge \vec{0}, \tag{6}$$

where $\Omega = \{\vec{\alpha} \in \mathcal{R}^{N_{tr}} \mid \vec{0} \le \vec{\alpha} \le C\vec{1},\ \vec{\alpha}^\top \vec{y} = 0\}$. Note that the centering constraint $\sum_{i,j} \mathcal{K}_{ij} = 0$ in (5) is not required for the MKL approach as the origin is automatically decided based on the base kernels and their weights.

Let p^m_{ij} denote the squared distance between samples x_i and x_j induced by the base kernel $\mathcal{K}^m$, i.e., $p^m_{ij} = \mathcal{K}^m_{ii} + \mathcal{K}^m_{jj} - \mathcal{K}^m_{ij} - \mathcal{K}^m_{ji}$. Let $J_1(\vec{\mu})$ and $J_2(\vec{\mu})$ represent the manifold-structure cost and the classifier cost respectively in (6). Then,

$$J_1(\vec{\mu}) = \sum_{i=1}^{N_{tr}} \sum_{j=i+1}^{N_{tr}} \zeta^2_{ij} = \sum_{i=1}^{N_{tr}} \sum_{j=i+1}^{N_{tr}} \left(\sum_{m=1}^{M} \mu_m p^m_{ij} - d^2_{ij} \right)^2,$$

$$J_2(\vec{\mu}) = \max_{\vec{\alpha} \in \Omega} \left(\vec{\alpha}^\top \vec{1} - \frac{1}{2} \vec{\alpha}^\top (\vec{y}\vec{y}^\top o \sum_{m=1}^{M} \mu_m \mathcal{K}^m_{tr,tr}) \vec{\alpha} \right). \tag{7}$$

Let Φ_m be the mapping corresponding to the kernel $\mathcal{K}^m$ and $\ell_h(f)$ be the hinge loss function: $\ell_h(f) = \max(0, 1 - f)$.

Theorem 2: $J_2(\vec{\mu}) = J_3(\vec{\mu})$, where

$$J_3(\vec{\mu}) = \min_{\vec{V}_m,\, b} \ \frac{1}{2} \sum_{m=1}^{M} \frac{\|\vec{V}_m\|_2^2}{\mu_m} + C \sum_{i=1}^{N_{tr}} \ell_h \left(y_i \left(\sum_{m=1}^{M} \vec{V}_m^\top \Phi_m(x_i) + b \right) \right). \tag{8}$$

Proof: The proof is based on Lagrangian duality. Please refer to [17] for details.

Let $h(\vec{\mu}) = \lambda J_1(\vec{\mu}) + J_3(\vec{\mu})$. Using Theorem 2, the optimization problem (6) can be written as

$$\min_{\vec{\mu}} \ h(\vec{\mu}) \quad \text{subject to } \vec{\mu} \ge \vec{0}. \tag{9}$$

Theorem 3: $h(\vec{\mu})$ is a differentiable convex function of $\vec{\mu}$ if $\mathcal{K}^m \succ 0$ for $m = 1, 2, ...M$.

Proof: $J_1(\vec{\mu})$ is a convex quadratic term and hence differentiable with respect to $\vec{\mu}$. As shown in [17], $J_3(\vec{\mu})$ is also convex and differentiable if all the base kernel matrices $\mathcal{K}^m$ are strictly positive definite. Hence $h(\vec{\mu})$ is a differentiable convex function of $\vec{\mu}$.

Using Theorem 3, the optimization problem (9) can be efficiently solved using the reduced gradient descent method [17] or any other standard algorithm used for solving constrained convex optimization problems. For any given $\vec{\mu}$, $J_1(\vec{\mu})$ can be evaluated directly using (7) and its gradient can be computed using

$$\frac{\partial J_1}{\partial \mu_m} = \sum_{i=1}^{N_{tr}} \sum_{j=i+1}^{N_{tr}} \left(2p^m_{ij} \left(\sum_{k=1}^{M} \mu_k p^k_{ij} - d^2_{ij} \right) \right). \tag{10}$$

Since $J_3(\vec{\mu}) = J_2(\vec{\mu})$, it can be computed by solving a standard SVM dual problem with $\mathcal{K} = \sum_{m=1}^{M} \mu_m \mathcal{K}^m$. The gradient of J_3 can be computed using [17]

$$\frac{\partial J_3}{\partial \mu_m} = -\frac{1}{2} \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{tr}} \alpha^*_i \alpha^*_j y_i y_j \mathcal{K}^m_{ij}, \tag{11}$$

where $\vec{\alpha}^*$ is the optimal solution for the SVM dual problem used for computing $J_3(\vec{\mu})$. Once the optimal $\vec{\mu}^*$ is computed, the classifier in the mapped space can be obtained by solving the SVM dual (4) with $\mathcal{K} = \sum_{m=1}^{M} \mu^*_m \mathcal{K}^m$. Note that Theorem 3 requires the Gram matrices $\mathcal{K}^m$ to be positive definite. To enforce this property a small ridge may be added to their diagonals.

4. Riemannian Manifolds in Computer Vision

In this section we briefly discuss the Riemannian geometry of two popularly used features, namely linear subspaces and covariance features, and show how these features are used in various computer vision applications.

4.1. Linear Subspaces - Grassmann Manifold

Grassmann manifold, denoted by $\mathscr{G}_{n,d}$, is the set of all d-dimensional linear subspaces of $\mathcal{R}^n$. An element S of $\mathscr{G}_{n,d}$ can be represented by any $n \times d$ orthonormal matrix Y_S such that span(Y_S) = S. The geodesic distance between two subspaces S_1 and S_2 on the Grassmann manifold is given by $\|\vec{\theta}\|_2$, where $\vec{\theta} = [\theta_1, \ldots, \theta_d]$ are the principal angles between S_1 and S_2. $\vec{\theta}$ can be computed using $\theta_i = \cos^{-1}(\alpha_i) \in [0, \frac{\pi}{2}]$, where α_i are the singular values of $Y^\top_{S1} Y_{S2}$. Other popularly used distances for the Grassmann manifold are the Procrustes metric given by $2(\sum_{i=1}^{d} \sin^2(\theta_i/2))^{1/2}$, and the Projection metric given by $(\sum_{i=1}^{d} \sin^2 \theta_i)^{1/2}$. We refer the interested readers to [7, 1] for further discussions on the Grassmann manifold.

Grassmann kernels: Grassmann manifold can be mapped to Euclidean spaces by using Mercer kernels [8]. One popularly used kernel [8, 9, 24] is the Projection kernel given by

$\mathcal{K}_P(Y_1, Y_2) = \|Y_1^\top Y_2\|_F^2$. The mapping corresponding to the Projection kernel is given by $\Phi_P(Y) = YY^\top$. Various kernels can be generated from $\mathcal{K}_P$ and Φ_P using

$$\begin{aligned} \mathcal{K}_P^{\text{rbf}}(Y_1, Y_2) &= \exp\left(-\gamma \|\Phi_P(Y_1) - \Phi_P(Y_2)\|_F^2\right), \\ \mathcal{K}_P^{\text{poly}}(Y_1, Y_2) &= \left(\gamma \mathcal{K}_P(Y_1, Y_2)\right)^d. \end{aligned} \tag{12}$$

We refer to the family of kernels K_P^{rbf} as projection-RBF kernels and the family of kernels K_P^{poly} as projection-polynomial kernels.

4.2. Covariance Features

The $d \times d$ symmetric positive definite (SPD) matrices, i.e., non-singular covariance matrices, can be formulated as a Riemannian manifold [16], and the resulting affine-invariant geodesic distance (AID) is given by $(\sum_{i=1}^d \ln^2 \lambda_i(C_1, C_2))^{1/2}$, where $\lambda_i(C_1, C_2)$ are the generalized Eigenvalues of SPD matrices C_1 and C_2. Another popularly used distance for SPD matrices is the log-Euclidean distance (LED) given by $\|log(C_1) - log(c_2)\|_F$, where log is the ordinary matrix logarithm and $\| \bullet \|_F$ denotes the matrix Frobenius norm. We refer the interested readers to [2, 16] for further details.

Kernels for SPD matrices: Similar to the Grassmann manifold, we can define kernels for the set of SPD matrices. One such kernel based on the log-Euclidean distance was derived in [24]: $\mathcal{K}_{log}(C_1, C_2) = \text{trace}[log(C_1)^\top log(C_2)]$. The mapping corresponding to $\mathcal{K}_{log}$ is given by $\Phi_{log}(C) = log(C)$. Various kernels can be generated from $\mathcal{K}_{log}$ and Φ_{log} using

$$\begin{aligned} \mathcal{K}_{log}^{\text{rbf}}(C_1, C_2) &= \exp\left(-\gamma \|\Phi_{log}(C_1) - \Phi_{log}(C_2)\|_F^2\right), \\ \mathcal{K}_{log}^{\text{poly}}(C_1, C_2) &= \left(\gamma \mathcal{K}_{log}(C_1, C_2)\right)^d. \end{aligned} \tag{13}$$

We refer to the family of kernels K_{log}^{rbf} as LED-RBF kernels and the family of kernels K_{log}^{poly} as LED-polynomial kernels.

4.3. Applications

Recognition using image sets: Given multiple images of the same face or object, they can be collectively represented [8, 9, 12, 13, 27] using a lower dimensional subspace obtained by applying the principal component analysis (PCA) on the feature vectors representing individual images. Let $S = [s_1, s_2, \ldots, s_N]$ be the mean-subtracted data matrix of an image set, where $s_i \in \mathcal{R}^n$ is an n-dimensional feature descriptor of i-th image. Let $V \Lambda V^T$ be the Eigen-decomposition of the data covariance matrix $C = SS^\top / N - 1$. Then the linear subspace spanned by the top d Eigenvectors can be used to represent the image set by a d-dimensional linear subspace. This d-dimensional linear subspace of the original n-dimensional space lies on the Grassmann manifold. Alternatively, the image set can also be represented using its natural second-order statistic [24], i.e., the covariance matrix C. Since covariance matrices are positive semi-definite in general, a small ridge may be added to their diagonals to make them positive definite.

Activity recognition using dynamical models: The autoregressive and moving average (ARMA) model is a dynamical model widely used in computer vision for modeling various kinds of time-series data [6, 3] and has been successfully used for activity recognition [21, 23, 5]. For an action video sequence ϕ, the ARMA model equations are given by

$$\begin{aligned} z_\phi(t+1) &= A(\phi) z_\phi(t) + v_\phi(t),\ v_\phi(t) \sim \mathcal{N}(\vec{0}, \Xi), \\ y_\phi(t) &= C(\phi) z_\phi(t) + w_\phi(t),\ \ w_\phi(t) \sim \mathcal{N}(\vec{0}, \Psi), \end{aligned} \tag{14}$$

where, $z_\phi(t) \in \mathcal{R}^d$ is the hidden state vector, $y_\phi(t) \in \mathcal{R}^p$ is the observed feature vector, $A(\phi) \in \mathcal{R}^{d \times d}$ and $C(\phi) \in \mathcal{R}^{p \times d}$ are the transition and measurement matrices. $v_\phi(t)$ and $w_\phi(t)$ are the noise components modeled as normal with zero mean and covariances $\Xi \in \mathcal{R}^{d \times d}$ and $\Psi \in \mathcal{R}^{p \times p}$ respectively. A closed form solution for parameters $(A(\phi), C(\phi))$ of the above model is available [6]. The expected observation sequence generated by a time-invariant model $(A(\phi), C(\phi))$, lies in the column space of the observability matrix $O_\infty(\phi) = [C(\phi),\ (C(\phi)A(\phi))^\top,\ (C(\phi)A(\phi)^2)^\top, \ldots]$ [21]. Following [21], instead of $O_\infty(\phi)$, we use a finite length approximation $O_m(\phi) \in \mathcal{R}^{mp \times d}$ given by $O_m^\top(\phi) = [C(\phi), (C(\phi)A(\phi))^\top, \ldots, (C(\phi)A(\phi)^{m-1})^\top]$ to represent the action sequence ϕ. The column space of $O_m(\phi)$ is a d-dimensional subspace of $\mathcal{R}^{mp}$ and hence is a point on the Grassmann manifold $\mathscr{G}_{mp,d}$. The orthonormal basis computed by Gram-Schmidt orthonormalization of $O_m(\phi)$ can be used to represent the action sequence ϕ as a point on the Grassmann manifold.

5. Experimental Evaluation

In this section, we evaluate the proposed approach using three applications where manifold features are used: (i) Face recognition using image sets, (ii) Object recognition using image sets and (iii) Human activity recognition from videos. We use two different manifold features, namely linear subspaces and covariance features.

5.1. Datasets and Feature Extraction

Face recognition – YouTube Celebrities [11]: This dataset has 1910 video clips of 47 subjects collected from the YouTube. Most of them are low resolution and highly compressed videos, making it a challenging dataset for face recognition. The face region in each image was extracted using a cascaded face detector, resized into 30 × 30 intensity image, and histogram equalized to eliminate lighting effects. Each video generated an image set of faces. Figure 2 shows some of the variations in an image set from this dataset.

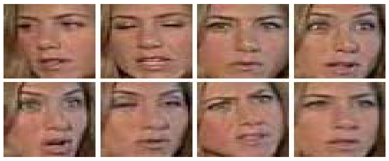

Figure 2: Variations in an image set from YouTube dataset.

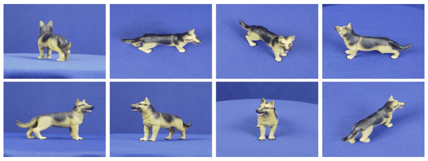

Figure 3: Variations in an image set from ETH80 dataset

Object recognition – ETH80 [15]: This benchmark dataset for object recognition task has images of 8 object categories with each category including 10 different object instances. Each object instance has 41 images captured under different views, which form an image set. All the images were resized into 20×20 intensity images. Figure 3 shows typical variations in an image set from this dataset.

For both of these datasets, we performed experiments with two different manifold features: covariance matrices and linear subspaces. As mentioned in section 4.3, to avoid matrix singularity, we added a small ridge δI to each covariance matrix C, where $\delta = 10^{-3} \times trace(C)$ and I is the identity matrix. For subspace representation, we used 20 dimensional linear subspaces spanned by the top 20 Eigenvectors of C.

Activity recognition – INRIA IXMAS [26]: This dataset consists of 10 actors performing 11 different actions, each action executed 3 times at varying rates while freely changing the orientation. We followed the same feature extraction procedure used in [21]. Specifically, for each segment of activity, we built a time series of motion history volumes using the segmentation results from [26]. Then each action was modeled as an ARMA process using the $16 \times 16 \times 16$ circular FFT features proposed by [26]. Following [21], the circular FFT features were reduced to 286 dimensions using PCA before building the dynamical model. The state space dimension d was chosen to be 5, and the observability matrix was truncated at $m = 5$.

5.2. Comparative Methods and Evaluation Settings

We compare our approach with the following methods:

(i) Nearest neighbor baseline (NN): We used three different distances for the Grassmann manifold, namely the geodesic distance, the Procrustes distance and the Projection metric. We report the best results among the three. For covariance features, we used two distances, namely the AID and the LED and report the best results among the two.

(ii) Standard MKL baseline (S-MKL) [17]: In the standard MKL approach, the kernel is learned as a convex combination of fixed base kernels ($\mathcal{K} = \sum_{m=1}^{M} \mu_m \mathcal{K}^m$, $\vec{\mu} \geq \vec{0}$, $\vec{\mu}^\top \vec{1} = 1$) , by minimizing the SVM cost (equation (4)) without manifold-based regularization.

iii) Statistical modeling (SM) [21]: This approach uses parametric (SM-P) and non-parametric (SM-NP) probability density estimation on the manifold followed by Bayes classification. For the parametric case, the Gaussian density was used in [21].

(iv) Grassmann discriminant analysis (GDA) [8]: Performs discriminant analysis followed by NN classification for the Grassmann manifold using the Projection kernel.

(v) PLS with the Projection kernel (Proj+PLS) [24]: Uses PLS combined with the Projection kernel for the Grassmann manifold.

(vi) Covariance discriminative learning (CDL) [24]: Uses discriminant analysis and PLS for covariance features using a kernel derived from the LED metric. Recently, state-of-the-art results were reported in [24] for image set-based face and object recognition tasks using this approach.

For the activity recognition experiment using the INRIA IXMAS dataset, we follow the round-robin (leave-one-person-out) experimental protocol used in [21]. For the object and face recognition experiments, we follow the settings used in [24]. For the YouTube dataset, for each person, we use 3 randomly chosen image sets for training and 6 for testing. For the ETH80 dataset, for each category, we use 5 randomly chosen image sets for training and 5 for testing. We report the results averaged over 10 random trials. The recognition rates reported for SM-P and SM-NP methods are taken from the original paper [21]. For GDA, Proj+PLS and CDL approaches, we use the recognition rates recently reported in [24].

5.3. Base Kernels and Parameters

For both the S-MKL and the proposed approach, we used several base kernels. For the experiments with linear subspaces, we used multiple projection-RBF and projection-polynomial kernels defined in (12). For each dataset, the values for the RBF parameter γ and the polynomial degree d were chosen based on their individual crossvalidation accuracy on the training data. Specifically, for the INRIA IXMAS dataset, we used 6 projection-polynomial kernels and 13 projection-RBF kernels. For the YouTube dataset, we used 10 projection-polynomial kernels and 15 projection-RBF kernels. For the ETH80 dataset, we used 10 projection-polynomial kernels and 13 projection-RBF kernels. The values for RBF kernel parameter γ were taken as $\frac{1}{n}2^{\delta}$, where n is the number of dimensions of Φ_P defined in section 4.1, and $\delta = \{-3, -1, \ldots, 19, 21\}$ for the INRIA IXMAS dataset, $\delta = \{-14, -12, \ldots, 12, 14\}$ for the YouTube dataset, $\delta = \{-5, -3, \ldots, 17, 19\}$ for the

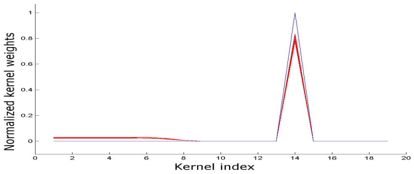

Figure 4: Normalized kernel weights for the S-MKL(blue) and the proposed method(red) on the INRIA IXMAS dataset

ETH80 dataset . Polynomial kernels were generated by taking $\gamma = \frac{1}{n}$ and varying the degree from 1 to 6 for the INRIA IXMAS dataset, and from 1 to 10 for the other two datasets.

For the experiments with covariance features, we used multiple LED-RBF and LED-polynomial kernels defined in (13), whose parameters were chosen based on their individual crossvalidation performance. Specifically, for the YouTube dataset, we used 10 LED-polynomial kernels and 15 LED-RBF kernels. For the ETH80 dataset, we used 10 LED-polynomial kernels and 20 LED-RBF kernels. The values for the RBF parameter γ were taken as $\frac{1}{n}2^{\delta}$, where n is the number of dimensions of Φ_{log} defined in section 4.2, and $\delta = \{-7, -6, \ldots, 6, 7\}$ for the YouTube dataset, $\delta = \{-10, -9, \ldots, 8, 9\}$ for the ETH80 dataset. For both datasets, polynomial kernels were generated by taking $\gamma = \frac{1}{n}$ and varying the degree from 1 to 10.

For both linear subspaces and covariance features, geodesic distances were used in the distance preserving constraints. In all the experiments, the parameters for the S-MKL method (SVM parameter C) and the proposed approach (SVM parameter C and the regularization parameter λ) were chosen using crossvalidation.

5.4. Results

Table 1 shows the recognition rates for human activity recognition using dynamical models. Tables 2 and 3 show the recognition rates for image set-based object and face recognition tasks using linear subspaces and covariance features respectively. We can see that the proposed approach clearly outperforms the nearest neighbor baseline method. On an average, the classification accuracy increases by 11.7%. This is expected as the simple nearest neighbor based classifier may not be powerful enough to handle the complex visual tasks considered. When compared to the S-MKL approach, the proposed approach performs better in four out of five experiments, with an average increase of 4.2% in the classification accuracy. This shows that the proposed manifold-based regularization is indeed helping in finding a better kernel for classification. On the INRIA IXMAS dataset, both the S-MKL and the proposed method gave same recognition rates. Figure 4 shows the normalized kernel weights for the S-MKL approach (blue) and the proposed approach (red) on the INRIA IXMAS dataset. The horizontal axis corresponds to the kernel index varying from 1 to 19 (13 RBF followed by 6 polynomial kernels). We can see that the kernel weights roughly follow the same pattern for both the approaches. This explains their similar performance on the INRIA IXMAS dataset. In the case of other datasets, the S-MKL approach mostly picked few base kernels (usually RBF kernels with high γ value or polynomial kernels of high degree d), whereas the weights for the proposed approach were distributed over many kernels.

Table 1: Recognition rates for human activity recognition on the INRIA IXMAS dataset using dynamical models

dataset	NN	S-MKL [17]	SM-P [21]	SM-NP [21]	Proposed approach
INRIA IXMAS	80.0	**90.0**	82.4	87.87	**90.0**

Table 2: Recognition rates for image set-based face and object recognition tasks using linear subspaces

dataset	NN	S-MKL [17]	GDA [8]	Proj + PLS [24]	Proposed approach
YouTube	62.8	64.3	65.7	67.7	**70.8**
ETH80	93.2	93.7	92.8	95.3	**96.0**

Table 3: Recognition rates for image set-based face and object recognition tasks using covariance features

dataset	NN	S-MKL [17]	CDL-LDA [24]	CDL-PLS [24]	Proposed approach
YouTube	40.7	69.7	67.5	70.1	**73.2**
ETH80	92.7	93.7	94.5	96.5	**98.2**

We can also see that the proposed approach clearly performs better than statistical and other kernel-based methods. The poor performance of SM-P method can be attributed to the Gaussian assumption. In the case of parametric density estimation, the mismatch between the assumed distribution and the actual underlying distribution often results in reduced performance. In the case of SM-NP, the poor performance could be due to the sub-optimal choice of the kernel width used in [21]. In general, non-parametric density estimation methods are sensitive to the choice of kernel width and a sub-optimal choice often results in poor performance [21]. The relatively lower performance of the other kernel-based methods suggests that, it is effective to jointly learn the kernel and the classifier directly from the data using the proposed framework.

Recently, covariance feature combined with PLS has been shown [24] to perform better than various other recent methods for image set-based recognition tasks. Our results show that the classification performance can be further improved by combining the covariance feature with the proposed approach.

6. Conclusion and Future Work

In this paper, we introduced a general framework for developing extrinsic classifiers for features that lie on Riemannian manifolds using the kernel learning approach. We proposed two criteria for learning a good kernel-classifier combination for manifold features. In the case of SVM classifier, based on the proposed criteria, we showed that the problem of learning a good kernel-classifier combination can be formulated as a convex optimization problem and efficiently solved following the multiple kernel learning approach. We performed experiments using two popularly used manifold features and obtained superior performance compared to other relevant approaches.

Though we focused on the SVM classifier in this paper, the proposed approach is general and we plan to extend it to other classifiers in the future. In this paper, the manifold structure has been used as a regularizer using simple distance-preserving constraints. Another possible direction of future work is to explore more sophisticated regularizers that can make use of the underlying manifold structure.

Acknowledgements: This research was supported by a MURI grant from the US Office of Naval Research under N00014-10-1-0934.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

[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 Applications*, 29(1):328–347, 2006.

[3] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto. Recognition of Human Gaits. In *CVPR*, 2001.

[4] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *JMLR*, 7:2399–2434, 2006.

[5] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of Oriented Optical Flow and Binet-Cauchy Kernels on Nonlinear Dynamical Systems for the Recognition of Human Actions. In *CVPR*, 2009.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic Textures. *IJCV*, 51(2):91–109, 2003.

[7] A. Edelman, T. A. Arias, and S. T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM J. Matrix Analysis Applications*, 20(2):303–353, 1998.

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

[9] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph Embedding Discriminant Analysis on Grassmannian Manifolds for Improved Image Set Matching. In *CVPR*, 2011.

[10] D. G. Kendall. Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

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

[12] T. K. Kim, O. Arandjelović, and R. Cipolla. Boosted Manifold Principal Angles for Image Set-Based Recognition. *Pattern Recognition*, 40(9):2475–2484, 2007.

[13] T. K. Kim, J. Kittler, and R. Cipolla. Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations. *PAMI*, 29(6):1005–1018, 2007.

[14] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *JMLR*, 5:27–72, 2004.

[15] B. Leibe and B. Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. In *CVPR*, 2003.

[16] X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. *IJCV*, 66(1):41–66, 2006.

[17] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

[18] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, 2005.

[19] J. F. Sturm. Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[20] P. K. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision. In *CVPR*, 2008.

[21] P. K. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video-Based Recognition. *PAMI*, 33(11):2273–2286, 2011.

[22] O. Tuzel, F. Porikli, and P. Meer. Pedestrian Detection via Classification on Riemannian Manifolds. *PAMI*, 30(10):1713–1727, 2008.

[23] A. Veeraraghavan, A. K. R. Chowdhury, and R. Chellappa. Matching Shape Sequences in Video with Applications in Human Movement Analysis. *PAMI*, 27(12):1896–1909, 2005.

[24] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification. In *CVPR*, 2012.

[25] K. Q. Weinberger and L. K. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. In *CVPR*, 2004.

[26] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition Using Motion History Volumes. *CVIU*, 104(2):249–257, 2006.

[27] O. Yamaguchi, K. Fukui, and K. ichi Maeda. Face Recognition Using Temporal Image Sequence. In *FG*, 1998.