Chapter 6
Dictionary Learning on Grassmann Manifolds

Mehrtash Harandi, Richard Hartley, Mathieu Salzmann and Jochen Trumpf

Abstract Sparse representations have recently led to notable results in various visual recognition tasks. In a separate line of research, Riemannian manifolds have been shown useful for dealing with features and models that do not lie in Euclidean spaces. With the aim of building a bridge between the two realms, we address the problem of sparse coding and dictionary learning in Grassmann manifolds, i.e, the space of linear subspaces. To this end, we introduce algorithms for sparse coding and dictionary learning by embedding Grassmann manifolds into the space of symmetric matrices. Furthermore, to handle nonlinearity in data, we propose positive definite kernels on Grassmann manifolds and make use of them to perform coding and dictionary learning.

6.1 Introduction

In the past decade, sparsity has become a popular term in neuroscience, information theory, signal processing, and related areas [7, 11, 12, 33, 46]. Through sparse representation and compressive sensing, it is possible to represent natural signals like images using only a few nonzero coefficients of a suitable basis. In computer vision, sparse and overcomplete image representations were first introduced for modeling the spatial receptive fields of simple cells in the human visual system by [33]. The

M. Harandi (✉) · R. Hartley · J. Trumpf
College of Engineering and Computer Science, Australian National University,
Canberra, ACT 2601, Australia
e-mail: mehrtash.harandi@anu.edu.au

R. Hartley
e-mail: richard.hartley@anu.edu.au

J. Trumpf
e-mail: jochen.trumpf@anu.edu.au

M. Salzmann
CVLab, EPFL, Lausanne, Switzerland
e-mail: mathieu.salzmann@epfl.ch

H.Q. Minh and V. Murino (eds.), *Algorithmic Advances in Riemannian Geometry and Applications*, Advances in Computer Vision and Pattern Recognition, DOI 10.1007/978-3-319-45026-1_6

linear decomposition of a signal using a few atoms of a dictionary has been shown to deliver notable results for various visual inference tasks, such as face recognition [46, 47], image classification [30, 48], subspace clustering [13] and image restoration [31] to name a few. While significant steps have been taken to develop the theory of the sparse coding and dictionary learning in Euclidean spaces, similar problems on non-Euclidean geometry have received comparatively little attention [8, 20, 22, 26]. This chapter discusses techniques to sparsely represent p-dimensional linear subspaces in $\mathbf{R}^d$ using a combination of linear subspaces.

Linear subspaces can be considered as the core of many inference algorithms in computer vision and machine learning. Examples include but not limited to modeling the reflectance function of Lambertian objects [4, 34], video analysis [9, 14, 18, 21, 41, 42], chromatic noise filtering [39], domain adaptation [16, 17], and object tracking [37]. Our main motivation here is to develop new methods for analyzing video data and image sets. This is inspired by the success of sparse signal modeling and related topics that suggest natural signals like images (and hence video and image sets as our concern here) can be efficiently approximated by superposition of atoms of a dictionary. We generalize the traditional notion of coding, which operates on vectors, to coding on subspaces. Coding with subspaces can then be seamlessly used for categorizing video data. Toward this, we first provide efficient solutions to the following two fundamental problems on Grassmann manifolds: (see Fig. 6.1 for a conceptual illustration):

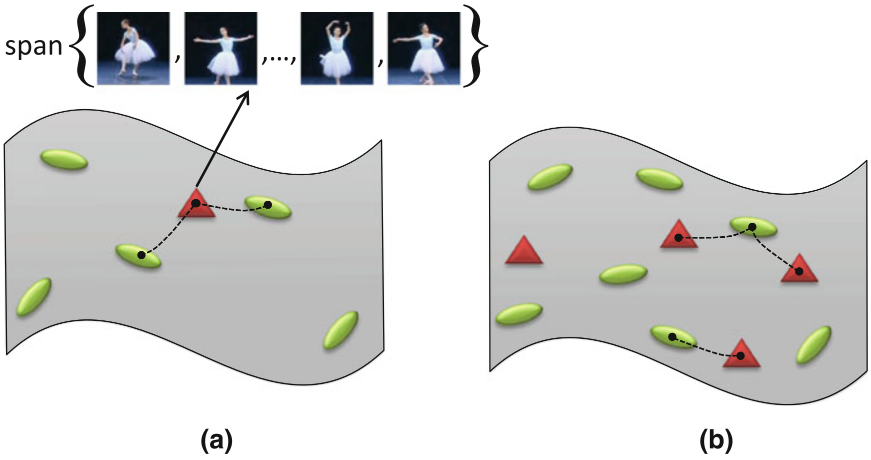

Fig. 6.1 A conceptual diagram of the problems addressed in this work. A video or an image set can be modeled by a linear subspace, which can be represented as a point on a Grassmann manifold. **a Sparse coding on a Grassmann manifold**. Given a dictionary (*green ellipses*) and a query signal (*red triangle*) on the Grassmann manifold, we are interested in estimating the query signal by a sparse combination of atoms while taking into account the geometry of the manifold (e.g, curvature). **b Dictionary learning on a Grassmann manifold**. Given a set of observations (*green ellipses*) on a Grassmann manifold, we are interested in determining a dictionary (*red triangles*) to describe the observations sparsely, while taking into account the geometry. This figure is best seen in color

1. **Coding**. Given a subspace $\mathscr{X}$ and a set $\mathbb{D} = \{\mathscr{D}_i\}_{i=1}^N$ with N elements (also known as atoms), where $\mathscr{X}$ and $\mathscr{D}_i$ are linear subspaces, how can $\mathscr{X}$ be approximated by a combination of atoms in $\mathbb{D}$?
2. **Dictionary learning**. Given a set of subspaces $\{\mathscr{X}_i\}_{i=1}^m$, how can a smaller set of subspaces $\mathbb{D} = \{\mathscr{D}_i\}_{i=1}^N$ be learned to represent $\{\mathscr{X}_i\}_{i=1}^m$ accurately?

Later, we tackle the problem of coding and dictionary learning on Grassmannian by embedding the manifold in Reproducing Kernel Hilbert Spaces (RKHS). To this end, we introduce a family of positive definite kernels on Grassmannian and make use of them to recast the coding problem in kernel spaces.

6.2 Problem Statement

In this section, we formulate the problem of coding and dictionary learning on the Grassmannian. Throughout this chapter, bold capital letters denote matrices (e.g X) and bold lowercase letters denote column vectors (e.g, x). The notation x_i (respectively $X_{i,j}$) is used to demonstrate the element in position i of the vector x (respectively (i, j) of the matrix X). $1_d \in \mathbf{R}^d$ and $0_d \in \mathbf{R}^d$ are vectors of ones and zeros. $\mathbf{I}_d$ is the $d \times d$ identity matrix. $\|x\|_1 = \sum_i |x_i|$ and $\|x\| = \sqrt{x^T x}$ denote the ℓ_1 and ℓ_2 norms, respectively, with T indicating transposition. $\|X\|_F = \sqrt{\mathrm{Tr}\left(X^T X\right)}$ designates the Frobenius norm, with $\mathrm{Tr}(\cdot)$ computing the matrix trace.

In vector spaces, by *coding* we mean the general notion of representing a vector x (the *query*) as some combination of other vectors d_i belonging to a *dictionary*. Typically, x is expressed as a linear combination $x = \sum_{j=1}^N y_j d_j$, or else as an *affine combination* in which the coefficients y_j satisfy the additional constraint $\sum_{j=1}^N y_j = 1$. (This constraint may also be written as $1^T y = 1$.)

In *sparse coding* one seeks to express the query in terms of a small number of dictionary elements. Given, a query $x \in \mathbf{R}^d$ and a dictionary $\mathbb{D}$ of size N, i.e, $\mathbb{D}_{d\times N} = \{d_1, d_2, \ldots, d_N\}$ with atoms $d_i \in \mathbf{R}^d$, the problem of coding x can be formulated as solving the minimization problem:

$$\ell_E(x, \mathbb{D}) \triangleq \min_y \left\| x - \sum\nolimits_{j=1}^N y_j d_j \right\|_2^2 + \lambda f(y). \tag{6.1}$$

The domain of y may be the whole of $\mathbf{R}^N$, so that the sum runs over all linear combinations of dictionary elements (or *atoms*), or alternatively, the extra constraint $1^T y = 1$ may be specified, to restrict to affine combinations.

The idea here is to (approximately) reconstruct the query x by a combination of dictionary atoms while forcing the coefficients of combination, i.e, y, to have some structure. The quantity $\ell_E(x, \mathbb{D})$ can be thought of as a coding cost combining the squared residual coding error, reflected in the energy term $\|\cdot\|_2^2$ in (6.1), along with a penalty term $f(y)$, which encourages some structure such as sparsity. The function

$f : \mathbf{R}^N \to \mathbf{R}$ could be the ℓ_1 norm, as in the Lasso problem [40], or some form of locality as proposed in [43].

The problem of dictionary learning is to determine $\mathbb{D}$ given a finite set of observations $\{x_i\}_{i=1}^m$, $x \in \mathbf{R}^d$, by minimizing the total coding cost for all observations, namely

$$h(\mathbb{D}) \triangleq \sum_{i=1}^{m} \ell_E(x_i, \mathbb{D}) \ , \tag{6.2}$$

while enforcing certain constraints on $\mathbb{D}$ to be satisfied to avoid trivial solutions. A "good" dictionary has a small residual coding error for all observations x_i while producing codes $y_i \in \mathbf{R}^N$ with the desired structure. For example, in the case of sparse coding, the ℓ_1 norm is usually taken as $f(\cdot)$ to obtain the most common form of dictionary learning in the literature. More specifically, the sparse dictionary learning problem may be written in full as that of jointly minimizing the total coding cost over all choices of coefficients and dictionary:

$$\min_{\{y_i\}_{i=1}^m, \mathbb{D}} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{N} y_{ij} d_j \right\|_2^2 + \lambda \sum_{i=1}^{m} \|y_i\|_1 . \tag{6.3}$$

A common approach to solving this is to alternate between the two sets of variables, $\mathbb{D}$ and $\{y_i\}_{i=1}^m$, as proposed for example by [2] (see [12] for a detailed treatment). Minimizing (6.3) over sparse codes y_i while dictionary $\mathbb{D}$ is fixed is a convex problem. Similarly, minimizing the overall problem over $\mathbb{D}$ with fixed $\{y_i\}_{i=1}^m$ is convex as well.

In generalizing the coding problem to a more general space $\mathcal{M}$, (e.g, Riemannian manifolds), one may write (6.1) as

$$\ell_{\mathcal{M}}(\mathcal{X}, \mathbb{D}) \triangleq \min_{y} \Big(d_{\mathcal{M}}\big(\mathcal{X},\ C(y, \mathbb{D})\big)^2 + \lambda f(y) \Big). \tag{6.4}$$

Here $\mathcal{X}$ and $\mathbb{D} = \{\mathcal{D}_j\}_{j=1}^N$ are points in the space $\mathcal{M}$, while $d_{\mathcal{M}}(\cdot, \cdot)$ is some distance metric and $C : \mathbf{R}^N \times \mathcal{M}^N \to \mathcal{M}$ is an *encoding function*, assigning an element of $\mathcal{M}$ to every choice of coefficients and dictionary. Note that (6.1) is a special case of this, in which $C(y, \mathbb{D})$ represents linear or affine combination, and $d_{\mathcal{M}}(\cdot, \cdot)$ is the Euclidean distance metric. To define the coding, one need only specify the metric $d_{\mathcal{M}}(\cdot, \cdot)$ to be used and the encoding function $C(\cdot, \cdot)$. Although this formulation may apply to a wide range of spaces, here we shall be concerned chiefly with coding on Grassmann manifolds.

A seemingly straightforward method for coding and dictionary learning is through embedding manifolds into Euclidean spaces via a fixed tangent space (the concepts related to differential geometry, such as tangent spaces will be shortly defined). The

Algorithm 1: Log-Euclidean sparse coding on Grassmann manifolds.

```
Input: Grassmann dictionary $\{\mathscr{D}_i\}_{i=1}^N$, $\mathscr{D}_i \in \mathscr{G}(p,d)$; the query sample $\mathscr{X} \in \mathscr{G}(p,d)$.
Output: The sparse code $y^*$.

Initialization.
    for $i \leftarrow 1$ to $N$ do
        $d_i \leftarrow \log_{\mathscr{P}}(\mathscr{D}_i)$;
    end
    $A \leftarrow [d_1|d_2|\cdots|d_N]$ ;

Processing.
    $x \leftarrow \log_{\mathscr{P}}(\mathscr{X})$;
    $y^* \leftarrow \arg\min_y \left\|x - A^T y\right\|_2^2 + \lambda \, \|y\|_1$;
```

embedding function in this case would be $\log_{\mathscr{P}}(\cdot)$, where $\mathscr{P}$ is some default base point.[1]

By mapping points in the manifold $\mathscr{M}$ to the tangent space, the problem at hand is transformed to its Euclidean counterpart. For example in the case of sparse coding, the encoding cost may be defined as follows:

$$\ell_{\mathscr{M}}(\mathscr{X}, \mathbb{D}) \triangleq \min_{y} \left\| \log_{\mathscr{P}}(\mathscr{X}) - \sum_{j=1}^{N} y_j \log_{\mathscr{P}}(\mathscr{D}_j) \right\|_{\mathscr{P}}^2 + \lambda \|y\|_1 \tag{6.5}$$

where the notation $\|\cdot\|_{\mathscr{P}}$ reminds us that the norm is in the tangent space at $\mathscr{X}$. We shall refer to this straightforward approach as *Log-Euclidean* sparse coding (the corresponding steps for Grassmann manifolds in Algorithm 1), following the terminology used in [3]. Since on a tangent space only distances to the base point are equal to true geodesic distances, the Log-Euclidean solution does not take into account the true structure of the underlying Riemannian manifold. Moreover, the solution is dependent upon the particular point $\mathscr{P}$ used as a base point.

Another approach is to measure the loss of $\mathscr{X}$ with respect to the dictionary $\mathbb{D}$, i.e, $\ell_{\mathscr{M}}(\mathscr{X}, \mathbb{D})$, by working in the tangent space of $\mathscr{X}$, rather than a fixed point $\mathscr{P}$ [8, 26]. The loss in this case becomes

$$\ell_{\mathscr{M}}(\mathscr{X}, \mathbb{D}) \triangleq \min_{\substack{y \in \mathbf{R}^N \\ 1^T y = 1}} \left\| \sum_{j=1}^{N} y_j \log_{\mathscr{X}}(\mathscr{D}_j) \right\|_{\mathscr{X}}^2 + \|y\|_1 \tag{6.6}$$

[1] The function that maps each vector $y \in T_{\mathscr{P}}\mathscr{M}$ to a point $\mathscr{X}$ of the manifold that is reached after a unit time by the geodesic starting at $\mathscr{P}$ with this tangent vector is called the *exponential map*. For complete manifolds, this map is defined in the whole tangent space $T_{\mathscr{P}}\mathscr{M}$. The *logarithm map* is the inverse of the exponential map, i.e, $y = \log_{\mathscr{P}}(\mathscr{X})$ is the smallest vector y such that $\mathscr{X} = \exp_{\mathscr{P}}(y)$.

Following [26], given a set of training data $\{\mathcal{X}_i\}_{i=1}^m$, $\mathcal{X}_i \in \mathcal{M}$, the problem of dictionary learning can be written as

$$\min_{\{y_i\}_{i=1}^m, \mathbb{D}} \sum_{i=1}^m \left\| \sum_{j=1}^N y_{ij} \log_{\mathcal{X}_i}(\mathcal{D}_j) \right\|^2 + \lambda \sum_{i=1}^m \|y_i\|_1 \tag{6.7}$$
$$\text{s.t.} \quad 1^T y_i = 1, \; i = 1, 2, \ldots, m.$$

Similar to the Euclidean case, the problem in (6.7) can be solved by iterative optimization over $\{y_i\}_{i=1}^m$ and $\mathbb{D}$. Computing the sparse codes $\{y_i\}_{i=1}^m$ is done by solving (6.6). To update $\mathbb{D}$, [26] proposed a gradient descent approach along geodesics. That is, the update of $\mathcal{D}_r$ at time t while $\{y_i\}_{i=1}^m$ and $\mathcal{D}_j, j \neq r$ are kept fixed has the form

$$\mathcal{D}_r^{(t)} = \exp_{\mathcal{D}_r^{(t-1)}}(-\eta\Delta). \tag{6.8}$$

In Eq. (6.8) η is a step size and the tangent vector $\Delta \in T_{\mathcal{D}_r}(\mathcal{M})$ represents the direction of maximum ascent. That is $\Delta = \operatorname{grad} \mathcal{J}(\mathcal{D}_r)$,[2] where

$$\mathcal{J} = \sum_{i=1}^m \left\| \sum_{j=1}^N y_{ij} \log_{\mathcal{X}_i}(\mathcal{D}_j) \right\|^2. \tag{6.9}$$

Here is where the difficulty arises. Since the logarithm map does not have a closed-form expression on Grassmann manifolds, an analytic expression for Δ in Eq. (6.8) cannot be sought for the case of interest in this work, i.e, Grassmann manifolds. Having this in mind, we will describe various techniques to coding and dictionary learning specialized for Grassmann manifolds.

6.3 Background Theory

This section overviews Grassmann geometry and provides the groundwork for techniques described in following sections. Since the term "manifold" itself is often used in computer vision in a somewhat loose sense, we emphasize that the word is used in this chapter in its strict mathematical sense.

One most easily interprets the Grassmann manifolds in the more general context of group actions, to be described first. Consider a transitive (left) group action of a group G on a set S. The result of applying a group element g to a point $x \in S$ is written as gx. Choose a specific point $x_0 \in S$ and consider its *stabilizer* $\mathrm{Stab}(x_0) =$

[2] On an abstract Riemannian manifold $\mathcal{M}$, the gradient of a smooth real function f at a point $x \in \mathcal{M}$, denoted by $\operatorname{grad} f(x)$, is the element of $T_x(\mathcal{M})$ satisfying $\langle \operatorname{grad} f(x), \zeta \rangle_x = Df_x[\zeta]$ for all $\zeta \in T_x(\mathcal{M})$. Here, $Df_x[\zeta]$ denotes the directional derivative of f at x in the direction of ζ. The interested reader is referred to [1] for more details on how the gradient of a function on Grassmann manifolds can be computed.

$\{g \in G \mid gx_0 = x_0\}$. The stabilizer is a subgroup of G, which we will denote by H, and there is a one-to-one correspondence between the left cosets gH and the elements of S, whereby a point $x \in S$ is associated with the coset $\{g \in G \mid gx_0 = x\}$. The set of all left cosets is denoted by G/H, which we identify with the set S under this identification.

The (real, unoriented) *Grassmannian* $\mathscr{G}(p, d)$, where $0 < p \leq d$, is the set of all p-dimensional linear subspaces (we shall usually call them p-planes, or simply planes) of the real vector space $\mathbf{R}^d$. A geometric visualization of a point in the Grassmannian is a p-plane through the origin of d-dimensional Euclidean space. From this geometric picture, it is obvious that the *orthogonal group* $O(d)$ of all real orthogonal transforms of $\mathbf{R}^d$ acts on $\mathscr{G}(p, d)$, as any element of $O(d)$ transforms a p-plane through the origin to a p-plane through the origin. The action is transitive, since any plane can be reached in this way from any given one. The set of elements of $O(d)$ that transforms a given p-plane $\mathscr{X}$ to itself, the stabilizer $\text{Stab}(\mathscr{X}) = \{U \in O(d) \mid U\mathscr{X} = \mathscr{X}\}$, is a subgroup of $O(d)$ and isomorphic to the product $O(p) \times O(d-p)$, where the factor $O(p)$ corresponds to in-plane transformations and the factor $O(d-p)$ corresponds to transformations that leave all points of the plane fixed. The Grassmannian can hence be identified with the coset space $O(d)/(O(p) \times O(d-p))$.

To make the above discussion more concrete, in terms of matrices, identify $O(d)$ as the group of orthogonal $d \times d$ matrices, a Lie group of dimension $d \times (d+1)/2$. In addition, think of p-planes in $\mathbf{R}^d$ as represented by *orthogonal Stiefel matrices*, that is, by rectangular $d \times p$-matrices X with orthonormal columns that form a basis of the plane. Since a given plane has many different orthogonal bases, two such matrices X and X' represent the same plane if and only if there exists a matrix $V \in O(p)$ such that $X' = XV$. This defines an equivalence relation between matrices X and X', and the planes may be identified with the equivalence classes of $d \times p$ matrices under this equivalence relation. The set of all such equivalence classes constitute the Grassmannian $\mathscr{G}(p, d)$.

A matrix U in $O(d)$ acts on a plane with representative Stiefel matrix X by left multiplication: $X \mapsto UX$. One immediately verifies that if X and X' represent the same plane, then so do UX and UX', so this defines a transitive left group action of $O(d)$ on $\mathscr{G}(p, d)$. Of particular interest is the element of the Grassmannian,

$$\mathscr{X}_0 = \text{Span}(X_0) = \text{Span}\begin{bmatrix} I_p \\ 0 \end{bmatrix},$$

where I_p is the $p \times p$ identity matrix. A transformation $U \in O(d)$ acts by left matrix multiplication and it is immediate that

$$\text{Stab}(\mathscr{X}_0) = \left\{ \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \in O(d) \;\middle|\; U_1 \in O(p),\ U_2 \in O(d-p) \right\}, \tag{6.10}$$

showing again that $\text{Stab}(\mathscr{X}_0) \simeq O(p) \times O(d-p)$.

Since the action of $O(d)$ is transitive on $\mathcal{G}(p, d)$, the elements of $\mathcal{G}(p, d)$ (planes) are in one-to-one correspondence with the set of left cosets of $\mathrm{Stab}(\mathcal{X}_0)$ in $O(d)$. We think of the Grassmannian as the coset space $O(d)/(O(d-p) \times O(p))$, where $O(d-p) \times O(p)$ is identified with the block-diagonal subgroup in Eq. (6.10).

In this way, it is seen that the Grassmann manifolds form a special case of coset spaces G/H, in which $G = O(d)$ and H is the subgroup $O(d-p) \times O(p)$. In the Grassmann case, the group $G = O(d)$ is a matrix Lie group, and hence has the topological structure of a compact manifold. In this situation, G/H inherits a topology, a smooth manifold structure, and indeed a Riemannian metric from $G = O(d)$.

We continue the discussion denoting by G the matrix Lie group $O(d)$ and $H = \mathrm{Stab}\mathcal{X}_0$, shown in Eq. (6.10), but the reader may bear in mind that the discussion holds equally well in the case of a general compact (sometimes also non-compact) Lie group G with Lie subgroup H.[3] This topic is treated in Chap. 21 of [29], which the reader may consult to fill in details.

The natural projection $\pi : G \to G/H \simeq \mathcal{G}(p, d)$ can now be used to equip the Grassmannian with quotient structures. For example, using the standard Lie group topology on G, the quotient as a quotient topology (the strongest topology such that π is continuous). Using the differential structure of the Lie group G, the quotient inherits a differential structure (the unique one that makes π a smooth submersion). With this differential structure, the action of G on G/H is smooth. With this smooth structure, the quotient space G/H is a manifold, according to the quotient manifold theorem (see Theorem 21.10 in [29]). Thus, the Grassmannian (or *Grassmann manifold*) is thus a *homogeneous space* of G. Its dimension is $p(d-p) = \dim(O(d)) - (\dim(O(d-p)) + \dim(O(p)))$.[4]

Tangent Space and Riemannian Metric

The homogeneous space structure of the Grassmannian can be used to equip it with a Riemannian metric, starting from a bi-invariant Riemannian metric on $G = O(d)$.

To this end, think of G as embedded in $\mathbf{R}^{d \times d}$ and equip the tangent space $T_I G$ at the identity with the Euclidean inner product inherited from that embedding. This defines a biinvariant Riemannian metric on G through right (or left) translation. For subgroup H of G define the Riemannian metric on the quotient G/H by

$$\langle X, Y \rangle_{UH} = \langle \tilde{X}, \tilde{Y} \rangle_U$$

where $U \in G$ and $X, Y \in T_{UH} G/H$. Further, $\tilde{X}$ and $\tilde{Y}$ in $T_U(G)$ are the horizontal lifts of X and Y with respect to π and the Riemannian metric on the group, that is, $\pi_U^*(\tilde{X}) = X$ and $\pi_U^*(\tilde{Y}) = Y$, and both $\tilde{X}$ and $\tilde{Y}$ are orthogonal to the kernel of π_U^* with respect to the inner product $\langle ., . \rangle_U$. It is easily verified that the above construction for (left) homogeneous spaces is well defined as long as the Riemannian metric on

[3] Another situation where this applies in Computer Vision is the study of the *essential manifold*, which may be envisaged as the coset space of $SO(3) \times SO(3)$ modulo a subgroup isomorphic to $SO(2)$. For details see [25].

[4] $O(d)$ has dimension $d(d-1)/2$, since its Lie algebra is the set of $n \times n$ skew-symmetric matrices.

the Lie group is right invariant under the action of the stabilizer. This is trivially the case for a biinvariant metric.

The tangent space to G at $U \in G$ is $T_U G = \{U\Omega \in \mathbf{R}^{d\times d} \mid \Omega \in \mathfrak{so}(d)\}$, where $\mathfrak{so}(d) = \{\Omega \in \mathbf{R}^{d\times d} \mid \Omega = -\Omega^\top\}$, the set of real skew-symmetric $d \times d$-matrices. The bi-invariant Riemannian metric on G inherited from the embedding of $T_I G$ in Euclidean d-space is given by

$$\langle U\Omega_1, U\Omega_2\rangle_U = -\operatorname{Tr}\Omega_1\Omega_2,$$

that is, by (left translations of) the Frobenius inner product. The tangent space to the Grassmannian $\mathscr{G}(p,d) \simeq G/H$ at UH is then the quotient

$$T_U G/UT_I H = \left\{ U\left(\begin{bmatrix} 0 & -\Omega_{21}^\top \\ \Omega_{21} & 0 \end{bmatrix} + T_I H\right) \mid \Omega_{21} \in \mathbf{R}^{(d-p)\times p} \right\}$$

where

$$T_I H = \left\{ \begin{bmatrix} \Omega_1 & 0 \\ 0 & \Omega_2 \end{bmatrix} \in \mathbf{R}^{d\times d} \mid \Omega_1 \in \mathfrak{so}(p),\, \Omega_2 \in \mathfrak{so}(d-p) \right\}.$$

The *horizontal subspace* of $T_U G$ formed by all horizontal lifts of tangent vectors to the Grassmannian $\mathscr{G}(p,d) \simeq G/H$ at UH is

$$V_U^\perp(G) = \left\{ U\begin{bmatrix} 0 & -\Omega_{21}^\top \\ \Omega_{21} & 0 \end{bmatrix} \in \mathbf{R}^{d\times d} \mid \Omega_{21} \in \mathbf{R}^{(d-p)\times p} \right\}.$$

This is most easily seen at $U = I$ where the elements of $V_I^\perp(G)$ are obviously perpendicular to the *vertical subspace* $V_I(G) = \operatorname{Ker}\pi_I^* = T_I H$ with respect to the Frobenius inner product. The normal Riemannian metric on the Grassmannian is then given by

$$\begin{aligned} &\left\langle U\left(\begin{bmatrix} 0 & -\Omega_1^\top \\ \Omega_1 & 0 \end{bmatrix} + T_I H\right), U\left(\begin{bmatrix} 0 & -\Omega_2^\top \\ \Omega_2 & 0 \end{bmatrix} + T_I H\right)\right\rangle_{UH} \\ &\quad = \operatorname{Tr}\Omega_1^\top\Omega_2 + \operatorname{Tr}\Omega_1\Omega_2^\top \\ &\quad = 2\operatorname{Tr}\Omega_1\Omega_2^\top \end{aligned}$$

in terms of the horizontal lifts.

Projective Representation of Grassmann

An alternative representation of the Grassmannian $\mathscr{G}(p,d)$ is not as a quotient space of G, but as a subset of $\operatorname{Sym}(d)$, the vector space of all real symmetric $d \times d$-matrices. In this representation, a p-plane $\mathscr{X}$ is represented by the symmetric projection operator $P\colon \mathbf{R}^d \to \mathbf{R}^d$ with image $\operatorname{Im}(P) = \mathscr{X}$. In terms of matrix representations, P is a rank p real symmetric $d \times d$ matrix with $P^2 = P$ and $\operatorname{colspan}(P) = \mathscr{X}$. For example,

$$\mathscr{X}_0 = \operatorname{colspan}(P_0) = \operatorname{colspan}\begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix}.$$

In general, if $\mathscr{X}$ is represented by the orthogonal Stiefel matrix V then it is also represented by the symmetric matrix $P = VV^\top$. We denote the set of rank p real symmetric $d \times d$ matrix with $P^2 = P$ by $\mathscr{PG}(d, p)$ and obtain a bijection $\mathscr{PG}(d, p) \to \mathscr{G}(p, d)$ via $P \mapsto \operatorname{colspan}(P)$.

The natural inclusion map

$$i \colon \mathscr{G}(p, d) \simeq \mathscr{PG}(d, p) \hookrightarrow \operatorname{Sym}(d)$$

can now be used to equip the Grassmannian with subspace structures. For example, the Grassmannian inherits a subspace topology (the coarsest topology such that i is continuous) and a differential structure (the unique one that makes i a smooth embedding) from the standard topology resp. differential structure on $\operatorname{Sym}(d)$. It turns out that both of these coincide with the respective quotient structures constructed above. The Grassmannian $\mathscr{G}(p, d)$ can hence equally be thought of as a homogeneous space of G or as an embedded submanifold of $\operatorname{Sym}(d)$.

Since $\operatorname{Sym}(d)$, as a linear subspace of $\mathbf{R}^{d\times d}$, carries a natural inner product, namely the restriction of the Frobenius inner product on $\mathbf{R}^{d\times d}$ to $\operatorname{Sym}(d)$, each tangent space $T_P\mathscr{PG}(d, p)$ at a point $P \in \mathscr{PG}(d, p) \subset \operatorname{Sym}(d)$ inherits this inner product via the inclusion $T_P\mathscr{PG}(d, p) \subset T_P\operatorname{Sym}(d) \simeq \operatorname{Sym}(d)$. This provides another construction for a Riemannian metric on $\mathscr{G}(p, d) \simeq \mathscr{PG}(d, p)$.

To make this construction more concrete, note that the tangent space

$$T_P\mathscr{PG}(d, p) = \{[P, \Omega] \mid \Omega \in \mathfrak{so}(d)\},$$

where $[P, \Omega] = P\Omega - \Omega P$ is the matrix commutator [24, Theorem 2.1]. In particular,

$$T_{P_0}\mathscr{PG}(d, p) = \left\{ \begin{bmatrix} 0 & \Omega_{12} \\ \Omega_{12}^\top & 0 \end{bmatrix} \in \operatorname{Sym}(d) \;\middle|\; \Omega_{12} \in \mathbf{R}^{p\times(d-p)} \right\}.$$

Now observe that $O(d)$ acts transitively on $\mathscr{PG}(d, p)$ by conjugation since $UPU^\top \in \mathscr{PG}(d, p)$ for every $U \in O(d)$ and every $P \in \mathscr{PG}(d, p)$, and every point in $\mathscr{PG}(d, p)$ can be reached thus from a given one. Note that this action of $O(d)$ is different to the action by left multiplication that has been used to define the above homogeneous space structure of $\mathscr{G}(p, d)$. Nevertheless, it can be used to more explicitly describe the tangent space

$$T_{UP_0U^\top}\mathscr{PG}(d, p) = \left\{ U \begin{bmatrix} 0 & \Omega_{12} \\ \Omega_{12}^\top & 0 \end{bmatrix} U^\top \in \operatorname{Sym}(d) \;\middle|\; \Omega_{12} \in \mathbf{R}^{p\times(d-p)} \right\}$$

at an arbitrary point $P = UP_0U^\top \in \mathscr{PG}(d, p)$. Note further that the first p columns of UP_0 form an orthogonal Stiefel matrix V (equal to the first p columns of U), and that $P = UP_0U^\top = VV^\top$ as observed before. The embedded Riemannian metric on $\mathscr{G}(p, d) \simeq \mathscr{PG}(d, p)$ is then given by

$$\langle U \begin{bmatrix} 0 & \Omega_1 \\ \Omega_1^\top & 0 \end{bmatrix} U^\top, U \begin{bmatrix} 0 & \Omega_2 \\ \Omega_2^\top & 0 \end{bmatrix} U^\top \rangle_{UP_0U^\top} = \operatorname{Tr} \Omega_1 \Omega_2^\top + \operatorname{Tr} \Omega_1^\top \Omega_2$$

in terms of this representation. It is not difficult to see that this Riemannian metric, in fact, is the same as the normal Riemannian metric constructed above [24, Proposition 2.3].

The unique geodesic starting at a point $U_0 P_0 U_0^\top \in \mathscr{PG}(d, p)$ in direction

$$U_0 \begin{bmatrix} 0 & \Omega \\ \Omega^\top & 0 \end{bmatrix} U_0^\top$$

is given by

$$P(t) = U_0 \operatorname{expm}\left(t \begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} \right) \begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix} \operatorname{expm}\left(-t \begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} \right) U_0^\top,$$

where expm denotes the matrix exponential [24, Theorem 2.2]. Alternatively, it is given by

$$U(t)H = U_0 \operatorname{expm}\left(t \begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} \right) H$$

in the quotient representation. In particular, the *Riemannian exponential map* on the Grassmannian is given by

$$\exp_{U_0 P_0 U_0^\top}\left(U_0 \begin{bmatrix} 0 & \Omega \\ \Omega^\top & 0 \end{bmatrix} U_0^\top \right) = U_0 \operatorname{expm} \begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix} \operatorname{expm} \begin{bmatrix} 0 & \Omega \\ -\Omega^\top & 0 \end{bmatrix} U_0^\top$$

in the embedded representation and by

$$\exp_{U_0 H}\left(U_0 \left(\begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} + T_I H \right) \right) = U_0 \operatorname{expm} \begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix} H$$

in the quotient representation.

Geodesics

The Grassmannian with the above Riemannian metric is a complete Riemannian manifold, hence any pair of points $\mathscr{X}_1$ and $\mathscr{X}_2$ on the Grassmannian can be connected by a length-minimizing geodesic. The *geodesic distance* $d_{\text{geod}}\,(\mathscr{X}_1, \mathscr{X}_2)$ is then defined as the length of this minimizing geodesic. Since points on the Grassmannian can be moved around arbitrarily by application of an orthogonal transformation $U \in O(d)$, and since the above Riemannian metric is invariant under such transformations, it is sufficient to compute the length of a minimizing geodesic connecting the special point $\mathscr{X}_0 = \operatorname{colspan}(P_0)$ to any other point $\mathscr{X} = \operatorname{colspan}(P)$. By the above formula for the exponential map

$$P = \operatorname{expm}\begin{bmatrix} 0 & -\Omega \\ \Omega^\top & 0 \end{bmatrix}\begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix}\operatorname{expm}\begin{bmatrix} 0 & \Omega \\ -\Omega^\top & 0 \end{bmatrix}$$
$$= \begin{bmatrix} \cos^2\sqrt{\Omega\Omega^\top} & \operatorname{sinc}\left(2\sqrt{\Omega\Omega^\top}\right)\Omega \\ \Omega^\top \operatorname{sinc}\left(2\sqrt{\Omega\Omega^\top}\right) & \sin^2\sqrt{\Omega^\top\Omega} \end{bmatrix}$$

for some $\Omega \in \mathbf{R}^{p\times(d-p)}$, where we have used [24, Eq. (2.68)] in the second line. The geodesic distance from $\mathscr{X}_0$ to $\mathscr{X}$ is then $d_{\text{geod}}(\mathscr{X}_0, \mathscr{X}) = \sqrt{2\operatorname{Tr}\Omega\Omega^\top}$, that is the length of the tangent vector

$$\begin{bmatrix} 0 & \Omega \\ \Omega^\top & 0 \end{bmatrix} \in T_{P_0}\mathscr{PG}(d, p)$$

under the Riemannian metric at P_0. In more explicit terms, starting with a block representation

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix},$$

where $P_1 \in \operatorname{Sym}(p)$, compute the eigenvalue decomposition $P_1 = U_1\operatorname{diag}(\lambda_1, \ldots, \lambda_p)U_1^\top$ with $U_1 \in O(p)$, then $U_1\operatorname{diag}(\lambda_1, \ldots, \lambda_p)U_1^\top = P_1 = \cos^2\sqrt{\Omega\Omega^\top}$ is equivalent to $\Omega\Omega^\top = U_1\operatorname{diag}(\arccos^2(\sqrt{\lambda_1}), \ldots, \arccos^2(\sqrt{\lambda_p}))U_1^\top$ and hence

$$d_{\text{geod}}(\mathscr{X}_0, \mathscr{X}) = \sqrt{2\operatorname{Tr}\Omega\Omega^\top} = \sqrt{2\sum_{i=1}^{p}\arccos^2(\sqrt{\lambda_i})},$$

cf. [24, Corollary 2.1].

A geometric interpretation of the above distance formula can be obtained as follows. Swapping back to the quotient representation and using [24, Eq. (2.66)], it follows that

$$\mathscr{X} = \operatorname{colspan}\begin{bmatrix} \cos\sqrt{\Omega\Omega^\top} \\ \frac{\sin\sqrt{\Omega^\top\Omega}}{\sqrt{\Omega^\top\Omega}}\Omega^\top \end{bmatrix},$$

where the columns of this matrix have unit length in the 2-norm since they are the first p columns of an orthogonal matrix. The *first principal angle* θ_1 between the subspaces $\mathscr{X}_0$ and $\mathscr{X}$ is given by

$$\begin{aligned}\cos\theta_1 &= \max_{u\in\mathscr{X}_0, v\in\mathscr{X}} \frac{u^\top v}{\|u\|_2\|v\|_2}\\ &= \max_{\|x\|_2=1,\|y\|_2=1} \begin{bmatrix} x^\top & 0\end{bmatrix} \begin{bmatrix} \cos\sqrt{\Omega\Omega^\top} \\ \frac{\sin\sqrt{\Omega^\top\Omega}}{\sqrt{\Omega^\top\Omega}}\Omega^\top \end{bmatrix} y\\ &= \max_{\|x\|_2=1,\|y\|_2=1} x^\top U_1 \text{diag}(\sqrt{\lambda_1},\ldots,\sqrt{\lambda_p})U_1^\top y\\ &= \sqrt{\lambda_1},\end{aligned}$$

assuming that the eigenvalues λ_i are ordered in nonincreasing order. Similarly, it can be shown that the ith principal angle $\theta_i = \sqrt{\lambda_i}$ for $i = 2,\ldots,p$. It follows that, in general,

$$d_{\text{geod}}\,(\mathscr{X}_1, \mathscr{X}_2) = \sqrt{2}\|\Theta\|_2,$$

where $\Theta = \begin{bmatrix}\theta_1 \ldots \theta_p\end{bmatrix}^\top$ is the vector of principal angles between $\mathscr{X}_1$ and $\mathscr{X}_2$. Note that some authors use a different scaling of the Frobenius inner product (usually an additional factor of $\frac{1}{2}$) to arrive at a formula for the geodesic distance without the factor of $\sqrt{2}$. Obviously, this does not change the geometry.

Principal angles. The geodesic distance has an interpretation as the magnitude of the smallest rotation that takes one subspace to the other. If $\Theta = [\theta_1, \theta_2, \ldots, \theta_p]$ is the sequence of principal angles between two subspaces $\mathscr{X}_1 \in \mathscr{G}(p, d)$ and $\mathscr{X}_2 \in \mathscr{G}(p, d)$, then $d_{\text{geod}}\,(\mathscr{X}_1, \mathscr{X}_2) = \|\Theta\|_2$.

Definition 1 (*Principal Angles*) Let X_1 and X_2 be two matrices of size $d \times p$ with orthonormal columns. The principal angles $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_p \le \pi/2$ between two subspaces $\text{Span}(X_1)$ and $\text{Span}(X_2)$, are defined recursively by

$$\begin{aligned}\cos(\theta_i) &= \max_{u_i\in\text{Span}(X_1)} \max_{v_i\in\text{Span}(X_2)} u_i^T v_i \qquad (6.11)\\ \text{s.t.:}\quad & \|u_i\|_2 = \|v_i\|_2 = 1\\ & u_i^T u_j = 0;\ j = 1, 2, \ldots, i-1\\ & v_i^T v_j = 0;\ j = 1, 2, \ldots, i-1\end{aligned}$$

In other words, the first principal angle θ_1 is the smallest angle between all pairs of unit vectors in the first and the second subspaces. The rest of the principal angles are defined similarly.

Two operators, namely the logarithm map $\log_x(\cdot) : \mathscr{M} \to T_x(\mathscr{M})$ and its inverse, the exponential map $\exp_x(\cdot) : T_x(\mathscr{M}) \to \mathscr{M}$ are defined over Riemannian manifolds to switch between the manifold and the tangent space at x. A key point here is the fact that both the logarithm map and its inverse do not have closed-form solutions for Grassmann manifolds. Efficient numerical approaches for computing both maps were proposed by [5, 15]. In this paper, however, the exponential and logarithm maps will only be used when describing previous work of other authors.

6.4 Dictionary Learning on Grassmannian

In this part, we propose to make use of the projective representation of Grassmannian to perform coding and dictionary learning on Grassmannian. We recall that working with $\mathscr{PG}(p, d)$ instead of $\mathscr{G}(p, d)$ has the advantage that each element of $\mathscr{PG}(p, d)$ is a single matrix, whereas elements of $\mathscr{G}(p, d)$ are equivalence classes of matrices. Hereinafter, we shall denote XX^T by $\widehat{X}$, the hat representing the action of the projection embedding. Furthermore, $\langle \cdot, \cdot \rangle$ represents the Frobenius inner product: thus $\langle \widehat{X}, \widehat{Y} \rangle = \mathrm{Tr}(\widehat{X}\widehat{Y})$. Note that in computing $\langle \widehat{X}, \widehat{Y} \rangle$ it is not necessary to compute $\widehat{X}$ and $\widehat{Y}$ explicitly (they may be large matrices). Instead, note that $\langle \widehat{X}, \widehat{Y} \rangle = \mathrm{Tr}(\widehat{X}\widehat{Y}) = \mathrm{Tr}(XX^T YY^T) = \mathrm{Tr}(Y^T XX^T Y) = \|Y^T X\|_F^2$. This is advantageous, since $Y^T X$ may be a substantially smaller matrix.

Apart from the geodesic distance metric, an important metric used in this paper is the *chordal metric*, defined by

$$d_{\mathrm{chord}}(\widehat{X}, \widehat{Y}) = \|\widehat{X} - \widehat{Y}\|_F \ , \tag{6.12}$$

This metric will be used in the context of (6.4) to recast the coding and consequently dictionary-learning problem in terms of chordal distance. Before presenting our proposed methods, we establish an interesting link between coding and the notion of weighted mean in a metric space.

6.4.1 Weighted Karcher Mean

The underlying concept of coding using a dictionary is to represent in some way a point in a space of interest as a combination of other elements in that space. In the usual method of coding in $\mathbf{R}^d$ given by (6.1), each x is represented by a linear combination of dictionary elements d_j, where the first term represents the coding error. For coding in a manifold, the problem to address is that linear combinations do not make sense. We wish to find some way in which an element $\mathscr{X}$ may be represented in terms of other dictionary elements $\mathscr{D}_j$ as suggested in (6.4). For a proposed method to generalize the $\mathbf{R}^d$ case, one may prefer a method that is a direct generalization of the Euclidean case in some way.

In $\mathbf{R}^d$, a different way to consider the expression $\sum_{j=1}^N y_j d_j$ in (6.1) is as a weighted mean of the points d_j This observation relies on the following fact, which is verified using a Lagrange multiplier method.

Lemma 1 *Given coefficients y with* $\sum_{i=1}^N y_i = 1$, *and dictionary elements* $\{d_1, \ldots, d_N\}$ *in* $\mathbf{R}^d$, *the point* $x^* \in \mathbf{R}^d$ *that minimizes* $\sum_{i=1}^N y_i \,\|x - d_i\|_F^2$ *is given by* $x^* = \sum_{i=1}^N y_i \, d_i$.

In other words, the affine combination of dictionary elements is equal to their weighted mean. Although linear combinations are not defined for points on manifolds or metric spaces, a weighted mean is.

Definition 2 Given points $\mathscr{D}_i$ on a Riemannian manifold $\mathscr{M}$, and weights y_i, the point $\mathscr{X}^*$ that minimizes $\sum_{i=1}^{N} y_i \, d_g(\mathscr{X}, \mathscr{D}_i)^2$, is called the weighted Karcher mean of the points $\mathscr{D}_i$ with weights y_i. Here, $d_g(\cdot, \cdot)$ is the geodesic distance on $\mathscr{M}$.

Generally, finding the Karcher mean [28] on a manifold involves an iterative procedure, which may converge to a local minimum, even on a simple manifold, such as $SO(3)$ [23, 32]. However, one may replace the geodesic metric with a different metric in order to simplify the calculation. To this end, we propose the *chordal metric* on a Grassman manifold, defined for matrices $\widehat{X}$ and $\widehat{Y}$ in $\mathscr{PG}(p, d)$ in Eq. (6.12). The corresponding mean, as in Definition 2 (but using the chordal metric) is called the *weighted chordal mean* of the points. In contrast to the Karcher mean, the weighted chordal mean on a Grassman manifold has a simple closed form.

Theorem 1 *The weighted chordal mean of a set of points* $\widehat{D}_i \in \mathscr{PG}(p, d)$ *with weights* y_i *is equal to* $\mathrm{Proj}(\sum_{i=1}^{m} y_i \widehat{D}_i)$, *where* $\mathrm{Proj}(\cdot)$ *represents the closest point on* $\mathscr{PG}(p, d)$ *[20].*

The function $\mathrm{Proj}(\cdot)$ has a closed-form solution in terms of the singular value decomposition. More specifically,

Lemma 2 *Let* X *be an* $d \times d$ *symmetric matrix with eigenvalue decomposition* $X = UDU^T$, *where* D *contains the eigenvalues* λ_i *of* X *in descending order. Let* U_p *be the* $d \times p$ *matrix consisting of the first* p *columns of* U. *Then* $\widehat{U}_p = U_p U_p^T$ *is the closest matrix in* $\mathscr{PG}(p, d)$ *to* X *(under the Frobenius norm) [20].*

The chordal metric on a Grassman manifold is not a geodesic metric (that is it is not equal to the length of a shortest geodesic under the Riemannian metric). However, it is closely related. In fact, one may easily show that for $\mathscr{G}(p, d) \ni \mathscr{X} = \mathrm{span}(X)$ and $\mathscr{G}(p, d) \ni \mathscr{Y} = \mathrm{span}(Y)$

$$\frac{2}{\pi} d_{\mathrm{geod}}(\mathscr{X}, \mathscr{Y}) \le d_{\mathrm{chord}}(\widehat{X}, \widehat{Y}) \le d_{\mathrm{geod}}(\mathscr{X}, \mathscr{Y}) \; .$$

Furthermore, the path-metric [23] induced by $d_{\mathrm{chord}}(\cdot, \cdot)$ is equal to the geodesic distance.

Sparse Coding

Given a dictionary $\mathbb{D}$ with atoms $\widehat{D}_j \in \mathscr{PG}(p, d)$ and a query sample $\widehat{X}$ the problem of sparse coding can be recast extrinsically as:

$$\ell(\mathscr{X}, \mathbb{D}) \triangleq \min_{y} \left\| \widehat{X} - \sum_{j=1}^{N} y_j \widehat{D}_j \right\|_F^2 + \lambda \|y\|_1 . \tag{6.13}$$

The formulation here varies slightly from the general form given in (6.4), in that the point $\sum_{j=1}^{N} y_j \widehat{D}_j$ does not lie exactly on the manifold $\mathscr{PG}(p,d)$, since it is not idempotent nor its rank is necessarily p. We call this solution an *extrinsic solution*; the point coded by the dictionary is allowed to step out of the manifold.

Expanding the Frobenius norm term in (6.13) results in a convex function in y:

$$\Big\|\widehat{X} - \sum_{j=1}^{N} y_j \widehat{D}_j\Big\|_F^2 = \|\widehat{X}\|_F^2 + \Big\|\sum_{j=1}^{N} y_j \widehat{D}_j\Big\|_F^2 - 2\,\langle \sum_{j=1}^{N} y_j \widehat{D}_j, \widehat{X}\rangle \ .$$

The sparse codes can be obtained without explicitly embedding the manifold in $\mathscr{PG}(p,d)$ using $\Pi(\mathscr{X})$. This can be seen by defining $[\mathscr{K}(X,\mathbb{D})]_i = \langle \widehat{X}, \widehat{D}_i \rangle$ as an N dimensional vector storing the similarity between signal X and dictionary atoms in the induced space and $[\mathbb{K}(\mathbb{D})]_{i,j} = \langle \widehat{D}_i, \widehat{D}_j \rangle$ as an $N \times N$ symmetric matrix encoding the similarities between dictionary atoms (which can be computed offline). Then, the sparse coding in (6.13) can be written as:

$$\ell(\mathscr{X},\mathbb{D}) = \min_{y}\, y^T \mathbb{K}(\mathbb{D}) y - 2 y^T \mathscr{K}(X,\mathbb{D}) + \lambda \|y\|_1 \ . \tag{6.14}$$

The symmetric matrix $\mathbb{K}(\mathbb{D})$ is positive semidefinite since for all $v \in \mathbf{R}^N$:

$$\begin{aligned} v^T \mathbb{K}(\mathbb{D}) v &= \sum_{i=1}^{N}\sum_{j=1}^{N} v_i v_j \,\langle \widehat{D}_i, \widehat{D}_j \rangle = \left\langle \sum_{i=1}^{N} v_i \widehat{D}_i,\ \sum_{j=1}^{N} v_j \widehat{D}_j \right\rangle \\ &= \Big\| \sum_{i=1}^{N} v_i \widehat{D}_i \Big\|_F^2 \geq 0. \end{aligned}$$

Therefore, the problem is convex and can be efficiently solved. The problem in (6.14) can be transposed into a vectorized sparse coding problem. More specifically, let $U \Sigma U^T$ be the SVD of $\mathbb{K}(\mathbb{D})$. Then (6.14) is equivalent to

$$\ell(\mathscr{X},\mathbb{D}) = \min_{y} \|x^* - Ay\|^2 + \lambda \|y\|_1, \tag{6.15}$$

where $A = \Sigma^{1/2} U^T$ and $x^* = \Sigma^{-1/2} U^T \mathscr{K}(X,\mathbb{D})$. This can be easily verified by plugging A and x^* into (6.15). Algorithm 2 provides the pseudo-code for performing Grassmann Sparse Coding (gSC).

A special case is sparse coding on the Grassmann manifold $\mathscr{G}(1,d)$, which can be seen as a problem on $d-1$ dimensional unit sphere, albeit with a subtle difference. More specifically, unlike conventional sparse coding in vector spaces, $x \sim -x, \forall x \in \mathscr{G}(1,d)$, which results in having antipodals points being equivalent. For this special case, the solution proposed in (6.13) can be understood as sparse coding in the higher dimensional quadratic space, i.e,$f : \mathbf{R}^d \to \mathbf{R}^{d^2}, f(x) = [x_1^2, x_1x_2, \ldots, x_d^2]^T$. We note that in the quadratic space, $\|f(x)\| = 1$ and $f(x) = f(-x)$.

Algorithm 2: Sparse coding on Grassmann manifolds (gSC).

```
Input: Grassmann dictionary $\{\mathscr{D}_i\}_{i=1}^N$, $\mathscr{D}_i \in \mathscr{G}(p,d)$ with $\mathscr{D}_i = \mathrm{span}(D_i)$; the query
    $\mathscr{G}(p,d) \ni \mathscr{X} = \mathrm{span}(X)$
Output: The sparse code $y^*$
Initialization.
    for $i, j \leftarrow 1$ to $N$ do
        $[\mathbb{K}(\mathbb{D})]_{i,j} \leftarrow \|D_i^T D_j\|_F^2$
    end
    $\mathbb{K}(\mathbb{D}) = U\Sigma U^T$ /* compute SVD of $\mathbb{K}(\mathbb{D})$ */
    $A \leftarrow \Sigma^{1/2} U^T$
Processing.
    for $i \leftarrow 1$ to $N$ do
        $[\mathscr{K}(X,\mathbb{D})]_i \leftarrow \|X^T D_i\|_F^2$
    end
    $x^* \leftarrow \Sigma^{-1/2} U^T \mathscr{K}(X,\mathbb{D})$
    $y^* \leftarrow \underset{y}{\arg\min}\ \|x^* - Ay\|^2 + \lambda\|y\|_1$
```

Classification Based on Coding

If the atoms in the dictionary are not labeled (e.g, if $\mathbb{D}$ is a generic dictionary not tied to any particular class), the generated sparse codes (vectors) for both training and query data can be fed to Euclidean-based classifiers like support vector machines [36] for classification. Inspired by the Sparse Representation Classifier (SRC) [46], when the atoms in sparse dictionary $\mathbb{D}$ are labeled, the generated codes of the query sample can be directly used for classification. In doing so, let

$$y_c = \begin{pmatrix} y_0\delta\big(l_0 - c\big) \\ y_1\delta\big(l_1 - c\big) \\ \vdots \\ y_N\delta\big(l_N - c\big) \end{pmatrix}$$

be the class-specific sparse codes, where l_j is the class label of atom $\mathscr{G}(p,d) \ni \mathscr{D}_j = \mathrm{span}(D_j)$ and $\delta(x)$ is the discrete Dirac function. An efficient way of utilizing class-specific sparse codes is through computing residual errors. In this case, the residual error of query sample $\mathscr{G}(p,d) \ni \mathscr{X} = \mathrm{span}(X)$ for class c is defined as:

$$\varepsilon_c(\mathscr{X}) = \left\| \widehat{X} - \sum_{j=1}^{N} y_j \widehat{D}_j \delta\big(l_j - c\big) \right\|_F^2 . \tag{6.16}$$

Alternatively, the similarity between query sample $\mathscr{X}$ to class c can be defined as $s(\mathscr{X}, c) = h(y_c)$. The function $h(\cdot)$ could be a linear function like $\sum_{j=1}^{N}(\cdot)$ or even a nonlinear one like $\max(\cdot)$. Preliminary experiments suggest that Eq. (6.16) leads to higher classification accuracies when compared to the aforementioned alternatives.

6.4.2 Dictionary Learning

Given a finite set of observations $\mathbb{X} = \{\mathscr{X}_i\}_{i=1}^m$, $\mathscr{G}(p, d) \ni \mathscr{X}_i = \operatorname{span}(X_i)$, the problem of dictionary learning on Grassmann manifolds is defined as minimizing the following cost function:

$$h(\mathbb{D}) \triangleq \sum_{i=1}^{m} \ell_{\mathscr{G}}(\mathscr{X}_i, \mathbb{D}), \tag{6.17}$$

with $\mathbb{D} = \left\{\mathscr{D}_j\right\}_{j=1}^N$, $\mathscr{G}(p, d) \ni \mathscr{D}_j = \operatorname{span}(D_j)$ being a dictionary of size N. Here, $\ell_{\mathscr{G}}(\mathscr{X}, \mathbb{D})$ is a loss function and should be small if $\mathbb{D}$ is "good" at representing $\mathscr{X}$. In the following text, we elaborate on how a Grassmann dictionary can be learned.

Aiming for sparsity, the ℓ_1-norm regularization is usually employed to obtain the most common form of $l_{\mathscr{G}}(\mathscr{X}, \mathbb{D})$ as depicted in Eq. (6.13). With this choice, the problem of dictionary learning on Grassmann manifolds can be written as:

$$\min_{\{y_i\}_{i=1}^m, \mathbb{D}} \sum_{i=1}^{m} \left\| \widehat{X}_i - \sum_{j=1}^{N} y_{ij} \widehat{D}_j \right\|_F^2 + \lambda \sum_{i=1}^{m} \|y_i\|_1. \tag{6.18}$$

Due to the non-convexity of (6.18) and inspired by the solutions in Euclidean spaces, we propose to solve (6.18) by alternating between the two sets of variables, $\mathbb{D}$ and $\{y_i\}_{i=1}^m$. More specifically, minimizing (6.18) over sparse codes y while dictionary $\mathbb{D}$ is fixed is a convex problem. Similarly, minimizing the overall problem over $\mathbb{D}$ with fixed $\{y_i\}_{i=1}^m$ is convex as well.

Therefore, to update dictionary atoms we break the minimization problem into N sub-minimization problems by independently updating each atom, $\widehat{D}_r$, in line with general practice in dictionary learning [12]. To update $\widehat{D}_r$, we write

$$\sum_{i=1}^{m} \left\| \widehat{X}_i - \sum_{j=1}^{N} y_{ij} \widehat{D}_j \right\|_F^2 = \sum_{i=1}^{m} \left\| \left(\widehat{X}_i - \sum_{j \neq r} y_{ij} \widehat{D}_j \right) - y_{ir} \widehat{D}_r \right\|_F^2. \tag{6.19}$$

All other terms in (6.18) being independent of $\widehat{D}_r$, and since $\|\widehat{D}_r\|_F^2 = p$ is fixed, minimizing this with respect to $\widehat{D}_r$ is equivalent to minimizing $\mathscr{J}_r = -2 \left\langle S_r, \widehat{D}_r \right\rangle$ where

$$S_r = \sum_{i=1}^{m} y_{ir} \left(\widehat{X}_i - \sum_{j \neq r} y_{ij} \widehat{D}_j \right). \tag{6.20}$$

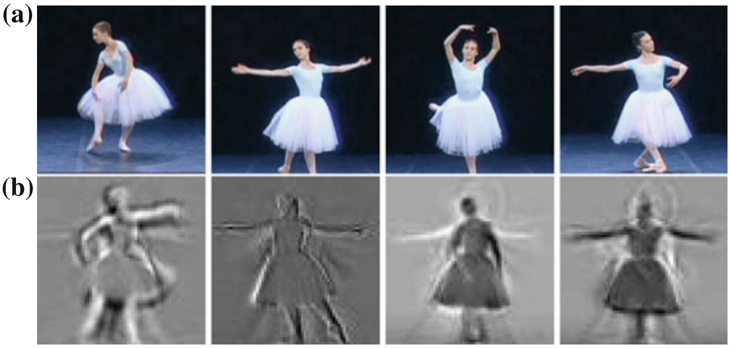

Fig. 6.2 **a** Examples of actions performed by a ballerina. **b** The dominant eigenvectors for four atoms learned by the proposed Grassmann Dictionary Learning (gDL) method (grayscale images were used in gDL)

Finally, minimizing $\mathscr{J}_r = -2\,\langle S_r,\, \widehat{D}_r\rangle$ is the same as minimizing $\|S_r - \widehat{D}_r\|$ over $\widehat{D}_r$ in $\mathscr{PG}(n, p)$. The solution to this problem is given by the p-leading eigenvectors of S_r according to the Lemma 2. Algorithm 3 details the pseudocode for learning a dictionary on Grassmann manifolds. Figure 6.2 shows examples of a ballet dance.

To perform coding, we have relaxed the idempotent and rank constraints of the mapping $\Pi(\cdot)$ since matrix addition and subtraction do not preserve these constraints. However, for dictionary learning, the orthogonality constraint ensures the dictionary atoms have the required structure.

6.5 Kernel Coding

In this section, we propose to perform coding and dictionary learning in a reproducing Kernel Hilbert space (RKHS). This has the twofold advantage of yielding simple solutions to several popular coding techniques and of resulting in a potentially better representation than standard coding techniques due to the nonlinearity of the approach. Before formulating our kernel solutions, we need to make sure that positive definite kernels on Grassmann manifolds are at our disposal. Formally,

Definition 3 (*Real-valued Positive Definite Kernels*) Let $\mathscr{X}$ be a nonempty set. A symmetric function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is a positive definite (***pd***) kernel on $\mathscr{X}$ if and only if $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$ for any $n \in \mathbb{N}$, $x_i \in \mathscr{X}$ and $c_i \in \mathbb{R}$.

Definition 4 (*Grassmannian Kernel*) A function $k : \mathscr{G}(p, d) \times \mathscr{G}(p, d) \to \mathbb{R}$ is a Grassmannian kernel, if it is well defined and *pd*. In our context, a function is well defined if it is invariant to the choice of basis, i.e, $k(XR_1, YR_2) = k(X, Y)$, for all $X, Y \in \mathscr{G}(p, d)$ and $R_1, R_2 \in \mathrm{SO}(p)$, where $\mathrm{SO}(p)$ denotes the special orthogonal group.

Algorithm 3: Grassmann Dictionary Learning (gDL)

```
Input: training set $\mathbb{X}=\{\mathcal{X}_i\}_{i=1}^{m}$, where each $\mathcal{G}(p,d) \ni \mathcal{X}_i = \operatorname{span}(X_i)$; nIter: number of
    iterations
Output: Grassmann dictionary $\mathbb{D} = \{\mathcal{D}_i\}_{i=1}^{N}$, where $\mathcal{G}(p,d) \ni \mathcal{D}_i = \operatorname{span}(D_i)$
Initialization.
    Initialize the dictionary $\mathbb{D}$ by selecting $N$ samples from $\mathbb{X}$ randomly
Processing.
    for $t = 1$ to nIter do
        // Sparse Coding Step using Algorithm 2
        for $i = 1$ to $m$ do
            $y_i \leftarrow \min_{y} \left\| \widehat{X}_i - \sum_{j=1}^{N} [y]_j \widehat{D}_j \right\|_F^2 + \lambda \|y\|_1$
        end
        // Dictionary update step
        for $r = 1$ to $N$ do
            Compute $S_r$ according to Eq. (6.20).
            $\{\lambda_k, v_k\} \leftarrow$ eigenvalues and eigenvectors of $S_r$
            $S_r v = \lambda v;\ \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$
            $D_r^* \leftarrow [v_1|v_2|\cdots|v_p]$
        end
    end
```

The most widely used kernel is arguably the Gaussian or radial basis function (RBF) kernel. It is therefore tempting to define a Radial Basis Grassmannian kernel by replacing the Euclidean distance with the geodesic distance. Unfortunately, although symmetric and well defined, the function $\exp(-\beta d_{\text{geod}}^2(\cdot,\cdot))$ is not *pd* [21]. Nevertheless, two Grassmannian kernels, i.e, the Binet–Cauchy kernel [45] and the projection kernel [18], have been proposed to embed Grassmann manifolds into RKHS. In this work, we are only interested in the projection kernels[5]

$$k_p(X,Y) = \left\| X^T Y \right\|_F^2 . \tag{6.21}$$

From the previous discussions, k_p, defined in Eq. (6.21) can be seen as a linear kernel in the space induced by the projection embedding. However, the inner products defined by the projection embedding can actually be exploited to derive many new Grassmannian kernels, including universal kernels.

Universal Grassmannian Kernels

Although often used in practice, linear kernels are known not to be universal [38]. This can have a crucial impact on their representation power for a specific task. Indeed, from the *Representer Theorem* [35], we have that, for a given set of training data $\{x_j\}$, $j \in \mathbb{N}_n$, $\mathbb{N}_n = \{1, 2, \ldots, n\}$ and a *pd* kernel k, the function learned by any

[5] In our experiments, we observed that the projection kernel almost always outperforms the Binet–Cauchy kernel.

algorithm can be expressed as

$$\hat{f}(x_*) = \sum_{j \in \mathbb{N}_n} c_j k(x_*, x_j) \; . \tag{6.22}$$

Importantly, only *universal kernels* have the property of being able to approximate any target function f_t arbitrarily well given sufficiently many training samples. Therefore, k_p may not generalize sufficiently well for certain problems. Below, we develop several universal Grassmannian kernels. To this end, we make use of negative definite kernels and of their relation to *pd* ones. Let us first formally define negative definite kernels.

Definition 5 (*Real-valued Negative Definite Kernels*) Let $\mathscr{X}$ be a nonempty set. A symmetric function $\psi : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is a negative definite (***nd***) kernel on $\mathscr{X}$ if and only if $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \leq 0$ for any $n \in \mathbb{N}$, $x_i \in \mathscr{X}$ and $c_i \in \mathbb{R}$ with $\sum_{i=1}^{n} c_i = 0$.

Note that, in contrast to positive definite kernels, an additional constraint of the form $\sum c_i = 0$ is required in the negative definite case. The most important example of *nd* kernels is the distance function defined on a Hilbert space. More specifically,

Theorem 2 ([27]) *Let $\mathscr{X}$ be a nonempty set, $\mathscr{H}$ be an inner product space, and $\psi : \mathscr{X} \to \mathscr{H}$ be a function. Then $f : (\mathscr{X} \times \mathscr{X}) \to \mathbb{R}$ defined by $f(x_i, x_j) = \|\psi(x_i) - \psi(x_j)\|^2_{\mathscr{H}}$ is negative definite.*

Therefore, being distances in Hilbert spaces, d^2_{chord} is a *nd* kernel. We now state an important theorem which establishes the relation between *pd* and *nd* kernels.

Theorem 3 (Theorem 2.3 in Chap. 3 of [6]) *Let μ be a probability measure on the half line $\mathbb{R}_+$ and $0 < \int_0^\infty t \mathrm{d}\mu(t) < \infty$. Let $\mathscr{L}_\mu$ be the Laplace transform of μ, i.e, $\mathscr{L}_\mu(s) = \int_0^\infty e^{-ts} \mathrm{d}\mu(t), \;\; s \in \mathbb{C}_+$. Then, $\mathscr{L}_\mu(\beta f)$ is positive definite for all $\beta > 0$ if and only if $f : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_+$ is negative definite.*

The problem of designing a *pd* kernel on the Grassmannian can now be cast as that of finding an appropriate probability measure μ. Below, we show that this lets us reformulate popular kernels in Euclidean space as Grassmannian kernels.

RBF Kernels.

Grassmannian RBF kernels can be obtained by choosing $\mu(t) = \delta(t - 1)$ in Theorem 3, where $\delta(t)$ is the Dirac delta function. This choice yields the Grassmannian RBF kernels (after discarding scalar constants)

$$k_{r,p}(X, Y) = \exp\left(\beta \left\| X^T Y \right\|_F^2\right) , \;\; \beta > 0 \; . \tag{6.23}$$

Table 6.1 The proposed Grassmannian kernels and their properties

Kernel	Equation	Properties
Linear	$k_p(X,Y) = \|X^T Y\|_F^2$	*pd*
RBF	$k_{r,p}(X,Y) = \exp\Big(\beta\|X^T Y\|_F^2\Big),\ \beta > 0$	*pd*, universal
Laplace	$k_{l,p}(X,Y) = \exp\Bigg(-\beta\sqrt{p - \|X^T Y\|_F^2}\Bigg), \beta > 0$	*pd*, universal

Laplace Kernels.

The Laplace kernel is another widely used Euclidean kernel, defined as $k(x,y) = \exp(-\beta\|x-y\|)$. To obtain heat kernels on the Grassmannian, we make use of the following theorem for *nd* kernels.

Theorem 4 (Corollary 2.10 in Chap. 3 of [6]) *If $\psi : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is negative definite and satisfies $\psi(x,x) \geqq 0$ then so is ψ^α for $0 < \alpha < 1$.*

As a result $d_{\text{chord}}(\cdot,\cdot)$ is *nd* by choosing $\alpha = 1/2$ in Theorem 4. By employing $d^2_{\text{chord}}(\cdot,\cdot)$ along with $\mu(t) = \delta(t-1)$ in Theorem 3, we obtain the Grassmannian heat kernels

$$k_{l,p}(X,Y) = \exp\Bigg(-\beta\sqrt{p - \|X^T Y\|_F^2}\Bigg),\quad \beta > 0\ . \tag{6.24}$$

As shown in [38], the RBF and heat kernels are universal for $\mathbb{R}^d, d > 0$. The kernels described above are summarized in Table 6.1. Note that many other kernels can be derived by, e.g, exploiting different measures in Theorem 3. However, the kernels derived here correspond to the most popular ones in Euclidean space, and we therefore leave the study of additional kernels as future work.

6.5.1 Kernel-Based Riemannian Coding

Let $\phi : \mathscr{M} \to \mathscr{H}$ be a mapping to an RKHS induced by the kernel $k(x,y) = \phi(x)^T\phi(y)$. Sparse coding in $\mathscr{H}$ can then be formulated by rewriting (6.1) as

$$\ell_\phi(x,\mathbb{D}) \triangleq \min_y \Big\|\phi\big(x\big) - \sum\nolimits_{j=1}^N [y]_j \phi\big(d_j\big)\Big\|_2^2 + \lambda\|y\|_1. \tag{6.25}$$

Expanding the reconstruction term in (6.25) yields

$$\begin{aligned}\left\|\phi(x)-\sum\nolimits_{j=1}^{N}[y]_j\phi(d_j)\right\|_2^2 &= \phi(x)^T\phi(x)\\ &-2\sum\nolimits_{j=1}^{N}[y]_j\phi(d_j)^T\phi(x)+\sum\nolimits_{i,j=1}^{N}[y]_i[y]_j\phi(d_i)^T\phi(d_j)\\ &= k(x,x)-2y^Tk(x,\mathscr{D})+y^TK(\mathscr{D},\mathscr{D})y,\end{aligned} \tag{6.26}$$

where $k(x,\mathscr{D})\in\mathbb{R}^N$ is the kernel vector evaluated between x and the dictionary atoms, and $K(\mathscr{D},\mathscr{D})\in\mathbb{R}^{N\times N}$ is the kernel matrix evaluated between the dictionary atoms.

This shows that the reconstruction term in (6.25) can be kernelized. More importantly, after kernelization, this term remains quadratic, convex, and similar to its counterpart in Euclidean space. To derive an efficient solution to kernel sparse coding, we introduce the following theorem.

Theorem 5 ([19]) *Consider the least-squares problem in an RKHS $\mathscr{H}$*

$$\begin{aligned}&\min_y \ \left\|\phi(x)-\sum\nolimits_{j=1}^{N}[y]_j\phi(d_j)\right\|_2^2 \Leftrightarrow\\ &\min_y \ y^TK(\mathscr{D},\mathscr{D})y-2y^Tk(x,\mathscr{D})+f(x)\ ,\end{aligned} \tag{6.27}$$

where $f(x)$ is a constant function (i.e, independent of α). Let $U\Sigma U^T$ be the SVD of the symmetric positive definite matrix $K(\mathscr{D},\mathscr{D})$. Then (6.27) is equivalent to the least-squares problem in $\mathbb{R}^N$

$$\min_{\alpha}\left\|\tilde{x}-\tilde{D}y\right\|_2^2\ , \tag{6.28}$$

with $\tilde{D}=\Sigma^{1/2}U^T$ and $\tilde{x}=\Sigma^{-1/2}U^Tk(x,\mathscr{D})$.

This theorem lets us write kernel sparse coding as

$$\min_y\left\|\tilde{x}-\tilde{D}y\right\|_2^2+\lambda\|y\|_1\ , \tag{6.29}$$

which is a standard linear sparse coding problem. Algorithm 4 provides the pseudocode for performing kernel Sparse Coding (kSC).

6.5.2 Kernel Dictionary Learning

To obtain a dictionary in $\mathscr{H}$, we follow an alternating optimization strategy to update the codes and the dictionary. Since obtaining the codes with a given dictionary was discussed in the previous part, here we focus on the dictionary update.

Algorithm 4: Kernel sparse coding (kSC).

```
Input: Dictionary $\mathscr{D} = \{d_i\}_{i=1}^N,\ d_i \in \mathscr{M}$; the query $x \in \mathscr{M}$, a positive definite kernel
    $k : \mathscr{M} \times \mathscr{M} \to \mathbb{R}$.
Output: The sparse codes $y^*$
Initialization.
    for $i, j \leftarrow 1$ to $N$ do
        $[K(\mathscr{D}, \mathscr{D})]_{i,j} \leftarrow k(d_i, d_j)$
    end
    $K(\mathscr{D}, \mathscr{D}) = U\Sigma U^T$ /*  apply SVD                    */
    $A \leftarrow \Sigma^{1/2}U^T$
Processing.
    for $i \leftarrow 1$ to $N$ do
        $[k(x, \mathscr{D})]_i \leftarrow k(x, d_i)$
    end
    $x^* \leftarrow \Sigma^{-1/2}U^T k(x, \mathscr{D})$
    $y^* \leftarrow \arg\min_y \|x^* - Ay\|^2 + \lambda\|y\|_1$
```

Algorithm 5: Learning a generic dictionary.

```
Input: Training data $\{x_i\}_{i=1}^M,\ x_i \in \mathscr{M}$; kernel function $k(\cdot,\cdot) : \mathscr{M} \times \mathscr{M} \to \mathbb{R}$; size of
    dictionary $N$.
Output: Dictionary $\phi(\mathscr{D})$ in the RKHS $\mathscr{H}$ described as $\phi(\mathscr{X})V$
Processing.
    /* Initialize $\phi(\mathscr{D})$ either randomly or through kernel
       k-means algorithm.                                            */
    for iter $\leftarrow 1$ to nIter do
        Compute kernel codes $y_i,\ i \in [1, \dots, M]$ using Algorithm 4.
        /* fix kernel codes $y_i$ and update dictionary.             */
        $\phi(\mathscr{D}) = \phi(\mathscr{X})A^\dagger$
        $K(\mathscr{D}, \mathscr{D}) \leftarrow (A^\dagger)^T K(\mathscr{X}, \mathscr{X})A^\dagger$
        $k(x_i, \mathscr{D}) \leftarrow (A^\dagger)^T k(x_i, \mathscr{X})$
    end
```

With fixed codes for the training data (and a fixed kernel parameter), learning the dictionary can be expressed as solving the optimization problem

$$\min_{\mathscr{D}} \frac{1}{M} \sum_{i=1}^{M} \ell_\phi(\mathscr{D}; x_i). \tag{6.30}$$

Here, we make use of the *Representer theorem* [35] which enables us to express the dictionary as a linear combination of the training samples in RKHS. That is

$$\phi(d_j) = \sum_{i=1}^{M} v_{i,j}\phi(x_i), \tag{6.31}$$

where $\{v_{i,j}\}$ is the set of weights, now corresponding to our new unknowns. By stacking these weights for the M samples and the N dictionary elements in a matrix $V_{M \times N}$, we have

$$\phi(\mathscr{D}) = \phi(\mathscr{X})V \ . \tag{6.32}$$

The only term that depends on the dictionary is the reconstruction error (i.e, the first term in the objective of (6.25)). Given the matrix of sparse codes $A_{N \times M} = [y_1|y_2|\cdots|y_M]$, this term can be expressed as

$$\begin{aligned} R(V) &= \left\|\phi(\mathscr{X}) - \phi(\mathscr{X})VA\right\|_F^2 \\ &= \mathrm{Tr}\left(\phi(\mathscr{X})(\mathbf{I}_M - VA)(\mathbf{I}_M - VA)^T\phi(\mathscr{X})^T\right) \\ &= \mathrm{Tr}\left(K(\mathscr{X},\mathscr{X})(\mathbf{I}_M - VA - A^TV^T + VAA^TV^T)\right) \ . \end{aligned} \tag{6.33}$$

The new dictionary, fully defined by V, can then be obtained by zeroing out the gradient of $R(V)$ w.r.t. V. This yields

$$\nabla R(V) = 0 \Leftrightarrow V = (AA^T)^{-1}A = A^{\dagger} \ . \tag{6.34}$$

6.6 Experiments

To compare and contrast the proposed techniques against state-of-the-art methods, we used the Ballet dataset [44] to classify actions from videos. The Ballet dataset contains 44 videos collected from an instructional ballet DVD [44]. The dataset consists of eight complex motion patterns performed by three subjects, The actions include: *'left-to-right hand opening'*, *'right-to-left hand opening'*, *'standing hand opening'*, *'leg swinging'*, *'jumping'*, *'turning'*, *'hopping'*, and *'standing still'*. Figure 6.3 shows examples. The dataset is challenging due to the significant intra-class variations in terms of speed, spatial and temporal scale, clothing, and movement.

We extracted 2200 image sets by grouping 6 frames that exhibited the same action into one image set. We described each image set by a subspace of order 6 with histogram of oriented gradients (HOG) as frame descriptor [10] using SVD. To this

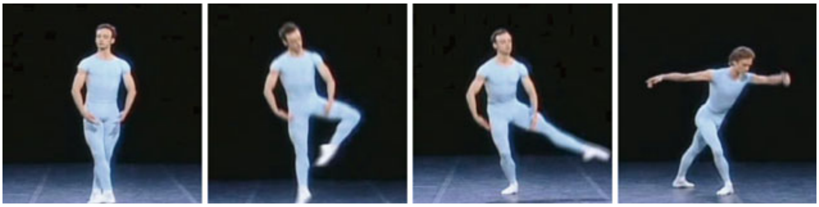

Fig. 6.3 Examples from the Ballet dataset [44]

Table 6.2 Average recognition rate on the Ballet dataset.

Method	gSC	kSC-RBF	kSC-Laplace	kSC-Poly
Accuracy	64.5	**69.7**	67.9	68.5

end, frame were first resized to 128 × 128 and HoG descriptor from four 64 × 64 nonoverlapping blocks were extracted. The HoG descriptors were concatenated to form the 124 dimensional frame descriptor.

Extracted subspaces were randomly split into training and testing sets (the number of image sets in both sets was even). The process of random splitting was repeated ten times and the average classification accuracy is reported.

Table 6.2 reports the average accuracies along their standard deviations for the studied methods. All the results were obtained by training a dictionary of size 128. To classify the sparse codes, we used a linear SVM. For the kSC algorithm, we used three different kernels, namely RBF, Laplace and a polynomial kernel of degree 2 as described in Sect. 6.5.

The highest accuracy is obtained by the universal RBF kernel. Interestingly, the polynomial kernel performs better than the Laplace kernel. All the kernel methods outperform the gSC algorithm, implying that the data is highly nonlinear.

References

1. P.A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds* (Princeton University Press, Princeton, 2008)
2. M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
3. V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magn. Reson. Med. **56**(2), 411–421 (2006)
4. R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Anal. Mach. Intell. **25**(2), 218–233 (2003)
5. E. Begelfor, M. Werman, Affine invariance revisited, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006), pp. 2087–2094
6. C. Berg, J.P.R. Christensen, P. Ressel, *Harmonic Analysis on Semigroups* (Springer, New York, 1984)
7. E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
8. H.E. Cetingul, M.J. Wright, P.M. Thompson, R. Vidal, Segmentation of high angular resolution diffusion MRI using sparse Riemannian manifold clustering. IEEE Trans. Med. Imaging **33**(2), 301–317 (2014)
9. S. Chen, C. Sanderson, M. Harandi, B.C. Lovell, Improved image set classification via joint sparse approximated nearest subspaces, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 452–459
10. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 886–893
11. D.L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)

12. M. Elad, *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing* (Springer, New York, 2010)
13. E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2765–2781 (2013)
14. M. Faraki, M. Harandi, F. Porikli, More about VLAD: a leap from Euclidean to Riemannian manifolds, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4951–4960
15. K.A. Gallivan, A. Srivastava, X. Liu, P. Van Dooren, Efficient algorithms for inferences on Grassmann manifolds, in *IEEE Workshop on Statistical Signal Processing* (2003), pp. 315–318
16. B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 2066–2073
17. R. Gopalan, R. Li, R. Chellappa, Unsupervised adaptation across domain shifts by generating intermediate data representations. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2288–2302 (2014). doi:10.1109/TPAMI.2013.249
18. J. Hamm, D.D. Lee, Grassmann discriminant analysis: a unifying view on subspace-based learning, in *Proceedings of the International Conference on Machine Learning (ICML)* (2008), pp. 376–383
19. M. Harandi, M. Salzmann, Riemannian coding and dictionary learning: kernels to the rescue, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 3926–3935
20. M. Harandi, R. Hartley, C. Shen, B. Lovell, C. Sanderson, Extrinsic methods for coding and dictionary learning on Grassmann manifolds. Int. J. Comput. Vis. **114**(2–3), 113–136 (2015)
21. M.T. Harandi, M. Salzmann, S. Jayasumana, R. Hartley, H. Li, Expanding the family of Grassmannian kernels: an embedding perspective, in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 8695, Lecture Notes in Computer Science, ed. by D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Springer International Publishing, Cham, 2014), pp. 408–423. doi:10.1007/978-3-319-10584-0_27
22. M.T. Harandi, R. Hartley, B.C. Lovell, C. Sanderson, Sparse coding on symmetric positive definite manifolds using Bregman divergences. IEEE Trans. Neural Netw. Learn. Syst. (TNNLS) **PP**(99), 1–1 (2015)
23. R. Hartley, J. Trumpf, Y. Dai, H. Li, Rotation averaging. Int. J. Comput. Vis. **103**(3), 267–305 (2013)
24. U. Helmke, K. Hper, J. Trumpf, Newton's method on Gramann manifolds (2007)
25. U. Helmke, K. Hüper, P.Y. Lee, J.B. Moore, Essential matrix estimation using Gauss-Newton iterations on a manifold. Int. J. Comput. Vis. **74**(2), 117–136 (2007). doi:10.1007/s11263-006-0005-0
26. J. Ho, Y. Xie, B. Vemuri, On a nonlinear generalization of sparse coding and dictionary learning, in *Proceedings of the International Conference on Machine Learning (ICML)* (2013), pp. 1480–1488
27. S. Jayasumana, R. Hartley, M. Salzmann, H. Li, M. Harandi, Kernel methods on Riemannian manifolds with Gaussian RBF kernels. IEEE Trans. Pattern Anal. Mach. Intell. **37**(12), 2464–2477 (2015). doi:10.1109/TPAMI.2015.2414422
28. H. Karcher, Riemannian center of mass and mollifier smoothing. Commun. Pure Appl. Math. **30**(5), 509–541 (1977)
29. J.M. Lee, *Introduction to Smooth Manifolds*, vol. 218 (Springer, New York, 2012)
30. J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2008), pp. 1–8
31. J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration. IEEE Trans. Image Process. (TIP) **17**(1), 53–69 (2008)
32. J.H. Manton, A globally convergent numerical algorithm for computing the centre of mass on compact lie groups. Int. Conf. Control Autom. Robot. Vis. **3**, 2211–2216 (2004)

33. B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**(6583), 607–609 (1996)
34. R. Ramamoorthi, Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object. IEEE Trans. Pattern Anal. Mach. Intell. **24**(10), 1322–1333 (2002)
35. B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, *Computational Learning Theory* (Springer, New York, 2001), pp. 416–426
36. J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, 2004)
37. S. Shirazi, M. Harandi, B. Lovell, C. Sanderson, Object tracking via non-Euclidean geometry: a Grassmann approach, in *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2014), pp. 901–908. doi:10.1109/WACV.2014.6836008
38. I. Steinwart, A. Christmann, *Support Vector Machines* (Springer, Berlin, 2008)
39. R. Subbarao, P. Meer, Nonlinear mean shift over Riemannian manifolds. Int. J. Comput. Vis. **84**(1), 1–20 (2009)
40. R. Tibshirani, Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**, 267–288 (1996)
41. P. Turaga, A. Veeraraghavan, A. Srivastava, R. Chellappa, Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2273–2286 (2011)
42. R. Vemulapalli, J.K. Pillai, R. Chellappa, Kernel learning for extrinsic classification of manifold features, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 1782–1789
43. J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 3360–3367
44. Y. Wang, G. Mori, Human action recognition by semilatent topic models. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1762–1774 (2009)
45. L. Wolf, A. Shashua, Learning over sets using kernel principal angles. J. Mach. Learn. Res. **4**, 913–931 (2003)
46. J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
47. J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition. Proc. IEEE **98**(6), 1031–1044 (2010)
48. J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 1794–1801