Stat Comput (2010) 20: 267–282
DOI 10.1007/s11222-009-9120-4

Optimal linear projections for enhancing desired data statistics

Evgenia Rubinshtein · Anuj Srivastava

Received: 26 June 2007 / Accepted: 12 March 2009 / Published online: 28 March 2009

Abstract Problems involving high-dimensional data, such as pattern recognition, image analysis, and gene clustering, often require a preliminary step of dimension reduction before or during statistical analysis. If one restricts to a linear technique for dimension reduction, the remaining issue is the choice of the projection. This choice can be dictated by desire to maximize certain statistical criteria, including variance, kurtosis, sparseness, and entropy, of the projected data. Motivations for such criteria comes from past empirical studies of statistics of natural and urban images. We present a geometric framework for finding projections that are optimal for obtaining certain desired statistical properties. Our approach is to define an objective function on spaces of orthogonal linear projections—Stiefel and Grassmann manifolds, and to use gradient techniques to optimize that function. This construction uses the geometries of these manifolds to perform the optimization. Experimental results are presented to demonstrate these ideas for natural and facial images.

Keywords Dimension reduction · Linear projection · Numerical optimization on Grassmann and Stiefel manifolds · Stochastic optimization · Optimization algorithm

E. Rubinshtein (✉)
Vladivostok State University of Economics and Service, Vladivostok, 690600, Russia
e-mail: fineasusual@mail.ru

A. Srivastava
Department of Statistics, Florida State University, Tallahassee, FL 32306, USA
e-mail: anuj@stat.fsu.edu

1 Introduction

In many applications involving pattern recognition, image analysis, meteorology, and environmental sciences, the presence of large datasets prohibit efficient use of statistical analysis. It becomes imperative to use a dimension-reduction technique either before or during statistical analysis of data. In the context of pattern analysis, one is often interested in extracting relevant features from observed data and the use of linear methods is prevalent for this feature extraction. In some applications, such as face recognition using images, the underlying variability in observed data is known to result from only a handful of physical variables, such as pose, shape, and illumination, and that also provides a strong motivation for seeking low-dimension representations of data. Such low-dimensional representations can also provide a useful immunity to observation noise, or clutter, that is typically high dimensional. In statistics, there is a great interest in variable selection for problems involving clustering and classification of high-dimensional data. In all these situations, the choice of feature, or the choice of projection leading to dimension reduction, is itself an important issue. In fact, a number of criteria have emerged in recent years that guide the process of dimension reduction. These criteria include combinations of properties such as sparseness, correlation, variance, kurtosis, and independence. Given one of these criteria how can we find a linear projection, or a basis, such that the data projected using this projection will achieve the given criterion? A solution to this problem is the subject of this paper.

Consider the following setup. Let $\mathbf{y}$ be an $n \times 1$ vector of random variables and we are interested in its statistical analysis—density estimation, modeling, testing, etc. In case n is very high, this analysis is intractable if tried directly on $\mathbf{y}$. For example, in analysis dealing with images

of size 100×100, n is 10^4, and a direct analysis of $\mathbf{y}$ is difficult. A common approach is to reduce dimension from n to d, with $d \ll n$, using a linear transformation. A linear transformation is a $d \times n$ matrix that pre-multiplies $\mathbf{y}$. It seems natural and efficient to restrict to matrices with linearly independent rows, or even further, to impose orthonormality constraint on the rows. For instance, let U be an $n \times d$ orthogonal matrix denoting an orthonormal basis of a d-dimensional subspace of $\mathbb{R}^n$. Then, the vector $\mathbf{z} = U^T\mathbf{y} \in \mathbb{R}^d$, also called the vector of coefficients, is a d-dimensional representation of $\mathbf{y}$ or a projection of $\mathbf{y}$.

In this paper we are concerned with the choice of U. Of course, depending upon the application and the data, the actual value of U will differ. The goal is to develop a principled approach where one chooses a criterion and then finds an optimal U under that criterion. Next, we present a number of criteria that have been used in selecting U.

1.1 Past criteria for dimension reduction

We start by listing some commonly used ideas.

1. *Principal component analysis*: One of the most commonly used method for dimension reduction is principal component analysis (PCA). In this approach, one chooses U in such a way that the sum of variances of the projected coefficients is maximized. That is,

$$\hat{U}_{\text{PCA}} = \underset{U}{\text{argmax}} \left(\sum_{i=1}^{d} \text{variance}(\mathbf{z}_i) \right).$$

   Another way to state this condition is: $\hat{U}_{\text{PCA}} = \text{argmax}_U E[\|\mathbf{y} - UU^T\mathbf{y}\|^2]$, where $\|\cdot\|$ implies the two norm of a vector and E denotes the expectation with respect to the joint probability density function of components of $\mathbf{y}$. In case the full density of $\mathbf{y}$ is not available, one maximizes the variance estimated from the samples of $\mathbf{y}$. Let Y be the observation matrix such that $Y_{i,l}$ denotes the lth observation of $\mathbf{y}_i$, $1 \le i \le n$ and $1 \le l \le k$, and define $Z = U^TY \in \mathbb{R}^{d \times k}$. Then, define

$$F_V = \frac{1}{k-1} \sum_{i=1}^{d} \sum_{l=1}^{k} (Z_{i,l} - \bar{z}_i)^2, \quad \bar{z}_i = \frac{1}{k} \sum_{l=1}^{k} Z_{i,l}. \tag{1}$$

   One reason for popularity of PCA is that the optimal projection, $\hat{U}_{PCA}$, can be determined analytically. The solution is obtained using the singular value decomposition (SVD) of the covariance of $\mathbf{y}$. Additionally, if $\mathbf{y}$ is multivariate normal, then the elements of $\mathbf{z}$ are statistically independent, and this provides a natural decomposition of factors influencing $\mathbf{y}$.
2. *Canonical correlation analysis* (CCA): For studying correlations between two given vectors $\mathbf{x}$ and $\mathbf{y}$ of random variables, with finite second moments, one seeks their linear projections such that the correlation between the projections are maximized (see for example Johnson and Wichern 2001). If Σ_x and Σ_y are the covariance matrices of $\mathbf{x}$ and $\mathbf{y}$, respectively, and Σ_{xy} is the cross-covariance, then the optimal projectors are related to the dominant eigen vectors of the matrix $\Sigma_x^{-1/2}\Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T\Sigma_x^{-1/2}$. Since the covariance matrices are non-negative definite and symmetric, the projections vectors can be considered as the columns of an orthogonal matrix U.
3. *Fisher's discriminant analysis*: In case of labeled data, i.e. the data consists of observations from different classes and the classes are known, the projection is chosen to maximize separation between the classes. A standard approach is the use of Fisher's discriminant analysis as follows. Define *between-class* scatter matrix by: $S_B = \sum_j [(\mu_j - \mu)(\mu_j - \mu)^T] \in \mathbb{R}^{n \times n}$, where j is the index for classes, $\mu_j = E[\mathbf{y}_j]$, and μ is the overall mean. The *within-class* scatter matrix is given by: $S_W = \sum_j E[(\mathbf{y}_j - \mu_j)(\mathbf{y}_j - \mu_j)^T]$, $S_W \in \mathbb{R}^{n \times n}$. The desired basis is now obtained by solving:

$$[\hat{U}] = \underset{[U]}{\text{argmax}} \frac{\det(U^T S_B U)}{\det(U^T S_W U)}, \tag{2}$$

   where $\det(\cdot)$ denotes matrix determinant. Like PCA and CCA, the solution can be obtained directly, using a generalized eigenvalue decomposition (Golub and Van Loan 1989).

   In contrast to PCA, CCA and FDA, there are some other criteria that do not result in an analytical solutions and require numerical strategies to find an optimal U. Some examples are listed next.
4. *Sufficient dimension reduction*: This idea is mainly used in linear regression and model building problems. Pioneered by Cook and colleagues (see Cook and Li 2002; Cook 2004 and references therein), the main idea in this approach is to find subspaces for projecting a large vector $\mathbf{x}$ such that, given the projected vector, the (univariate) response variable y is independent of $\mathbf{x}$. This is considered a projection of $\mathbf{x}$ without loss of any information about y. The smallest such subspace is called the central subspace. Several methods have been proposed for finding the central subspace, some of which can be stated as problems in optimization over the space of all projectors.
5. *Independent component analysis*: Here the goal is to find a projection such that the projected components are statistically independent. There are several ways to formulate ICA (Hyvärinen et al. 2001); one way is to use the cost function:

$$\text{KL}(P(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d) || P_1(\mathbf{z}_1) P_2(\mathbf{z}_2) \ldots P_d(\mathbf{z}_d)), \tag{3}$$

   where KL denotes the Kullback-Leibler divergence. In this definition, P denotes the joint probability, and P_is

denote the marginal probabilities of the random variables $\mathbf{z}_i$s. The desired transformation is obtained by finding a minimizer of this cost function. Since it is difficult to estimate KL divergence using observations of random variables, several approximations of Kullback-Leibler function have been applied to obtain ICA in the literature (Hyvärinen et al. 2001). One idea is to maximize kurtosis of the projected variables and that is one of the main ideas pursued in this current paper. Comon (1994) proposed the use of negentropy, and further its polynomial approximation, to approximate minimization of mutual information and Bell and Sejnowski (1995) used a stochastic gradient technique to solve such optimization problems. It must be noted that some of these formulations do not require orthogonality of basis; in fact, they often use over-complete or non-orthogonal bases. Hyvärinen (1999) proposed a "FastICA" algorithm for computing ICA using an over-complete basis.

1.2 More recent criteria for feature extraction

Some additional ideas have been proposed in the recent years are presented next. Many of them are motivated in part by applications in image analysis where empirical studies have shown that image statistics, under a variety of representations, exhibit certain striking properties. As summarized in Srivastava et al. (2003), these properties are: (i) estimated densities are unimodal with modes at zeros, (ii) the underlying random quantities are leptokurtic, i.e. their kurtosis are much larger than that of a Gaussian and the tails are heavier. Consequently, there is interest in seeking representations that emphasize these properties.

1. *Maximal kurtosis*: There are several motivations to seek projections that maximize kurtosis. Firstly, kurtosis has been proposed earlier as an objective function for independent component analysis (Hyvärinen et al. 2001). Secondly, experiments indicate that the level of non-Gaussianity of pixel values in a image seems to relate to information content of images (Srivastava et al. 2003). Therefore, there is interest in seeking linear projections that maximize non-Gaussianity and result in heavy-tailed distributions. Of course, a difficult question is: How should one measure non-Gaussianity? There are several ideas but perhaps the simplest one is to use kurtosis. The kurtosis has the nice property that it is invariant to certain transformations such as translation and scale of the original image vector $\mathbf{y}$; these transformations are considered as nuisance variables in image analysis and may result from changes in intensity of illumination or color maps. In other words, scaling of pixels, or adding a constant to pixels, does not often change the information content, and hence the basis search criterion should be invariant to them.

   For an $n \times d$ matrix U, let $\mathbf{z} = U^T\mathbf{y}$ be the d-dimensional projection of $\mathbf{y}$ into $\mathbb{R}^d$. We are interested in choosing U that maximizes

$$\sum_{i=1}^{d} \text{kurt}(\mathbf{z}_i),$$

$$\text{where kurt}(\mathbf{z}_i) = \frac{E[(\mathbf{z}_i - \mu_i)^4]}{E[(\mathbf{z}_i - \mu_i)^2]^2} \text{ and } \mu_i = E[\mathbf{z}_i].$$

   Note that $\mathbf{z}$ is a linear transformation of $\mathbf{y}$, so the moments of $\mathbf{z}$ can in principle be computed from the moments of $\mathbf{y}$. However, in many practical situations we do not have exact moments of $\mathbf{y}$, and instead are given its observations. So we will focus on estimated quantities, such as the sample kurtosis, throughout this paper. As earlier, let Y be the observation matrix such that $Y_{i,l}$ denotes the lth observation of $\mathbf{y}_i$, $1 \le i \le n$ and $1 \le l \le k$, and define $Z = U^T Y \in \mathbb{R}^{d\times k}$. Then, the total estimated kurtosis of $\mathbf{z}$ is given by:

$$F_K = \sum_i F_{K,i}, \quad F_{K,i} \equiv \frac{(k-1)^2}{k} \frac{\sum_{l=1}^{k}(Z_{i,l} - \bar{z}_i)^4}{(\sum_{l=1}^{k}(Z_{i,l} - \bar{z}_i)^2)^2},$$

$$\text{where } \bar{z}_i = \frac{1}{k}\sum_{l=1}^{k} Z_{i,l}. \tag{4}$$

   The definition of sample kurtosis sets up the optimization problem for maximizing kurtosis: $\hat{U} = \text{argmax}_U F_K(U; Y)$.
2. *Maximal sparseness*: Another criterion of interest in feature extraction and dimension reduction is sparseness. A collection of random variables is considered sparse if the observations of that collection contains only a few non-zero values with a high probability. Motivated by the studies of human visual system and by a growing understanding of its efficiency, researchers have focused on sparsity of the projected data as a criterion for dimension reduction. Empirical studies of natural images show that the distributions of their wavelet coefficients are typically sparse (Donoho and Flesia 2001). This means that the energy of the image is mostly concentrated in a small proportion of wavelet coefficients (Mallat 1989; Olshausen and Field 1996). This result seems intuitively relevant because natural images may generally be described in terms of a small number of structural primitives—for example, edges, lines, or other elementary features. One of the ways to quantify sparseness of the random variable v is (Olshausen and Field 1996; Field 1994): $\text{spars}(v) = -E\{\log(1 + v^2)\}$. The sample

sparseness of $\mathbf{z}_i$ is given by: $-\frac{1}{k}\sum_{j=1}^{k}\log(1+Z_{i,j}^2)$, and the total sparseness of the vector $\mathbf{z}$ is given by:

$$F_S(U;Y) = -\frac{1}{k}\sum_{i=1}^{d}\sum_{j=1}^{k}\log(1+Z_{i,j}^2), \quad Z = U^T Y. \tag{5}$$

Maximizing sparseness is to solve the problem: $\hat{U} = \text{argmax}_U\, F_S(U;Y)$. An obvious solution, in case of unconstrained optimization, is $Z_{i,j} = 0, i = 1,\ldots,d,\ j = 1,\ldots,k$. Therefore, sparseness is seldom used alone as a criterion for dimension reduction; it is used in conjunction with other criteria to form composite objective functions.

3. *Optimal entropy*: In physics, entropy is considered to be a measure of chaos or uncertainty in a dynamic system. Similarly, in information theory, entropy provides a measure of information contained in a random quantity. Low entropy implies larger information and vice-versa. Entropy also plays a very important role in independent component analysis (Hyvärinen et al. 2001). For a continuous scalar random variable v, with probability density function $f(v)$, the (differential) entropy is defined as $H(v) = -\int f_v(t)\log f_v(t)dt$. One use of entropy is in defining the mutual information between two random variables v and w, according to:

$$I(v;w) = H(v) - H(v|w),$$

where $H(v|w)$ is conditional entropy and denotes the uncertainty about v when w is known. Thus, $I(v;w)$ is the reduction in uncertainty about v due to the knowledge of w. One can use this information-theoretic framework in dimension reduction as follows. We may seek projections such that the mutual information between two random vectors is maximized. Or, we might seek projections that make two random vectors independent of each other. In either case, the idea is to maximize or minimize an entropy function, conditional or unconditional, by choosing optimal projections. We will focus on such a subproblem in the current paper. The evaluation of entropy requires the knowledge of underlying probability density function. In case one only has samples, an estimate of the density function is used instead. For instance, a kernel density estimator for $\mathbf{z}_i$, using the Gaussian kernel, is:

$$\hat{p}_i(x) = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-Z_{i,j})^2}{2\sigma^2}}, \tag{6}$$

where σ is the bandwidth of the kernel. To find the total entropy associated with the vector $\mathbf{z}$, we need to estimate their joint density function. To avoid that calculation, we make a gross approximation and consider the sum of individual entropies:

$$\hat{H} = -\sum_{i=1}^{d}\left(\int_{\mathbb{R}}\hat{p}_i(x)\log(\hat{p}_i(x))\right) \equiv \sum_{i=1}^{d}H_i, \tag{7}$$

and the optimization problem is to find $\hat{U} = \text{argmax}_U \sum_{i=1}^{d} H_i(U;Y)$. We must point out that in the literature on independent component analysis (ICA), one rarely uses an estimated density function to study entropy. It is most often approximated using lower order moments and polynomials involving them (Hyvärinen et al. 2001).

Additional criteria can be generated by taking convex combinations of the individual criteria listed above. In this paper we consider a few combinations described next.

- As a first combination, we study a convex combination of kurtosis and variance. The cost function to minimize is given by $F_{KV} = \lambda F_K + (1-\lambda)F_V$, for a $0 < \lambda < 1$. For small values of λ we expect the maximizer to be similar to U_{PCA} but for other values the solution will differ from U_{PCA}.
- Another possibility is to use a convex combination of kurtosis and sparseness. Thus, maximizing F_{KS} will result in a basis that not only increase kurtosis but also sparseness: $F_{KS} = \lambda F_K + (1-\lambda)F_S$.
- One can also study a convex combination of variance and sparseness, resulting in the goal function F_{SV}: $F_{SV} = \lambda F_S + (1-\lambda)F_V$.

Although we consider these composite criteria as problems in joint optimization, they can also be treated as Bayesian problems, or penalized likelihood problems, with one of the terms providing a prior density with the other specifying the likelihood function (Hyvärinen et al. 2001).

1.3 Our approach: optimization over manifolds

Based on the previous discussion one can envision a variety of criteria that can be used for finding a suitable projection, and the choice of an appropriate criterion depends on the nature of the problem. Given such a criterion, how does one find an optimal projection U? Our approach is to optimize the associated goal function over the space of all possible orthogonal projections U. This amounts to searching for $\hat{U}$, where

$$\hat{U} = \underset{U}{\text{argmax}}\, F(U;Y), \tag{8}$$

where F is a scalar-valued function. Since analytical solutions for Fs of interest are not known, we will take a numerical approach to search for $\hat{U}$. What is the set over which this optimization should be performed? There are two possibilities:

1. U is an $n \times d$ orthogonal matrix and the required space could be the set of all such matrices. This set is called a Stiefel manifold:

$$\mathcal{S}_{n,d} = \{U \in \mathbb{R}^{n \times d} | U^T U = I_d\}. \quad (9)$$

2. In some cases the goal function depends on the subspace and not a particular basis we choose to represent it. In other words, $F(U) = F(UO)$ where O is a $d \times d$ rotation matrix. For instance, this is the case for the variance function F_V. In this case one searches over the space of all subspaces rather than searching over the space of all orthogonal bases. This set is called the Grassmann manifold $\mathcal{G}_{n,d}$.

Both Stiefel and Grassmann manifolds are *nonlinear spaces*, i.e. they are not vector spaces, and the traditional optimization techniques, such as those used in Hyvärinen et al. (2001), do not apply directly. We will use the differential geometry of these two manifolds to construct gradient processes to solve optimization problems. Several papers have addressed the problem of solving numerical optimization on Stiefel and Grassmann manifolds. Of those, we note the seminal paper by Edelman et al. (1998) which utilizes the geometry of these manifolds to derive deterministic gradient approaches such as Newton-Raphson method. In earlier works, we have applied a stochastic gradient search algorithm to maximize classification performance in image analysis (Liu et al. 2003; Srivastava and Liu 2005). Similar problems have also been studied by Fiori and colleagues (Fiori 2002; Fiori 2002), especially for independent component analysis. The approach taken in the current paper, but for cost functions involved in pattern recognition and classification, has earlier been explored by Srivastava and Liu (2005).

The rest of this paper is organized as follows. Section 2 describes the representation of orthogonal linear projections as elements of Stiefel and Grassmann manifolds. Section 3 introduces basic elements from differential geometry of these manifolds that are important in our approach, and Section 4 describes our solution to the optimization problems formulated in Section 1.2. Finally, Section 5 presents some experimental results using natural and face image databases.

2 Representations of linear projections

We are interested in linear transformations that can be used for reducing data size. Such transformations can be denoted by $n \times d$ non-singular matrices. If the columns are forced to be linearly independent, which seems natural for studying linear transformations, an efficient representation is obtained by further assuming that the columns are orthogonal with unit length. Denoting such a linear transformation via a matrix $U \in \mathbb{R}^{n \times d}$, U satisfies the property that $U^T U = I_d$, where I_d is the $d \times d$ identity matrix. This orthogonality constraint sets up our representation spaces as follows.

1. *Stiefel manifold*: The set of all $n \times d$ orthogonal matrices forms a Stiefel manifold $\mathcal{S}_{n,d}$, as stated in Eq. 9. Each element of $\mathcal{S}_{n,d}$ provides an orthonormal basis for a d-dimensional subspace of $\mathbb{R}^n$. $\mathcal{S}_{n,d}$ can also be viewed as a quotient space of $SO(n)$, where $SO(n) = \{Q \in \mathbb{R}^{n \times n} | Q^T Q = I_n, \det(Q) = 1\}$, as follows. First, consider $SO(n-d)$ as a subset of $SO(n)$ using the embedding: $\phi_1 : SO(n-d) \mapsto SO(n)$, defined by

$$\phi_1(A) = \begin{bmatrix} I_d & 0 \\ 0 & A \end{bmatrix} \in SO(n), \quad A \in SO(n-d).$$

Accordingly, $SO(n-d)$ here consists of those rotations in $SO(n)$ that rotate only the last $(n-d)$ components in $\mathbb{R}^n$, leaving the first d unchanged. In this notation, $\mathcal{S}_{n,d}$ can be viewed as the quotient space $\mathcal{S}_{n,d} = SO(n)/\phi_1(SO(n-d))$ or simply $SO(n)/SO(n-d)$.

2. *Grassmann manifold*: As stated earlier, a Grassmann manifold is the set of all d-dimensional subspaces of $\mathbb{R}^n$. Let $SO(d) \times SO(n-d)$ be a subset of $SO(n)$ using the embedding $\phi_2 : (SO(d) \times SO(n-d)) \mapsto SO(n)$:

$$\phi_2(A_1, A_2) = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \in SO(n),$$
$$A_1 \in SO(d), \; A_2 \in SO(n-d).$$

Then, $\mathcal{G}_{n,d}$ is a quotient space $\mathcal{S}_{n,d}/SO(d)$ or $SO(n)/\phi_2(SO(d) \times SO(n-d))$, or simply $SO(n)/(SO(d) \times SO(n-d))$.

For an orthogonal matrix $U \in \mathbb{R}^{n \times d}$, we will use $[U]$ to denote an element of $\mathcal{G}_{n,d}$, where

$$[U] = \{UO \in \mathbb{R}^{n \times d} | O \in SO(d)\}.$$

That is, $[U]$ denotes the equivalence class of all orthogonal bases spanning the same d-dimensional subspace of $\mathbb{R}^n$.

In summary, (i) elements of $SO(n)$ form full rotations in $\mathbb{R}^n$, (ii) elements of $\mathcal{S}_{n,d}$ form a subset where rotations within an $(n-d)$-dimensional subspace, corresponding to the last $n-d$ components of $\mathbb{R}^n$, are ignored, and (iii) elements of $\mathcal{G}_{n,d}$ form a subset where, additionally, rotations within the first d components are also ignored. Consequently, many properties of $\mathcal{S}_{n,d}$ and $\mathcal{G}_{n,d}$ are inherited from $SO(n)$. Both are compact manifolds and continuous functions defined on them attain their maximum (or minimum) values on the manifolds.

We emphasize the choice of orthogonal bases in representing linear transformations as it leads to a significant reduction in computational cost. Solving for an orthogonal

basis on $\mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$ leads to a smaller search space as compared to searching for optimal linear transformations on (nd)-dimensional space of $n \times d$ matrices. It also provides stability to iterative optimization algorithms by ensuring that the basis vectors remain unit length and the basis matrix is always full ranked.

3 Tools for gradient searches

Our approach is to use stochastic gradient to solve the optimization problem stated in Eq. 8. Before we describe the final algorithm, we present some basic tools from differential geometry of $\mathcal{S}_{n,d}$ that are needed in this optimization. In particular, we are interested in defining tangent spaces, gradient vector fields, and gradient flows.

3.1 Tangent spaces of Stiefel and Grassmann manifolds

In a gradient-based search we need to define and compute the gradient of F with respect to the elements of $\mathcal{S}_{n,d}$ and $\mathcal{G}_{n,d}$. Since these manifolds are nonlinear, this is accomplished using tangent spaces, whose elements also act as derivatives of functions. Nonlinearity of these spaces causes the tangent spaces to differ from point to point on.

1. *Stiefel case*: Let $J \in \mathbb{R}^{n\times d}$ be a tall-skinny matrix, made up of the first d columns of I_n; J acts as the "identity" element in $\mathcal{S}_{n,d}$. Let $Q \in SO(n)$ be a matrix that rotates the columns of U to align with the columns of J, i.e. $QU = J$, or $Q^T = [U\ V]$, where $V \in \mathbb{R}^{n\times(n-d)}$ is an orthogonal basis of the null space of U. Note that the choice of Q is not unique. In this notation, the space of vectors tangent to $\mathcal{S}_{n,d}$ at a point U, denoted $T_U(\mathcal{S}_{n,d})$, can be stated as follows:

$$\begin{aligned} &T_U(\mathcal{S}_{n,d}) \\ &= \left\{ Q^T \begin{bmatrix} C & B \\ -B^T & 0 \end{bmatrix} J \mid C = -C^T, C \in \mathbb{R}^{d\times d}, \right. \\ &\quad \left. B \in \mathbb{R}^{d\times(n-d)} \right\} \\ &= \left\{ Q^T \begin{bmatrix} C \\ -B^T \end{bmatrix} \mid C = -C^T, C \in \mathbb{R}^{d\times d}, \right. \\ &\quad \left. B \in \mathbb{R}^{d\times(n-d)} \right\} \\ &= \{UC - VB^T \mid C = -C^T, C \in \mathbb{R}^{d\times d}, \\ &\quad B \in \mathbb{R}^{d\times(n-d)}\}. \end{aligned} \tag{10}$$

Later on, we will be interested in projecting an arbitrary matrix $D \in \mathbb{R}^{n\times d}$ onto the tangent space $T_U(\mathcal{S}_{n,d})$ for a given point $U \in \mathcal{S}_{n,d}$. According to Eq. 10, an element of $T_U(\mathcal{S}_{n,d})$ takes the form $UC - VB^T$, we need to find an appropriate C and B such that $\|D - UC + VB^T\|^2$ becomes zero. This leads to: $C^* = \frac{(-D^TU+U^TD)}{2}$ and $B^* = -D^TV$. In other words, the orthogonal projection of D onto the tangent space $T_U(\mathcal{S}_{n,d})$ is given by $\Pi_1 : \mathbb{R}^{n\times d} \mapsto T_U(\mathcal{S}_{n,d})$:

$$\begin{aligned} \Pi_1(D) = UC^* - VB^{*T} &= U\frac{(-D^TU + U^TD)}{2} \\ &+ VV^TD. \end{aligned} \tag{11}$$

2. *Grassmann case*: The tangent space at $[U] \in \mathcal{G}_{n,d}$ is given by:

$$\begin{aligned} T_{[U]}(\mathcal{G}_{n,d}) &= \left\{ Q^T \begin{bmatrix} 0 & B \\ -B^T & 0 \end{bmatrix} J \mid B \in \mathbb{R}^{d\times(n-d)} \right\} \\ &= \{-VB^T \mid B \in \mathbb{R}^{d\times(n-d)}\}. \end{aligned} \tag{12}$$

The orthogonal projection of D onto the tangent space $T_U(\mathcal{G}_{n,d})$ is given by $\Pi_2 : \mathbb{R}^{n\times d} \mapsto T_U(\mathcal{G}_{n,d})$:

$$\Pi_2(D) = VV^TD, \quad \text{i.e. } B^* = -D^TV. \tag{13}$$

The formulas are very similar in the two cases except $C = 0$ in the second case.

3.2 Gradient vector fields

We can now define gradient vector fields associated with the given functions F on $\mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$. A gradient vector field is a map from a space to its tangent spaces such that it assigns a gradient vector at each point. In other words, for any $U \in \mathcal{S}_{n,d}$, $G(U) \in T_U(\mathcal{S}_{n,d})$ is the gradient of F at U. We remind the reader that the gradient at a point is the direction of maximal increase in the value of F at that point. There are several ways of computing G. We will take an extrinsic approach where we will first compute the gradient of F in the ambient space $\mathbb{R}^{n\times d}$. Then, we will project this full gradient to $T_U(\mathcal{S}_{n,d})$ to obtain the gradient on $\mathcal{S}_{n,d}$.

Let $D = \frac{dF}{dU}$ be the gradient of F in $\mathbb{R}^{n\times d}$ for a goal function F, i.e. $D_{l,p} = \frac{\partial F}{\partial U_{l,p}}$. We can compute D using the chain rule as follows:

$$D_{l,p} = \frac{dF}{dU_{l,p}} = \frac{dF}{dZ}\frac{dZ}{dU_{l,p}} = \sum_{i=1}^{d}\sum_{j=1}^{k} \frac{\partial F}{\partial Z_{i,j}}\frac{dZ_{i,j}}{dU_{l,p}}. \tag{14}$$

Recall that $Y_{l,j}$ is the lth observation of $\mathbf{y}_j$, U is the projection matrix and $Z = U^TY$ is the matrix of observations of $\mathbf{z}$. The partial derivative $\frac{\partial Z_{i,j}}{\partial U_{l,p}}$ can be shown to be $\delta_{i,p}Y_{l,j}$, where $\delta_{i,p}$ is the Kronecker delta. Combining these terms, we find that the term $\frac{\partial Z_{i,j}}{\partial U}$ is an $n \times d$ matrix that contains $Y_{\cdot,j} \in \mathbb{R}^n$ in its ith column and zero everywhere else. Therefore, the matrix D simplifies to:

$$D = YW^T,$$

where the entries of W are $W_{i,j} = \frac{\partial F}{\partial Z_{i,j}}$. (15)

The remaining issue is to compute the matrix W for a given objective function F. Once we have W and, thus, $D = YW^T$, D can be projected onto the tangent spaces $T_U(\mathcal{S}_{n,d})$ and $T_{[U]}(\mathcal{G}_{n,d})$ using the projection Π_1 and Π_2, respectively, to obtain a gradient vector field on $\mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$.

Next, we study the calculation of W for some of the goal functions considered earlier in Sect. 1.

- *Kurtosis:* For F_K given by Eq. 4, its derivative with respect to the (elements of) matrix Z is given by:

$$(W_K)_{i,j} = \frac{\partial F_K}{\partial Z_{i,j}} = \frac{(k-1)^2}{k} \frac{a-b}{(\sum_{l=1}^k (Z_{i,l} - \bar{z}_i)^2)^3}, \quad (16)$$

where $i = 1, \ldots, d$, $j = 1, \ldots, k$, and where

$$a = 4\left(\sum_{l=1}^k (Z_{i,l} - \bar{z}_i)^3 \left(\delta_{l,j} - \frac{1}{k}\right)\right) \sum_{l=1}^k (Z_{i,l} - \bar{z}_i)^2,$$

$$b = 4\left(\sum_{l=1}^k (Z_{i,l} - \bar{z}_i)^4\right)\left(\sum_{l=1}^k (Z_{i,l} - \bar{z}_i)\left(\delta_{l,j} - \frac{1}{k}\right)\right).$$

- *Sparseness*: For F_S given in Eq. 5, its derivative with respect to the matrix Z is:

$$(W_S)_{i,j} = \frac{\partial F_S}{\partial Z_{i,j}} = -\frac{2}{k} \frac{Z_{i,j}}{1 + Z_{i,j}^2}. \quad (17)$$

- *Variance*: For F_V given by Eq. 1, its derivative with respect to the matrix Z is:

$$(W_V)_{i,j} = \frac{\partial F_V}{\partial Z_{i,j}} = \frac{2}{k-1}\left[(Z_{i,j} - \bar{z}_i) - \frac{1}{k}\sum_{l=1}^k (Z_{i,l} - \bar{z}_i)\right]. \quad (18)$$

- *Entropy*: If we replace the estimated density function in Eq. 7 with its discrete approximations, the integral is replaced by a summation, with the total entropy being:

$$H = -\sum_{r=1}^d \left(\sum_{l=1}^N \hat{p}_r^l \log(\hat{p}_r^l)\right),$$

where $\hat{p}_r^l$ is the estimated values of pdf of $\mathbf{z}_r$, evaluated on the lth bin denoted by t_l, and N is the number of bins. $\hat{p}_r^l$ is estimated using the rth row of the matrix $Z = U^T Y$. Now, we can calculate the required derivative $W_H = \frac{dH}{dZ}$ as follows:

$$(W_H)_{i,j} = \frac{dH}{dZ_{i,j}} = \sum_{r=1}^d \sum_{l=1}^N \frac{dH}{d\hat{p}_r^l} \frac{d\hat{p}_r^l}{dZ_{i,j}}$$

$$= -\frac{1}{k\sqrt{2\pi}\sigma^3} \sum_{l=1}^N (1 + \log(\hat{p}_i^l))(t_l - Z_{i,j}) \times e^{-\frac{(t_l - Z_{i,j})^2}{2\sigma^2}}, \quad (19)$$

where $\hat{p}_i^l = \frac{1}{k\sqrt{2\pi}\sigma} \sum_{j=1}^k e^{-\frac{(t_l - Z_{i,j})^2}{2\sigma^2}}$.

- *Kurtosis and variance*: If the objective function is given by $\mathbf{F} \equiv \lambda \mathbf{F}_K + (1-\lambda)\mathbf{F}_V$, then its derivative with respect to Z is given by $W_{KV} = \lambda W_K + (1-\lambda) W_V$.
- *Variance and sparseness*: For the objective function $\mathbf{F} \equiv \lambda \mathbf{F}_S + (1-\lambda)\mathbf{F}_V$, the derivative with respect to elements of Z is: $W_{VS} = \lambda W_S + (1-\lambda) W_V$.
- *Kurtosis and sparseness*: For the function $\mathbf{F} \equiv \lambda \mathbf{F}_K + (1-\lambda)\mathbf{F}_S$, the derivative with respect to elements of Z is: $W_{KS} = \lambda W_K + (1-\lambda) W_S$.

In each of these cases, starting from W, one can compute the actual gradient of F on Stiefel $\mathcal{S}_{n,d}$ or Grassmann $\mathcal{G}_{n,d}$ as follows. First, compute D, the full derivative of F in the space $\mathbb{R}^{n \times d}$, using Eq. 15. Then,

1. In case of a Stiefel manifold, project D onto $T_U(\mathcal{S}_{n,d})$ using Π_1 given in Eq. 11. Call the projected element $G(U)$. This establishes a gradient vector field of F on $\mathcal{S}_{n,d}$.
2. In case of a Grassmann manifold, project D onto $T_U(\mathcal{G}_{n,d})$ using Π_2 given in Eq. 13. Call the projected element $G(U)$. This establishes a gradient vector field of F on $\mathcal{G}_{n,d}$.

3.3 Gradient flows on Stiefel and Grassmann manifolds

For the purpose of this discussion, we focus on the Stiefel manifold $\mathcal{S}_{n,d}$; the case of the Grassmann manifold can be obtained simply by restricting the Stiefel case.

Given a gradient vector field G on a manifold $\mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$, a process $X(t) \in \mathcal{S}_{n,d}$ is called its gradient flow if it satisfies the relation

$$\frac{dX(t)}{dt} = G(X(t)). \quad (20)$$

An important issue here is: Given a smooth vector field G, how to solve Eq. 20 for the flow $X(t)$? On a computer, one can approximate the solution using the following discretization: for a small step size $\delta > 0$, one can generate a discrete-time process $\{X(t\delta), s = 1, 2, \ldots\}$ that will approximate the solution of Eq. 20 as δ gets smaller. As stated in Sect. 3.1, $G(U) \in T_U(\mathcal{S}_{n,d})$ takes the form:

$$G(U) = Q^T \begin{bmatrix} C & B \\ -B^T & 0 \end{bmatrix} J \in \mathbb{R}^{n \times d},$$

where C is skew-symmetric and $Q^T = [U\ V]$. Let the inner skew-symmetric matrix be called $A = \left[\begin{smallmatrix} C & B \\ -B^T & 0 \end{smallmatrix}\right] \in \mathbb{R}^{n \times n}$. It

can be shown that a discrete approximation of $X(t)$ is obtained using the update:

$$X_{(t+1)\delta} = Q^T \exp(\delta A) J, \tag{21}$$

where exp denotes the matrix exponential. Note that both Q and A depend on the current location $X(t\delta)$ although we have not shown this dependence explicitly. A discrete implementation of gradient search involves starting from an initial condition, and iteratively updating using Eq. 21.

Note that A is an $n \times n$ matrix, n being rather large in practice, and the computation of matrix exponential is an order $O(n^3)$ operation. However, the matrix A here has a structure that can be exploited to reduce this computational cost. If the submatrix $C = 0$, then A reduces to a convenient form that can be exponentiated using $O(nd^2)$ computations (see Sect. 3.4 for details of this idea). With a non-zero C, we do not know of any efficient, i.e. order $O(nd^2)$, algorithm to compute $\exp(A)$. Therefore, we decompose the update in two ordered steps: Let $U \equiv X(t\delta)$ be the current state and Q, V be as defined earlier. Let D be the full gradient of F in $\mathbb{R}^{n\times d}$. The two steps are as follows:

1. *Update subspace*: In this step, we flow perpendicular to the sets $[U]$ by setting $C = 0$ in the skew-symmetric matrix A. This update is given by:

$$\tilde{X}_{(t+1)\delta} = Q^T \exp\left(\delta \begin{bmatrix} 0 & B \\ -B^T & 0 \end{bmatrix}\right) J. \tag{22}$$

   Recall that $B = -D^T V \in \mathbb{R}^{d\times(n-d)}$. This exponential is computed efficiently as described in Sect. 3.4.
2. *Update basis*: In this step, we flow parallel to the orbit $[U]$ and we update the basis of the current subspace (spanned by columns of $\tilde{X}_{(t+1)\delta}$), while keeping that subspace fixed. It essentially rotates the current axes in the direction specified by the gradient of F. This update is given by:

$$X_{(t+1)\delta} = \tilde{X}_{(t+1)\delta} \exp(\delta_1 \tilde{C}), \tag{23}$$

   where $\tilde{C}$ is a $d \times d$ skew-symmetric matrix that captures the gradient direction of function $\tilde{F}(O) = F(UO)$ at $O = I_d$, and where $U = \tilde{X}_{(t+1)\delta}$, δ_1 is a gradient step size chosen to update the basis. One can use the Baker-Campbell-Hausdorff formula for relating $\tilde{C}$ to C but we have not explored that relation. Instead, we have decomposed the gradient direction in the larger space $(\mathcal{S}_{n,d})$ into two components: one for updating the subspace in $\mathcal{G}_{n,d}$ and the other for updating the basis in $SO(d)$. Note that the step sizes δ and δ_1 can be different for two different gradient processes. Their values are chosen empirically—not too small as the convergence will be slow and not too large to avoid overshooting the extremum. It can be shown that: $\tilde{C} = (S - S^T)/2$, where $S = \tilde{X}^T_{(t+1)\delta} Y W^T$. **Exponential of $\tilde{C}$ is $O(d^3)$ operation and can be performed fast since d is rather small in our applications.**

To construct a gradient flow on a Grassmann manifold, we simply remove the second step in the update process.

3.4 Computational issues

There is a computational step in the previous section that requires further consideration. In this section we study an efficient strategy for that update step that is central to our gradient search. This idea was presented earlier in papers (Edelman et al. 1998; Srivastava and Liu 2005) but is repeated here for convenience.

Exponential map Given a matrix A of the type: $A = \begin{bmatrix} 0 & B \\ -B^T & 0 \end{bmatrix}$, with $B \in \mathbb{R}^{d\times(n-d)}$, the goal is to compute $\exp(A)J$ efficiently without resorting to full matrix exponential in $n \times n$. This can be computed using the following algorithm.

Algorithm 1

1. Compute singular value decomposition of the matrix B: $B = H_1 \Theta H_2^T$, where Θ is a $d \times (n-d)$ diagonal matrix.
2. Set matrix H_{21} to the first d columns of the matrix H_2.
3. Set matrix Θ_1 to the first d columns of the matrix Θ; $\Theta_1 \in \mathbb{R}^{d\times d}$, diagonal.
4. Compute matrices $\Gamma = \cos(\Theta_1)$ and $\Sigma = \sin(\Theta_1)$. The matrices Γ, $\Sigma \in \mathbb{R}^{d\times d}$ are also diagonal.
5. Compute the matrix $\exp(A)J$ as

$$\exp(A)J = \begin{bmatrix} H_1 \Gamma H_1^T \\ -H_{21} \Sigma H_1^T \end{bmatrix}. \tag{24}$$

4 Optimization algorithm

In each of the applications stated in Sect. 1, the goal of finding an optimal dimension-reduction transformation reduces to solving an optimization problem on a Stiefel or a Grassmann manifold. In this section, we use the tools introduced in the last section to develop a (stochastic) gradient type approach to solving such problems. The goal here is to construct a stochastic gradient process, governed by Markov chain dynamics, in such a way that the process converges to a global optimum in the limit (Geman and Hwang 1987). A useful idea in this context, that has been pursued earlier in Liu et al. (2003) for optimization on Grassmann manifolds and in Srivastava (2000) for MCMC-type random sampling on Grassmann manifolds, is to utilize a Metropolis-Hastings type acceptance-rejection step. Here, the stochastic gradient part provides candidates for updating estimates, but they are

accepted or rejected according to a probability density function that depends upon F. It uses randomly-perturbed versions of the gradient directions to find candidates for updating the chain; these candidates are accepted and rejected according to a certain probability. The search for global solutions, in general, is a hard problem. One commonly used solution is simulated annealing. We adapt our Metropolis algorithm to result in an annealing framework as follows: (i) we introduce a temperature T that is multiplied to the random perturbations of gradient, and (ii) acceptance/rejection function is governed by T. As iterations proceed, T_t is decreased slowly according to a slow cooling schedule, index t indicates the current temperature on step number t. Of course, if we start with $T = 0$, we will get a purely deterministic gradient algorithm.

For $M = \mathcal{S}_{n,d}$ or $\mathcal{G}_{n,d}$, let $F : M \mapsto \mathbb{R}_+$ be a performance function such that we seek an optimal point of F. One can define a vector field on M associated with gradient of F. In general this vector field has to be smooth although we can allow to a finite set of points in M where this field is discontinuous. This is because the probability of reaching this set in practice is zero. The gradient flow is approximated by Eq. 21, and has the limitation that it converges to a local maximum of the function F. Define an orthogonal basis of the set of skew-symmetric matrices using the elements:

$$E_{ij}(k,l) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k = i, l = j; \\ -\frac{1}{\sqrt{2}}, & \text{if } k = j, l = i; \\ 0, & \text{otherwise,} \end{cases} \qquad E_{ij} \in \mathbb{R}^{n \times n} \tag{25}$$

where $1 \le i < j \le n$. If we restrict i, j to $1 \le i < j \le d$, then the set $\{E_{ij}\}$ spans the set of matrices of type $\left[\begin{smallmatrix} C & 0 \\ 0 & 0 \end{smallmatrix}\right]$, and we call these basis matrices E^C_{ij}. If we restrict $1 \le i \le d$, $d+1 \le j \le n$ then the set $\{E_{ij}\}$ spans the set of matrices of type $\left[\begin{smallmatrix} 0 & B \\ -B^T & 0 \end{smallmatrix}\right]$, and we call these basis matrices E^B_{ij}. We can use this notation to add random terms to submatrices C and B separately and still preserve the structure of A. Let A be the skew-symmetric matrix included in the gradient $G(U)$ of a function F on $\mathcal{S}_{n,d}$. A random perturbation of A is given by:

$$\tilde{A} = A + \sqrt{2T_t} \sum_{i=1}^{d} \sum_{j=d+1}^{n} r_{i,j} E^B_{ij} + \sqrt{2T_t} \sum_{i=1}^{d} \sum_{j=1}^{d} r_{i,j} E^C_{ij} \tag{26}$$

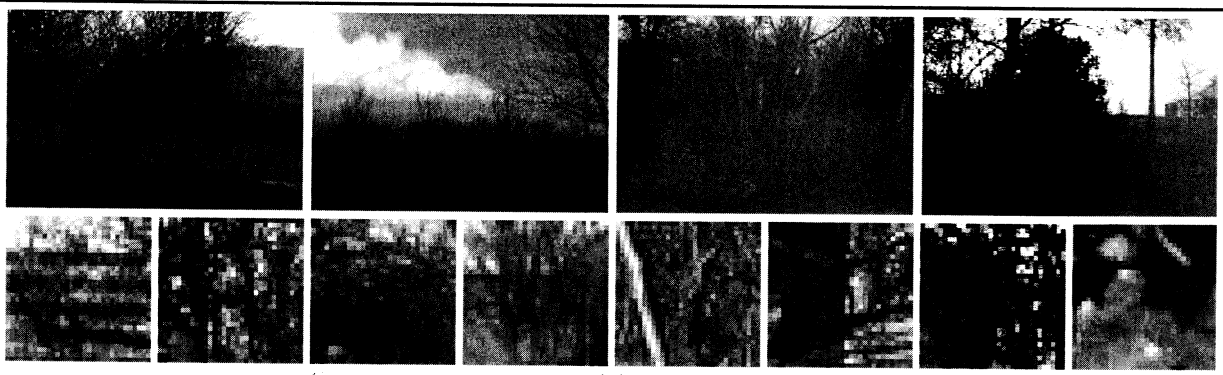
where $r_{i,j}$ are distributed normally with mean zero and variance $\frac{1}{\delta}$. In case of a Grassmann manifold, the last term is zero. T_t is the temperature for simulated annealing and follows a slow cooling schedule during the evolution of the algorithm. If we update the states using $\tilde{A}$, instead of A, we obtain a stochastic perturbation of the gradient update. However, we add a step of acceptance/rejection that decides whether the point suggested by $\tilde{A}$ is accepted or not. The acceptance/rejection function is simply $\min\{e^{\frac{F(U_{new})-F(U_{old})}{T_t}}, 1\}$, where U_{old} is the previous point and U_{new} is the candidate point generated by $\tilde{A}$. This is when F is being maximized, otherwise the signs for the two terms in the exponent are changed. Initially, when T_t is high, the candidates are accepted more frequently while later on only the good candidate points have a high probability of being accepted.

The full algorithm is presented next.

Algorithm For a given objective function F, this algorithm updates the current state $X_t \in \mathcal{G}_{n,d}$ $(\mathcal{S}_{n,d})$ to the state $X_{(t+1)} \in \mathcal{G}_{n,d}$ $(\mathcal{S}_{n,d})$ using the following sequence of steps.

1. Update the space:
   (a) Compute the matrix D according to Eq. 15, where the matrix W is computed using the formula appropriate for the chosen F.
   (b) For $U = X_{t\delta}$, compute the matrix $V = null(U^T)$.
   (c) Compute the elements of the tangent vector according to $B = -D^T V$.
   (d) Generate $r_{i,j} \sim N(0, \frac{1}{\delta})$ and calculate matrix $\hat{B} = B + \sqrt{2T_t} \sum_{i=1}^{d} \sum_{j=d+1}^{n} r_{i,j} E^B_{ij}$.
   (e) Compute $\tilde{X}_{(t+1)\delta} = Q_t^T e^{\delta \hat{A}} J$ using fast computation of $e^{\delta \hat{A}} J$ in Eq. 24, where $\hat{A} = \left[\begin{smallmatrix} 0 & \hat{B} \\ -\hat{B}^T & 0 \end{smallmatrix}\right]$.
2. In case the optimization is on Stiefel manifold, the following steps are added:
   (a) For the current state, $\tilde{X}_{(t+1)\delta}$, compute $S = \tilde{X}^T_{(t+1)\delta} \times YW^T$, and $\tilde{C} = (S - S^T)/2$.
   (b) Generate $r_{i,j} \sim N(0, \frac{1}{\delta})$ and form $\hat{C} = \tilde{C} + \sqrt{2T_t} \times \sum_{i=1}^{d} \sum_{j=1}^{d} r_{i,j} E^C_{ij}$.
   (c) Generate a candidate for the next state according to $\tilde{X}_{(t+1)\delta} \to \tilde{X}_{(t+1)\delta} e^{\delta_1 \hat{C}}$.
3. Set U_{cand} to be $\tilde{X}_{(t+1)\delta}$.
4. Generate $u \sim Uniform(0, 1)$, and calculate $p = \min\{e^{\frac{\Delta F}{T_t}}, 1\}$, where $\Delta F = F(U_{cand}) - F(X_{t\delta})$. If $u < p$, then $X_{(t+1)\delta} = U_{cand}$, and if $u \ge p$ then $X_{(t+1)\delta} = X_{(t)\delta}$.
5. Set $T_{t+1} = \frac{T_t}{\gamma}$ and $t = t + 1$. Go to Step 1.

Here $\gamma > 1$ is the cooling ratio for simulated annealing with a typical value of 1.0025. This algorithm is an example of a larger family of algorithms that perform optimization over manifolds with nonlinear constraints. It is also a particularization of Algorithm A.20 (p. 200) (Robert and Casella 1999), where some asymptotic properties of the resulting Markov chain are discussed. These convergence results rely on sufficiently slow decrease in annealing temperature, a condition that is difficult to establish in a practical situation. Therefore, one relies on experimental results to

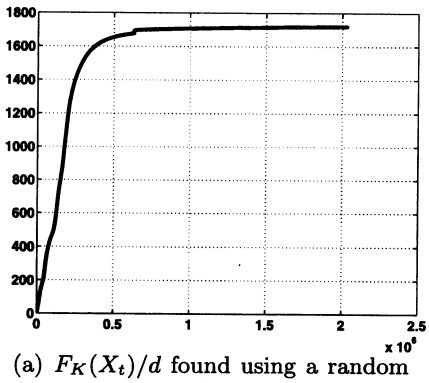


(a)

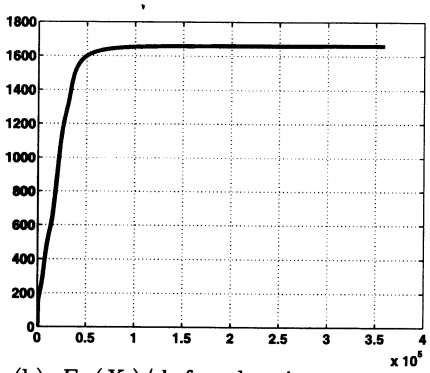
(b)

Fig. 1 (**a**) Examples of original natural images and some down-sampled images, (**b**) face images used in the experiments presented here

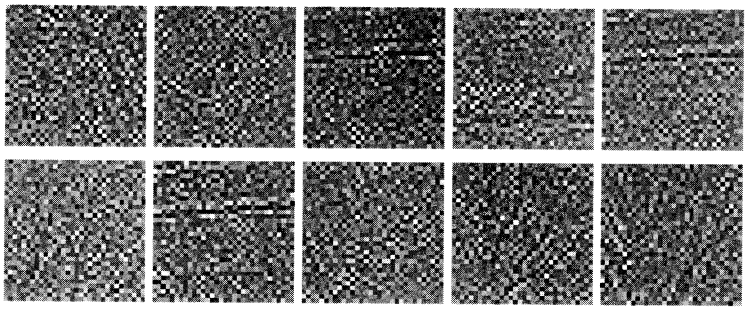
Fig. 2 The results of the experiments with goal function F_K using the stochastic gradient method on $\mathcal{S}_{n,d}$



(a) $F_K(X_t)/d$ found using a random initial condition and stochastic gradient is plotted vs t. The local maximal value of F_K/d is 1719.9.



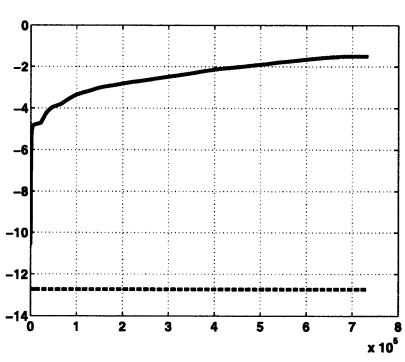
(b) $F_K(X_t)/d$ found using a random initial condition and deterministic gradient is plotted vs t. The local maximal value of F_K/d is 1662.721.



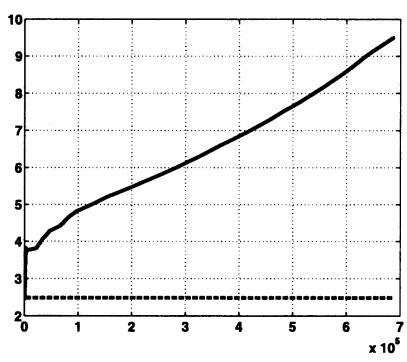
(c) The images of the basis at the point of convergence found using the stochastic search.

evaluate algorithmic performance. Experimental results presented in the next section point to the success of this algorithm in solving some of the problems targeted in this paper.

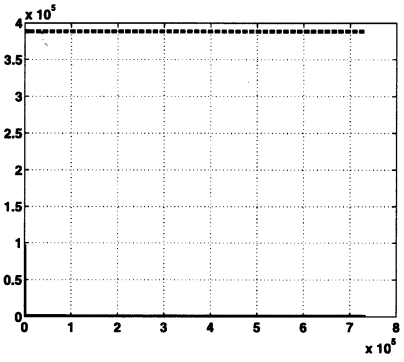
Similar to any other numerical procedure, the performance of Algorithm is ultimately tied to the choice of parameters such as δ and the cooling schedule. It must be noted that this



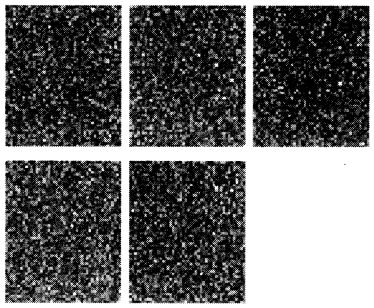
(a) The goal function $F_S(X_t)/d$ is plotted vs t. The local maximal value of F_S/d is -1.49.



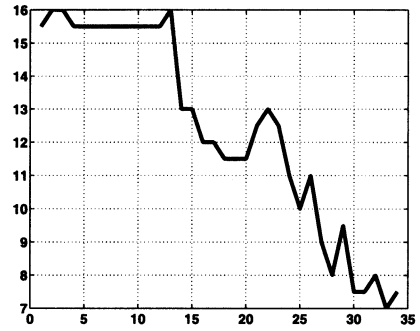
(b) $F_K(X_t)/d$ is plotted vs t.



(c) $F_V(X_t)/d$ is plotted vs t.



(d) The images of of the basis vectors at the point of convergence.



(e) The evolution of the recognition rate.

Fig. 3 The results of the experiments with goal function F_S using the stochastic gradient method on $\mathcal{S}_{n,d}$

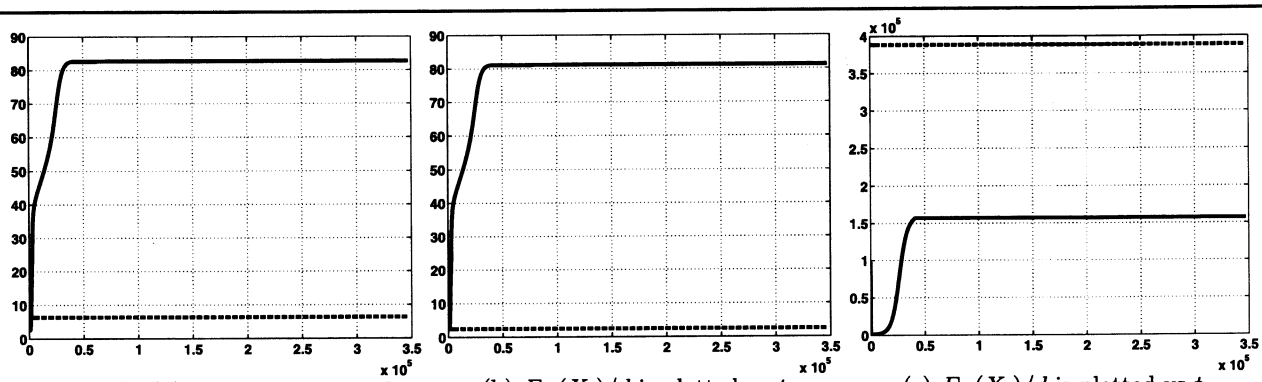
dependence on parameters may render it ineffective in some practical situations.

5 Experimental results

For the experimental results presented in this section, we have used two publicly available databases.

- The first one is the database of natural images obtained from the home page of Hans van Hateren's Lab (url is hlab.phys.rug.nl/imlib/index.html). This contains images of natural scenes: trees, roads, buildings, and fields, and the original images are much larger in size; we have down-sampled them for our experiments. We extract patches of size 32×32 from these larger images to form observations of $\mathbf{y}$. Some examples of these original images and the down-sampled images are shown in Fig. 1(a), top and bottom rows, respectively. In this setting, the dimension of the observation space is $n = 32 \times 32 = 1024$, and we use $k = 2400$ of such images in our experiments.
- The second type is "The ORL Database of Faces", a database of face images obtained from the site http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. There are ten different images of each of 40 distinct subjects; each image has size 112×92 and is taken under varying lighting conditions, pose, scale, facial expressions and the presence/absence of glasses. The reason for selecting face images is the possibility of studying the problem of human recognition. In addition to optimize different criteria mentioned earlier, we can also monitor the recognition performance under different projections. This data set was downsized to the dimensionality 56×46, resulting in $n = 2576$, and we divided images of this face database into two disjoint sets. One half (200) of these images were used as a training set, so $k = 200$, and the remaining half were used as a test set. The training set contains images of 40 people with 5 facial expressions, and the test set consists of images of the same 40 people with 5 different facial expressions. Some examples of these images are given in Fig. 1(b). We used the nearest neighbor classifier for recognizing (classifying) test images although any such classifier can be used here.

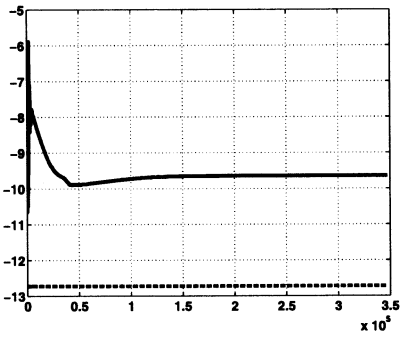
Throughout these experiments, the choice of n and d is determined according to computational convenience, rather than a precise guiding principle.



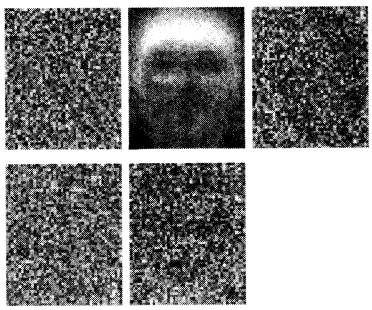
(a) $F_{KV}(X_t)/d$ is plotted vs t. The local maximal value of F_{KV}/d is 82.7197

(b) $F_K(X_t)/d$ is plotted vs t.

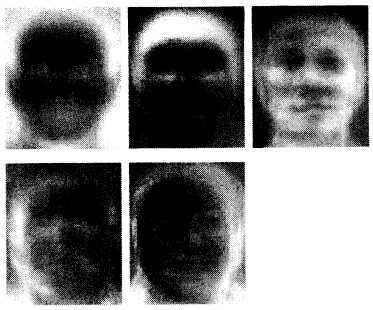
(c) $F_V(X_t)/d$ is plotted vs t.



(d) $F_S(X_t)/d$ is plotted vs t.



(e) The images of the basis vectors at the point of convergence.

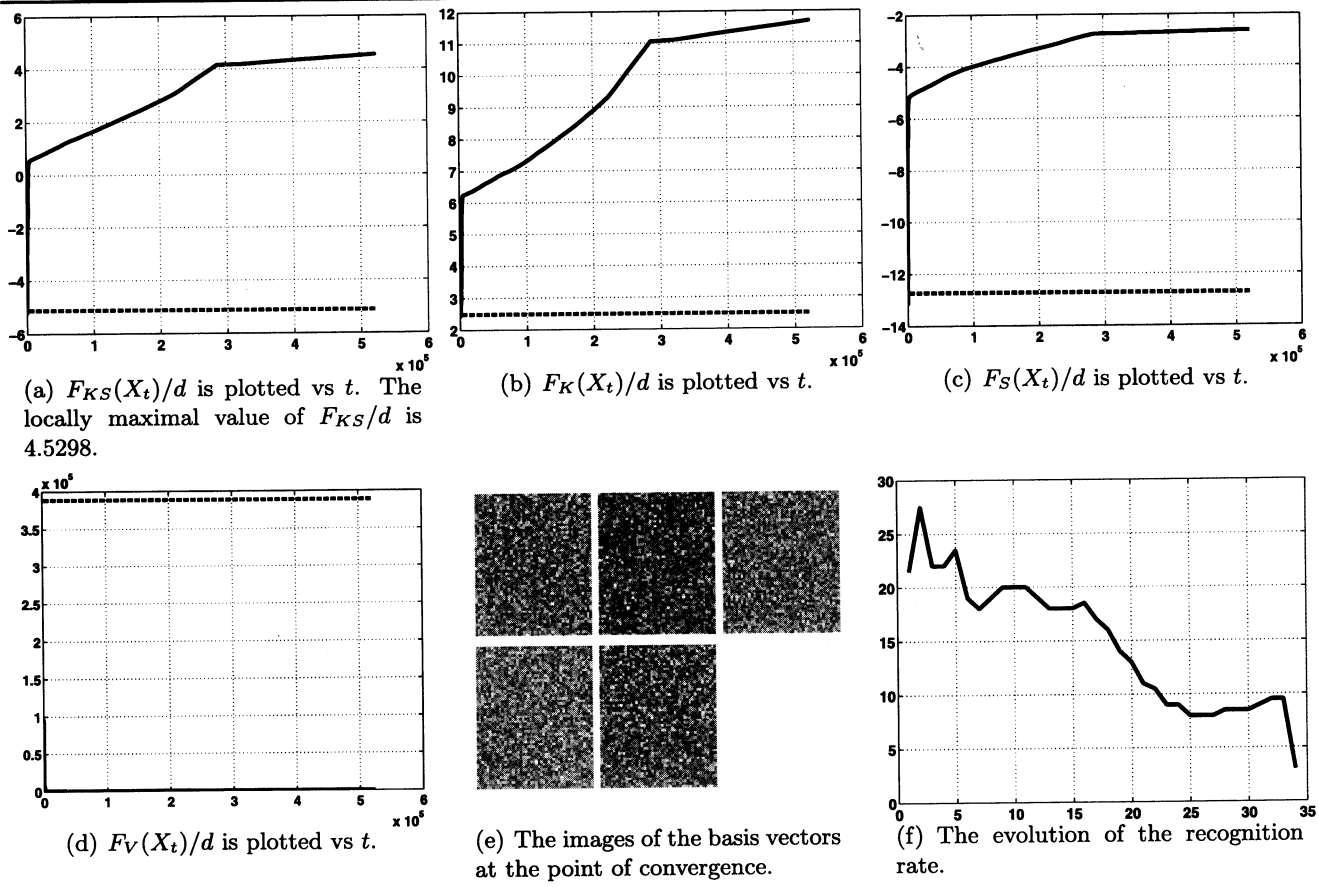


(f) The images of the basis vectors found using the PCA method.

Fig. 4 The results of the experiments with the goal function F_{KV}, $\lambda = 1 - 10^{-5}$

Maximizing kurtosis To study maximization of F_K, given in Eq. 4, we used a set of natural images, with $d = 10$, $n = 1024$, and $k = 2400$. In this experiment, we used the stochastic gradient method to maximize F_K on $\mathcal{S}_{n,d}$ with three different initial points: (i) random initial condition, (ii) initial condition generated by PCA method, and (iii) initial condition generated by ICA method. The ICA algorithm used here is the FastICA which could be downloaded from the site of the Department of Computer Science and Engineering at Helsinki University of Technology (www.cis.hut.fi/projects/ica/fastica/). The initial temperature for simulated annealing was $T = 10$ for the random initial condition and the PCA initial condition, while it was $T = 100$ for the experiments with the ICA initial condition. The evolution of the goal function F_K looks similar for all three cases, it increases and then stabilizes. As an example, we show the evolution of the function F_K for the random initial condition in Fig. 2(a). As a comparison, we also present the evolution of the same function using a random initial condition but a deterministic gradient search. The stochastic search results in a slightly improvement in the result and, hence, we use that method in the rest of the paper. The images formed by re-arranging individual columns of U at the point of (stochastic search) convergence are shown in Fig. 2(c). A few conclusions can be drawn from these results. Firstly, the algorithm finds a stable maximum for each of the three initial conditions, although the convergence seems to be more local than global. Although the search performance improves, over a deterministic gradient approach, due to the presence of a stochastic components, the convergence to a global solution is far from guaranteed. Secondly, in terms of the resulting basis vectors, their images seem to contain edge-like structures at different angles that may represent frequently occurring boundaries in the original images.

Maximizing sparseness The experiment on maximizing F_S, as given in Eq. 5, was conducted using the Face Image database, with $n = 2576$, $d = 5$, $k = 200$. Again, the search was conducted using the stochastic gradient method on a Stiefel manifold with random initial condition. To show the results, first we plot the evolution of the sum of sparseness F_S versus the iteration index in Fig. 3(a). Dashed lines in Fig. 3 (and all Figures now on) show the values achieved by the PCA basis. Additionally, we monitor the evolution of F_K (Fig. 3(b)) and F_V (Fig. 3(c)) for the process that is maximizing F_S. We can see from the resulting plots that F_S increases at first and then stabilizes; the resulting value



(a) $F_{KS}(X_t)/d$ is plotted vs t. The locally maximal value of F_{KS}/d is 4.5298.

(b) $F_K(X_t)/d$ is plotted vs t.

(c) $F_S(X_t)/d$ is plotted vs t.

(d) $F_V(X_t)/d$ is plotted vs t.

(e) The images of the basis vectors at the point of convergence.

(f) The evolution of the recognition rate.

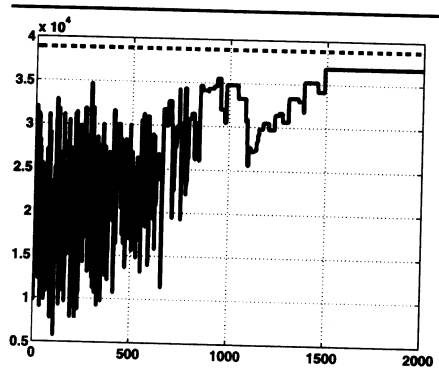
Fig. 5 The results of the experiments with the goal function F_{KS}, $\lambda = 0.5$

is much higher than that achieved by a PCA basis. It is well known that the PCA projections do not provide optimal sparsity in the projected data. It is also interesting to note the increase in F_K even though it is not a part of the optimization process. It shows that two criteria F_K and F_S are closely related. Since this experiment involved face database, we also studied the changes in recognition performance generated using a nearest neighbor classifier. It can be seen in Fig. 3(e) that the recognition performance goes down as the sparsity increases. This implies that image representations that result in sparse coefficients are generally not good for use in face recognition and classifications. The images of the basis vectors at the point of convergence are shown in Fig. 3(d).

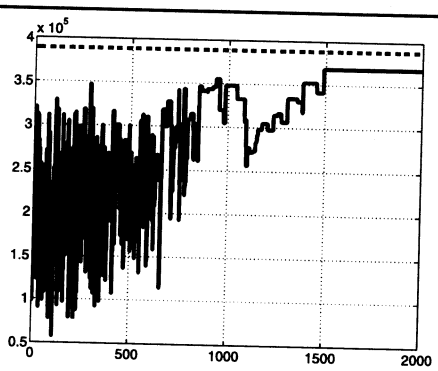
Maximizing kurtosis and variance jointly We used the facial data for these experiments using the stochastic gradient search on $\mathcal{S}_{n,d}$. The goal function used here is F_{KV} is for $\lambda = 1 - 10^{-5}$. The initial conditions were chosen randomly. Figure 4(a) shows the evolution of the goal function F_{KV}. To study the evolution of other quantities for this gradient search, we plot the functions F_K in Fig. 4(b), F_V in Fig. 4(c), F_S in Fig. 4(d), and the images of the basis vectors at the point of convergence in Fig. 4(e). Since F_{KV} is a linear combination of F_K and F_V, it is reasonable to expect an increase in both these functions during the maximization of F_{KV}. Also, as mentioned earlier, an increase in variance tends to decrease the level of sparseness associated with a representation. This is also reflected here in the fact that F_S decreases. In terms of comparisons with the PCA basis, the solution obtained by the optimization process provides higher kurtosis and higher sparseness, but smaller variance.

All the vectors of the basis at the point of convergence look similar, but the second vector looks similar to the images of the basis vectors found by the PCA method. The images of the PCA vectors are given in Fig. 4(f). This similarity appears because F_V is a part of the goal function, it is maximized together with kurtosis, and PCA maximizes F_V.

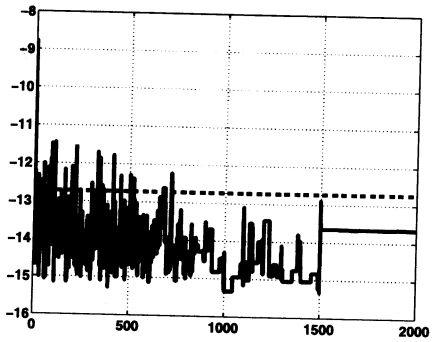
Maximizing kurtosis and sparseness jointly In this case, we form F_{KS} with $\lambda = 0.5$. The evolution of the goal function F_{KS} is shown in Fig. 5(a) and it shows a steady increase in F_{KS} as the algorithm evolves. The next two plots in this figure show the evolution of the functions F_K (Fig. 5(b)) and F_S (Fig. 5(c)). Since they both contribute in the definition of F_{KS}, we see an expected increase in their values



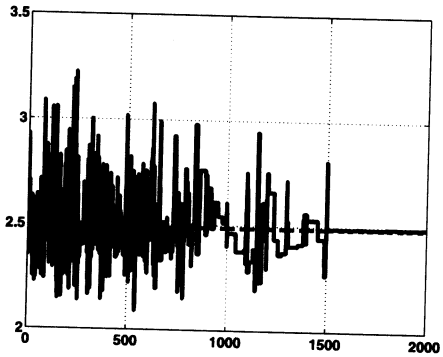
(a) $F_{SV}(X_t)/d$ is plotted vs t. The locally maximal value of F_{SV}/d is 3.6840×10^4.



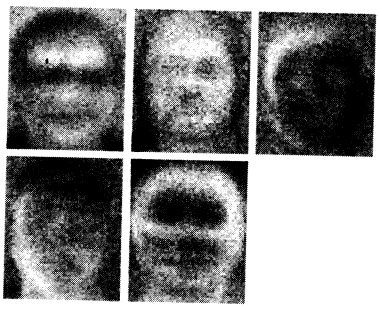
(b) $F_V(X_t)/d$ is plotted vs t.



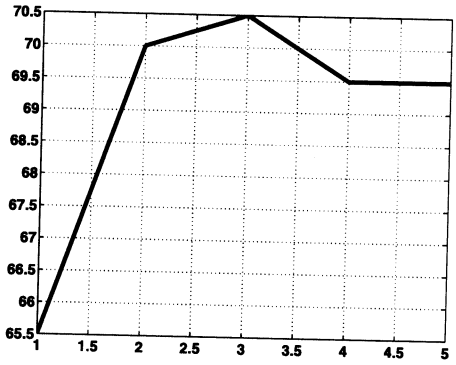
(c) $F_S(X_t)/d$ is plotted vs t.



(d) $F_K(X_t)/d$ is plotted vs t.



(e) The images of the basis vectors at the point of convergence.



(f) The evolution of the recognition rate.

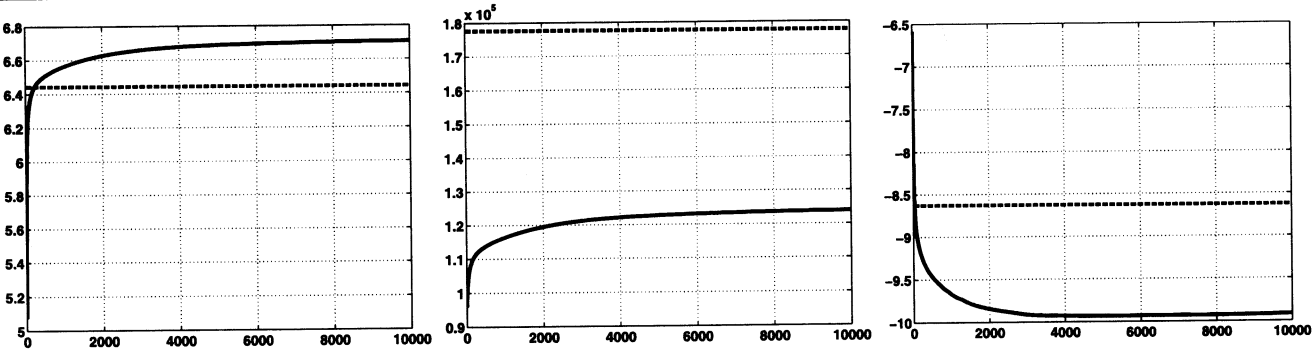
Fig. 6 The results of the experiments with the goal function F_{SV}, $\lambda = 0.9$

as the algorithm proceeds. The evolution of F_V is shown in Fig. 5(d) and it shows a sharp decrease in F_V right at the start of the algorithm. This is expected as both the kurtosis and the sparseness typically steer the algorithm towards a decrease in the variance. The images of the basis vectors at the point of convergence are shown in Fig. 5(e), while the rate of recognition is shown in Fig. 5(f).

Maximizing sparseness and variance jointly Here we describe the results for $\lambda = 0.9$. Figure 6(a) shows the evolution of the goal function F_{SV}, while Fig. 6(b) plots the evolution of F_V, Fig. 6(c) plots F_S, and Fig. 6(d) plots the change in F_K. In this experiment we used a higher initial temperature T and, therefore, we see a larger fluctuation in the process initially. As seen in these results, neither F_V nor F_{SV} reached the levels found by the PCA method. This is obvious, because the PCA method produces the maximum value of F_V. The sparseness term bounces for some time and then stabilizes. The images of the basis vectors at the end of the optimization are given in Fig. 6(e). The vectors of the basis at the point of convergence look like the images, which we found by the PCA method, but grainy. For a comparison, the PCA images are given in Fig. 4(f). The evolution of the recognition rate is given in Fig. 6(f). It increases slightly from 65.5% to 69.5%.

Entropy The data set used for this experiment is the set of natural images. We used 1000 images, so that $n = 1024$, $d = 10$, and $k = 1000$. Here we present results from a deterministic maximization of entropy on $\mathcal{G}_{n,d}$ with random initial condition. Figure 7(a) shows the evolution of the goal function H. One can see that the stabilized value of H is higher than that achieved by a PCA basis. The PCA method maximizes variance, which is the measure of the variability; and the entropy is the measure of uncertainty, so they are positively related. Figure 7(b) shows the variance F_V and Fig. 7(c) shows the sparseness F_S.

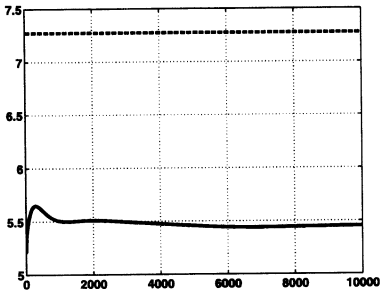
The H is rapidly increasing at the very beginning, then the increase becomes slower, and the sum of the entropy stabilizes. The sum of the sparseness F_S decreases at a high rate at the beginning, then rate becomes lower and value stabilizes. The graphs of these two functions look as a mirror reflection of each other with horizontal line as axis of symmetry. The sum of the sparseness F_S stabilizes at a level which is much lower than that for the PCA method. The sum of the variances increases as the function H; as ex-



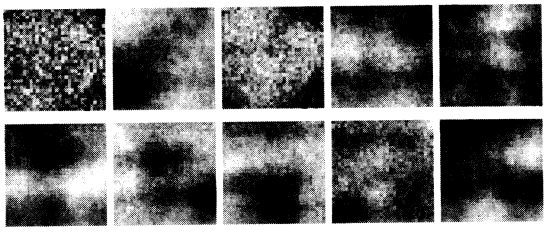
(a) $H(X_t)/d$ is plotted vs t. The locally maximal value of H/d is 6.7061.

(b) $F_V(X_t)/d$ is plotted vs t.

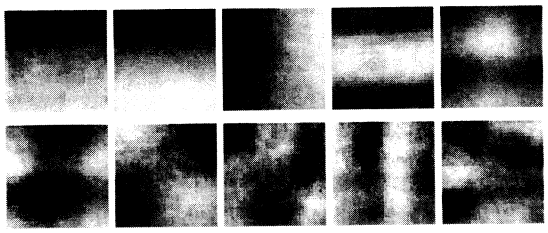
(c) $F_S(X_t)/d$ is plotted vs t.



(d) $F_K(X_t)/d$ is plotted vs t.



(e) The images of the basis vectors at the point of convergence.



(f) The images of the basis vectors found by the PCA method.

Fig. 7 The results of the experiments with the goal function H

pected they change in the same direction. The evolution of the sum of the kurtosis is given in Fig. 7(d). Figure 7(e) shows the images of the basis vectors at the point of convergence and Fig. 7(f) shows the images of the components of a PCA basis. One can see that these images look similar: they have geometrical structures on them, which are lighter spots. In the case of the goal function H images are grainy, especially images of the first, third, and ninth vectors.

6 Summary

We have presented the problem of dimension reduction of the data as a problem of the choice of a linear projection. The basic idea was to define a criterion which might include combinations of the properties of the data such as sparseness, variance, kurtosis, and independence and find a linear projection or basis such that the projected data will achieve the optimal value of the given criterion. We introduced the problem of dimension reduction as an optimization problem on the Stiefel or Grassmann manifold and utilized differential geometry of these manifolds to construct a stochastic search to solve this problem. This search used a multi-flow approach. An algorithm for finding a local optimal point was presented. We illustrated the algorithm using two different collections of images—one of natural images and other of facial images.

Acknowledgements We thank the producers of the ORL Face database and the van Hateren natural image database for making them public. This research was supported in part by the grants ARO W911NF-04-01-0268 and AFOSR FA9550-06-1-0324.

References

Bell, A.J., Sejnowski, T.J.: An information maximization approach to blind separation and blind deconvolution. Neural Comput. **7**, 1129–1159 (1995)

Comon, P.: Independent component analysis, a new concept? Signal Process. Special issue on higher-order statistics **36**(3), 287–314 (1994)

Cook, D.: Testing predictor contributions in sufficient dimension reduction. Ann. Stat. **32**(3), 1062–1092 (2004)

Cook, D., Li, B.: Dimension reduction for conditional mean in regression. Ann. Stat. **30**(2), 455–474 (2002)

Donoho, D.L., Flesia, A.G.: Can recent innovations in harmonic analysis "explain" key findings in natural image statistics? Netw. Comput. Neural Syst. **12**(3), 371–393 (2001)

Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)

Field, D.J.: What is the goal of sensory coding? Neural Comput. **6**(4), 559–601 (1994)

Fiori, S.: A minor subspace algorithm based on neural Stiefel dynamics. Int. J. Neural Syst. **19**(5), 339–350 (2002a)

Fiori, S.: A theory for learning based on rigid bodies dynamics. IEEE Trans. Neural Netw. **13**(3), 521–531 (2002b)

Geman, S., Hwang, C.-R.: Diffusions for global optimization. SIAM J. Control Optim. **24**(5), 1031–1043 (1987)

Golub, G.H., Van Loan, C.: Matrix Computations. The John Hopkins University Press, Baltimore (1989)

Hyvärinen, A.: Fast and robust fixed-point algorithm for independent component analysis. IEEE Trans. Neural Netw. **10**(3), 626–634 (1999)

Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, New York (2001)

Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall, New York (2001)

Liu, X., Srivastava, A., Gallivan, K.A.: Optimal linear representations of images for object recognition. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 229–234 (2003)

Mallat, S.G.: Theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989)

Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**, 607–609 (1996a)

Olshausen, B.A., Field, D.J.: Natural image statistics and efficient coding. Netw. Comput. Neural Syst. **7**, 333–339 (1996b)

Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics. Springer, New York (1999)

Srivastava, A.: A Bayesian approach to geometric subspace estimation. IEEE Trans. Signal Process. **48**(5), 1390–1400 (2000)

Srivastava, A., Lee, A.B., Simoncelli, E.P., Zhu, S.-C.: On advances in statistical modeling of natural images. J. Math. Imaging Vis. **18**, 17–33 (2003)

Srivastava, A., Liu, X.: Tools for application-driven linear dimension reduction. J. Neurocomput. **67**, 136–160 (2005)