Sparsity Preserving Projections with Applications to Face Recognition

Lishan Qiao[1,2], Songcan Chen[1,*] Xiaoyang Tan[1]

[1] *Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, 210016, Nanjing, P.R. China*
[2] *Department of Mathematics Science, Liaocheng University, 252000, Liaocheng, P.R. China*

Abstract: Dimensionality reduction methods (DRs) have commonly been used as a principled way to understand the high-dimensional data such as face images. In this paper, we propose a new unsupervised DR method called Sparsity Preserving Projections (SPP). Unlike many existing techniques such as Local Preserving Projection (LPP) and Neighborhood Preserving Embedding (NPE), where local neighborhood information is preserved during the DR procedure, SPP aims to preserve the sparse reconstructive relationship of the data, which is achieved by minimizing a L1 regularization-related objective function. The obtained projections are invariant to rotations, rescalings and translations of the data, and more importantly, they ***contain natural discriminating information*** even if no class labels are provided. Moreover, SPP chooses its neighborhood automatically and hence can be more conveniently used in practice compared to LPP and NPE. The feasibility and effectiveness of the proposed method is verified on three popular face databases (Yale, AR and Extended Yale B) with promising results.

Key words: Dimensionality reduction; sparse representation; compressive sensing; face recognition.

1 Introduction

In many application domains, such as appearance-based object recognition, information retrieval and text categorization, the data are usually provided in high-dimensional form. Dimensionality reduction is an effective approach to deal with such data, due to its potential to mitigate the so-called "curse of dimensionality" [1] and to improve the computational efficiency. Up to now, researchers have developed a variety of dimensionality reduction methods (DRs) under supervised, unsupervised and semi-supervised scenarios. The supervised DRs include typically Linear Discriminant Analysis(LDA)[2], Marginal Fisher Analysis(MFA)[3] and Maximum Margin Criterion(MMC)[4,5], etc.; the unsupervised DRs include Principal Component Analysis(PCA)[6], Locality Preserving Projections(LPP)[7], etc.; and the semi-supervised DRs include Semi-supervised Dimensionality Reduction(SSDR)[8], Semi-supervised Discriminant Analysis(SDA) [9], just to name a few. In this paper, we only focus on unsupervised scenario, mainly for justifying its effectiveness and feasibility as a new DR method for classification, though our algorithm presented here can be easily and straightforwardly extended to include supervised information (e.g. class-label information and must/cannot-link pair-wise constraints) following the existing semi-supervised dimensionality reduction framework [8,10].

In the unsupervised DRs, PCA seems to be the most popular one. It is very simple and effective to some practical applications throughout science and engineering. However, PCA may fail to discover essential data structures that are nonlinear. Although the kernel-based techniques such as KPCA [11] can implicitly deal

[*] Corresponding author: Tel: +86-25-84896481 Ext. 12106; Fax: +86-25-84498069; E-mail: s.chen@nuaa.edu.cn (S. Chen)

with nonlinear DR problems, most of them do not explicitly treat the manifold structure of the data. Furthermore, how to select kernel and assign optimal kernel parameter is generally difficult and unsolved fully in many practical applications.

Another technique for nonlinear DR is manifold learning. In the past decade years, a variety of manifold-based techniques such as Isomap [12], LLE [13], Laplacian Eigenmaps [14] and their variations [15] have been developed to explicitly discover the nonlinear manifold structure concealed in the data. However, some desirable virtues the traditional PCA possesses are not inherited. For example, 1) how to evaluate the map for unseen test samples is not as natural as PCA, and thus special tricks [16] are required to handle the "out-of-sample" problem; 2) a recent research [15] has shown that nonlinear techniques perform well on some artificial data sets, but do not necessarily outperform the traditional PCA for real-world tasks yet; 3) it is generally difficult to select suitable values for the hyper-parameters (e.g., the neighborhood size) in such models. One effective approach to overcome the above limitations is approximating the nonlinear DRs using linear ones. For example, LPP [7] is a linearized version of Laplacian Eigenmaps; Neighborhood Preserving Embedding (NPE) [18] and Locally Linear Embedded Eigenspace Analysis (LEA) [19] are two linearized counterparts of LLE; Isometric Projection (IsoProjection) [20] can be seen as a linearized Isomap. Most of these linearized versions can generally outperform PCA on real-world data due to the simplicity (linearity) of these models and their capability to preserve spatial consistency between the input space and the output space. In addition, the "out-of-sample" problem is usually addressed as well in these methods. However, it is still unclear how to select the neighborhood size and how to assign optimal values for other hyper-parameters for them.

In this paper, motivated by the recent development of sparse representation [21, 22, 23, 24], we propose a simple dimensionality reduction method called Sparsity Preserving Projections (SPP). Specifically, in the proposed algorithm, an "adjacent" weight matrix of the data set is firstly constructed based on a modified sparse representation framework, and then the low-dimensional embedding of the data is evaluated to best preserve such weight matrix. Although supervised information is not needed, SPP tends to find the discriminative mapping since the sparsest representation has natural discriminating power: taking face images into account, the most compact expression of a certain face image is generally given by the face images from the same class [21]. We now enumerate several characteristics of our presented algorithm as follows:

1） SPP shares some advantages of both LPP and many other linear DRs. For example, it is linear and defined everywhere, thus the "out-of-sample" problem is naturally solved. In addition, the weight matrix is kept sparse like in most locality preserving algorithms, which is beneficial to computational tractability.

2） SPP does not have to encounter model parameters such as the neighborhood size and heat kernel width incurred in LPP and NPE, etc, which are generally difficult to set in practice. Although cross-validation technique [10,40] can be used in these cases, it is very time-consuming and tends to waste the limited training data. In contrast, SPP does not need to deal with such parameters, which makes it very simple to use in practice.

3） Although SPP belongs to global methods in nature, it owns some local properties due to the sparse representation procedure. In section 4, we will show that SPP has some factual connection with

several popular locality preserving algorithms under certain conditions.

4） This technique proposed here can be easily extended to supervised and semi-supervised scenarios based on the existing dimensionality reduction framework [8,10,25].

The rest of the paper is organized as follows: Section 2 reviews PCA, LPP and NPE, three popular linear DRs. The Sparsity Preserving Projections (SPP) algorithm is introduced in Section 3. In Section 4, we compare SPP with some related works. The experimental results are presented in Section 5. Finally, we provide some concluding remarks and future work in Section 6.

2 Linear Unsupervised Dimensionality Reduction

In this paper, we mainly focus on linear approaches though our SPP can be easily kernelized as a nonlinear one [10, 25]. In fact, up to now, linear techniques are still important research subject in pattern recognition and machine learning mainly due to their simplicity, mathematical tractability, efficiency and effectiveness for many real-world problems such as face recognition. In the numerous linear DRs, PCA, LPP and NPE are three popular ones. In face recognition, they are known as Eigenface [6], Laplacianface [26] and NPEface [18] respectively.

2.1 Principal Component Analysis (PCA)

PCA seeks a low-dimensional representation of the data to retain as much of the variance in the data as possible. Given a set of data points $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in R^m$ is an *m*-dimensional column vector, we expect to get their low-dimensional images $\{y_i\}_{i=1}^n$ by projecting each $\mathbf{x}_i$ onto the direction vector $\mathbf{w} \in R^m$. The objective function of PCA is defined as follows:

$$\max_{\|\mathbf{w}\|=1} \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{1}$$

where $y_i = \mathbf{w}^T \mathbf{x}_i$, and $\bar{y}$ is the mean of $\{y_i\}_{i=1}^n$. Eq.(1) can be rewritten as

$$\max_{\|\mathbf{w}\|=1} \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \tag{2}$$

where $\boldsymbol{\Sigma}$ is the sample covariance matrix. The eigenvectors of $\boldsymbol{\Sigma}$ corresponding to the largest *d* eigenvalues span the optimal subspace of PCA. In face recognition, $\mathbf{x}_i$ represents a face image, and the eigenvectors are so-called ***Eigenfaces***.

2.2 Locality Preserving Projections (LPP)

While PCA aims to preserve the global structure of the data, LPP aims to preserve the local (i.e., neighborhood) structure of the data. Intuitively, LPP may keep more discriminating information than PCA, assuming that the samples from the same class are likely close to each other in the input space. With the same mathematical notations as in PCA, the objective function of LPP is defined as follows:

$$\min_{\mathbf{w}} \sum_{i,j} (y_i - y_j)^2 p_{ij} \tag{3}$$

where $y_i = \mathbf{w}^T \mathbf{x}_i$, $i = 1,2,\cdots,n$, and $\mathbf{P} = (p_{ij})_{n \times n}$ is a similarity matrix defined as follows:

$$p_{ij} = \begin{cases} \exp(-\| \mathbf{x}_i - \mathbf{x}_j \|^2 / t), & \textit{if } \mathbf{x}_i \textit{ is among kNN of } \mathbf{x}_j \\ & \textit{or if } \mathbf{x}_j \textit{ is among kNN of } \mathbf{x}_i \\ 0 & , \quad \textit{otherwise} \end{cases}$$

Minimizing (3) aims to encourage that if two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other in the input space, then so should be in the corresponding output space. With simple formulation, the objective function is equivalent to minimizing

$$\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 p_{ij} = \frac{1}{2}\sum_{i,j}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 p_{ij} = \mathbf{w}^T\mathbf{X}(\mathbf{D}-\mathbf{P})\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w} \tag{4}$$

where $\mathbf{D}$ is a diagonal matrix with its entries being the row (or column since $\mathbf{P}$ is symmetric) sums of $\mathbf{P}$, i.e., $d_{ii} = \sum_j p_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{P}$ is the Laplacian matrix. By imposing a constraint $\mathbf{w}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w} = 1$, LPP reduces to

$$\min_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w}} \tag{5}$$

The optimal **w** is given by the minimum eigenvalue solution to the following generalized eigenvalue problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w} \tag{6}$$

2.3 Neighborhood Preserving Embedding (NPE)

Similar to LPP, NPE also aims at preserving the local neighborhood structure of the data. However, NPE evaluates the affinity weight matrix using local least squares approximation instead of defining it directly as in LPP. The local approximation error in NPE is measured by minimizing the cost function [18]:

$$\phi(\mathbf{N}) = \sum_i \| \mathbf{x}_i - \sum_j \mathbf{N}_{ij}\mathbf{x}_j \|^2 \tag{7}$$

where $\mathbf{x}_j$'s are *k* neighbors of $\mathbf{x}_i$. A reasonable criterion for choosing a "good" projection is minimizing the cost function [18]:

$$\Phi(\mathbf{w}) = \sum_i (\mathbf{w}^T\mathbf{x}_i - \sum_j \tilde{\mathbf{N}}_{ij}\mathbf{w}^T\mathbf{x}_j)^2 \tag{8}$$

where $\tilde{\mathbf{N}}_{ij}$ is the optimal solution of Eq.(7). By removing an arbitrary scaling factor, minimizing Eq.(8) leads to:

$$\min_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w}} \tag{9}$$

where $\mathbf{M} = (\mathbf{I}-\mathbf{N})^T(\mathbf{I}-\mathbf{N})$. The optimization problem boils down to a generalized eigenvalue problem as in LPP.

3 Sparsity Preserving Projections

Recently some researchers have shown that most of the existing DRs can be explained from the kernel view [27] and unified under a graph framework [25], where constructing a specific graph and its affinity

weight matrix plays a key role. Although, according to the celebrated "No Free Lunch" theorem [29], there is no clear evidence that any of affinity weight matrix is always superior to the others, the weight matrices in most locality-based DRs such as LPP and NPE have a common characteristic: ***sparsity*** [17]. The sparsity is an important way to encode the domain knowledge thus helpful to improve the generalization capability of the model. Motivated by this, here we present a new method to design the weight matrix straightforwardly based on sparse representation theory [35,36,41], through which the sparsity can be optimally and naturally derived. For completeness, we briefly review the concept of Sparse Representation (SR) before going into the details of our method.

3.1 Sparse representation

Sparse representation is initially proposed as an extension to traditional signal representations such as Fourier representation and wavelet representation. In the past few years, SR has been successfully applied to solve many practical problems in signal processing, statistics, and pattern recognition. For example, in signal and image processing fields, SR is used to signal compression and coding [30], image denoising [31], image super-resolution[32], etc.; in statistics, SR is an effective tool to variable selection, and keeps close relation with the popular LASSO[33,34]; in machine learning and pattern recognition communities, SR is used to objection detection and classification tasks[22,23]. In the emerging field of Compressive Sensing [35,36], as a very attractive theory challenging Shannon-Nyquist sampling theorem, SR seeks to recover the signal from the compressed measures in a most economical way. Especially, recent researches [21,37] showed that classifier based on SR is exceptionally effective and achieves by far the best recognition rate on some face databases.

SR has compact mathematical expression. Given a signal (or an image with vector pattern) $\mathbf{x} \in R^m$, and a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in R^{m \times n}$ containing the elements of an overcomplete dictionary [28] in its columns, the goal of SR is to represent $\mathbf{x}$ using as few entries of $\mathbf{X}$ as possible. This can be formally expressed as follows:

$$\begin{aligned} &\min_{\mathbf{s}} \| \mathbf{s} \|_0 \\ &s.t. \quad \mathbf{x} = \mathbf{X}\mathbf{s} \end{aligned} \tag{10}$$

where $\mathbf{s} \in R^n$ is the coefficient vector, and $\| \mathbf{s} \|_0$ is the pseudo-ℓ_0 norm which is equal to the number of non-zero components in $\mathbf{s}$. Unfortunately, this criterion is not convex, and finding the sparsest solution of Eq.(10) is NP-hard. This difficulty can be bypassed by convexizing the problem and solving

$$\begin{aligned} &\min_{\mathbf{s}} \| \mathbf{s} \|_1 \\ &s.t. \quad \mathbf{x} = \mathbf{X}\mathbf{s} \end{aligned} \tag{11}$$

where ℓ_1 is used instead of ℓ_0. It can be shown that if the solution $\mathbf{s}^0$ sought is sparse enough, the solution of ℓ_0 minimization problem is equal to the solution of ℓ_1 minimization problem [35,36]. Fig. 1 shows that the sparse solution can be found by solving a ℓ_1 minimization problem but may not by other traditional strategies like ℓ_2 minimization.

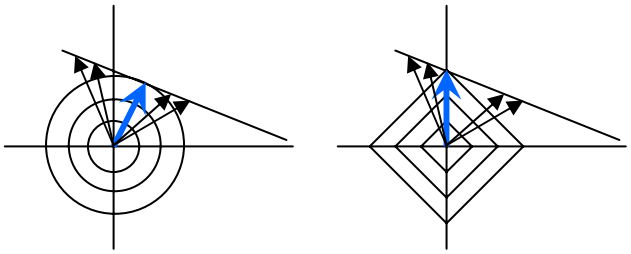


Fig. 1 A 2D example of optimization under (left) ℓ_2 minimization and (right) ℓ_1 minimization. The skew line denotes the feasible solution space, i.e., $\{\mathbf{s} \in R^2 \mid \mathbf{x} = \mathbf{X}\mathbf{s}\}$ under 2D case. The two bold arrow lines denote the optimal solutions of ℓ_2 and ℓ_1 minimization problem respectively.

In fact, suboptimal solutions can be found by a variety of approaches such as greedy [38] algorithms and Bayesian [39] strategies. However, the equivalence of the ℓ_0 and ℓ_1 problem has been studied deeply from a mathematical perspective, which makes the ℓ_1 approximate strategy more reliable than others for practical applications. In general, the ℓ_1 minimization problem can be solved by standard linear programming [41].

In many practical problems, the signal $\mathbf{x}$ is generally noisy, thus the constraint $\mathbf{x} = \mathbf{X}\mathbf{s}$ in Eq.(11) does not always hold. According to [21], at least two robust extensions can be used to handle this problem: 1) relax the constraint to $\| \mathbf{x} - \mathbf{X}\mathbf{s} \| < \varepsilon$, where ε can be seen as an error tolerance; 2) simply replace $\mathbf{X}$ with $[\mathbf{X}, \mathbf{I}]$, where $\mathbf{I}$ is an *m*-order identity matrix. Both of the strategies are considered in this paper. Although the second strategy is often used to deal with occlusion and corruption [21], our experiments show that it also works well in our algorithm even when there are no occlusion and corruption in face images. This is partially due to that the strategy can provide illumination compensation for representing a given face image. (See the next subsection and Fig.2 for more details).

3.2 Sparse reconstructive weights

Since DR is mainly characterized by specific affinity weight matrix of the data, we try to construct the matrix based on a Modified Sparse Representation (MSR) framework, and then explain why it is helpful to both the compact representation of data and the subsequent classification task.

Given a set of training samples $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in R^m$, let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in R^{m \times n}$ be the data matrix including all the training samples in its columns. We expect to reconstruct each sample $\mathbf{x}_i$, e.g., a face image, using as few samples as possible. Hence we firstly seek a ***sparse reconstructive weight vector*** $\mathbf{s}_i$ for each $\mathbf{x}_i$ through the following modified ℓ_1 minimization problem:

$$\begin{aligned} &\min_{\mathbf{s}_i} \| \mathbf{s}_i \|_1 \\ &s.t. \quad \mathbf{x}_i = \mathbf{X}\mathbf{s}_i \\ &\qquad 1 = \mathbf{1}^T \mathbf{s}_i \end{aligned} \tag{12}$$

where $\mathbf{s}_i = [s_{i1}, \cdots, s_{i,i-1}, 0, s_{i,i+1}, \cdots, s_{in}]^T$ is a *n*-dimensional vector in which the *i*-th element is equal to zero (implying that the $\mathbf{x}_i$ is removed from $\mathbf{X}$), and the elements $s_{ij}, j \neq i$ denote the contribution of each $\mathbf{x}_j$

to reconstructing $\mathbf{x}_i$; $\mathbf{1} \in R^n$ is a vector of all ones.

The MSR problem can be solved by standard linear programming as original ℓ_1 minimization problem Eq.(11), since the sum-to-one constraint $1 = \mathbf{1}^T \mathbf{s}_i$ is also linear. We will explain the reason for this constraint shortly.

After computing the weight vector $\mathbf{s}_i$ for each $\mathbf{x}_i$, $i = 1,2,\cdots,n$, we can define the ***sparse reconstructive weight matrix*** $\mathbf{S} = (\tilde{s}_{ij})_{n\times n}$ as follows:

$$\mathbf{S} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_n]^T \tag{13}$$

where $\tilde{\mathbf{s}}_i$ is the optimal solution of Eq.(12). The element $\tilde{s}_{ij}$ in $\mathbf{S}$ is not simple similarity measure between samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and in this sense $\mathbf{S}$ is essentially different from the adjacency weigh matrix in LPP.

Now we give some insights into the effectiveness of $\mathbf{S}$ as a weight matrix for dimensionality reduction and the subsequent recognition task.

1) Each weight vector $\mathbf{s}_i$ obeys an important symmetry: it is invariant to rotations and rescalings due to the first constraint in Eq.(12), and invariant to translations due to the sum-to-one constraint $1 = \mathbf{1}^T \mathbf{s}_i$. As a result, the weight matrix $\mathbf{S}$ reflects some intrinsic geometric properties[1] of the data.

2) Discriminant information can be naturally preserved in the weight matrix $\mathbf{S}$, even if no class-labels are provided. Let us take face recognition as an example. One particularly simple but effective assumption in face recognition is that the samples from the same class lie on a linear subspace (so-called face subspace). Given a face image $\mathbf{x}_i^j$ from the *j*-th class, $\mathbf{x}_i^j$ can be theoretically represented using the samples from the *j*-th class according to the subspace assumption. That is,

$$\mathbf{x}_i^j = 0 \cdot \mathbf{x}_1^1 + \cdots + \alpha_{i,i-1}\mathbf{x}_{i-1}^j + \alpha_{i,i+1}\mathbf{x}_{i+1}^j + \cdots + 0 \cdot \mathbf{x}_n^c \tag{14}$$

where $j = 1,\cdots,c$, denotes the class label. The weight vector $\mathbf{s}_i^0 = [0,\cdots,\alpha_{i,i-1},0,\alpha_{i,i+1},\cdots,0]^T$ is sparse, since class number is generally large[2] in most face recognition problems. Although the Eq.(14) does not always hold due to insufficient sampling, our experiments show the sparse $\mathbf{s}_i^0$ can actually be approximated by the optimal solutions $\tilde{\mathbf{s}}_i$ (see Fig. 2). In other words, the non-zero entries in $\tilde{\mathbf{s}}_i$ mostly correspond to the samples from the *j*-th class, which implies that $\tilde{\mathbf{s}}_i$ may help to distinguish that class from the others. Therefore, the weight vector $\tilde{\mathbf{s}}_i$, constructed using all the samples with sparsity constraint, tends to include potential discriminant information.

As described previously, in some real-world applications, the constraint $\mathbf{x}_i = \mathbf{X}\mathbf{s}_i$ in Eq.(12) does not

[1] These properties can also be got from the popular LLE algorithm. We will discuss the similarities and differences between LLE and our proposed algorithm in section 4.

[2] For example, if *c*=10, then at least 90% of the entries in $\mathbf{s}_i$ should be zeros.

always hold. By considering the two strategies mentioned in subsection 3.1, the MSR problem can be extended to the following two stable versions, (15) and (16). The first extension is defined as

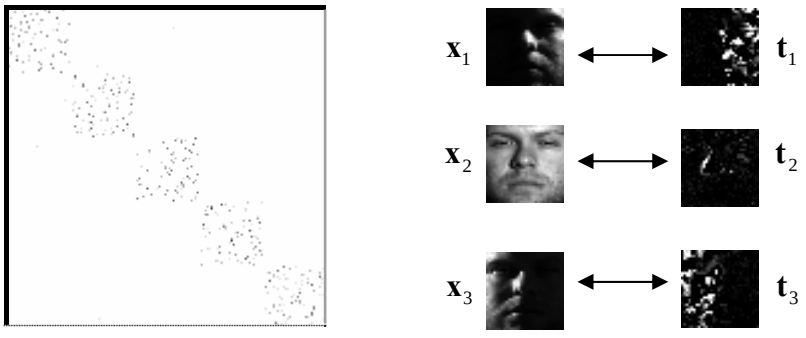
$$\begin{aligned} &\min_{\mathbf{s}_i,\mathbf{t}} \| \mathbf{s}_i \|_1 \\ &s.t. \quad \| \mathbf{x}_i - \mathbf{X}\mathbf{s}_i \| < \varepsilon \\ &\qquad 1 = \mathbf{1}^T \mathbf{s}_i \end{aligned} \tag{15}$$

where ε is the error tolerance and generally fixed across various instances of the problem[21]. It is easy to validate that its optimal solution still reflects some intrinsic geometric properties (e.g. invariant to translations and rotations) of the original data. Another extension[3] of MSR can be expressed as

$$\begin{aligned} &\min_{[\mathbf{s}_i^T \ \mathbf{t}^T]^T} \| [\mathbf{s}_i^T \ \mathbf{t}_i^T]^T \|_1 \\ &s.t. \quad \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{I} \\ \mathbf{1}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{s}_i \\ \mathbf{t}_i \end{bmatrix} \end{aligned} \tag{16}$$

where $\mathbf{t}_i$ is an *m*-dimensional vector, $\mathbf{0}$ is an *m*-dimensional vector of all zeros. The optimal solution of (16) is also invariant to translations, but the invariance to rotations and rescalings does not rigorously hold any longer. But our experiments indicate that such a loss of the invariance does not much invoke influence on the final classification performance and conversely increases robustness to lighting change. Here, we take extended Yale B face database (see section 5.2 for special description about this database) as an example to intuitively show why the Eq.(16) may work.

Let us assume the training data matrix $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{38}]$, where $\mathbf{X}_i$ denotes the data samples that belong to the *i*-th class. Then, we calculate the weight matrix $\mathbf{S}$ based on Eq.(16). For space limitation, only a sub-block of $\mathbf{S}$ corresponding to the first 5 classes is shown in Fig. 2(a) with the gray level[4] denoting the value of the element $\tilde{s}_{ij}$. In Fig. 2(b) we show 3 $\mathbf{t}_i$'s (i.e., $\mathbf{t}_1$, $\mathbf{t}_2$ and $\mathbf{t}_3$), which are respectively associated with 3 samples from the first class. From the example, we find most of the non-zero adjacency weights link the samples from the same class, and intuitively the $\mathbf{t}_i$ plays a role to compensate the illumination.



(a) (b)

Fig. 2 (a) A sub-block of the weight matrix $\mathbf{S}$ constructed by Eq.(16). (b) The optimal $\mathbf{t}_i$'s for 3 different samples.

[3] In fact, we can further consider the trade-off between $\mathbf{s}_i$ and $\mathbf{t}_i$ in (16) to design a more general version. However, we empirically find such extension is generally not helpful on our used databases.

[4] For convenience of display, the black pixel denotes 1, while the white pixel denotes 0.

3.3 Preserving sparse reconstructive weights

By the above design, the sparse weight matrix $\mathbf{S}$ can reflect intrinsic geometric properties of the data to some extent and contains natural discriminating information. We thereby expect that the desirable characteristics in the original high-dimensional space can be preserved in the low-dimensional embedding subspace. Therefore, similar to LLE and NPE, we define the following objective function to seek the projections which best preserve the optimal weight vector $\tilde{\mathbf{s}}_i$.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \| \mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{X} \tilde{\mathbf{s}}_i \|^2 \tag{17}$$

With simple algebric formulation, we can get

$$\begin{aligned} &\sum_{i=1}^{n} \| \mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{X} \tilde{\mathbf{s}}_i \|^2 \\ &= \mathbf{w}^T \left(\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{X}\tilde{\mathbf{s}}_i)(\mathbf{x}_i - \mathbf{X}\tilde{\mathbf{s}}_i)^T \right) \mathbf{w} \end{aligned} \tag{18}$$

Let $\mathbf{e}_i$ be a *n*-dimensional unit vector with the *i*-th element 1, 0 otherwise, then Eq.(18) is equal to

$$\begin{aligned} &\mathbf{w}^T \left(\sum_{i=1}^{n} (\mathbf{X}\mathbf{e}_i - \mathbf{X}\tilde{\mathbf{s}}_i)(\mathbf{X}\mathbf{e}_i - \mathbf{X}\tilde{\mathbf{s}}_i)^T \right) \mathbf{w} \\ &= \mathbf{w}^T \mathbf{X} \left(\sum_{i=1}^{n} (\mathbf{e}_i - \tilde{\mathbf{s}}_i)(\mathbf{e}_i - \tilde{\mathbf{s}}_i)^T \right) \mathbf{X}^T \mathbf{w} \\ &= \mathbf{w}^T \mathbf{X} \left(\sum_{i=1}^{n} \mathbf{e}_i \mathbf{e}_i^T - \tilde{\mathbf{s}}_i \mathbf{e}_i^T - \mathbf{e}_i \tilde{\mathbf{s}}_i^T + \tilde{\mathbf{s}}_i \tilde{\mathbf{s}}_i^T \right) \mathbf{X}^T \mathbf{w} \\ &= \mathbf{w}^T \mathbf{X} (\mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}^T \mathbf{S}) \mathbf{X}^T \mathbf{w} \end{aligned} \tag{19}$$

To avoid degenerate solutions, we constrain $\mathbf{w}^T \mathbf{X}\mathbf{X}^T \mathbf{w}$ =1. Thus, the objective function can be recast as the following optimization problem:

$$\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X} (\mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}^T \mathbf{S}) \mathbf{X}^T \mathbf{w}}{\mathbf{w}^T \mathbf{X}\mathbf{X}^T \mathbf{w}} \tag{20}$$

For compact expression, the minimization problem can further be transformed to an equivalent maximization problem as follows:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X} \mathbf{S}_\beta \mathbf{X}^T \mathbf{w}}{\mathbf{w}^T \mathbf{X}\mathbf{X}^T \mathbf{w}} \tag{21}$$

where $\mathbf{S}_\beta = \mathbf{S} + \mathbf{S}^T - \mathbf{S}^T \mathbf{S}$. Another benefit of this transform is that the maximum formulation in some case can get a more numerically stable solution [17]. Then, the optimal **w**'s are the eigenvectors corresponding to the largest *d* eigenvalues of the following generalized eigenvalue problem:

$$\mathbf{X} \mathbf{S}_\beta \mathbf{X}^T \mathbf{w} = \lambda \mathbf{X}\mathbf{X}^T \mathbf{w} \tag{22}$$

3.4 SPP algorithm

Based on the above discussion, we summarize the proposed algorithm as follows:

Algorithm: Sparsity Preserving Projections

Step 1: Construct weight matrix $\mathbf{S}$ using MSR(12) or stable MSR(15), (16);

Step 2: Calculate the projection vectors using (22), and the eigenvectors corresponding to the largest *d* eigenvalues span the optimal subspace.

Fig. 3 SPP algorithm

The algorithm is simple, since it does not involve any hyper-parameters except the subspace dimension *d*. For example, in step 1, MSR can be efficiently solved by standard Linear Programming (LP) using publicly available packages such as *l*1-magic[5]. In addition, if the sparsity is well considered, the sparse representation problem can be more efficiently solved [42]. In step 2, one can directly calculate eigenvectors for most practical applications or resort to the recent proposed techniques such as spectral regression [17] and Density-weighted Nystrom method [51] for large scale problems.

For some high-dimensional data, the matrix $\mathbf{X}\mathbf{X}^T$ is generally singular since the training sample size is much smaller than the feature dimensions. To address this problem, the training set can be first projected onto a PCA subspace spanned by its leading eigenvectors: $\mathbf{W}_{pca} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{d'}]$. The matrix $\mathbf{X}\mathbf{X}^T$ is then approximated by $\hat{\mathbf{X}}\hat{\mathbf{X}}^T$ ($\hat{\mathbf{X}} = \mathbf{W}_{pca}^T\mathbf{X}$), which is obviously nonsingular.

4 Comparison with Related Works

4.1 PCA

PCA can be seen as a globality preserving DR method in that a single hyper-plane is used to represent the data, hence not facing problem of selecting appropriate neighborhood size. SPP also doesn't need to worry about this since it actually uses all the training samples to construct the weight matrix without explicitly setting the neighborhood size. But compared to PCA, SPP has an extra advantage that it is capable of implicitly and naturally employing the "local" structure of the data by imposing the sparsity prior.

4.2 NPE and other Locality preserving DRs

SPP has a similar objective function to NPE (c.f., Eq.(9) and (21). Both of them can be closely related to LLE. In fact, NPE is a directly linearized version of LLE, while our SPP constructs the "affinity" weight matrix in a completely different manner from LLE. In particular, SPP constructs the weight matrix using all the training samples (with sparsity constraint) instead of *k* nearest neighbors, preventing it from suffering from the difficulty of parameter selection as in the case of NPE and other locality preserving DRs. Despite of such difference, SPP can actually be thought as a regularized extension of NPE through the modified ℓ_1-regularization problem. From the Bayesian [43] view, such regularization essentially encodes prior knowledge of sparsity, allowing it to extract more discriminating information from the data than NPE does.

[5] The web site (http://www.dsp.ece.rice.edu/cs/) provides many practical toolboxes and recent research works to solve the sparse representation problem. In our experiments, *l1-Magic* toolbox is used due to its simplicity.

For clarity, three key components (i.e., model parameters, "affinity" weight matrices and the way to construct them) of several common DR methods are summarized in Table 1.

Table 1 The construction manners of weight matrices for different DR methods.

	Model parameters	weight matrices	construction manners
PCA	No	**E**	globality*
LPP	*k*, *t*	**P**	local neighborhood distance
NPE	*k*	**N**	local neighborhood reconstruction
SPP	No	**S**	global sparse reconstruction

*__E__ denotes the matrix of all ones, and its corresponding adjacency graph is given in [25].

4.3 Sparse Subspace Learning

Sparse Subspace Learning (SSL) [44] is a special family of DR methods which also consider "sparsity". The representative SSL methods include Sparse Principal Component Analysis (SPCA) [45], Nonnegative Sparse PCA [46] and Sparse Nonnegative Matrix Factorization [47], etc.. Although having different objective functions, they share a common goal, i.e., to find a subspace spanned by ***sparse base vectors***. Different from those methods whose sparsity is encoded in the projection $\mathbf{w}$ and associated with the *feature dimension*, SPP aims at the sparse reconstructive weight $\mathbf{s}_i$ associated with the *sample size*. Naturally, we can also enforce the projection $\mathbf{w}$ sparse in SPP, but that is beyond the focus of this paper.

4.4 Sparse Representation Classifier (SRC)

SRC [21,37] is a recently proposed supervised classification framework based on sparse representation. Surprisingly, [21] shows that the classification performance of most meaningful features converges when the feature dimension increases if a SCR classifier is used. Although this does provide some new insight into the role of feature extraction played in a pattern classification task, we argue that designing effective and efficient feature extractor is still of great importance since the classification algorithm could become simple and tractable.

Here we note several remarkable differences between SPP and SRC: 1) SPP is a feature extractor, while SRC is a classifier. As a result, SPP can be used as a preprocessor for any typical classifiers, such as 1NN, SVM and even SRC. 2) SPP is an unsupervised algorithm, meaning that SPP may enjoy wider applications such as data visualization (e.g., Fig.5) and other unsupervised learning tasks. 3) SRC is essentially a lazy classifier and uses time-consuming sparse reconstruction for each test sample. By contrast, in SPP, the sparse reconstruction is involved only in the training process. Once the low-dimensional projection vectors are obtained, they can be used for both the training data and the test data, thus being able to effectively improve the efficiency of recognition.

5 Experiments

5.1 Illustrative examples

In this section, we first use two simple data sets including a toy data set and a real-word data set to intuitively show how and why our algorithm works.

5.1.1 Toy problem

Let's consider the following toy binary classification problem on a 3D space where the samples from

each class lie on an intrinsic 1D subspace (see Fig. 4(b0), where each class is denoted by a bar): We randomly sample 10 points from each class, and half of them are added by Gaussian white noise with standard deviation 0.1, which makes the samples from each class actually lie on the 3D space instead of the 1D one (Fig. 4(a0)). Then we construct four 1D subspaces for these data using PCA, LPP, NPE and SPP respectively. For LPP and NPE, we empirically[6] set the neighborhood size $k = \min\{n_i\} - 1$, where n_i denotes the number of the i-th class samples. The heat kernel parameter t in LPP is empirically chosen as the mean norm of the samples. The results are plotted in Fig. 4(a1-a4). We repeat the experiments using another dataset with 100 points generated in the same way above and the results are shown in Fig. 4(c1-c4).

From Fig.4, we have the following observations:

1) For PCA, LPP and NPE, the samples from two classes are overlapped together, but the degree of overlapping is different for these methods respectively. The PCA suffers most since it tries to use a single hyper-plane to model the data according to the directions of large sample variance. Both LPP and NPE improve over this by explicitly taking the local structure of data into account. But the Euclidean distance measure and the predefined neighborhood size in these methods fail to identify the real local structure they are supposed.

2) On the contrary, one can see from Fig. 4(a4,c4) that, with SPP, the two classes can be perfectly separated in the low-dimensional subspace. This can be explained from the angle of sparse prior, which assumes that a point should be best explained by a set of samples as small as possible. Further, this illustrates that SPP can effectively and implicitly use the subspace assumption, even when the two classes are close to each other and the data are noisy.

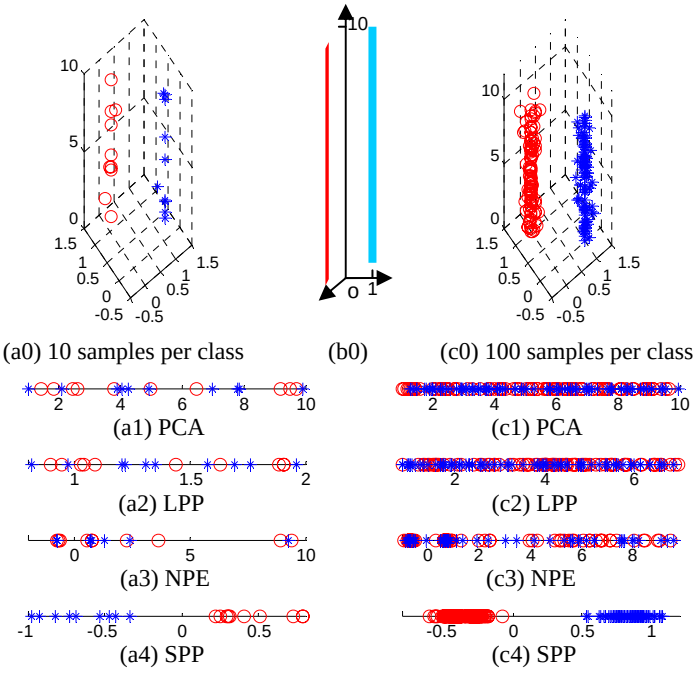


Fig. 4 The toy data ant their 1D images based on 4 DRs above mentioned algorithms

5.1.2 *Wine* dataset from UCI

[6] In the experiments on the toy problem and the following real-world UCI dataset, we attempt to assign the parameter values by searching from a large range of candidates, but most of them can not achieve satisfying results.

Now we use *Wine* dataset, a real-life dataset from the UCI machine learning repository[7], to give SPP further explanation. *Wine* has 13 features, 3 classes and 178 instances. The basic statistics including means, variances and ranges of the 13 features are shown in Table 2.

Table 2 The means, variances and ranges of the 13 features on *Wine* dataset

Features	1	2	3	4	5	6	7	8	9	10	11	12	13
Mean	13.0	2.3	2.4	19.5	99.7	2.3	2.0	0.4	1.6	5.1	1.0	2.6	**746.9**
Variance	0.7	1.2	0.08	11.2	204.0	0.4	1.0	0.02	0.3	5.4	0.05	0.5	**99166.7**
Range	3.8	5.1	1.9	19.4	92	2.9	4.7	0.5	3.2	11.7	1.2	2.7	**1402**

It is easy to see from Table 2 that the last feature should play a decisive role in the data distribution due to its large range and variance. Here, we apply PCA, LPP, NPE and the proposed SPP respectively to project the data onto a 2D subspace. The hyper-parameters in LPP and NPE are chosen by following the same scheme as in the toy problem. According to the results shown in Fig. 5, we have the following observations:

1) The leading projection directions of PCA are decided by the 13th feature, since it has a much larger variance and range than the others. Thus the 2D data distribute in a large range along the direction dominated by the last feature, which makes the projected data mixed up.
2) The locality preserving methods such as LPP and NPE suffer from the same problem as in PCA, even though the local information is considered. This is because the neighborhood of a certain sample point is still dominated by the 13th feature due to its large variance and range.
3) The data projected by SPP form a point-cloud distribution instead of a "linear" one. This reflects that the SPP gives other features besides the 13th one a chance to play a role in capturing a reasonable structure of the data.

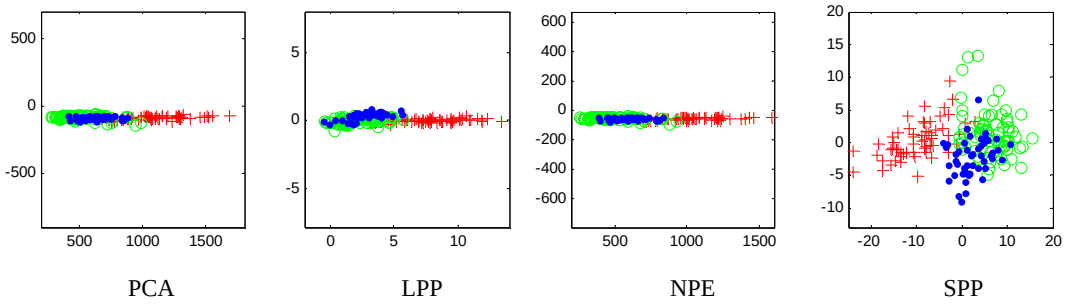


Fig. 5 The 2D results of *Wine* dataset based on 4 different DR methods.

From the two illustrative examples, we can see that sparsity actually works, especially when the data from each class lie on a subspace. Furthermore, SPP is not quite sensitive to the imbalance of the feature distribution which incurs the failure of LPP and NPE, due to the fact that neighborhood is mainly decided by the features with large range and variance.

5.2 Face representation and recognition

PCA, LPP and NPE are three popular unsupervised DR methods. They have been successfully applied to face recognition where they are known as Eigenface [6], Laplacianface [26] and NPEface [18] respectively. In what follows, we test the performance of the three popular algorithms and our proposed algorithm on three

[7] http://archive.ics.uci.edu/ml

face databases.

5.2.1 Datasets and Experimental Settings

We firstly give simple descriptions of the datasets used later.

Yale: This database [2] contains 165 face images of 15 individuals. There are 11 images per subject, and these 11 images are respectively under the following different facial expression or configuration: center-light, wearing glasses, happy, left-light, wearing no glasses, normal, right-light, sad, sleepy, surprised, and wink. In our experiment, the images are cropped to a size of 32x32, and the gray level values of all images are rescaled to [0 1]. ***AR:*** This database consists of over 4000 face images of 126 individuals. For each individual, 26 pictures were taken in two sessions (separated by two weeks) and each section contains 13 images. These images include front view of faces with different expressions, illuminations and occlusions. In our experiments here, we use a subset of the AR face database provided and preprocessed by Martinez [49]. This subset contains 1400 face images corresponding to 100 person (50 men and 50 women), where each person has 14 different images with illumination change and expressions. The original resolution of these image faces is 165x120. Here, for computational convenience, we resize them to 66x48, and the gray level values are rescaled to [0 1]. ***Extended YaleB:*** This database [51] contains 2414 front-view face images of 38 individuals. For each individual, about 64 pictures were taken under various laboratory-controlled lighting conditions. In our experiments, we use the cropped images with the resolution of 32x32, which is directly downloaded from http://www.cs.uiuc.edu/homes/dengcai2. Fig. 6 shows some face images from the three face databases used here.

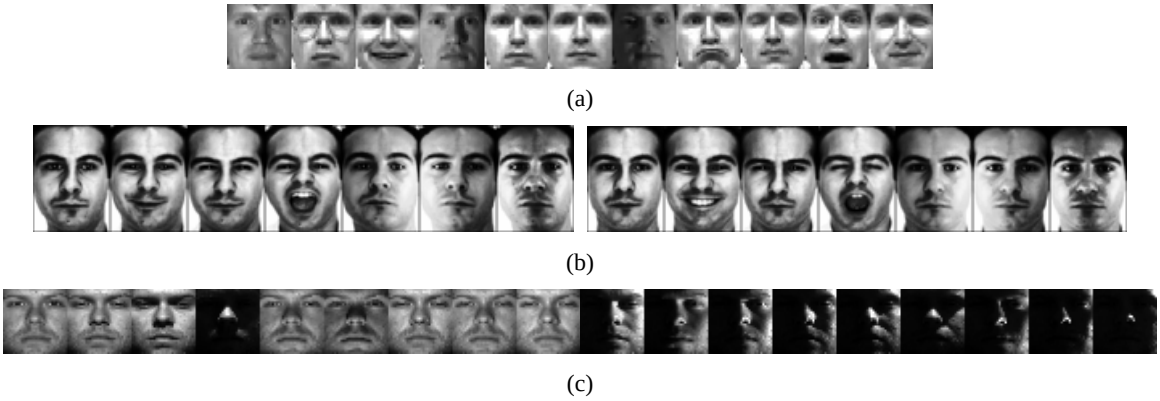


Fig. 6 (a) All the 11 images of the first person in Yale database. (b) All the 14 images of the first person in the subset of AR. The first 7 images are from the first session, and the last 7 image are from the second session. (c) Partial images of the first person in Extended Yale B database.

For these databases, we randomly select half of the images per class for training (i.e., 6, 7 and about 32 images per subject for Yale, AR and Extended Yale B databases, respectively), and the remaining for test. Since AR database has naturally been partitioned into two sessions, we also consider this case in our experiments. We simply use "AR_fixed" to denote the AR database partitioned based on two fixed sessions, and "AR_random" to the one partitioned randomly. In particular, with the given training set, the projection matrix $\mathbf{W}$ is learned by PCA, LPP, NPE and SPP respectively, and the test samples are subsequently transformed by the learned projection matrix. Then, specific classifiers are employed to evaluate the recognition rates on the test data. In the experiments, 20 training/test splits are randomly generated and the

average classification accuracies over these splits are reported. The codes of PCA, LPP[8] and NPE are all from http://www.cs.uiuc.edu/homes/dengcai2.

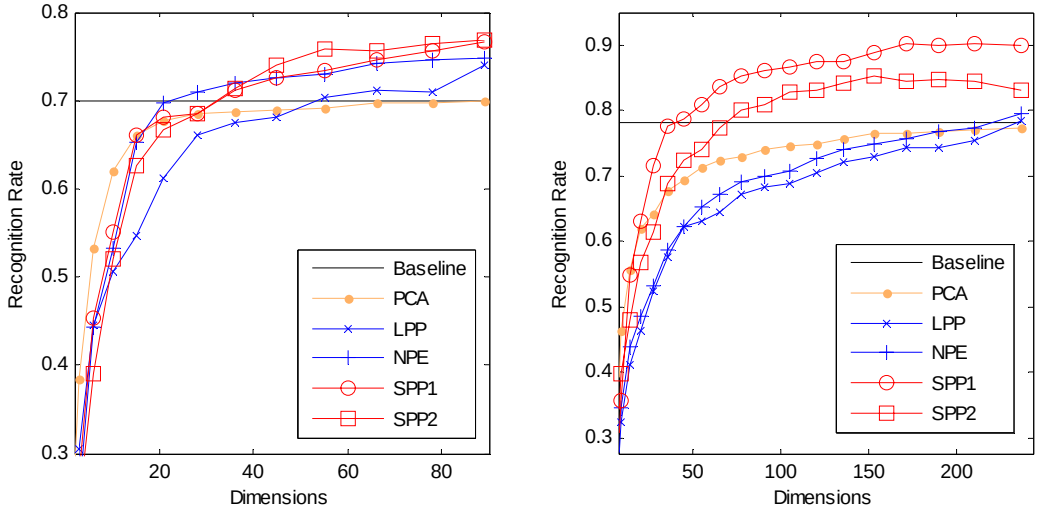
5.2.2 Parameter Selection

For PCA, the only model parameter is the subspace dimension. For LPP, the model parameters include neighborhood size k and kernel width t. In our experiments, we set their appropriate values by search in a large range of candidates and report the best results. A similar strategy is used to decide the neighborhood size k in NPE. In particular, for Yale and AR databases, the neighborhood size k is searched from $\{1,2,\cdots,l-1\}$; for Extended Yale B database, from $\{1,2,5,10,\cdots,l-1\}$, where l is the number of the training samples in each class. The kernel width t is empirically set as the mean norm t_0 of the training data, or the adjacency weight is directly calculated based on "cosine" distance. For the proposed SPP, when directly using the MSR (12) or the stable MSR (16) to construct the adjacency weight matrix, it is parameter-free; when using another stable version (15), it has an error tolerance parameter ε which is generally fixed across various instances of the problem [37], and thus in our experiments we simply set it to 0.05 as in [37].

5.2.3 Experimental Results

A. Based on 1-NN classifier

To verify the effectiveness of the proposed method, in this series of experiments we evaluate the performance of the proposed method and compare it to that of several methods using the simplest Nearest Neighbor (1-NN) classifier. As a baseline, we also give the classification results of 1-NN classifier directly using the raw data without dimensionality reduction. The recognition rates are shown in Fig. 7, where SPP1 denotes the SPP algorithm based on the stable MSR (15), SPP2 denotes the SPP based on (16). We also summarize the best results of these methods in Table 3. Furthermore, based on the training set of AR database, the first 10 Eigenfaces, Laplacianfaces, NPEfaces and SPPfaces are shown in Fig. 8.



Yale AR_fixed

[8] We use the unsupervised LPP instead of the supervised extension, since we only focus on unsupervised dimensionality reduction in this paper.

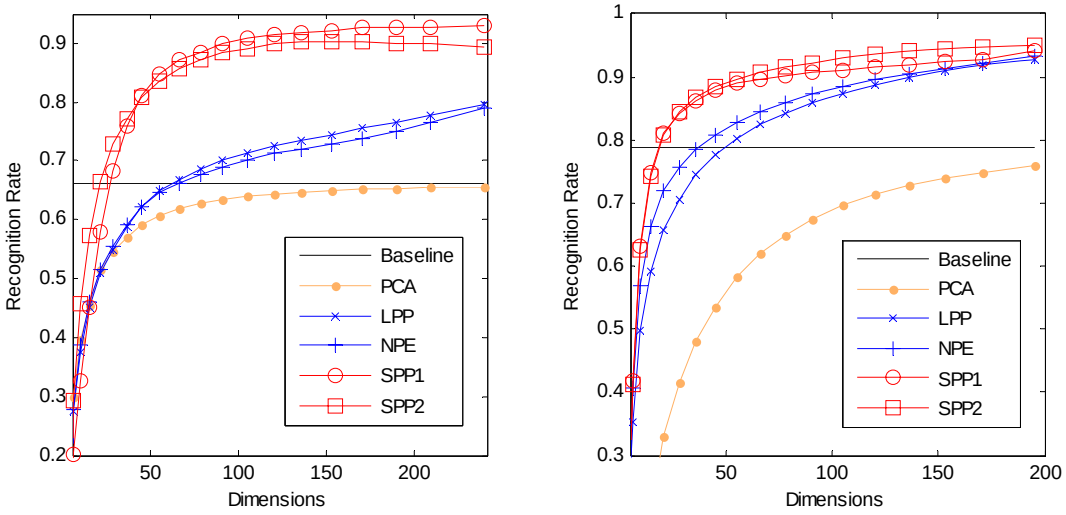


Fig. 7 The recognition rates of 1-NN classifier based on several mentioned DRs.

Table 3 The best recognition rates of 1NN classifier based on different DRs.

Yale (PCAratio=1)

DRs	Baseline	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.6993	0.6993	0.7407	0.7513	0.766	**0.7680**
Dimensions	1024	86	89	86	89	81
Parameters	no	no	k=1,cosine	k=5	no	no

AR_fixed (PCAratio=0.98)

DRs	Baseline	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.7814	0.7729	0.7843	0.7943	**0.9057**	0.8557
Dimensions	3168	232	235	236	216	176
Parameters	no	no	k=1, $t=t_0$	k=3	no	no

AR_random (PCAratio=0.98)

DRs	Baseline	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.6627	0.6564	0.7961	0.7904	**0.9308**	0.9041
Dimensions	3168	240	240	240	227	159
Parameters	no	no	k=1, $t=t_0$	k=1	no	no

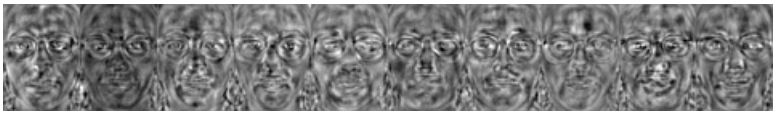
Extended Yale B (PCAratio=0.98)

DRs	Baseline	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.7887	0.7589	0.9273	0.9319	0.9414	**0.9518**
Dimensions	1024	195	190	195	195	193
Parameters	no	no	k=2, $t=t_0$	k=2	no	no

* PCAratio denotes the energy ratio kept in the PCA preprocessing step.

(a) Eigenface

(b) Laplacianface



(c) NPEface

(d) SPP1face

(e) SPP2face

Fig. 8 The first 10 basis vectors of PCA, LPP, NPE and SPP calculated from the training set of AR database.

B. Based on other classifiers

We further evaluate the above mentioned DRs based on several other popular classifiers including k-NN, SVM and SRC. Here, we just report the experimental results on Yale database since the simple 1-NN classifier does not work well on it. One can refer to [21] for some related discussions on AR and Extended Yale B databases. The experimental results are shown in Table 4. For k-NN classifier, we empirically set neighborhood size k=6 (i.e., the training sample size per subject). For SVM, we simply use the linear kernel following the scheme of [21].

Table 4 The recognition rates on Yale database based on different feature extractor and classifier pairs.

k-NN

DRs	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.7087	0.7400	0.7887	**0.8447**	0.8220
Dimensions	86	46	87	88	89

SVM

DRs	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.8293	0.9073	0.9073	0.9593	**0.9613**
Dimensions	89	89	89	86	88

SRC

DRs	PCA	LPP	NPE	SPP1	SPP2
Accuracy	0.9493	**0.9560**	0.9520	0.9493	0.9493
Dimensions	89	89	89	89	89

5.2.4 Overall observations and discussion of the above experiment results

1) PCA is simple to perform, but it generally gets much worse performance than LPP, NPE and SPP. Based on 1-NN classifier, its recognition rates are just close to the baseline on all the used databases, which is consistent with the results in many publications such as recent [40].
2) LPP and NPE always outperform PCA when the subspace dimension exceeds a certain threshold. This shows that by preserving the local structure of the data, the recognition rate can be improved. That is, when nearest neighbor search is considered, local structure seems to be more important than global structure. However, LPP and NPE are less tractable than PCA due to the difficulty of parameter selection involved. Results shown in Table 3 indicate that one has to adjust the values of parameters in LPP and NPE for different training set in order to achieve good performance. However, we have also observed that small neighborhood size k in LPP and NPE empirically tends to perform better on the data sets used.
3) On the tested databases, the two versions of SPP consistently outperform PCA, LPP and NPE with

1-NN classifier, even though no parameter needs to be adjusted. This suggests that the projections found by SPP can preserve more discriminating information than those of the compared methods. Furthermore, the performance of the two versions is data-driven. Concretely, SPP1 achieves better performance on AR database, while SPP2 generally outperforms SPP1 on Yale and Extended Yale B databases.

4) As shown in Table 4, the classifiers also affect the recognition performance significantly. However, the proposed SPP can generally achieve better performance than PCA, LPP and NPE based on most of the classifiers (i.e., 1NN, k-NN and SVM). The only exception is that for SRC classifier, all the mentioned DRs achieve comparable performance. This further illustrates that SRC is insensitive to different feature extractors as pointed out in [21]. Furthermore, by combining the SPP algorithm and the SVM classifier, we can achieve the best performance among all the combinations of the compared feature extractors and classifiers.

6 Conclusions and Future Work

In this paper, based on sparse representation, we propose a new algorithm called Sparsity Preserving Projections (SPP) for unsupervised dimensionality reduction. In the proposed algorithm, the projections of SPP are sought such that the sparse reconstructive weights can be best preserved. SPP is shown to outperform PCA, LPP and NPE on all the data sets used here, and is very simple to perform like PCA by avoiding the difficulty of parameter selection as in LPP and NPE. Since it remains unclear how to define the "locality" theoretically for many locality-based algorithms like LPP and NPE, SPP can be considered as an alternative to them.

However, each approach has its own advantages and disadvantages. SPP is sensitive to large variations in pose as many whole-pattern based feature extractors such as PCA, LPP and NPE. Therefore, in this paper, we only focus on front-view face images with variations in illumination and expression. In the future work, we will try to overcome this limitation using sub-pattern based strategy [48] and absorb supervised information into the algorithm to further improve its performance.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. This work was partly supported by National Natural Science Foundation of China (60773061, 60773060), the Innovation Foundation of NUAA (Y0603-042) and Project sponsored by SRF for ROCS, SEM. Also thank Deng Cai et al for providing the codes of LPP and NPE on their homepage.

References

[1] A. Jain, R. Duin and J. Mao, Statistical pattern recognition: A review, IEEE Trans. Pattern Anal. Mach. Intell. 22(1) (2000) 4-37

[2] P. Belhumeur, J. Hepanha and D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19(7) (1997) 711-720

[3] D. Xu, S. Yan, D. Tao, S. Lin and H. Zhang, Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval, IEEE Trans. Image Processing, 16(11) (2007) 2811-2821

[4] H. Li, T. Jiang and K. Zhang, Efficient and robust feature extraction by maximum margin criterion, IEEE Trans. Neural Networks, 17(1) (2006) 157-165

[5] J. Liu, S. Chen, X. Tan and D. Zhang, Comments on “Efficient and robust feature extraction by maximum margin criterion”, IEEE Trans. Neural Networks, 18(6) (2007) 1862-1864

[6] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience, 3(1) (1991) 71-86

[7] X. He, P. Niyogi, Locality preserving projections, Proc. Conf. Advances in Neural Information Processing Systems (NIPS), 2003

[8] D. Zhang, Z. Zhou and S. Chen, Semi-supervised dimensionality reduction, in SIAM Conference on Data Mining (ICDM), 2007

[9] D. Cai, X. He and J. Han, Semi-supervised discriminant analysis, Proc. in International Conference on Computer Vision (ICCV), 2007

[10] Y. Song, F. Nie, C. Zhang and S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognition, 41(9) (2008) 2789-2799

[11] B. Scholkopf, A. Smola, and K. Muller, Kernel principal component analysis, Advances in Kernel Methods-Support Vector Learning, B. Scholkopf, C. Burges, and A. Smola, Ed. Cambridge, MA: MIT Press, 1999, pp. 327-352

[12] J. Tenenbaum, Mapping a manifold of perceptual observations, Advances in Neural Information Processing Systems (NIPS), 1998

[13] S. Roweis, L, Saul, Nonlinear dimensionality reduction by Locally Linear Embedding, Science, 290(5500) (2000) 2323-2326

[14] M. Belkin, P. Niyogi, Laplacian Eigenmaps for dimensionality reduction and data representation, Neural Computation, 15(6) (2003) 1373-1396

[15] L. Maaten, E. Postma, and H. Herik, Dimensionality reduction: A comparative review. Submitted to Neurocomputing, 2009

[16] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux and M. Ouimet, Out-of-sample Extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering, Advances in Neural Information Processing Systems (NIPS), 2004

[17] D. Cai, X. He and J. Han, Spectral regression for dimensionality reduction, Technical report, Computer Science Department, UIUC, UIUCDCS-R-2007-2856, May 2007

[18] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, Proc. in International Conference on Computer Vision (ICCV), 2005

[19] Y. Fu, T. Huang, Locally linear embedded eigenspace analysis, IFP-TR, Univ. of Illinois at Urbana-Champaign, Jan.2005

[20] D. Cai, X. He and J. Han, Isometric Projection, In Proc. AAAI Conf. on Artificial Intelligence, 2007

[21] J. Wright, A. Yang, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31(2) (2009) 210-227

[22] K. Huang, S. Aviyente, Sparse representation for signal classification, Advances in Neural Information Processing Systems (NIPS), 2006

[23] M. Davenport, M. Duarte, M. Wakin, D. Takhar, K. Kelly and R. Baraniuk, The smashed filter for compressive classification and target recognition, in Proc. IS&T/SPIE Symposiumon Electronic Imaging: Computational Imaging, Jan. 2007

[24] M. Davenport, M. Wakin and R. Baraniuk, Detection and estimation with compressive measurements. Technical Report, January 24, 2007

[25] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang and S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29(1) (2007) 40-51

[26] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, Face recognition using Laplacianfaces, IEEE. Trans. Pattern Analysis and Machine Intelligence, vol. 27, No. 3, 2005

[27] J. Ham, D. Lee, S. Mika, and B. Scholkopf, A kernel view of the dimensionality reduction of manifolds, Proc. Int'l Conf. Machine Learning, pp. 47-54, 2004

[28] J. Murray, K. Kreutz-Delgado. Visual recognition and inference using dynamic overcomplete sparse learning. Neural Computation, 19 (2007) 2301–2352

[29] R. Duda, P. Hart, D. Stork, Pattern classication, 2nd ed. John Wiley & Sons, NY, 2001

[30] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek. An overview of jpeg-2000, Proc. of the Data Compression Conference, 2000

[31] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. on Image Processing 15(12) (2006) 3736-3745

[32] J. Yang, J. Wright, Y. Ma and T. Huang, Image super-resolution as sparse representation of raw image patches, in Computer Vision and Pattern Recognition (CVPR), 2008

[33] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society B, 58(1) (1996) 267-288

[34] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society Series B, 67(2) (2005) 301–320

[35] R. Baraniuk, A Lecture on compressive sensing, IEEE Signal Processing Magazine, 24(4) (2007) 118-121

[36] D. Donoho, Compressed sensing, IEEE Trans. Inform. Theory, 52(4) (2006) 1289-1306

[37] A. Yang, J. Wright, Y. Ma, S. Sastry, Feature selection in face recognition: A sparse representation perspective, UC Berkeley Tech Report UCB/EECS-2007-99, 2007

[38] Mallat S, Zhang Z, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process, 41(12) (1993) 3397-3415

[39] S. Ji, Y. Xue and L. Carin, Bayesian compressive sensing, IEEE Trans. Signal Process, 56(6) (2008) 2346-2356

[40] M. Wu, Kai. Yu, S. Yu and B. Scholkopf, Local learning projections, in International Conference on Machine Learning (ICML), 2007

[41] S. Chen, D. Donoho and M. Saunders, Atomic decomposition by basis pursuit, SIAM Review, 43(1) (2001) 129-159

[42] D. Donoho, Y. Tsaig, Fast solution of l1-norm minimization problems when the solution may be sparse, Technical Report, Institute for Computational and Math. Eng., Stanford University, 2006.

[43] M. E. Tipping, Sparse bayesian learning and the relevance vector machine, Journal of Machine Learning Research, 1 (2001) 211-244

[44] D. Cai, X. He and J. Han, Spectral regression: A unified approach for sparse subspace learning, in Proc.

Int. Conf. on Data Mining (ICDM), 2007

[45] H. Zhou, T. Hastie, and R. Tibshirani, Sparse principle component analysis, Technical report, Statistics Department, Stanford University, 2004

[46] R. Zass, A. Shashua, Non-negative sparse PCA, Advances in Neural Information Processing systems(NIPS), 2007

[47] P. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research, 5 (2004) 1457–1469

[48] S. Chen, Y. Zhu, Subpattern-based principal component analysis, Pattern Recognition, 37(1) (2004) 1081-1083

[49] A. M. Martinez, A. C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23(2) (2001) 228-233

[50] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27(5) (2005) 684-698

[51] K. Zhang, J.T. Kwok. Density-weighted Nystrom method for computing large kernel eigen-systems, Neural Computation, 21(1) (2009) 121-146