A Linear Subspace Learning Approach via Sparse Coding

Lei Zhang[a], Pengfei Zhu[a], Qinghua Hu[b] and David Zhang[a]
[a]Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[b]Harbin Institute of Technology, Harbin, China
cslzhang@comp.polyu.edu.hk

Abstract

Linear subspace learning (LSL) is a popular approach to image recognition and it aims to reveal the essential features of high dimensional data, e.g., facial images, in a lower dimensional space by linear projection. Most LSL methods compute directly the statistics of original training samples to learn the subspace. However, these methods do not effectively exploit the different contributions of different image components to image recognition. We propose a novel LSL approach by sparse coding and feature grouping. A dictionary is learned from the training dataset, and it is used to sparsely decompose the training samples. The decomposed image components are grouped into a more discriminative part (MDP) and a less discriminative part (LDP). An unsupervised criterion and a supervised criterion are then proposed to learn the desired subspace, where the MDP is preserved and the LDP is suppressed simultaneously. The experimental results on benchmark face image databases validated that the proposed methods outperform many state-of-the-art LSL schemes.

1. Introduction

As a popular dimensionality reduction and feature extraction technique, linear subspace learning (LSL) has been successfully used in various computer vision and pattern recognition applications, for example, appearance based face recognition (FR). Representative LSL methods include principal component analysis (PCA), e.g., Eigenface [1], Fisher linear discriminant analysis (FLDA) [2-4], the manifold learning [5-6] based locality preserving projection (LPP) [7], local discriminant embedding (LDE) [8], graph embedding [9], etc. According to if the class label information of the training samples is exploited, the LSL methods can be categorized into unsupervised methods (e.g., PCA and LPP) and supervised methods (e.g., FLDA [2], regularized LDA (RLDA) [4] and LDE).

Generally speaking, LSL methods learn the desired subspace or projections by optimizing a certain criterion function. For example, PCA seeks for an optimal subspace in which the image variable vector is de-correlated, while FLDA seeks for an optimal subspace by maximizing the ratio of between-class scatter to within-class scatter. Considering the fact that high dimensional data often reside on a low dimensional manifold, the LSL methods such as LPP [7] learn the subspace by preserving the geometric graph of the original high dimensional data.

One key step in LSL is the estimation of sample scatter matrices, with which the liner projections are computed. On the other hand, the learned projections decompose the training samples into different components, which have different contributions to recognition tasks. Most of the existing LSL methods estimate the scatter matrices directly from the original training samples. That is, the subspace is learned for image decomposition, or at most we could say that the subspace learning and image decomposition are accomplished simultaneously. However, the different contributions of different components to image recognition cannot be effectively exploited by these methods because the original training samples are used in statistics calculation. For example, the noise and trivial structures in face images should have little contribution to FR because they could not represent the intrinsic and stable features of the subject. Intuitively, why don't we decompose the image first and then use the different image components to guide the subspace learning? It is of high interest and importance to investigate new LSL schemes by considering the characteristics of different image components.

It has been found that natural images can be generally represented by a small number of basis functions chosen out of an over-complete code set [10]. With the development of l_0- and l_1-minimization techniques [11-12], in recent years the sparse coding (or sparse representation) methods have been well studied for solving the inverse problems in image reconstruction and separation, such as compressive signal recovery [11-12], morphological component analysis [13-14] and dictionary learning [15-16]. Suppose that $\boldsymbol{x} \in \Re^n$ is the target signal to be coded, and $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m]$ is the given dictionary of atoms $\boldsymbol{\phi}_i$, the sparse coding of $\boldsymbol{x}$ over $\boldsymbol{\Phi}$ is to find a sparse vector $\boldsymbol{\alpha}$ (i.e., most of the coefficients in $\boldsymbol{\alpha}$ are close to zero) such that $\boldsymbol{x} \approx \boldsymbol{\Phi\alpha}$. If l_1-norm is used to measure the sparsity, the sparse coding problem can be formulated as $\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1$ s.t.

$\|\boldsymbol{x}-\boldsymbol{\Phi\alpha}\|_2^2 \le \varepsilon$, where ε is a constant. Alternatively, the Lagrangian formulation of the above minimization is often used: $\min_{\alpha}\left\{\|\boldsymbol{x}-\boldsymbol{\Phi\alpha}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_1\right\}$, where constant λ is to balance the reconstruction error and the sparsity constraint.

Sparse coding has also been used for pattern classification. In [17], a signal is coded over a set of redundant bases and is then classified based on its sparse coding vector. In [18], a sparse representation based classification (SRC) scheme is proposed for robust FR. The query face image is sparsely coded over the training samples, and then it is classified to the class which yields the least coding error. The SRC is improved in [19] by learning a dictionary from the original training samples. In [20], the l_1-graph is established by sparsely coding one sample over the other samples for classification. In [21], an LSL scheme called sparsity-preserving projection (SPP) is proposed for FR. It aims to preserve the l_1-graph after the linear dimension reduction. Compared with LPP, which aims to preserve the l_2-graph of the training samples, SPP shows some superiority. However, very recently Zhang *et al*. [22] showed that it is not the l_1-norm sparsity but the collaborative representation mechanism that truly helps FR in such sparsity based FR methods.

Different from most of the existing LSL methods such as PCA, FLDA/RLDA, LPP and SPP, where the subspace is learned for image decomposition, in this paper we propose a new LSL framework, where the images are decomposed for subspace learning. The sparse coding is used as a tool for adaptive image decomposition in the learning stage. First, a patch based dictionary $\boldsymbol{D}$ is learned from the training samples. Suppose that $\boldsymbol{D}$ has k atoms. By coding each image patch over $\boldsymbol{D}$ (note that this process is actually accomplished in the dictionary learning stage), the whole training sample can be written as a linear combination of k components. We then group the k components into a more discriminative part (MDP) and a less discriminative part (LDP). Finally, we seek for a subspace where the MDP is preserved while the LDP is suppressed. Once the projection matrix $\boldsymbol{P}$ is trained, for a query image we only need to project it onto $\boldsymbol{P}$ for classification. The sparse coding is only employed in the learning stage.

The rest of the paper is organized as follows. Section 2 presents the methodology. Section 3 performs extensive experiments and Section 4 concludes the paper.

2. Subspace learning via sparse coding

2.1. Motivation and flowchart

One important advantage of LSL methods is their efficiency and simplicity. With the learned projection matrix $\boldsymbol{P}$, the dimensionality of training samples can be significantly reduced so that the storage space can be greatly reduced. Meanwhile, in the dimensionality reduced subspace, the main features of the images can be enhanced so that the recognition accuracy can be improved. Apart from the projection matrix $\boldsymbol{P}$, the classifier is another important issue in practical FR systems. In this paper we prefer to use the simple nearest neighbor (NN) classifier for its efficiency.

To preserve the discriminative features in the subspace defined by $\boldsymbol{P}$, a step of feature grouping can be very helpful. In the case of unsupervised LSL, we could group the features into a more informative group and a less informative group, while in the case of supervised LSL, the Fisher criterion can be applied to group the features into a more discriminative group and a less discriminative group. The linear subspace can then be computed by preserving the more informative/discriminative components and suppressing the less informative/discriminative components simultaneously. In order for feature grouping, we use the dictionary learning and sparse coding technique to decompose the training samples over a set of adaptive and redundant bases so that there is more flexibility for image (feature) representation. The flowchart of the proposed LSL approach is shown in Fig. 1.

Figure 1: The procedures of the proposed linear subspace learning approach.

2.2. Dictionary learning and sparse coding

Denote by $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_m]\in\Re^{n\times m}$ the dataset of training samples. We'd like to decompose each training sample $\boldsymbol{x}_i$, i=1,2,…,m, into two parts, a part $\boldsymbol{x}_i^a$ formed by the more informative (discriminative) components and a part $\boldsymbol{x}_i^b$ formed by the less informative (discriminative) components. Then the dataset $\boldsymbol{X}$ can be written as $\boldsymbol{X}=\boldsymbol{X}_a+\boldsymbol{X}_b$, where $\boldsymbol{X}_a=[\boldsymbol{x}_1^a,\boldsymbol{x}_2^a,...,\boldsymbol{x}_m^a]$ and $\boldsymbol{X}_b=[\boldsymbol{x}_1^b,\boldsymbol{x}_2^b,...,\boldsymbol{x}_m^b]$. With a linear projection matrix $\boldsymbol{P}$, there is $\boldsymbol{PX}=\boldsymbol{PX}_a+\boldsymbol{PX}_b$, and we intend to make the features in $\boldsymbol{X}_a$ be preserved and the features in $\boldsymbol{X}_b$ be suppressed after the projection by $\boldsymbol{P}$. Finally, when a testing face image $\boldsymbol{y}$ comes, we can directly project it onto $\boldsymbol{P}$, and compute the distance between $\boldsymbol{Py}$ and $\boldsymbol{PX}$ to judge which class $\boldsymbol{y}$ belongs to according to the NN classification rule.

One important issue in the proposed scheme is how to decompose the training sample images for grouping. There are many existing tools for image decomposition, such as Fourier transform, wavelet transform, etc. However, these transforms are universal to all types of images, and they

may not be most effective to face images. It is desired that a transformation which is adaptive to the face dataset $\boldsymbol{X}$ can be used. PCA is a kind of adaptive transformation, whose bases are adaptively calculated from $\boldsymbol{X}$. Nonetheless, PCA is an orthogonal transform aiming to de-correlate $\boldsymbol{X}$, and it concentrates most of the energy of $\boldsymbol{X}$ into only several major components. This makes the feature grouping of $\boldsymbol{X}$ in the PCA domain infeasible.

Inspired by the success of sparse coding in image processing, we propose to learn an adaptive dictionary $\boldsymbol{D}$ from $\boldsymbol{X}$ and use it to represent $\boldsymbol{X}$. Since the dimensionality of original face image $\boldsymbol{x}_i$ is often very high, it is hard to learn a redundant dictionary directly for $\boldsymbol{X}$ under the sparse coding framework. As in [15-16], we learn a patch based dictionary. Each training sample $\boldsymbol{x}_i$ is partitioned into q overlapped patches, and totally there are $h=m\times q$ patches. Suppose that the dimension of each patch vector $\boldsymbol{t}_j$, $j=1,2,\ldots,h$, is l, then an $l\times h$ data matrix $\boldsymbol{T}=[\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_h]$ is established. From $\boldsymbol{T}$, we aim to learn a dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k] \in \Re^{l\times k}$, where $\boldsymbol{d}_z^T\boldsymbol{d}_z=\boldsymbol{1}$, $z=1,2,\ldots,k$, such that

$$J_{\boldsymbol{D},\boldsymbol{\Lambda}} = \arg\min_{\boldsymbol{D},\boldsymbol{\Lambda}} \left\{ \left\| \boldsymbol{T} - \boldsymbol{D}\boldsymbol{\Lambda} \right\|_F^2 + \lambda \left\| \boldsymbol{\Lambda} \right\|_1 \right\} \quad (1)$$

where $\boldsymbol{\Lambda}=[\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_h]\in\Re^{k\times h}$ and $\boldsymbol{\alpha}_j$ is the coding vector of $\boldsymbol{t}_j$ over $\boldsymbol{D}$, and λ is a scalar that balances the sparsity and reconstruction error.

Eq. (1) is a joint optimization problem of the dictionary $\boldsymbol{D}$ and the coefficient matrix $\boldsymbol{\Lambda}$. Although it is not jointly convex to $\boldsymbol{D}$ and $\boldsymbol{\Lambda}$, it is convex with respect to each of them when the other one is fixed. Therefore, a local minimum of Eq. (1) can be obtained by optimizing $\boldsymbol{D}$ and $\boldsymbol{\Lambda}$ alternatively. In this paper, we adopted the dictionary learning algorithm in [19] to solve Eq. (1).

In learning the dictionary $\boldsymbol{D}$, the sparse coding matrix $\boldsymbol{\Lambda}$ is computed simultaneously. For each patch $\boldsymbol{t}_j$, we have

$$\boldsymbol{t}_j \approx \boldsymbol{D}\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_j(1)\cdot\boldsymbol{d}_1 + \boldsymbol{\alpha}_j(2)\cdot\boldsymbol{d}_2 + \ldots + \boldsymbol{\alpha}_j(k)\cdot\boldsymbol{d}_k$$

That is, each patch can be written as the summation of k components

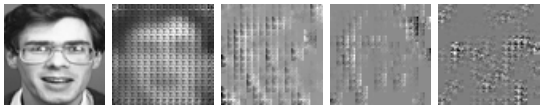
$$\boldsymbol{t}_j \approx \boldsymbol{t}_{j,1} + \boldsymbol{t}_{j,2} + \ldots + \boldsymbol{t}_{j,k} \quad (2)$$

where $\boldsymbol{t}_{j,z}=\boldsymbol{\alpha}_j(z)\cdot\boldsymbol{d}_z$. By combining all the patches, image $\boldsymbol{x}_i$ can be written as the summation of k components:

$$\boldsymbol{x}_i \approx \boldsymbol{x}_{i,1} + \boldsymbol{x}_{i,2} + \ldots + \boldsymbol{x}_{i,k} \quad (3)$$

where $\boldsymbol{x}_{i,z}$ is just the concatenation of those $\boldsymbol{t}_{j,z}$ belong to this image, and the overlapped pixels of the patches are simply averaged for $\boldsymbol{x}_{i,z}$.

Fig. 2 shows an example of sparse coding. We set k=64 in the dictionary learning. Fig. 2(a) is the original face image; Fig. 2(b) ~ Fig. 2(e) show the decomposed components $\boldsymbol{x}_{i,z}$ corresponding to the 1st, 11th, 21th, and 41th atoms. We can see that the representations on different atoms are different, which implies that the discrimination ability of different components is different. This lays the foundation for feature grouping in next subsection. Please note that in our methods the sparse coding is used for image decomposition but not for local feature extraction. The LSL methods to be developed are holistic feature based, while they can be extended to local feature based methods.



(a) (b) (c) (d) (e)

Figure 2: (a) A face image, and (b) ~ (e) some decomposed components of it via dictionary learning and sparse coding.

(a) (b) (c)

Figure 3: (a) A face image; and its (b) more informative part and (c) less informative part after unsupervised feature grouping.

2.3. Unsupervised subspace learning

After sparse coding, each face image $\boldsymbol{x}_i$ is decomposed into k feature images $\boldsymbol{x}_{i,z}$. We can then group them for more effective subspace learning. Depending on whether or not the class labels of training samples are known, we have different grouping and learning criteria. In this subsection, we discuss the unsupervised learning, while the supervised learning is discussed in next subsection.

2.3.1. Feature grouping. In unsupervised learning, we do not know the class label of each face image $\boldsymbol{x}_i$. Or we can view each image $\boldsymbol{x}_i$ as a class. Hence if the feature $\boldsymbol{x}_{i,z}$ has a bigger variance, we can think that this feature is more informative to separate the samples. Based on this heuristic, we can group those feature images into a more informative group and a less informative group.

Denote by $\bar{\boldsymbol{x}}_z$ the mean of feature images $\boldsymbol{x}_{i,z}$. The variance of feature image $\boldsymbol{x}_{i,z}$ is

$$\sigma_z = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{x}_{i,z} - \bar{\boldsymbol{x}}_z\right)^2 \quad (4)$$

We want to put those feature images having larger σ_z into the more informative group, and the remaining into the less informative group. To this end, we re-order $\boldsymbol{x}_{i,z}$ according to their variances σ_z in descending order. Then the first $\tau\cdot k$ feature images, where τ is a constant, are grouped into the more informative group, while the remaining images are grouped into the less informative group.

For the convenience of expression, we suppose that features $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,z1}\}$ fall into the more informative group and the remaining features $\{\boldsymbol{x}_{i,z1+1}, \boldsymbol{x}_{i,z1+2}, \ldots, \boldsymbol{x}_{i,k}\}$ fall into the less informative group. Then we define the more discriminative part (MDP) $\boldsymbol{x}_i^a$ and the less discriminative part (LDP) $\boldsymbol{x}_i^b$ of each image $\boldsymbol{x}_i$ as

$$\boldsymbol{x}_i^a = \boldsymbol{x}_{i,1} + \boldsymbol{x}_{i,2} + \ldots + \boldsymbol{x}_{i,z1} \quad (5\text{-a})$$

$$\boldsymbol{x}_i^b = \boldsymbol{x}_{i,z1+1} + \boldsymbol{x}_{i,z1+2} + \ldots + \boldsymbol{x}_{i,k} \quad (5\text{-b})$$

Fig. 3 shows the MDP and LDP of the face image in Fig. 2. We can see that the MDP image preserves the main appearance of the original face image.

2.3.2. Subspace learning. After feature grouping, each training sample can be written as $\boldsymbol{x}_i = \boldsymbol{x}_i^a + \boldsymbol{x}_i^b$, and thus we have $\boldsymbol{X} = \boldsymbol{X}_a + \boldsymbol{X}_b$. It is experimentally validated that $\boldsymbol{X}_a$ could lead to much higher FR rate than $\boldsymbol{X}_b$. However, if we learn the projection matrix $\boldsymbol{P}$ only based on $\boldsymbol{X}_a$, the result cannot be very satisfying because $\boldsymbol{X}_b$ is also useful for determining the projection direction. To effectively exploit the information in both $\boldsymbol{X}_a$ and $\boldsymbol{X}_b$, we propose to learn a subspace where the energy in $\boldsymbol{X}_a$ is well preserved and the energy in $\boldsymbol{X}_b$ is suppressed.

Denote by $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{x}}^a$ and $\bar{\boldsymbol{x}}^b$ the mean vectors of $\boldsymbol{X}$, $\boldsymbol{X}_a$ and $\boldsymbol{X}_b$, respectively, and let $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$, $\bar{\boldsymbol{x}}_i^a = \boldsymbol{x}_i^a - \bar{\boldsymbol{x}}^a$ and $\bar{\boldsymbol{x}}_i^b = \boldsymbol{x}_i^b - \bar{\boldsymbol{x}}^b$ be the centralized image vectors. Accordingly we have the centralized datasets $\bar{\boldsymbol{X}}$, $\bar{\boldsymbol{X}}_a$, and $\bar{\boldsymbol{X}}_b$. Clearly, we have $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}^a + \bar{\boldsymbol{x}}^b$, $\bar{\boldsymbol{x}}_i = \bar{\boldsymbol{x}}_i^a + \bar{\boldsymbol{x}}_i^b$ and $\boldsymbol{P}\bar{\boldsymbol{x}}_i = \boldsymbol{P}\bar{\boldsymbol{x}}_i^a + \boldsymbol{P}\bar{\boldsymbol{x}}_i^b$. After projection, the average energy of $\boldsymbol{P}\bar{\boldsymbol{x}}_i^a$ is

$$\begin{aligned} E_a &= \tfrac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{P}\bar{\boldsymbol{x}}_i^a\right\|_2^2 = \tfrac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{P}\bar{\boldsymbol{x}}_i^a\right)^T\left(\boldsymbol{P}\bar{\boldsymbol{x}}_i^a\right) \\ &= tr\left\{\boldsymbol{P}\left(\tfrac{1}{m}\bar{\boldsymbol{X}}_a\bar{\boldsymbol{X}}_a^T\right)\boldsymbol{P}^T\right\} = tr\left\{\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T\right\} \end{aligned} \quad (6)$$

where $\boldsymbol{S}_a = \tfrac{1}{m}\bar{\boldsymbol{X}}_a\bar{\boldsymbol{X}}_a^T$ is the total scatter matrix (i.e. the covariance matrix) of $\boldsymbol{X}_a$, and "*tr*" is the matrix trace operator. Similarly, the average energy of $\boldsymbol{P}\bar{\boldsymbol{x}}_i^b$ is

$$E_b = \tfrac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{P}\bar{\boldsymbol{x}}_i^b\right\|_2^2 = tr\left\{\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T\right\} \quad (7)$$

where $\boldsymbol{S}_b = \tfrac{1}{m}\bar{\boldsymbol{X}}_b\bar{\boldsymbol{X}}_b^T$ is the total scatter matrix of $\boldsymbol{X}_b$.

To preserve the MDP $\boldsymbol{X}_a$ while suppressing the LDP $\boldsymbol{X}_b$, we seek for a projection matrix $\boldsymbol{P}$ to maximize the energy E_a while minimizing the energy E_b by solving the following optimization problem:

$$J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}}\frac{E_a}{E_b} = \arg\max_{\boldsymbol{P}}\frac{tr\left(\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T\right)}{tr\left(\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T\right)} \quad (8)$$

An equivalent form of Eq. (8) is

$$J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} tr\left(\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T\right) \text{ s.t. } \boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T = \boldsymbol{I} \quad (9)$$

Apparently, the desired $\boldsymbol{P}$ can be computed by using generalized eigenvalue decomposition, i.e., matrix $\boldsymbol{P}$ is composed of the generalized eigenvectors of $\boldsymbol{S}_a\boldsymbol{w}=\lambda\boldsymbol{S}_b\boldsymbol{w}$ corresponding to the p largest eigenvalues. We can see that the conventional PCA method is a special case of the proposed method without applying sparse coding and feature grouping to the training images. In this case the scatter matrix $\boldsymbol{S}_b$ does not exist and the scatter matrix $\boldsymbol{S}_a$ becomes the total scatter matrix $\boldsymbol{S}$ of all training samples. Therefore, the objective function in Eq. (9) reduces to $J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} tr\left(\boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T\right)$ s.t. $\boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T = \boldsymbol{I}$, which is exactly the objective function of PCA.

2.4. Supervised subspace learning

2.4.1. Feature grouping. In supervised learning, the class label of $\boldsymbol{x}_i$ is available. In this case, the Fisher ratio can be utilized to evaluate features. If the feature $\boldsymbol{x}_{i,z}$ has a bigger Fisher ratio, this feature is more discriminative to separate the samples. Based on this heuristic, we can group the feature images $\boldsymbol{x}_{i,z}$ into a more discriminative group and a less discriminative group.

We denote by $\boldsymbol{X}_c$ the set of samples of the c^{th} class, by $\bar{\boldsymbol{x}}_z$ the mean of feature images $\boldsymbol{x}_{i,z}$ and by $\bar{\boldsymbol{x}}_{z,c}$ the mean of feature images $\boldsymbol{x}_{i,z}$ that belong to class c, c=1,2,…,C. The fisher ratio f_z of feature images $\boldsymbol{x}_{i,z}$ is

$$f_z = \frac{\sigma_b}{\sigma_w} = \frac{\sum_{c=1}^{C}(\bar{\boldsymbol{x}}_z - \bar{\boldsymbol{x}}_{z,c})^2}{\sum_{c=1}^{C}\frac{1}{m_c}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_c}(\boldsymbol{x}_{i,z} - \bar{\boldsymbol{x}}_{z,c})^2} \quad (10)$$

where m_c is the number of samples belong to the c^{th} class.

Similar to unsupervised LSL, those feature images which have larger f_z are added up for the MDP image $\boldsymbol{x}_i^a$, and the remaining features are added up for the LDP image $\boldsymbol{x}_i^b$. Fig. 4 shows the MDP and LDP images of the face image in Fig. 2.

(a) (b) (c)

Figure 4: (a) A face image; and its (b) more discriminative part and (c) less discriminative part after supervised feature grouping.

2.4.2. Subspace learning. After supervised feature grouping, each training sample can be written as $\boldsymbol{x}_i = \boldsymbol{x}_i^a + \boldsymbol{x}_i^b$, and we have $\boldsymbol{X} = \boldsymbol{X}_a + \boldsymbol{X}_b$. Our goal is still to train a projection matrix $\boldsymbol{P}$ so that for a query image $\boldsymbol{y}$ we can use $\boldsymbol{P}\boldsymbol{y}$ as the feature for classification. Denote by $\bar{\boldsymbol{x}}_c^a$ the mean vector of the MDP images in the c^{th} class, and denote by $\bar{\boldsymbol{x}}^a$ the mean vector of the MDP images in all classes. We can construct the between-class and within-class scatter matrices of MDP images as follows

$$\boldsymbol{S}_B^a = \tfrac{1}{m}\sum_{c=1}^{C} m_c\left(\bar{\boldsymbol{x}}_c^a - \bar{\boldsymbol{x}}^a\right)\left(\bar{\boldsymbol{x}}_c^a - \bar{\boldsymbol{x}}^a\right)^T \quad (11)$$

$$\boldsymbol{S}_W^a = \tfrac{1}{m}\sum_{c=1}^{C}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_c}\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c^a\right)\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c^a\right)^T \quad (12)$$

Denote by $\bar{\boldsymbol{x}}^b$ the mean vector of the LDP images in all classes. The total LDP scatter matrix is

$$\boldsymbol{S}^b = \frac{1}{m}\sum_{c=1}^{C}\sum_{\boldsymbol{x}_i \in X_c}\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}^b\right)\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}^b\right)^T \quad (13)$$

The LDP scatter matrix $\boldsymbol{S}^b$ should be minimized in the subspace, i.e., the desired $\boldsymbol{P}$ should minimize $tr\{\boldsymbol{P}\boldsymbol{S}^b\boldsymbol{P}^T\}$. Meanwhile, to better separate the different classes in the subspace, the MDP between-class scatter matrix should be maximized while the MDP within-class scatter matrix should be minimized, i.e., maximize $tr\{\boldsymbol{P}\boldsymbol{S}_B^a\boldsymbol{P}^T\}$ and minimize $tr\{\boldsymbol{P}\boldsymbol{S}_W^a\boldsymbol{P}^T\}$. In total, the supervised subspace learning criterion can be defined as follows:

$$J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} \frac{tr\{\boldsymbol{P}\boldsymbol{S}_B^a\boldsymbol{P}^T\}}{\alpha \cdot tr\{\boldsymbol{P}\boldsymbol{S}_W^a\boldsymbol{P}^T\} + (1-\alpha)\,tr\{\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T\}}$$

where scalar α is used to balance the MDP within-class scatter and the LDP total scatter. The above criterion is equivalent to

$$J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} \frac{tr\{\boldsymbol{P}\boldsymbol{S}_B^a\boldsymbol{P}^T\}}{tr\left\{\boldsymbol{P}\left(\alpha\boldsymbol{S}_W^a + (1-\alpha)\boldsymbol{S}_b\right)\boldsymbol{P}^T\right\}} \quad (14)$$

Clearly, the row vector of desired $\boldsymbol{P}$ can be chosen as the p generalized eigenvectors of $\boldsymbol{S}_B^a\boldsymbol{w} = \lambda[\alpha\boldsymbol{S}_W^a + (1-\alpha)\boldsymbol{S}_b]\boldsymbol{w}$ corresponding to the first p largest eigenvalues. It is not difficult to see that the conventional FLDA is a special case of Eq. (14) without applying sparse coding and feature grouping to the training images and let α=0.

3. Experimental results

We denote the proposed unsupervised LSL method via sparse coding as USCP (unsupervised sparse coding based projection), and the supervised version as SSCP (supervised sparse coding based projection). The performance of USCP and SSCP is evaluated on three representative facial image databases: the AR database, the extended Yale B database and the Multi-PIE database. The representative LSL methods, including PCA (Eigenface), LPP (Laplacianface) [7], FLDA (Fisherface) [2], RLDA [4] and SPP [21], are used for comparison. The code of the proposed USCP and SSCP methods can be downloaded at http://www4.comp.polyu.edu.hk/~cslzhang/code.htm.

In all the experiments, the size of face images is resized to 32×32. In dictionary learning, the size of each patch is 8×8 and the overlap between two neighboring patches is 4 pixels. The number of atoms, i.e., k, in the dictionary is 64. The NN classifier with Euclidean distance is employed for classification. The parameter α in Eq. (14) for SSCP is chosen by 10-fold cross-validation on training set. As to the parameter τ in feature grouping, it is set as 0.8 for both USCP and SSCP. On each database, the program is run for 100 times and the average results are reported.

Since the dimension of the image vector space is much larger than the number of training samples, all the methods (except for PCA) involve a PCA (or KL transform) phase on the three databases.

3.1. Experiments on the extended Yale B database

(a) Training samples

(b) Testing samples

Figure 5: Some randomly selected training and testing images of a subject in the Extended Yale B database.

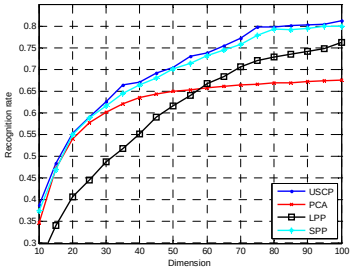


Figure 6: Recognition rates of PCA, LLP, SPP and USCP versus dimensions on the Extended Yale B database.

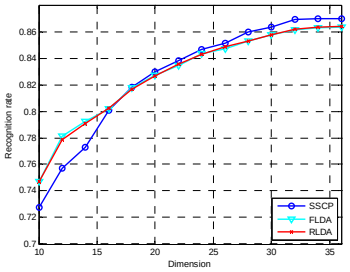


Figure 7: Recognition rates of FLDA, RLDA and SSCP versus dimensions on the Extended Yale B database.

The extended Yale B face database [25] contains 38 human subjects under 9 poses and 64 illumination conditions. The 64 images of a subject in a particular pose were acquired at a rate of 30 frames/second so that there are only small changes in head pose and facial expression. All frontal-face images marked with P00 were used in our experiment. Each image is resized and pre-processed by histogram equalization. In each of the 100 runs, we randomly selected 5 images from the first 32 images per subject for training, and 5 images from the remaining 32 images for testing. Fig. 5 shows some example training and testing samples. In the PCA phase of methods FLDA, LPP, etc, we selected the number of principal axes as 100. The K-nearest neighborhood parameter K is chosen as 1 in LPP method.

The average recognition rates (over 100 runs) versus dimensions are illustrated in Fig. 6 and Fig. 7. We can see

that the recognition rate of USCP is much higher than PCA and LPP and is higher than SPP, while SSCP performs better than FLDA and RLDA.

3.2. Experiments on the Multi-PIE database

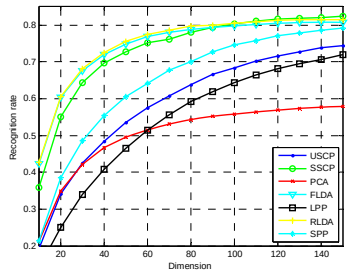


Figure 8: Recognition rates of all competing methods versus dimensions on the Multi-PIE database.

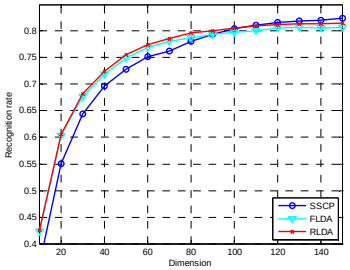


Figure 9: Recognition rates of the supervised methods SSCP, FLDA and RLDA versus dimensions on the Multi-PIE database.

The Multi-PIE database [24] contains 337 subjects, captured under 15 viewpoints and 19 illumination conditions in four recording sessions for a total of more than 750,000 images. We selected a subset that contains images from 249 individuals from session 1, each providing 11 different images. In the experiment, we randomly selected three images of each class for training and the remaining eight images for testing. In the PCA phase of methods FLDA, LPP, etc, we selected the number of principal axes as 200. The average recognition rates versus the variation of dimensions are illustrated in Fig. 8 and Fig. 9. We can see that the performances of USCP and SSCP are very competitive.

3.3. Experiments on the AR database

The AR face database [23] contains over 4,000 color face images of 126 people (70 men and 56 women), including frontal views of faces with different facial expressions, lighting conditions and occlusions. The pictures of most people were taken in two sessions (separated by two weeks). Each session contains 14 color images and 120 individuals (65 men and 55 women) participated in both sessions. The images of these 120 persons were selected in our experiment. Only the full facial images were considered here (no attempt was made to handle occluded face recognition in each session). We manually cropped the face portion and normalized it to 32×32 pixels. The normalized images of one person are shown in Fig. 10, where (a)~(g) are from Session 1, and (n)~(t) are from Session 2. The details of the images are: (a) neutral expression, (b) smile, (c) anger, (d) scream, (e) left light on; (f) right light on; (g) all sides light on; and (n)~(t) were taken under the same conditions as (a)~(g).

Figure 10: Sample images for one subject of the AR database.

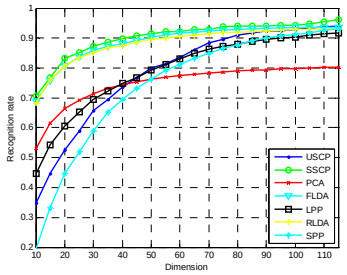


Figure 11: Recognition rates of all competing methods versus dimensions on the AR database.

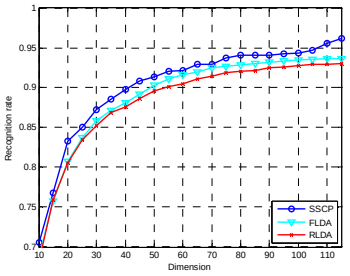


Figure 12: Recognition rates of supervised methods SSCP, FLDA and RLDA versus dimensions on the AR database.

In the experiment, we randomly selected two images from Section 1 and the corresponding two images from Section 2 for training. The remaining ten images were used for testing. In the PCA phase, we selected the number of principal axes as 200. The K-nearest neighborhood parameter K is chosen as 3 in the LPP method. From Fig. 11 and Fig. 12, we can see that SSCP consistently outperforms other methods, and USCP performs better than the unsupervised methods.

Table 1 summarizes the top average recognition rates of all competing methods on the three databases. We can

conclude that USCP and SSCP work stably across all the databases. The top average recognition rates of USCP are much higher than LPP and PCA, comparable to SSP, and even better than the supervised FLDA and RLDA methods on some databases. As to the supervised method SSCP, it consistently outperforms all the competing methods on all the databases, validating that the proposed subspace learning methods via sparse coding is effective.

Table 1: The top average recognition rates (%) and the associated dimensionality of different methods.

Method	Yale B	MPIE	AR
PCA	67.5±17.2	57.9±10.1	80.2±13.6
	100	150	115
LLP	76.2±16.6	71.9±7.3	91.8±3.5
	100	150	115
SSP	79.9±17.7	**79.2±7.3**	93.2±2.8
	95	150	115
USCP	**81.2±16.8**	**74.5±8.8**	**94.0±4.4**
	100	150	115
FLDA	86.3±12.8	80.7±8.8	93.7±8.6
	36	130	115
RLDA	86.4±12.9	81.5±9.5	93.1±9.2
	36	150	115
SSCP	**87.0±13.5**	**82.4±11.6**	**96.2±3.2**
	36	150	115

4. Conclusion

In this paper, we proposed a novel linear subspace learning (LSL) method via sparse coding and feature grouping. A patch based dictionary with *k* atoms was first learned from the training set. Then each training image can be decomposed as a linear combination of *k* components. These components were grouped into two parts: a more discriminative part (MDP) and a less discriminative part (LDP). Finally, a desired linear subspace was sought by preserving the MDP component while weakening the LDP component. The experimental results on benchmark face databases showed that the proposed sparse coding induced LSL methods outperform many representative and state-of-the-art LSL methods.

5. References

[1] M. Turk and A. Pentland. Eigenfaces for recognition. J. Cognitive Neuroscience, 3(1):71-86, 1991.

[2] P. Belhumeur, J. Hespanha, and D. Kriengman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE TPAMI, 19 (7): 711-720, 1997.

[3] J. Yang and J.Y. Yang. Why can LDA be performed in PCA transformed space. Pattern Recognition, 36(2):563-566, 2003.

[4] J. Lu, K. Plataniotis and A. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. Pattern Recognition Letters, 26(2): 181–191, 2005.

[5] J. Tenenbaum, V. deSilva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290: 2319-2323, 2000.

[6] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290: 2323-2326, 2000.

[7] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. IEEE TPAMI, 27(3): 328-340, 2005.

[8] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In CVPR, 2005.

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

[10] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? Vision Research, 37(23):3311–3325, 1997.

[11] E. Candès and J. Romberg. l_1-magic: Recovery of sparse signals via convex programming. http://www.acm.cal tech.edu/l1magic/, 2005.

[12] D. Donoho. For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. Comm. Pure and Applied Math., 59(6):797–829, 2006.

[13] J. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of the sparse representation and a variation approach. IEEE TIP, 14(10): 1570-1582, 2005.

[14] J. Bobin, J. Starck, J. Fadili, Y. Moudden and D. Donoho. Morphological component analysis: An adaptive thresholding strategy. IEEE TIP, 16(11):2675-2681, 2007.

[15] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. IEEE TIP, 54(11):4311-4322, 2006.

[16] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. IEEE TIP, 17(1):53–69, 2008.

[17] K. Huang and S. Aviyente. Sparse representation for signal classification. In NIPS, 2006.

[18] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. IEEE TPAMI, 31(2):210–227, 2009.

[19] M. Yang, L. Zhang, D. Zhang, and J. Yang. Metaface learning for sparse representation based face recognition. In ICIP 2010.

[20] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with l_1-graph for image analysis. IEEE Trans. on Image Processing, 19(4):858-866, 2010.

[21] L. Qiao, S. Chen, and X. Tan. Sparsity preserving projections with applications to face recognition. Pattern Recognition, 43(1):331-341, 2010.

[22] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: which helps face recognition? In ICCV 2011.

[23] A. Martinez and R. Benavente. The AR face database. CVC Technical Report, June 1998.

[24] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. Image and Vision Computing, 28(5):807-813, 2010.

[25] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE TPAMI, 23(6): 643-660, 2001.