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

Motivation and flowchart

Most of the existing LSL methods estimate sample scatter matrices directly from the original training samples. The subspace learning and image decomposition are accomplished simultaneously.

PCA | LDA → **subspace learning** | **image decomposition**

P

PX + (grey block)

(blue block) intrinsic and stable features + the noise and trivial structures (grey block)

Different contributions of different components to image recognition cannot be effectively exploited by the present methods. why don't we decompose the image first and then use the different image components to guide the subspace learning?

→ **image decomposition** → **subspace learning** → P

preserve

suppress

Dictionary Learning and Sparse Coding	A patch based dictionary $\boldsymbol{D}$ (k atoms) is learned from the training samples. By coding each image patch over $\boldsymbol{D}$, the whole training sample can be written as a linear combination of k components.
Feature Grouping	In unsupervised and supervised cases, variance and fisher ratio can be used to group the features into LDP and MDP respectively.
Linear Subspace Learning	We seek for a subspace where the MDP is preserved while the LDP is suppressed.

Dictionary learning and sparse coding

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 $\boldsymbol{d}_z^T\boldsymbol{d}_z=\mathbf{1}$, $z=1,2,\ldots,k$, such that

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

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 $\boldsymbol{D}$.

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

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

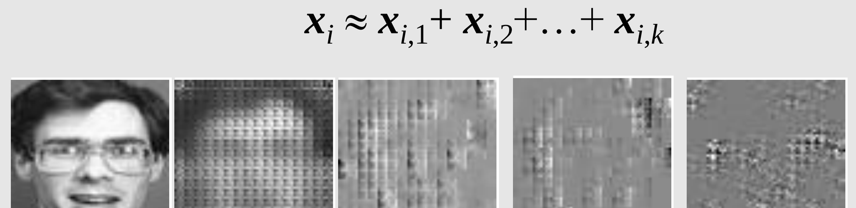
That is, each patch can be written as the summation of k components

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

where $\boldsymbol{t}_{j,z}=\boldsymbol{\alpha}_j(z)\cdot\boldsymbol{d}_z$.

By combining all the patches, each image $\boldsymbol{x}_i$ can be written as the summation of k components:

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



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

(a) is the original face image; (b) ~ (e) show the decomposed components $\boldsymbol{x}_{i,z}$ corresponding to the 1st, 11th, 21th, and 41th atoms.

Unsupervisded subspace learning

variance $\sigma_z = \frac{1}{m}\sum_{i=1}^{m}\left(x_{i,z}-\bar{x}_z\right)^2$ (a) a face image

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.

MDP $\boldsymbol{x}_i^a = \boldsymbol{x}_{i,1}+ \boldsymbol{x}_{i,2}+ \ldots+ \boldsymbol{x}_{i,z1}$ (b) more informative part — preserve

LDP $\boldsymbol{x}_i^b = \boldsymbol{x}_{i,z1+1}+\boldsymbol{x}_{i,z1+2}+\ldots+\boldsymbol{x}_{i,k}$ (c) less informative part — suppress

average energy of $\boldsymbol{P}\boldsymbol{x}_i^a$: $E_a = \frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{P}\boldsymbol{x}_i^a\|_2^2 = tr\{\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T\}$

average energy of $\boldsymbol{P}\boldsymbol{x}_i^b$: $E_b = \frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{P}\boldsymbol{x}_i^b\|_2^2 = tr\{\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T\}$

optimization objective: $J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} \frac{E_a}{E_b} = \arg\max_{\boldsymbol{P}} \frac{tr(\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T)}{tr(\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T)}$

equivalent form: $J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} tr(\boldsymbol{P}\boldsymbol{S}_a\boldsymbol{P}^T)$ s.t. $\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T=\boldsymbol{I}$

PCA: $J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} tr(\boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T)$ s.t. $\boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T=\boldsymbol{I}$

Solution: The row vector of desired $\boldsymbol{P}$ can be chosen as the p generalized eigenvectors $\boldsymbol{S}_a\boldsymbol{w}=\lambda\boldsymbol{S}_b\boldsymbol{w}$ corresponding to the first p largest eigenvalues

Supervised subspace learning

In supervised learning, 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.

Fisher ratio: $f_z = \frac{\sigma_b}{\sigma_w} = \frac{\sum_{c=1}^{C}(\bar{x}_z-\bar{x}_{z,c})^2}{\sum_{c=1}^{C}\frac{1}{m_c}\sum_{x_i\in X_c}(x_{i,z}-\bar{x}_{z,c})^2}$

Original MDP LDP

MDP within-class scatter matrices: $\boldsymbol{S}_W^a = \frac{1}{m}\sum_{c=1}^{C}\sum_{x_i\in X_c}\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}_c^a\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}_c^a\right)^T$ — suppress

MDP between-class scatter matrices: $\boldsymbol{S}_B^a = \frac{1}{m}\sum_{c=1}^{C}m_c\left(\bar{\boldsymbol{x}}_c^a-\bar{\boldsymbol{x}}^a\right)\left(\bar{\boldsymbol{x}}_c^a-\bar{\boldsymbol{x}}^a\right)^T$ — preserve

LDP scatter matrix: $\boldsymbol{S}^b = \frac{1}{m}\sum_{c=1}^{C}\sum_{x_i\in X_c}\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}^b\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}^b\right)^T$ — suppress

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

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

Solution: The row vector of desired $\boldsymbol{P}$ can be chosen as the p generalized eigenvectors $\boldsymbol{S}_B^a\boldsymbol{w}=\lambda[\alpha\boldsymbol{S}_W^a+(1-\alpha)\boldsymbol{S}_b]\boldsymbol{w}$ corresponding to the first p largest eigenvalues.

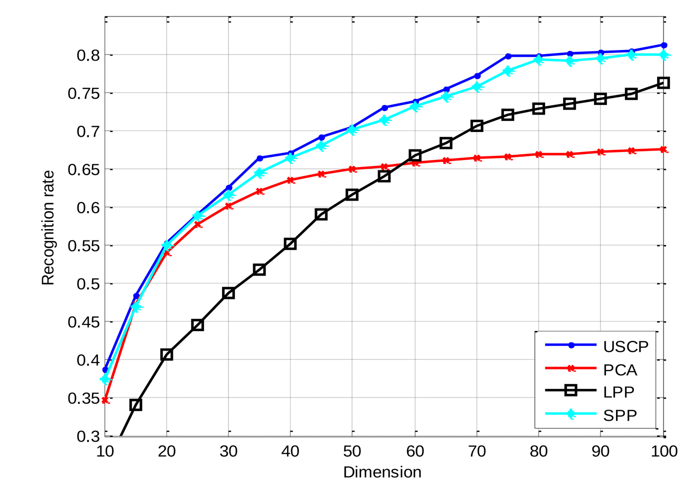
FLDA: $J_{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} \frac{tr\{\boldsymbol{P}\boldsymbol{S}_B\boldsymbol{P}^T\}}{tr\{\boldsymbol{P}\boldsymbol{S}_b\boldsymbol{P}^T\}}$ without applying sparse coding and feature grouping to the training images and let $\alpha=0$

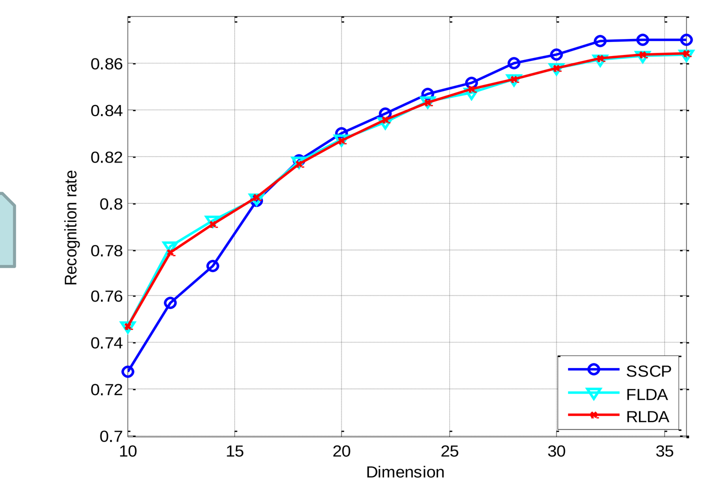
Experimental result

Databases: the AR database, the extended Yale B database and the Multi-PIE database

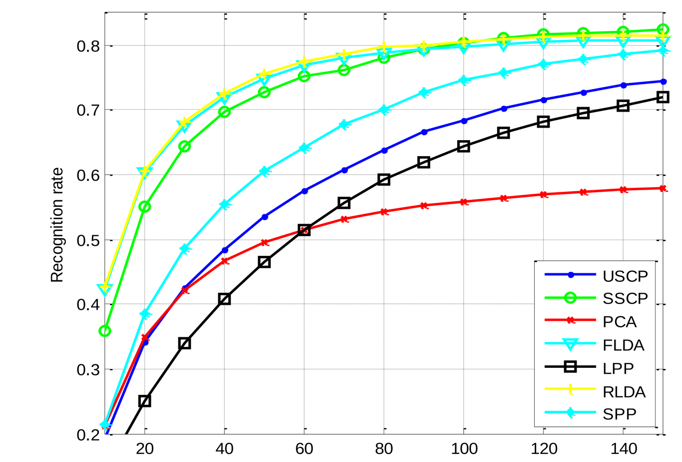
unsupervised | supervised

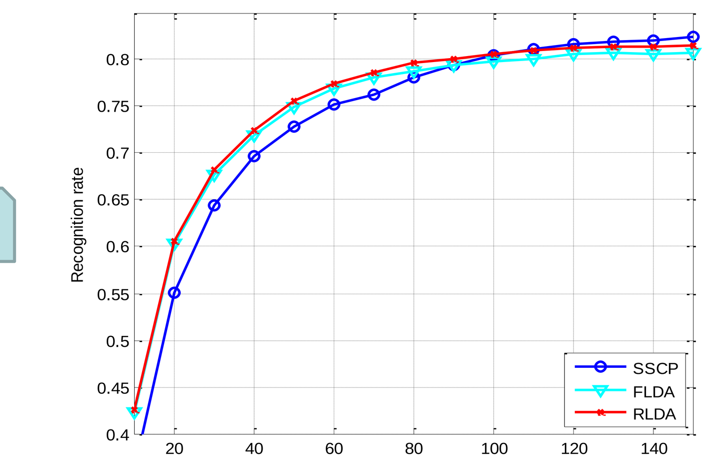
YaleB



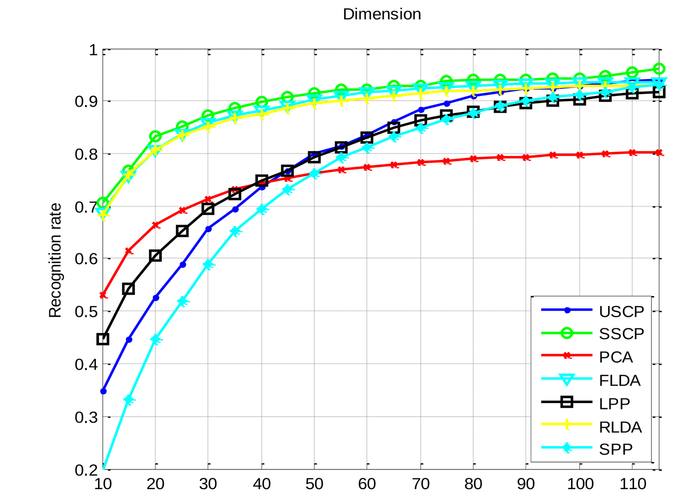


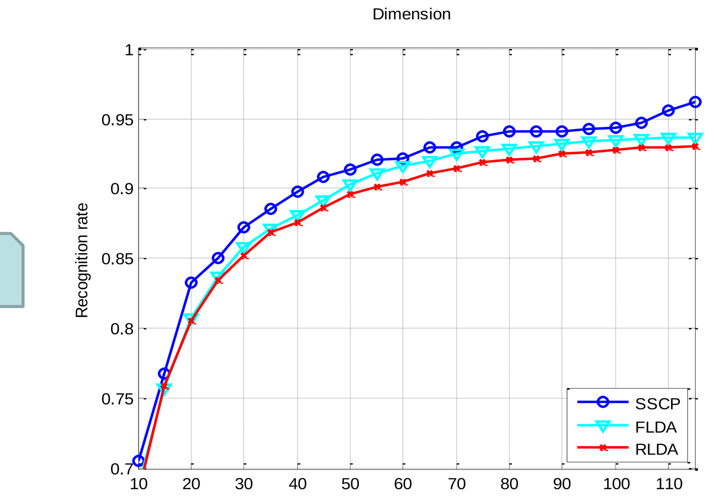
mpie





AR





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

Conclusion

In this paper, we proposed a novel linear subspace learning (LSL) method via sparse coding and feature grouping. 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 LSI methods.