Linear Discriminative Image Processing Operator Analysis

Toru Tamaki, Bingzhi Yuan, Kengo Harada, Bisser Raytchev, Kazufumi Kaneda

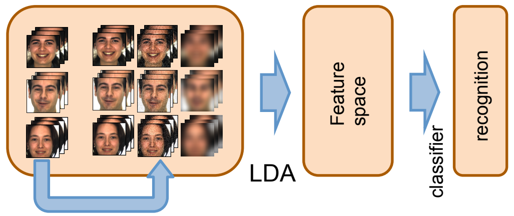
Most discriminative image processing operators (IPOs)

Goal
Find a most *discriminative* set of image processing operations for LDA.

Motivation
For a small sample size problem, many studies use an approach to increase training samples by synthetically generating new training samples. But, HOW ?

✗ Ad-hoc… ✓ *discriminatively* !

Contribution
Simultaneous estimation of **both** LDA feature space and a set of discriminative *generating matrices*.



$$x_j = G_j x$$

Generating matrices
(image processing operators)

Linear IPO + LDA = LDA with increased samples

Training sample

a **generating matrix** (an image processing operator)

increased sample $x_j = G_j x$ an original sample

average of image processing operations

Mean of class i for increased samples

$$m'_i = \frac{1}{J n_i}\sum_{j=1}^{J}\sum_{x \in \mathcal{X}_i} G_j x = \frac{1}{J}\sum_{j=1}^{J} G_j m_i = \bar{G} m_i$$

Mean of all increased samples

$$m' = \bar{G} m$$

Given $\{G_j\}$, we don't need to actually increase training samples.
But, need more memory to store…

Scatter matrices

Scatter matrix of class i for increased samples

$$S'_i = \bar{G}(S_i - R_i)\bar{G}^T + \frac{1}{J}\sum_{j=1}^{J} G_j R_i G_j^T$$

Within-class scatter matrix for increased samples

$$S'_W = \bar{G}(S_W - R_W)\bar{G}^T + \frac{1}{J}\sum_{j} G_j R_W G_j^T$$

Between-class scatter matrix for increased samples

$$S'_B = \bar{G} S_B \bar{G}^T$$

S_W, S_B are scatter matrices for original (non-increased) samples

$$R_W = \sum_{i}^{c} R_i \qquad R_i = \frac{1}{n_i}\sum_{x \in \mathcal{X}_i} x x^T$$

$y_j = A^T P^T x_j = A^T P^T G_j x$
$\tilde{m}'_i = A^T P^T \bar{G} m_i$
$\tilde{m}' = A^T P^T m' = A^T P^T \bar{G} m$

Dimensionality reduction

PCA

Covariance matrix

$$X' = \bar{G}(X - R_{\text{all}})\bar{G}^T + \frac{1}{J}\sum_{j=1}^{J} G_j R_{\text{all}} G_j^T$$

PCA projection matrix P

Feature space

scatter matrices

$$\tilde{S}'_i = A^T P^T S'_i P A$$
$$\tilde{S}'_W = A^T P^T S'_W P A$$
$$\tilde{S}'_B = A^T P^T S'_B P A$$

LDA

Rayleigh quotient

$$\frac{\mathrm{tr}(\tilde{S}'_B)}{\mathrm{tr}(\tilde{S}'_W)}$$

Generalized Eigenvalue problem

$$(P^T S'_W P)^{-1} P^T S'_B P$$

Analysis of IPO: the spectral decomposition

Q: To reduce the memory cost of generating matrices, can we use a decomposition for operators just like for images?

A: **Yes**.

Definition 1 *Let $f(x), g(x) \in L^2(R^2)$ be complex-valued 2D functions where $x \in R^2$. The inner product is defined as*

$$(f, g) \equiv \int_{R^2} f(x)\overline{g(x)}dx,$$

where $\bar{g}$ is the complex conjugate of g. An operator $G : f \mapsto g$ is linear if it satisfies $G(af + bg) = aG(f) + bG(g)$, $\forall a, b \in R$. G^* is the adjoint operator of G if it satisfies $(Gf, g) = (f, G^* g)$.

Proposition 1 *A filtering is defined as*

$$Gf(x) = \int G(x, y) f(y) dy,$$

where the kernel is symmetric $G(x, y) = G(y, x)$ and real valued. G is an Hermite operator which satisfies $G^* = G$.

Proposition 2 *A geometric (affine) transformation G is defined as*

$$Gf(x) = |A|^{1/2} f(Ax + t),$$

where $|A| \neq 0$. G is a unitary operator which satisfies $G^* G = I$.

Corollary 1 *Filtering or geometric transformation operators G are normal operators which satisfy $G^* G = G G^*$.*

$$G = \sum \lambda_i P_i$$

A normal operator can be decomposed into projection operators!

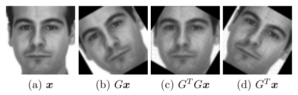
But, is it feasible for a generating *matrix*? Yes!

Is a fltering Hermite?

Almost symmetric $||G - G^T|| < 10^{-6}$

Is a geometric trans. Unitary?

Transpose is *apparently* inverse



Are eigenvalues complex? Use Hermite decomposition.

$$G = H_1 + iH_2 \qquad i = \sqrt{-1}$$

an operator / two Hermite operators (which have real eigenvalues)

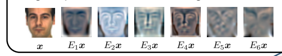
$$H_1 = \frac{G + G^T}{2}, \quad H_2 = \frac{G - G^T}{2i}$$

So, two step approximation.

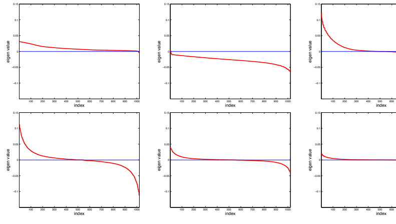
$$G \simeq \sum_j a_j E_j = \sum_j a_j (H_{1j} + iH_{2j}) \simeq \sum_j a_j \sum_i (\lambda_{1ji} P_{1ji} + i\lambda_{2ji} P_{2ji})$$

Examples

Eigenoperators transorm images to variants.

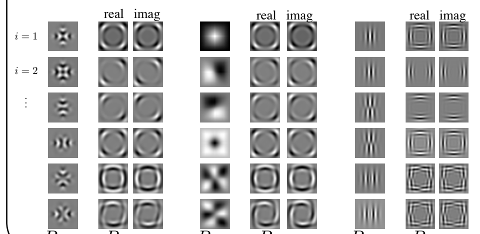


Real eigenvalues can be small so that we can compress them.



(a) H_{11}, H_{21} (b) H_{12}, H_{22} (c) H_{13}, H_{23}

Eigenprojections of eigenoperators transorm images to … wavelets?



$P_{11i}x$ $P_{21i}x$ $P_{12i}x$ $P_{22i}x$ $P_{13i}x$ $P_{23i}x$

LDA + IPO = LDIPOA: find a set of discriminative IPOs

At each step k, estimate a single generating matrix represented as a linear combination.

$$G^{(k)} = \sum_j^J \alpha_j^{(k)} G_j \qquad (\alpha_1^{(k)}, \alpha_2^{(k)}, \ldots, \alpha_J^{(k)})^T = \alpha^{(k)}$$

The proposed algorithm iteratively estimates

- α (coeffs. of generating matrices)
- P (PCA)
- A (LDA)

at the same time.

Algorithm 1 LDIPOA

1: Compute PCA P and LDA A. $G^0 \leftarrow I$.
2: **for** $k = 1, \ldots,$ **do**
3: **repeat**
4: α step: $\alpha^{(k)} = \mathrm{argmax}_\alpha E(A, P, \alpha)$
5: PCA step: Compute P with $\alpha^{(k)}$.
6: LDA step: $A = \mathrm{argmax}_A E(A, P, \alpha^{(k)})$
7: **until** E converges
8: **end for**

The Rayleigh quotient

alpha step

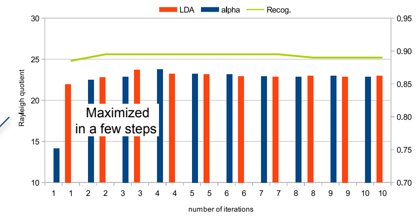
$$\tilde{S}_W^{\prime(k)} = D^T \tilde{G}^{(k)} (S_W - R_W) \tilde{G}^{(k)T} D + \frac{1}{k+1}\sum_{l=0}^{k} D^T G^{(l)} R_W G^{(l)T} D$$
$$\tilde{S}_B^{\prime(k)} = D^T \tilde{G}^{(k)} S_B \tilde{G}^{(k)T} D$$
$$\tilde{G}^{(k)} = \frac{1}{k+1}\sum_{l=0}^{k} G^{(l)} \quad D = PA$$

PCA step

$$X'^{(k)} = \tilde{G}^{(k)} (X - R_{\text{all}}) \tilde{G}^{(k)T} + \frac{1}{k+1}\sum_{l=0}^{k} G^{(l)} R_{\text{all}} G^{(l)T}$$

LDA step

$$S_W^{\prime(k)} = \tilde{G}^{(k)} (S_W - R_W) \tilde{G}^{(k)T} + \frac{1}{k+1}\sum_{l=0}^{k} G^{(l)} R_W G^{(l)T}$$
$$S_B^{\prime(k)} = \tilde{G}^{(k)} S_B \tilde{G}^{(k)T}$$



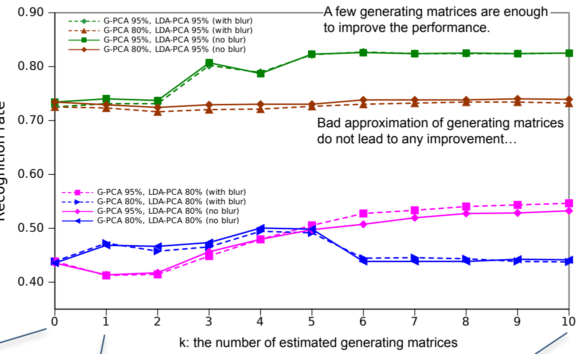
Experiments with FERET dataset

Size of images: 32x32
Size of generating matrices: 1024x1024
Number of classes: 1001 (fa)
Training images per class: 1 (fa)
Test images per class: 1 (fb)

Eigen-generating matrices: 96
Initial generating matrices: 567
(3 scaling, 7 rotations, 3 Gaussian and 9 motion blurs)

Classifiers: nearest neighbor

PCA rates: 80% and 95%
for eigen-generating matrices (G-PCA)
for PCA step (LDA-PCA)



No generating matrices are used (normal LDA)

1 generating matrix is used to increase the dataset double.

10 generating matrices are used to increase the dataset 11 times.