The Role of Principal Angles in Subspace Classification

Jiaji Huang, Qiang Qiu, Robert Calderbank

Department of Electrical Engineering, Duke University, Durham, NC 27708

Abstract

- Union of subspaces models a wide range of data
- Classification: each class is modeled as a subspace
- How the subspace geometry influences classification performance?
- How to learn "better" feature for classification?

Geometric Framework

Two subspaces $\mathcal{U}, \mathcal{V} \subset \mathbb{R}^n$, dimension: $\ell, s (\ell \leq s)$.
$\mathbf{U}, \mathbf{V}$: orthonormal bases for $\mathcal{U}$ and $\mathcal{V}$.

- **Principal angles:**

$$\theta_1 = \min_{\mathbf{u}_1 \in \mathcal{U}, \mathbf{v}_1 \in \mathcal{V}} \arccos\left(\frac{\mathbf{u}_1^\top \mathbf{v}_1}{\|\mathbf{u}_1\|\|\mathbf{v}_1\|}\right),$$

$$\vdots$$

$$\theta_j = \min_{\substack{\mathbf{u}_j \in \mathcal{U}, \mathbf{v}_j \in \mathcal{V} \\ \mathbf{u}_j \perp \mathbf{u}_1, \ldots, \mathbf{u}_{j-1} \\ \mathbf{v}_j \perp \mathbf{v}_1, \ldots, \mathbf{v}_{j-1}}} \arccos\left(\frac{\mathbf{u}_j^\top \mathbf{v}_j}{\|\mathbf{u}_j\|\|\mathbf{v}_j\|}\right), \; j = 2, \ldots, \ell.$$

Principal vectors $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ and $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$

- $\dim(\mathcal{U} \cap \mathcal{V})$ = multiplicity of zero principal angle

The MAP Classifier for GMM

Setup: Binary Classification,

$$p(\mathbf{x}) = \frac{1}{2}p(\mathbf{x}|1) + \frac{1}{2}p(\mathbf{x}|2),$$

where $p(\mathbf{x}|i) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_i)$, $i = 1, 2$.
$\boldsymbol{\Sigma}_i = \mathbf{U}_i \boldsymbol{\Lambda}_i \mathbf{U}_i^\top + \sigma^2 \mathbf{I}$.
$\dim(\mathbf{U}_1) = \dim(\mathbf{U}_2) = d$, $\dim(\mathbf{U}_1 \cap \mathbf{U}_2) = r$.
principal angles between $\mathbf{U}_1, \mathbf{U}_2$ are $\theta_1, \ldots, \theta_d$.

Classification error

$$P_e = \frac{1}{2}\int \min\{p(\mathbf{x}|1), p(\mathbf{x}|2)\}\, d\mathbf{x}$$

Bhattacharyya Bound

$$P_e \leq \frac{1}{2}e^{-K}, \text{ where } K = \frac{1}{2}\ln \frac{\det\left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)}{\sqrt{\det \boldsymbol{\Sigma}_1 \cdot \det \boldsymbol{\Sigma}_2}}$$

high SNR

As $\sigma \to 0$,

$$P_e \leq c_1 (\sigma^2)^{\frac{d-r}{2}} \left(\prod_{i=r+1}^{d} \sin^2\theta_i\right)^{-\frac{1}{2}} + o\left((\sigma^2)^{\frac{d-r}{2}}\right)$$

Low SNR

When σ^2 is sufficiently large,

$$P_e \leq \frac{1}{2}\exp\left\{-\frac{1}{\sigma^4}\left(c_2 - \frac{1}{8}\lambda_{1,1}\lambda_{2,1}\sum_{i=1}^{d}\cos^2\theta_i\right)\right\}$$

Moderate SNR

If $\frac{p}{c(p)} \leq \frac{\lambda_{1,i}}{\sigma^2}, \frac{\lambda_{2,i}}{\sigma^2} \leq p$, then

$$P_e \leq \frac{c_3}{2}\exp\left\{\frac{\lambda_{1,1}\lambda_{2,1}}{4\sigma^4(1+p)^2}\sum_i \cos^2\theta_i - c_4(2d - r)\right\}$$

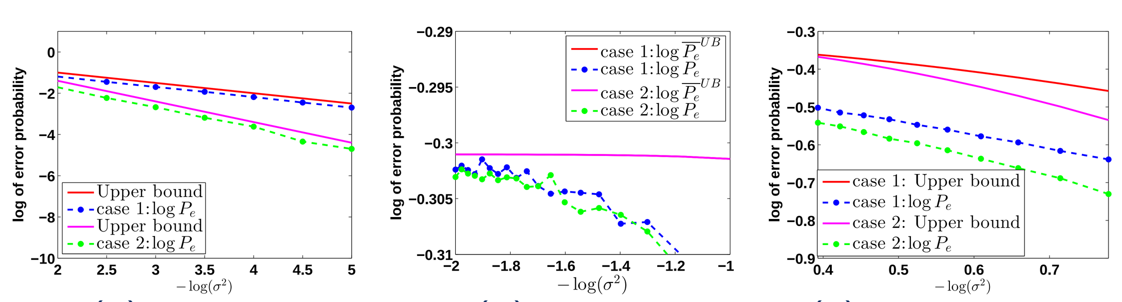
Numerical Example

case 1

$$\mathbf{U}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^\top \quad \mathbf{U}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^\top.$$

case 2

$$\mathbf{U}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^\top \quad \mathbf{U}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix}^\top.$$



(a) high SNR (b) low SNR (c) moderate SNR

Extending GMM

Setup:

$$p(\mathbf{x}|i) = \int \mathcal{N}(\mathbf{x}; \mathbf{U}_i\boldsymbol{\alpha}, \sigma^2\mathbf{I})p_i(\boldsymbol{\alpha})d\boldsymbol{\alpha}, \; i = 1, 2.$$

Nearest Subspace Classifier (**NSC**):

$$\|\mathbf{U}_1^\top \mathbf{x}\|^2 \underset{''2''}{\overset{''1''}{\gtrless}} \|\mathbf{U}_2^\top \mathbf{x}\|^2$$

NSC performance Bound

As $\sigma^2 \to 0$, the classification error is upper bounded as

$$P_e \leq \int E(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)\frac{p_1(\boldsymbol{\alpha}) + p_2(\boldsymbol{\alpha})}{2}d\boldsymbol{\alpha}$$

where $E(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2) = \frac{1}{2}\exp\left[-\frac{\left(\sum_{i=1}^{d}\sin^2\theta_i\alpha_i^2\right)^2}{8\sigma^2\sum_{i=1}^{d}\sin^2\theta_i(\alpha_i^2 + \sigma^2)}\right]$.

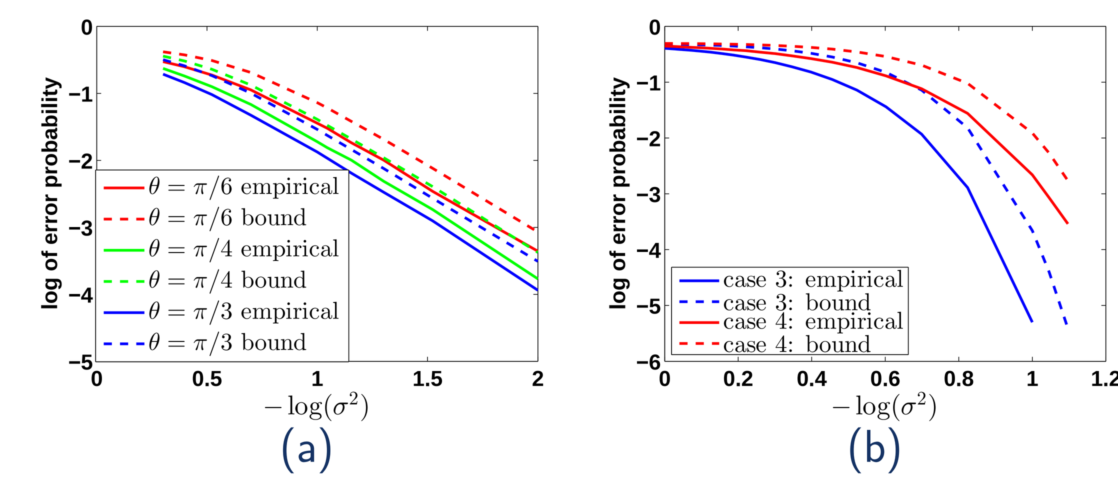
Numerical Example

(a) $\mathbf{U}_1 = \left[\mathbf{I}_2, \mathbf{0}_4\right]^\top$, $\mathbf{U}_2 = \begin{bmatrix} \cos\theta & 0 & 0 & 0 & \sin\theta & 0 \\ 0 & \cos\theta & 0 & 0 & 0 & \sin\theta \end{bmatrix}^\top$,

(b) $\mathbf{U}_1 = \left[\mathbf{I}_2, \mathbf{0}_4\right]^\top$

$$\mathbf{U}_2 = \begin{bmatrix} \cos(\pi/6) & 0 & 0 & 0 & \sin(\pi/6) & 0 \\ 0 & \sin(\pi/6) & 0 & 0 & 0 & \cos(\pi/6) \end{bmatrix}^\top$$

case 3: $|\alpha_1| < |\alpha_2|$ case 4: $|\alpha_1| > |\alpha_2|$



(a) (b)

(a)Principal angles ↗ classification error↘; (b) Proportionate of signal energy to larger principal angles, classification error↘

Feature Learning

Tunable Recognition Adapted to Intra-class Target (**TRAIT**)

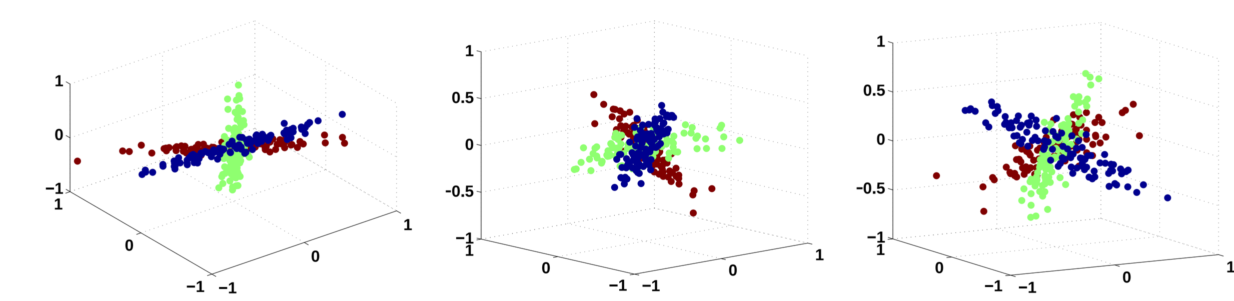
$$\min_{\mathbf{A} \in \mathbb{R}^{m \times n}} \frac{1}{N^2}\|(\mathbf{AX})^\top(\mathbf{AX}) - \mathbf{T}\|_F^2,$$

where $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_c]$ and $\mathbf{T}$ is block diagonal.
Example: $\mathbf{T} = \text{diag}\{\mathbf{X}_1\mathbf{X}_1^\top, \ldots, \mathbf{X}_c\mathbf{X}_c^\top\}$

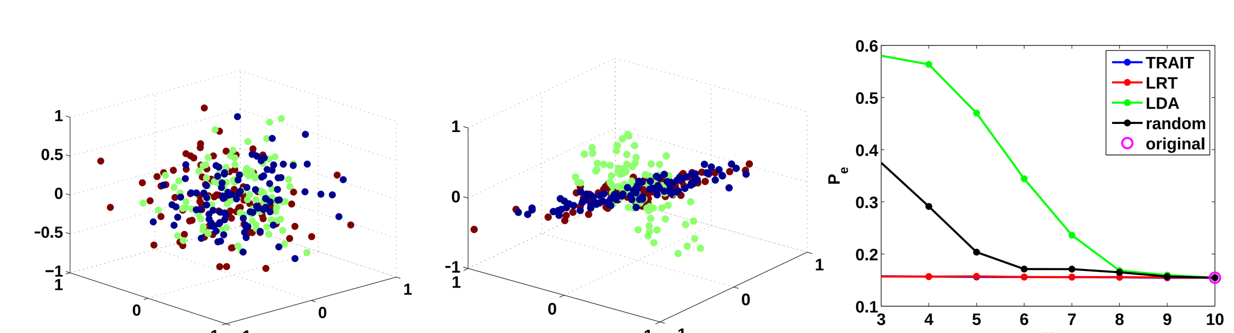
Experiments

TRAIT enlarges principal angles

classify three subspaces: $d = 1$



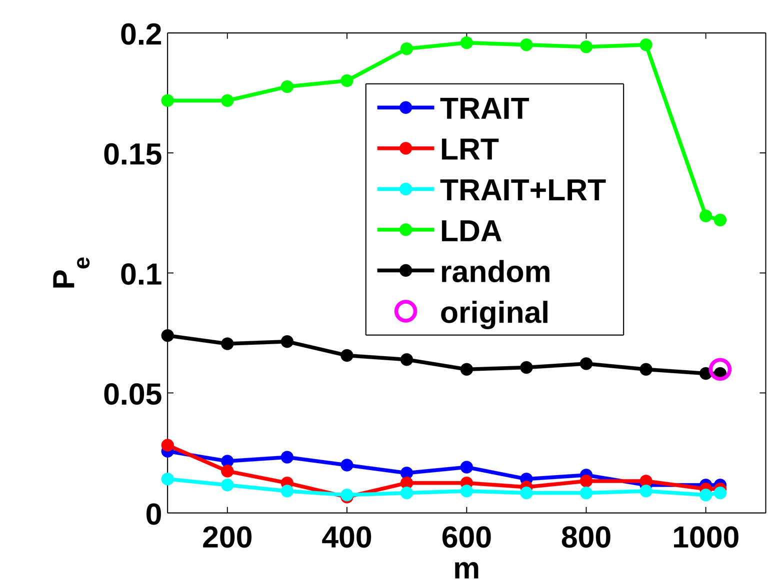
(a) Original, angles: 66.1°, 21.8°, 70.0° (b) TRAIT, angles: 87.8°, 72.9°, 88.7° (c) LRT [1], angles: 86.0°, 77.1°, 83.0°



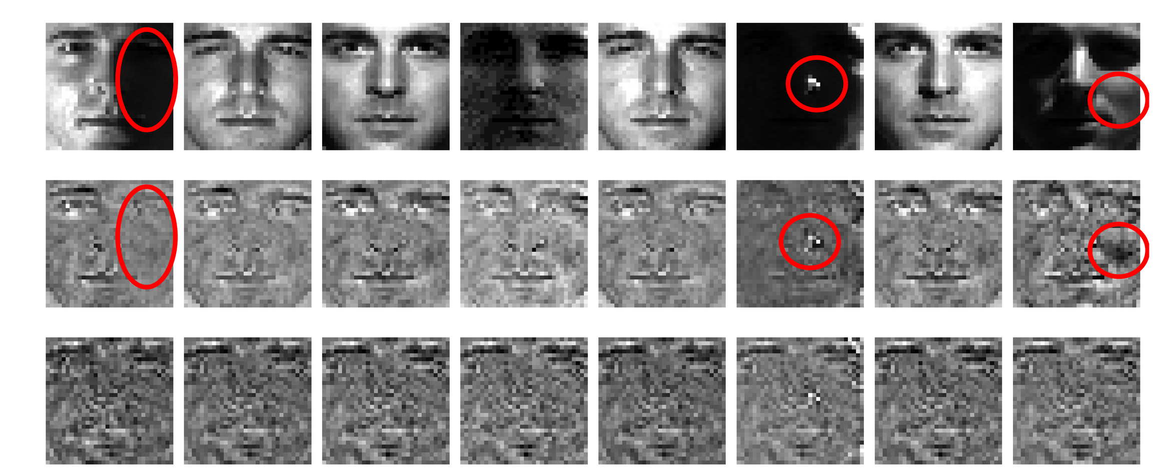
(d) LDA, angles: 47.0°, 86.4°, 87.8° (e) Random, angles: 72.4°, 2.5°, 73.0° (f) Classification error with different feature dimensions

TRAIT preserves intra-class structure

Yale B Human frontal face images

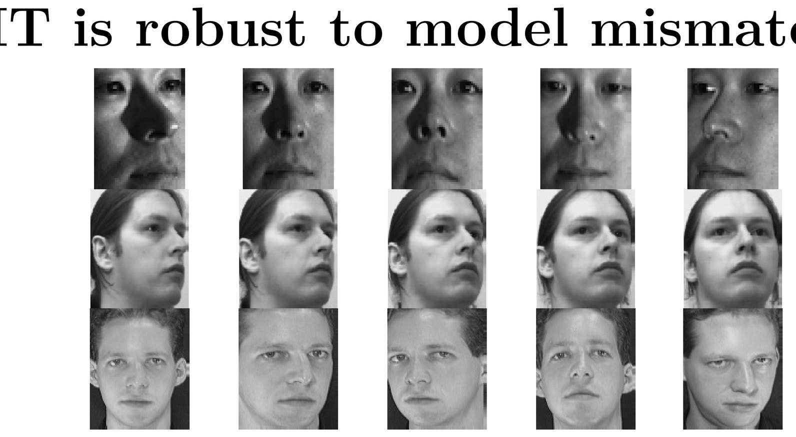


(a) classification accuracy on yaleB



(b) Comparison of original images (top) with TRAIT transformed images (middle) and LRT transformed images (bottom)

TRAIT is robust to model mismatch



From top to bottom row: subjects in PIE, UMIST and ORL database, taken under different poses

NSC accuracy on original and 1000 dimensional (compressed) extracted features

	PIE	UMIST	ORL
Original	74.57%	96.14%	95.50%
random	72.14%	95.44%	94.50%
LDA	40.10%	84.91%	92.00%
LRT	70.80%	96.84%	95.00%
TRAIT	**76.11%**	**97.90%**	**97.00%**

Reference

[1] Q. Qiu, G. Sapiro, Learning Transformations for Clustering and Classification, Journal of Machine Learning Research, 16:187-225, 2015.