Dimensionality Reduction

Lecturer: Javier Hernandez Rivera

30th September 2010

MAS 622J/1.126J: Pattern Recognition & Analysis

"A man's mind, stretched by new ideas, may never return to it's original dimensions"
Oliver Wendell Holmes Jr.

Outline

- Before
  - Linear Algebra
  - Probability
  - Likelihood Ratio
  - ROC
  - ML/MAP
- Today
  - Accuracy, Dimensions & Overfitting (DHS 3.7)
  - Principal Component Analysis (DHS 3.8.1)
  - Fisher Linear Discriminant/LDA (DHS 3.8.2)
  - Other Component Analysis Algorithms

Accuracy, Dimensions & Overfitting

- Classification accuracy depends upon the dimensionality?

$$P(e) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2}\, du,$$

$$r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2).$$

$$r^2 = \sum_{i=1}^{d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2 \quad \text{if} \quad \Sigma = diag(\sigma_1^2, ..., \sigma_d^2)$$

- Complexity (computational/temporal)

Big O notation $f(x) = O(g(x))$ as $x \to \infty$

$$|f(x)| \le M|g(x)| \text{ for all } x > x_0.$$

$$f(x) = a_0 + a_1 x + a_2 x^2; \text{ in that case we have } f(x) = O(x^2)$$

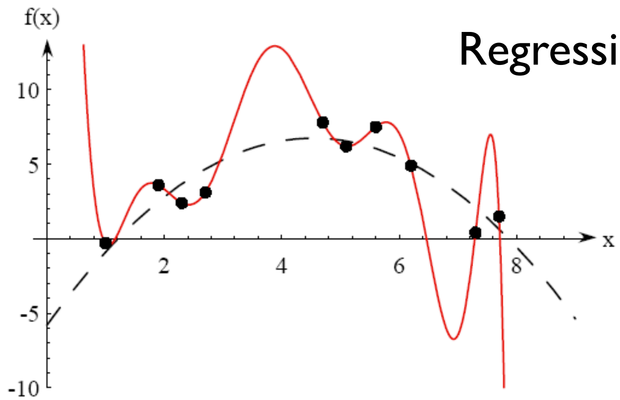
O(c) < O(log n) < O(n) < O(n log n) < O(n^2) < O(n^k) < O(2^n) < O(n!) < O(n^n)

Training $$g(\mathrm{x}) = -\frac{1}{2}(\mathrm{x} - \underset{\uparrow}{\overset{O(dn)}{\hat{\mu}}})^t \overbrace{\widehat{\Sigma}^{-1}}^{O(nd^2)} (\mathrm{x} - \hat{\mu}) - \overbrace{\frac{d}{2} \ln 2\pi}^{O(1)} - \overbrace{\frac{1}{2} \ln |\widehat{\Sigma}|}^{O(d^2 n)} + \overbrace{\ln P(\omega)}^{O(n)}$$

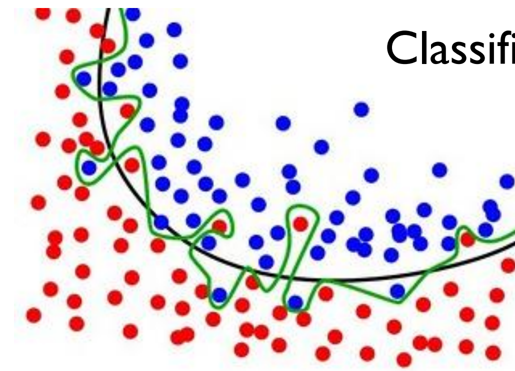
Testing $(\mathrm{x} - \hat{\mu})$, an $O(d)$

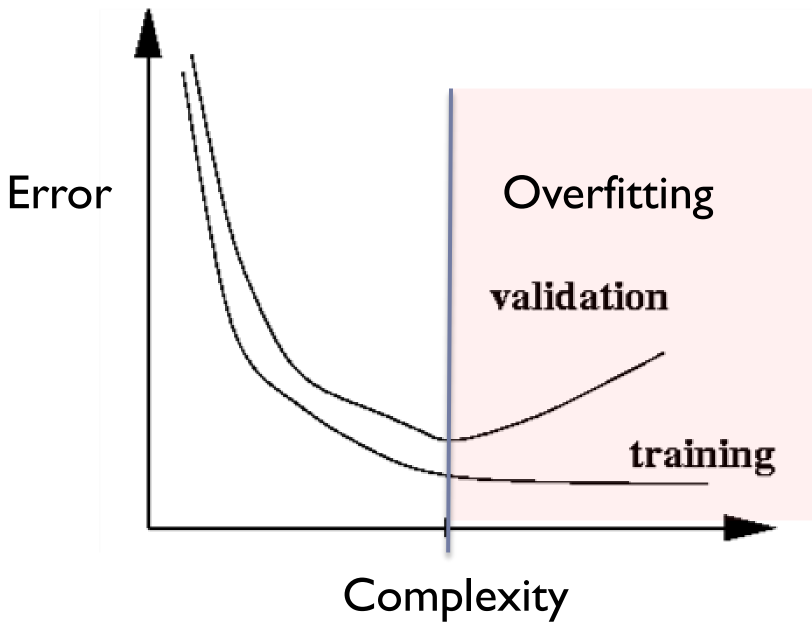
Overfitting

Regression



Classification

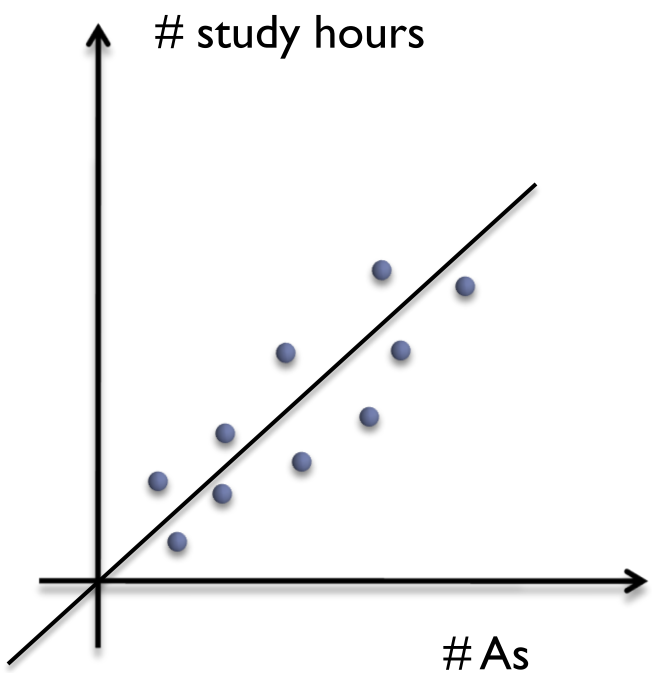


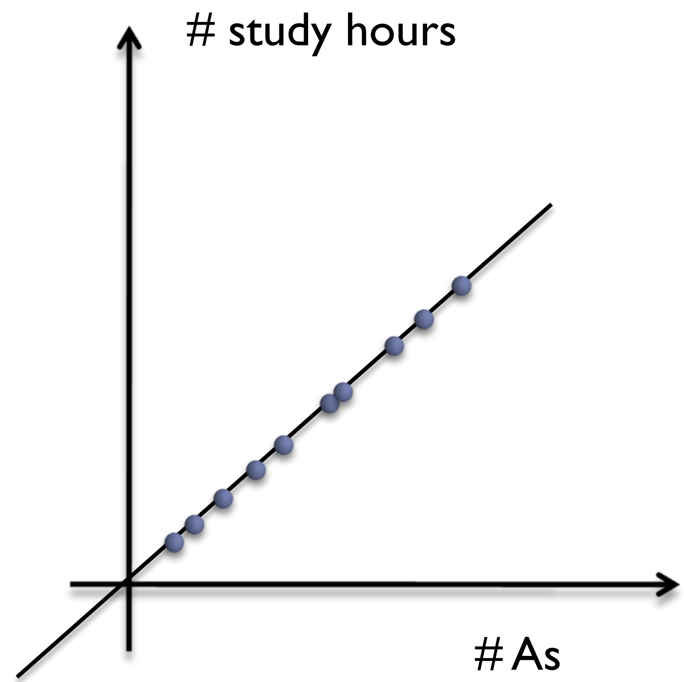


Dimensionality Reduction

- Optimal data representation

  2D: Evaluating students' performance

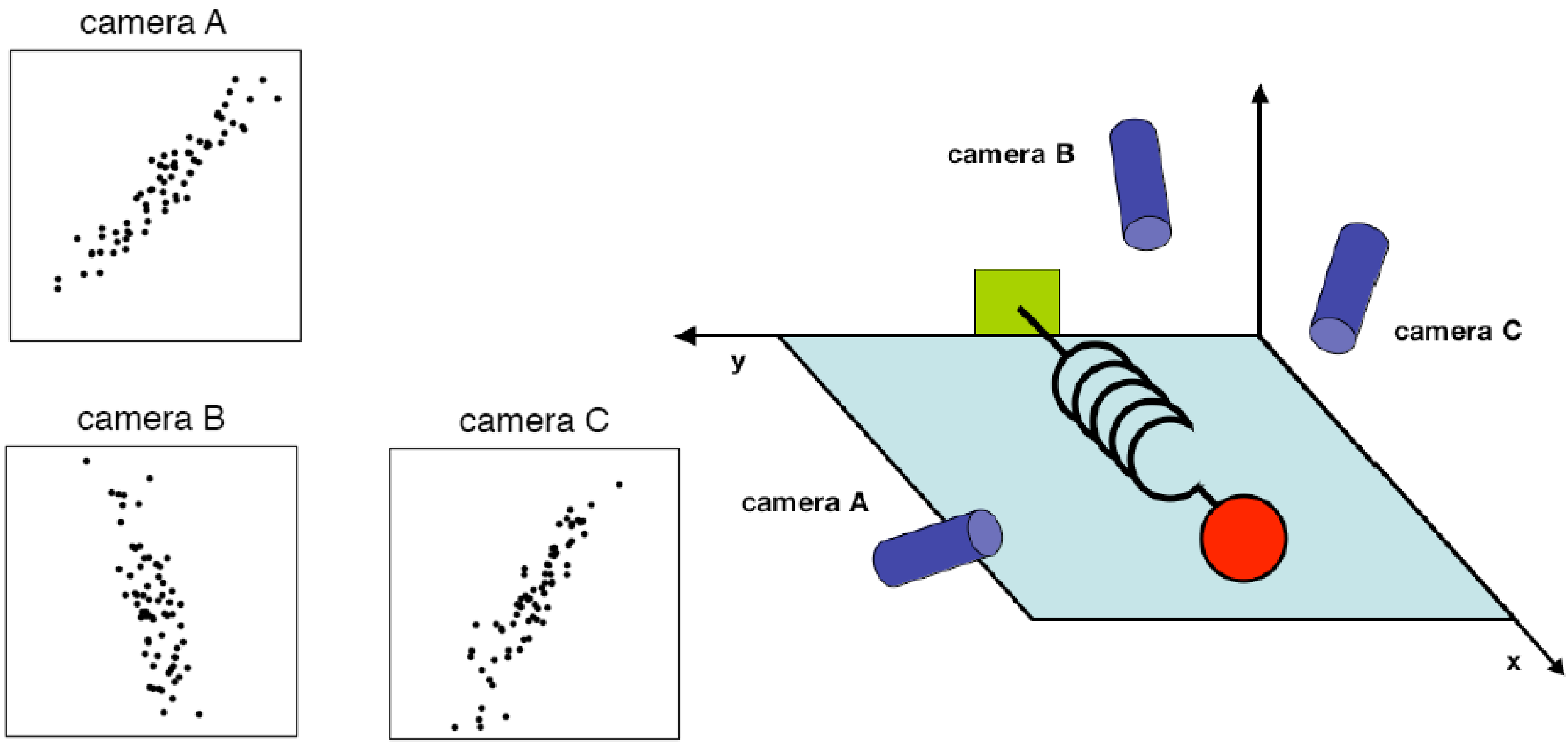




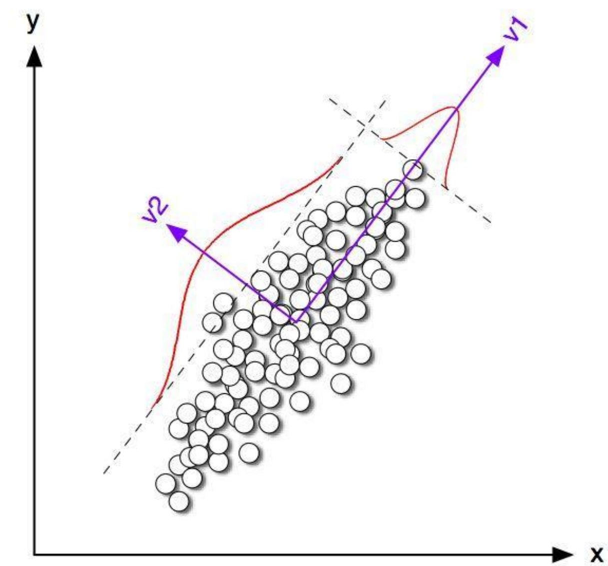
Unnecessary complexity

Dimensionality Reduction

- Optimal data representation
  - 3D: Find the most informative point of view



Principal Component Analysis



- Assumptions for new basis:
  - Large variance has important structure
  - Linear projection
  - Orthogonal basis

$$w_j^T w_i = 1 \quad i = j \qquad w_j^T w_i = 0 \quad i \neq j$$

$$\mathbf{Y} = \mathbf{W^T X}$$

$\mathbf{X} \in \mathbf{R^{d,n}}$ d dim, n samples

x_{ij} dim i of sample j

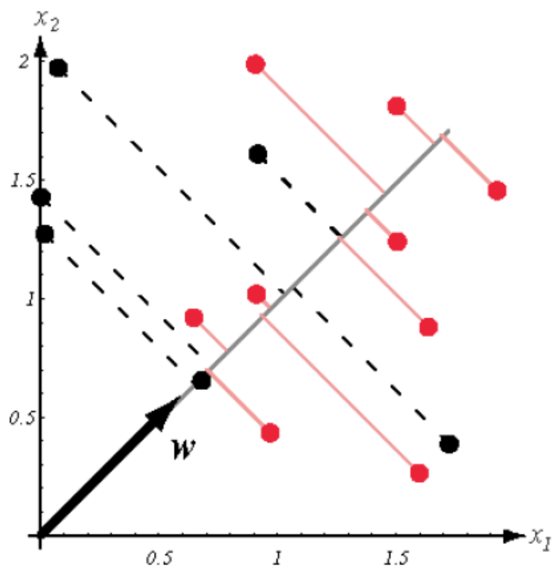
$\mathbf{Y} \in \mathbf{R^{k,n}}$ projected data

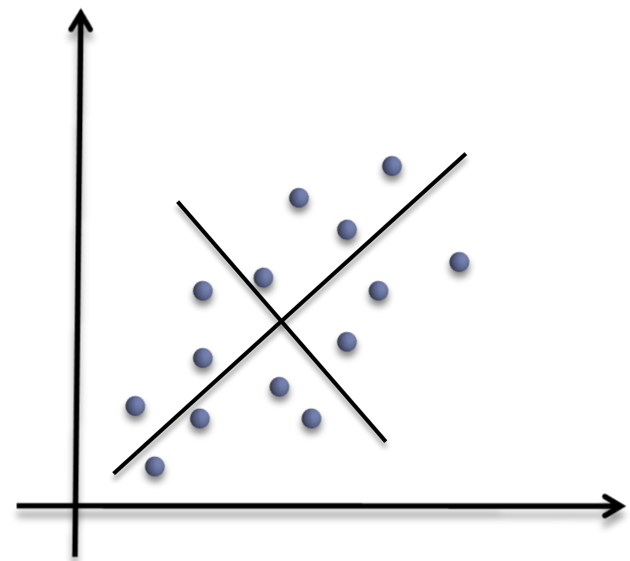
$\mathbf{W^T} \in \mathbf{R^{k,d}}$ k basis of d dim

Principal Component Analysis

- Optimal W?
  1. Minimize projection cost (Pearson, 1901)

$$min||x - \hat{x}||$$



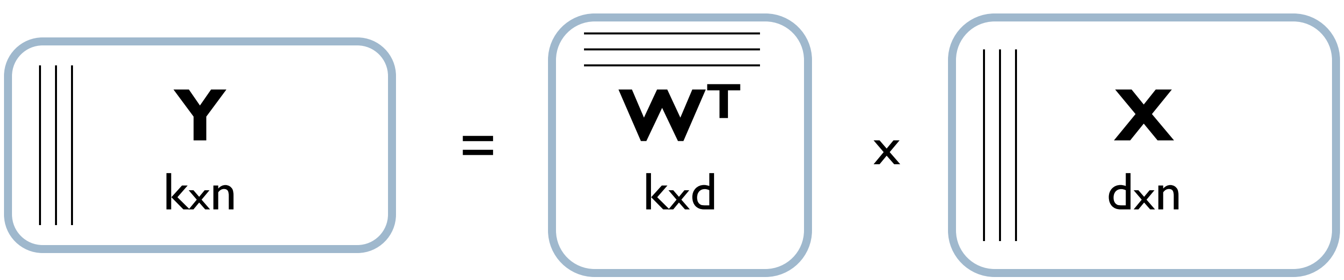


  2. Maximize projected variance (Hotelling, 1933)

Principal Component Analysis

Covariance Algorithm

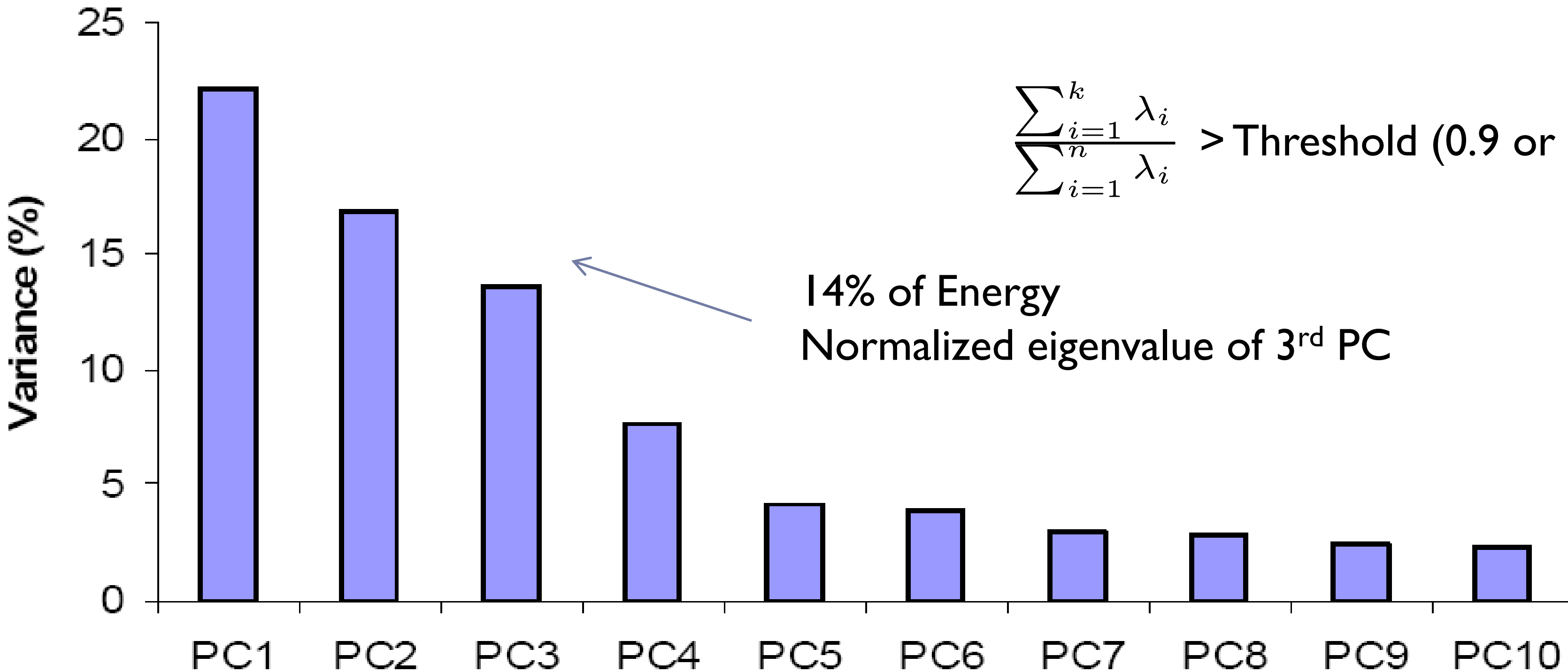
1. Zero mean: $x_{ij} = x_{ij} - \mu_i$ $\qquad \mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}$
2. Unit variance: $x_{ij} = \frac{x_{ij}}{\sigma_j}$ $\qquad \sigma_i = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ij} - \mu_i)^2$
3. Covariance matrix: $\Sigma = \mathbf{X}\mathbf{X}^{\mathbf{T}}$
4. Compute eigenvectors of $\sum$: W^T
5. Project X onto the k principal components



k < d

Principal Component Analysis

- How many PCs? $\underset{k\times n}{\mathbf{Y}} = \underset{k\times d}{\mathbf{W}}\underset{d\times n}{\mathbf{X}}$



$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{n}\lambda_i}$ > Threshold (0.9 or 0.95)

14% of Energy
Normalized eigenvalue of 3rd PC

There are d eigenvectors
Choose first p eigenvectors based on their eigenvalues
Final data has only k dimensions

PCA Applications: EigenFaces

Images from faces:

- Aligned
- Gray scale
- Same size
- Same lighting

$\Rightarrow$ **X** dxn $\xrightarrow{\text{eig vect } (X X^T)}$ **W** dxd

d = # pixels/image

n = # images

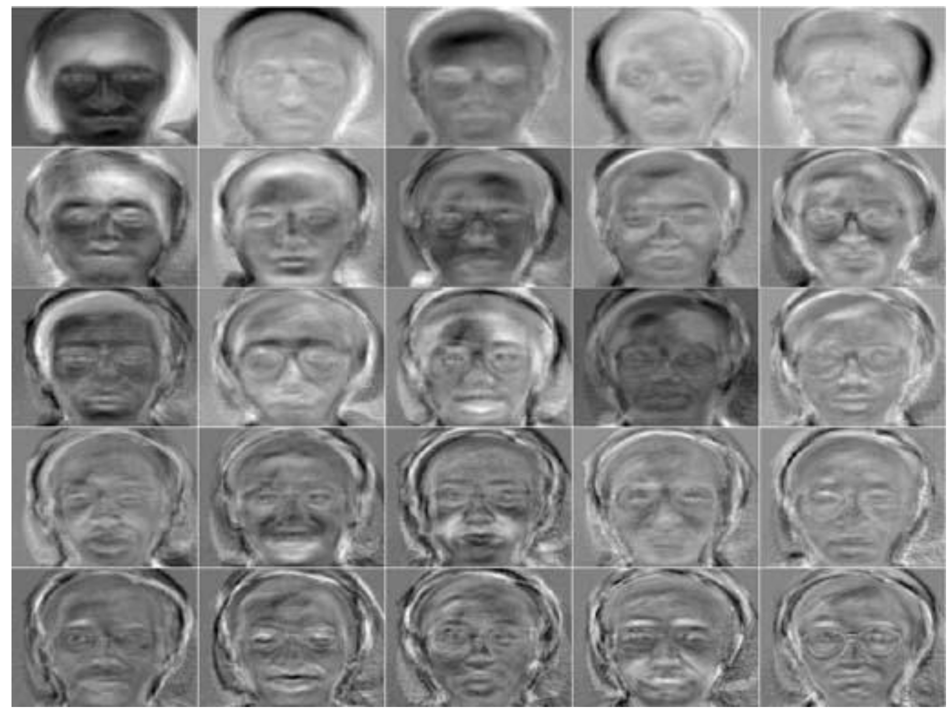
eigenFaces

Sirovich and Kirby, 1987
Matthew Turk and Alex Pentland, 1991

PCA Applications: EigenFaces

Faces

EigenFaces

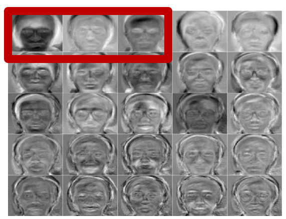


- Eigenfaces are standardized face ingredients
- A human face may be considered to be a combination of these standard faces

PCA Applications: EigenFaces

Dimensionality Reduction

W **x**



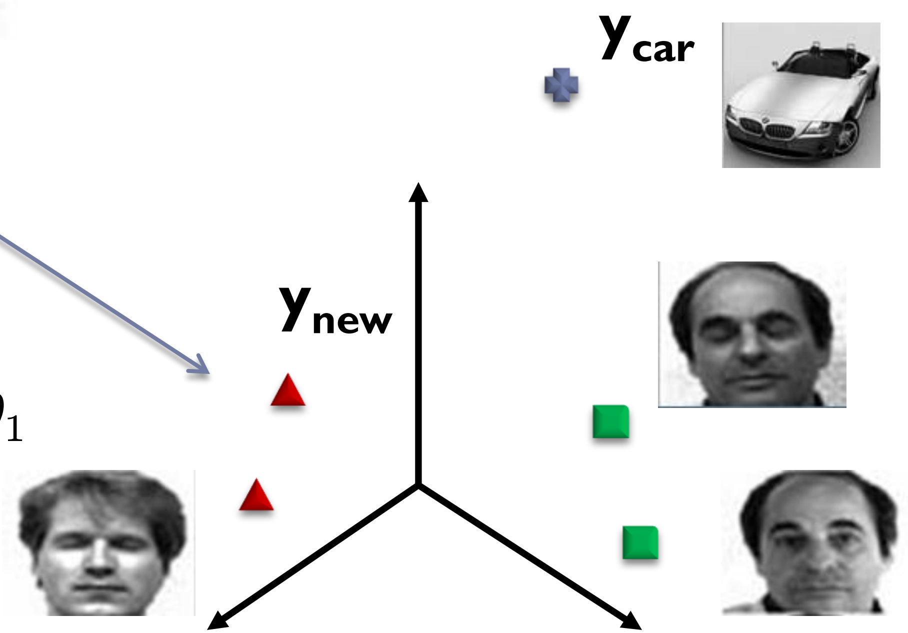
x

Outlier Detection

$$argmax_c \|y_{car} - y^c\| > \theta_2$$

Classification

$$argmin_c \|y_{new} - y^c\| < \theta_1$$



PCA Applications: EigenFaces

Reconstruction

x  = y_1 **w₁**  + y_2 **w₂**  + … + y_k **w_k** 

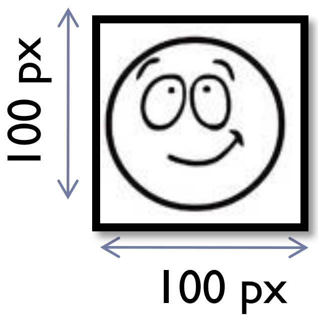
Reconstructed images (adding 8 PC at each step)

Reconstructing with partial information

Principal Component Analysis

- Any problems?

  Let’s define a samples as an image of100x100 pixels



d = 10,000 → $X X^T$ is 10,000 x 10,000

→ $O(d^2 d^3)$

computing eigenvectors becomes infeasible

- Possible solution (just if $n << d$)

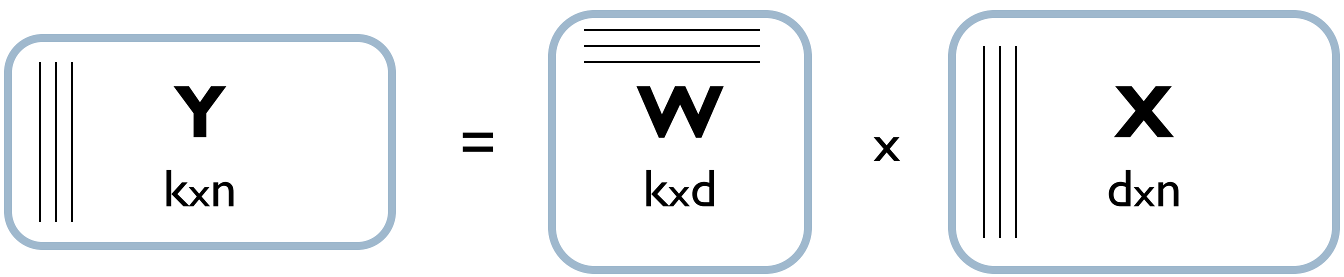
  Use $W^T = (XE)^T$

  where $E \in R^{n,k}$ are k eigenvectors of $X^T X$

Principal Component Analysis

SVD Algorithm

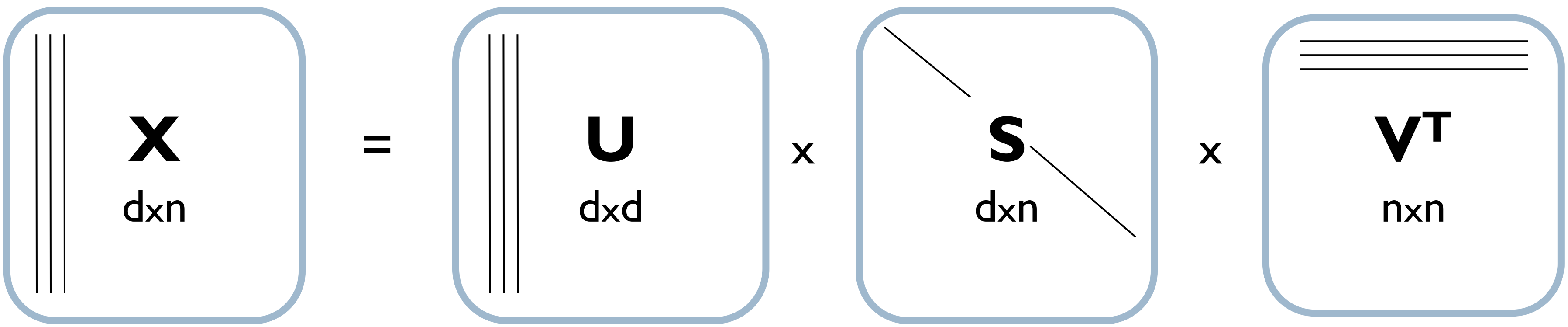
1. Zero Mean: $x_{ij} = x_{ij} - \mu_i$ $\qquad \mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}$
2. Unit Variance: $x_{ij} = \frac{x_{ij}}{\sigma_j}$ $\qquad \sigma_i = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ij} - \mu_i)^2$
3. Compute SVD of X: $\mathbf{X} = \mathbf{U\Sigma V^T}$
4. Projection matrix: $\mathbf{W} = \mathbf{U}$
5. Project data



k <= n < d

Why W = U?

Principal Component Analysis



Data
Columns are data points

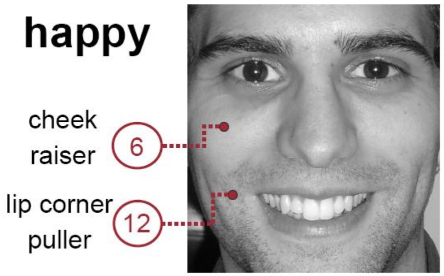
Right Singular Vectors
Columns are eigenvectors of XX^T

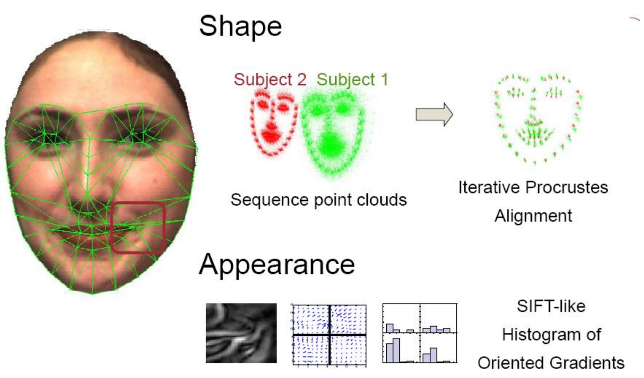
Singular Values
Diagonal matrix of sorted values

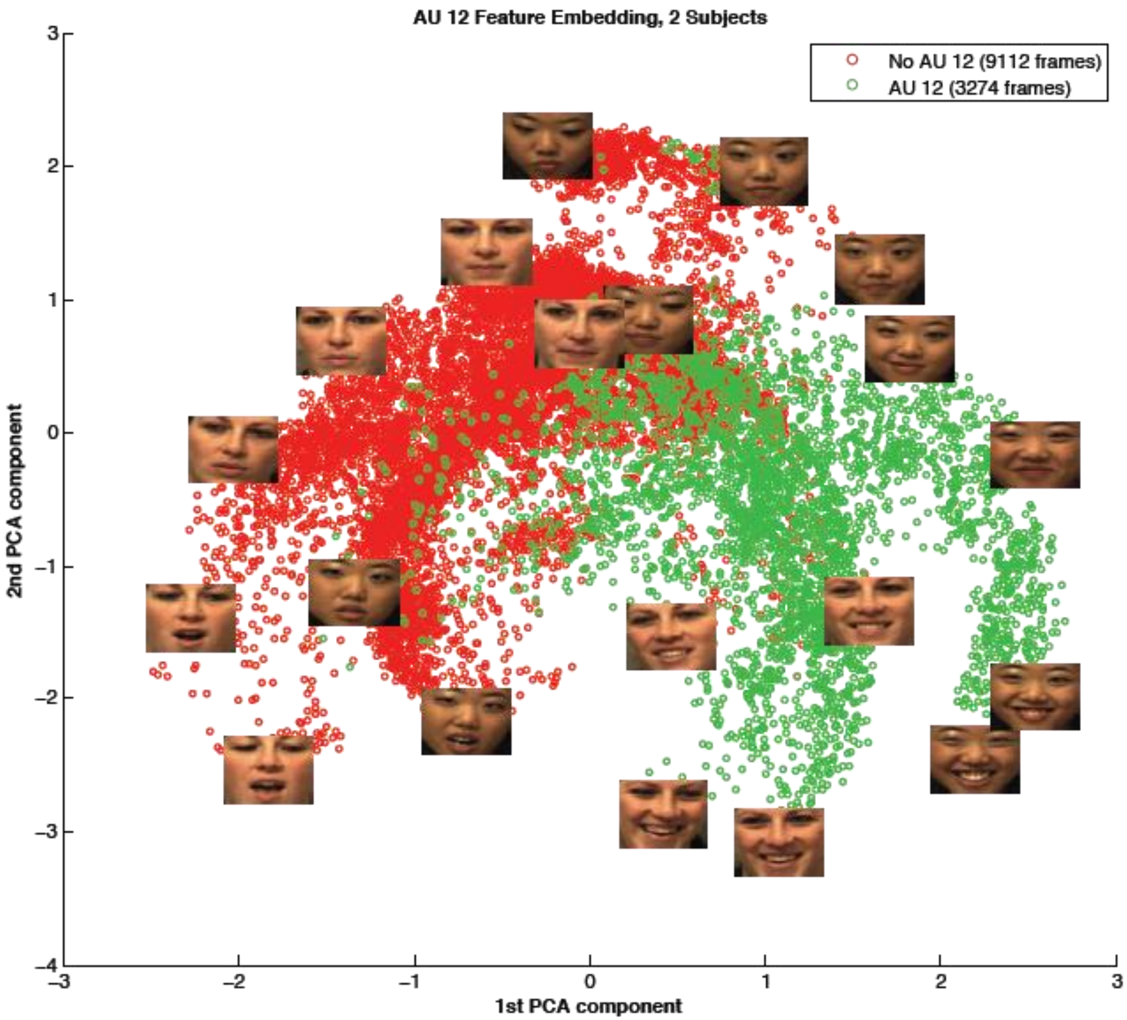
Left Singular Vectors
Rows are eigenvectors of X^TX

MATLAB: [U S V] = svd(A);

PCA Applications: Action Unit Recognition







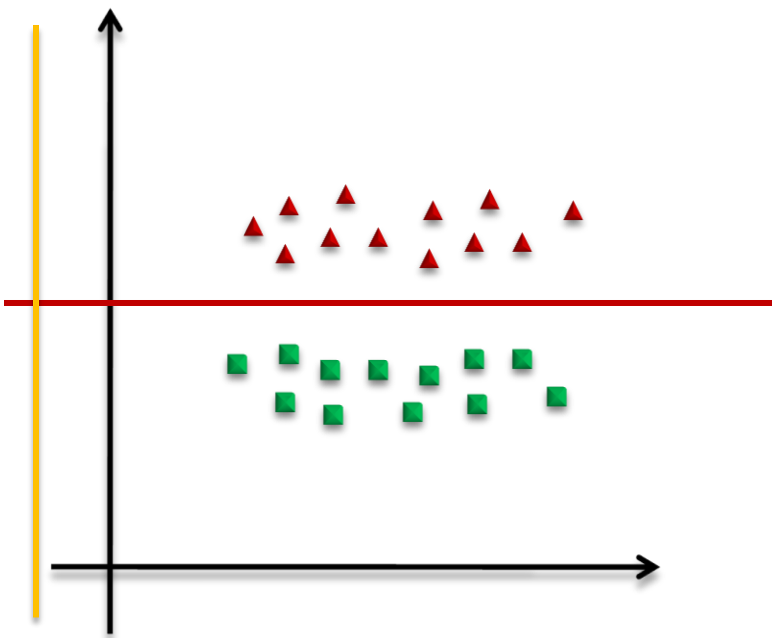
T. Simon, J. Hernandez, 2009

Applications

- Visualize high dimensional data (e.g. AnalogySpace)
- Find hidden relationships (e.g. topics in articles)
- Compress information
- Avoid redundancy
- Reduce algorithm complexity
- Outlier detection
- Denoising

Demos:
http://www.cs.mcgill.ca/~sqrt/dimr/dimreduction.html

PCA for pattern recognition



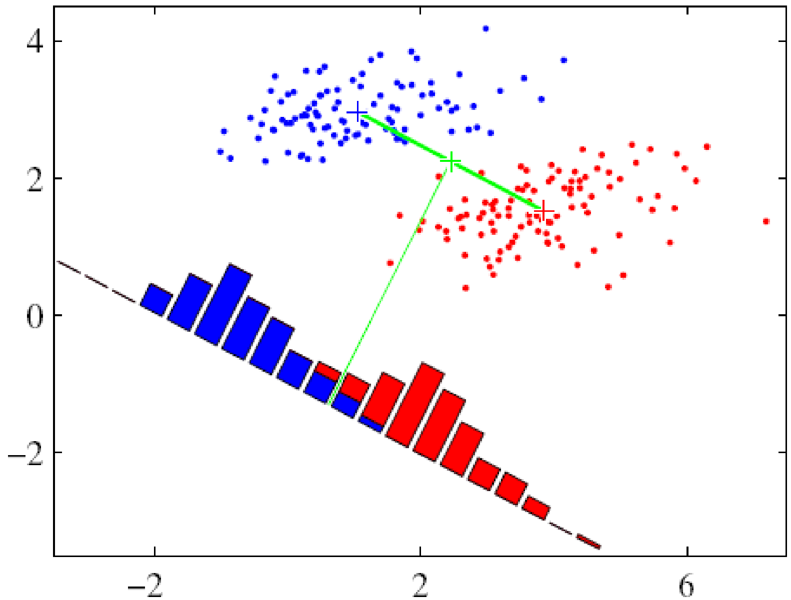
Principal Component Analysis

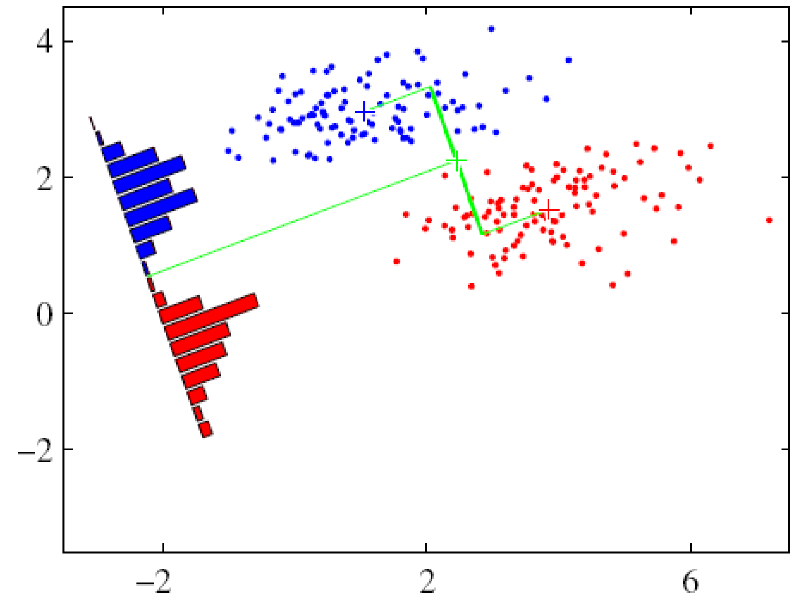
- higher variance
- bad for discriminability

Fisher Linear Discriminant
Linear Discriminant Analysis

- smaller variance
- good discriminability

Linear Discriminant Analysis





$$y = w^T x$$

- Assumptions for new basis:
  - Maximize distance between projected class means
  - Minimize projected class variance

Linear Discriminant Analysis

Objective

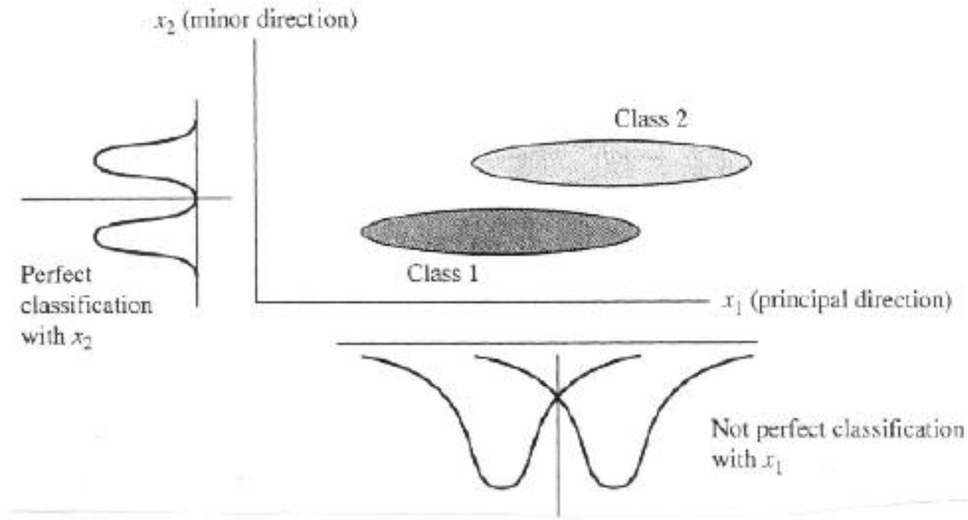
$$argmax_w J(w) = \frac{w^T S_B w}{w^T S_W w}$$

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$$
$$S_B = (m_2 - m_1)(m_2 - m_1)^T$$
$$S_W = \sum_j^2 \sum_{x \in C_j} (x - m_j)(x - m_j)^T$$

Algorithm

1. Compute class means
2. Compute $w = S_W^{-1}(m_2 - m_1)$
3. Project data $y = w^T x$

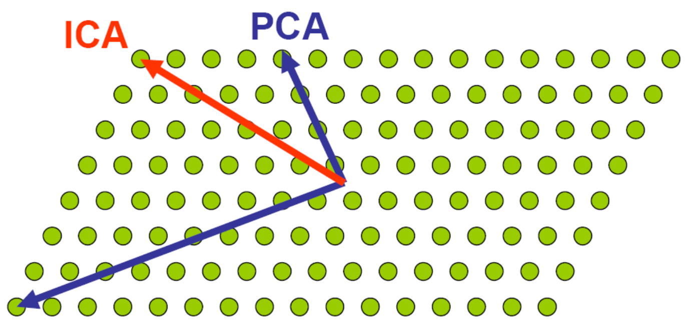
PCA vs LDA



- **PCA**: Perform dimensionality reduction while preserving as much of the **variance** in the high dimensional space as possible.
- **LDA**: Perform dimensionality reduction while preserving as much of the class **discriminatory** information as possible.

Other Component Analysis Algorithms

- Independent Component Analysis (ICA)



PCA: uncorrelated PCs
ICA: independent PCs

Blind source separation

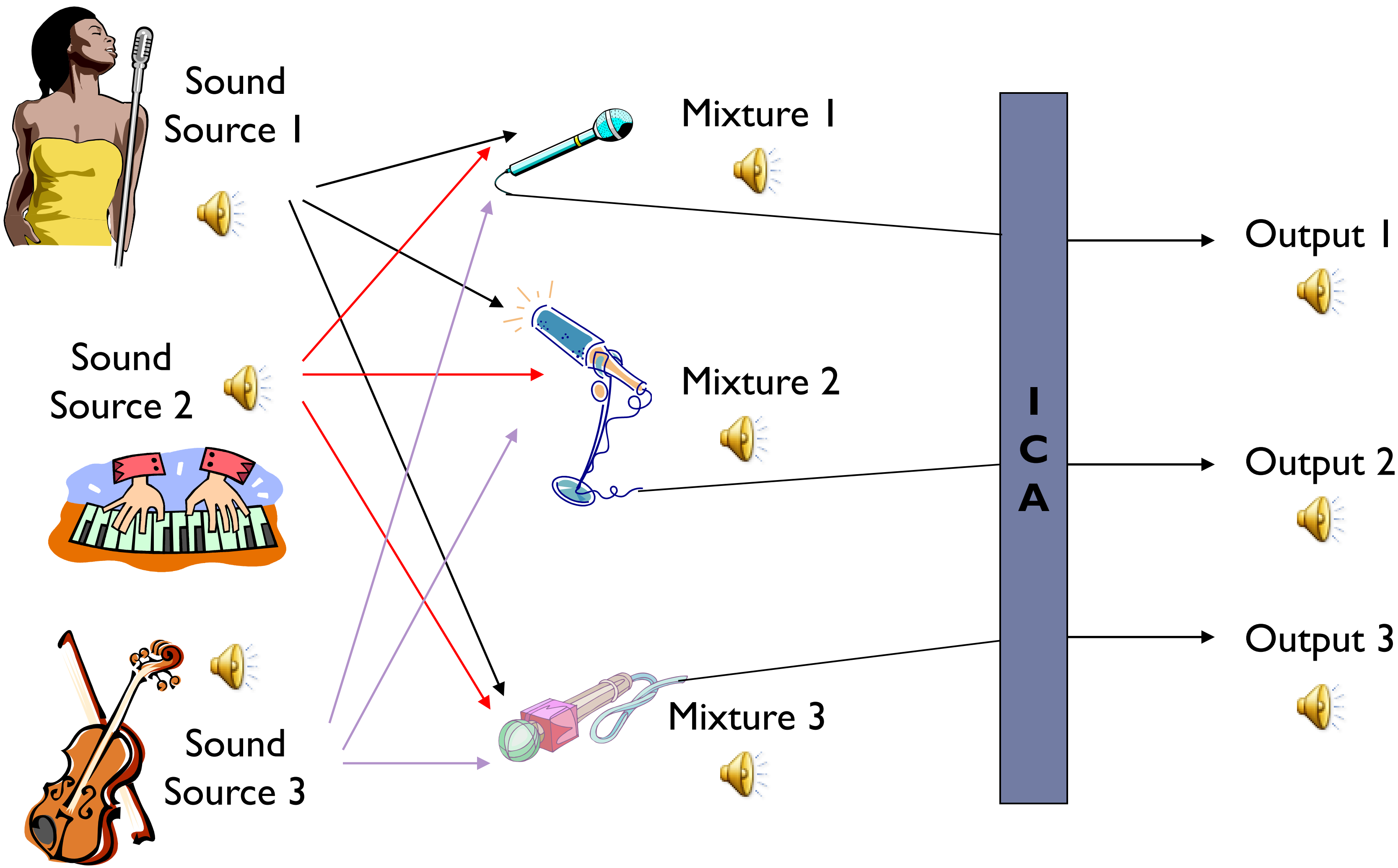
M. Poh, D. McDuff, and R. Picard 2010

Recall:

independent → uncorrelated

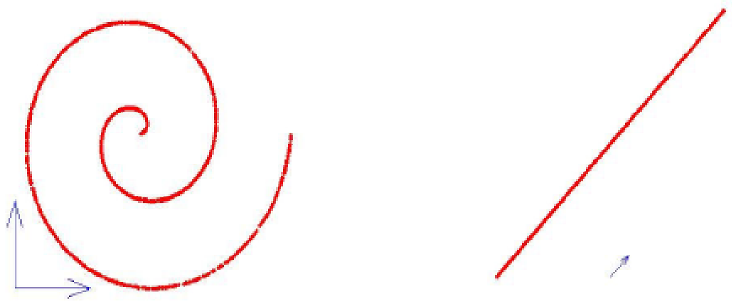
uncorrelated X independent

Independent Component Analysis

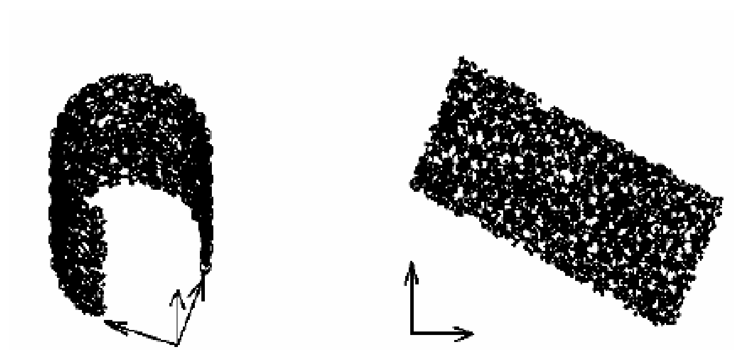


Other Component Analysis Algorithms

- Curvilinear Component Analysis (CCA)



Non linear projection of a spiral



Non linear projection of a horseshoe

PCA/LDA: linear projection

CCA: non-linear projection
while preserving proximity
between points

What you should know?

- Why dimensionality reduction?
- PCA/LDA assumptions
- PCA/LDA algorithms
- PCA vs LDA
- Applications
- Possible extensions

Recommended Readings

- Tutorials
  - **A tutorial on PCA**. J. Shlens
  - **A tutorial on PCA**. L. Smith
  - **Fisher Linear Discriminat Analysis**. M. Welling
  - **Eigenfaces** (http://www.cs.princeton.edu/~cdecoro/eigenfaces/) C. DeCoro

- Publications
  - **Eigenfaces for Recognition**. M. Turk and A. Pentland
  - **Using Discriminant Eigenfeatures for Image Retrieval**. D. Swets, J. Weng
  - **Eigenfaces vs. Fisherfaces**: Recognition Using Class Specific Linear Projection. P. Belhumeur, J. Hespanha, and D. Kriegman
  - **PCA versus LDA**. A. Martinez, A. Kak