EE 6885 Statistical Pattern Recognition

Fall 2005
Prof. Shih-Fu Chang
http://www.ee.columbia.edu/~sfchang

Lecture 18 (11/28/05)

- Reading
  - Feature Dimension Reduction
    - PCA, ICA, LDA, Chapter 3.8, 10.13
    - ICA Tutorial:
      - **Aapo Hyvärinen and Erkki Oja, "Independent Component Analysis: Algorithms and Applications,"** ***Neural Networks*****, 13(4-5):411-430, 2000**
- Final Exam
  - Dec. 16th Friday 1:10-3 pm, Mudd Rm 644

Review Lecture #3: Multi-variate Gaussian

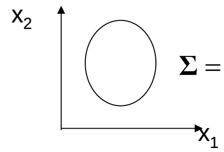
- Multivariate Gaussian, $N(\mu, \Sigma)$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\boldsymbol{\Sigma}|}} e^{\left(\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}$$

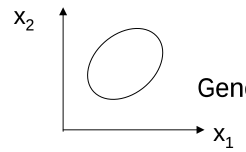
where $\mathbf{x}, \boldsymbol{\mu}$ are D-dimensional vectors

$\boldsymbol{\Sigma} : D \times D$ matrix

$|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$



$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$



General $\boldsymbol{\Sigma}$

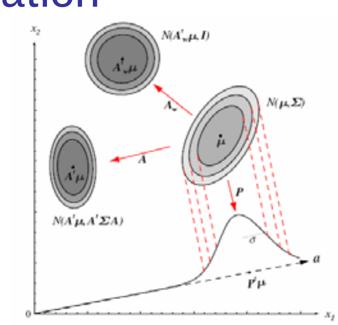
$$(\sigma_{ij})^2 = \Sigma(i, j) = \text{cov}(x(i), x(j))$$
$$= E[(x(i) - \mu(i))(x(j) - \mu(j))]$$

Effect of Linear Transformation

- Linear transformation of Gaussian

$y = A^t x \qquad y : k \times 1,\ A : d \times k,\ x : d \times 1$

$y \sim N(A^t \mu, A^t \Sigma A)$



- Whitening transform

$\Sigma = \Phi \Lambda \Phi^t$ (SVD, Eigenvectors)

$\Phi : [\phi_1 | \phi_2 | \ldots | \phi_d]$ columns are eigenvectors

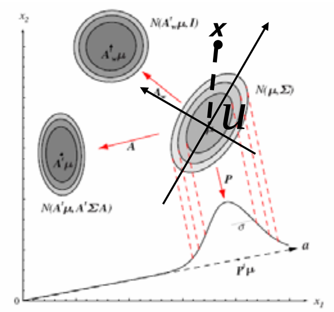
$$A^t \Sigma A = A^t \Phi \Lambda \Phi^t A = I$$

Whitening Trans. $A_w = \Phi \Lambda^{-1/2}$

also PCA Transform

$y = A_w{}^t x \sim N(A_w{}^t \mu,\ I)$

- Malalanobis distance from point x to the mean of a Gaussian



$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\boldsymbol{\Sigma}|}} \exp(\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

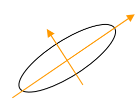
$$= \frac{1}{(2\pi)^{D/2}\sqrt{|\boldsymbol{\Sigma}|}} \exp(\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \Phi\Lambda^{-1}\Phi^t(\mathbf{x}-\boldsymbol{\mu}))$$

$$= \frac{1}{(2\pi)^{D/2}\sqrt{|\boldsymbol{\Sigma}|}} \exp(\frac{-1}{2}\underbrace{(A_w^t(\mathbf{x}-\boldsymbol{\mu}))^t(A_w^t(\mathbf{x}-\boldsymbol{\mu}))}_{r^2})$$

r is the Mahalanobis distance
is also the Euclidean dist in the PCA space

PCA for feature dimension reduction

- Approximate data with reduced dimensions



1-D approximation $\hat{\mathbf{x}} = \mathbf{m} + a\mathbf{e}$, $\mathbf{m}$: mean

Approximation Error $J_1(\mathbf{e}) = \sum_{k=1}^{n} \|\hat{\mathbf{x}}_k - \mathbf{x}_k\| = \sum_{k=1}^{n} \|(\mathbf{m} + a_k\mathbf{e}) - \mathbf{x}_k\|^2$

$$= \sum_{k=1}^{n} a_k^2 \|\mathbf{e}\|^2 - 2\sum_{k=1}^{n} a_k \mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \quad = -\sum_{k=1}^{n} \left[\mathbf{e}^t(\mathbf{x}_k - \mathbf{m})\right]^2 + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\mathbf{e}^t \underbrace{\left[\sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t\right]}_{} \mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2 \quad = -\mathbf{e}^t \mathbf{S} \mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

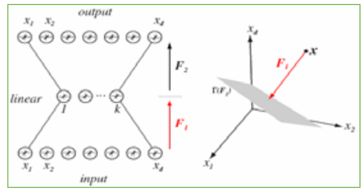
S: scatter matrix $= (n-1) \times$ sample covariance

Optimal **e** minimizing error J_1 -- eigenvector of **S** with the largest eigenvalue

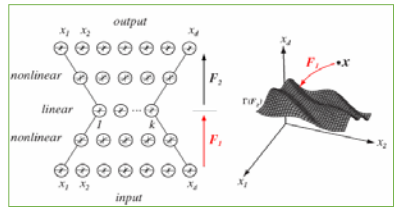
Multi-Dim. approximation $\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d'} a_i \mathbf{e}_i$ → what are the optimal $\boldsymbol{e}_i$?

Extension to Non-Linear Components

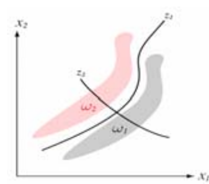
- **PCA bases can be found by backpropagation of multi-layer Neural Network (details in Chap 6)**



- **PCA can be extended to Nonlinear Component Analysis (NLCA) by adding using a multi-layer NN (Chap 6)**



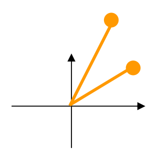
- **nonlinear components useful for approximating data**
- **But may not be good for classification**



EE6887-Chang

18-7

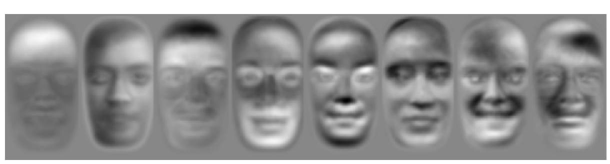
Eigenface (Pentland et al '91)



- Treat each image as a 1-D vector
- Find the dimensions with the largest variation
- How to classify? Pros and Cons?

- Samples from DARPA FERET Data Set

- Sample Eigenfaces



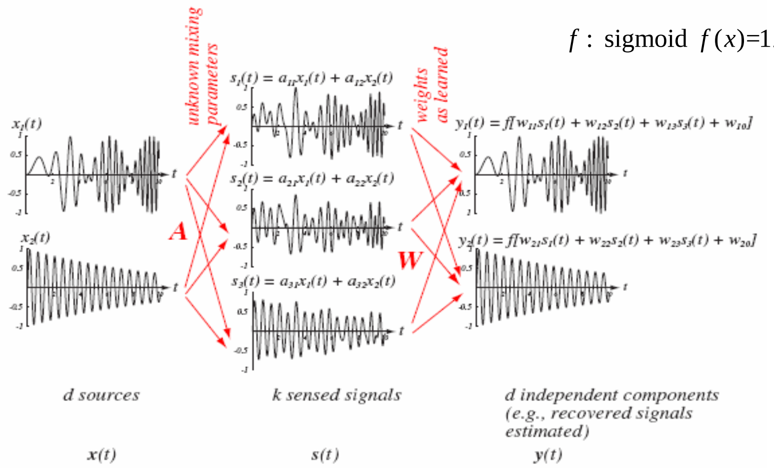
EE6887-Chang

18-8

Independent Component Analysis

- Seek most independent directions, instead of minimize representation errors (sum-squared-error) as in PCA
- Blind source separation in speech mixture

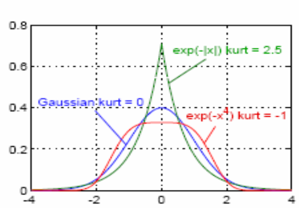
f : sigmoid $f(x)=1/(1+e^{-x})$



- Find the best weights to make the output components independent
- How to measure independence?
  - Linear combination of random variables leads to Normal distribution
  - Use the high-order statistics to measure the divergence from Gaussian

- **Measures of deviations from Gaussianity: 4th moment is Kurtosis ("bulging")**

$$kurt(y) = E\left[\left(\frac{y-\mu}{\sigma}\right)^4\right] - 3$$



(from Ellis)

-kurtosis of Gaussian is zero (this def.)
-'heavy tails' → $kurt > 0$
-closer to uniform dist. → $kurt < 0$

•Directly related to KL divergence from Gaussian PDF

- Use gradient decent method to refine weights to reduce Gaussianity
- FastICA Matlab package : http://www.cis.hut.fi/projects/ica/fastica/

- Or... maximize the independence measure
  - Joint entropy

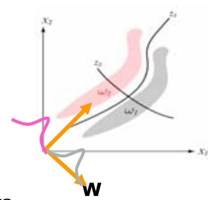
$$H(\mathbf{y}) = -E\left[\ln p_y(\mathbf{y})\right] \;\; = E\left[\ln|\mathbf{J}|\right] - \underbrace{E\left[\ln p_s(\mathbf{s})\right]}_{\text{independent of w}}$$

$$p_y(\mathbf{y}) = \frac{p_s(\mathbf{s})}{|\mathbf{J}|} \qquad \text{Jacobian } J = \begin{bmatrix} \frac{\delta y_1}{\delta s_1} & \cdots & \frac{\delta y_d}{\delta s_1} \\ \vdots & \ddots & \vdots \\ \frac{\delta y_1}{\delta s_d} & \cdots & \frac{\delta y_d}{\delta s_d} \end{bmatrix}$$

- Gradient decent $\Delta \mathbf{W} \propto \frac{\delta H(\mathbf{y})}{\delta \mathbf{W}}$
- **Pros :**
  - **non-orthogonal, variable number, combined with nonlinear function**
- **Cons:**
  - **no order of importance, sensitive to noise, temporal dimension not considered**

LDA: Linear Discriminant Analysis

Given a set of data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, and their class labels
Find the best projection dimension, $y_i = \mathbf{w}^t \mathbf{x}_i$
so that y_i are most separable



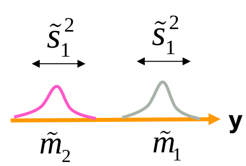
$$\tilde{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{w}^t \mathbf{x} = \mathbf{w}^t \mathbf{m}_i$$

$\mathbf{m}_i$: sample means
$\tilde{m}_i$: sample means of projected points

$$\tilde{s}_i^2 = \frac{1}{n_i} \sum_{\mathbf{y} \in Y_i} (y - \tilde{m}_i)^2$$

$\tilde{s}_1^2 + \tilde{s}_2^2$: within-class scatter

LDA maximizes criterion function: $J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$



LDA Scatter Matrices

before projection: $\mathbf{S}_i = \sum_{\mathbf{x}\in D_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t$

after projecttion: $\tilde{s}_i^2 = \mathbf{w}^t \mathbf{S}_i \mathbf{w}$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^t(\mathbf{S}_1 + \mathbf{S}_2)\mathbf{w} = \mathbf{w}^t \mathbf{S}_\mathbf{w} \mathbf{w}$$

$\mathbf{S}_\mathbf{w} = \mathbf{S}_1 + \mathbf{S}_2$: within-class scatter matrix

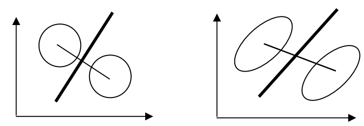
Similarly, between-class scatter matrix $\mathbf{S}_\mathbf{B} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$

$$\Rightarrow J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_\mathbf{B} \mathbf{w}}{\mathbf{w}^t \mathbf{S}_\mathbf{w} \mathbf{w}}$$

$\mathbf{S}_\mathbf{w}$: usually nonsingular $\mathbf{S}_\mathbf{B}$: rank 1

$$\Rightarrow \mathbf{w}_{opt} = \arg\max J(\mathbf{w}) = \mathbf{S}_\mathbf{w}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

Remember the Gaussian Cases



EE6887-Chang

18-13