Lecture Slides for

INTRODUCTION TO

Machine Learning

ETHEM ALPAYDIN

Edited for CS 536 Fall 2005 – Rutgers University
Ahmed Elgammal

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml

CHAPTER 6:

Dimensionality Reduction

Why Reduce Dimensionality?

1. Reduces time complexity: Less computation
2. Reduces space complexity: Less parameters
3. Saves the cost of observing the feature
4. Simpler models are more robust on small datasets
5. More interpretable; simpler explanation
6. Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions

Feature Selection vs Extraction

- Feature selection: Choosing $k<d$ important features, ignoring the remaining $d - k$

  Subset selection algorithms
- Feature extraction: Project the

  original x_i , $i = 1,...,d$ dimensions to

  new $k<d$ dimensions, z_j , $j = 1,...,k$

  Principal components analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Subset Selection

- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially Ø.
  - At each iteration, find the best new feature
    $j = \text{argmin}_i E (F \cup x_i)$
  - Add x_j to F if $E (F \cup x_j) < E (F)$
- Hill-climbing O(d^2) algorithm
- Backward search: Start with all features and remove one at a time, if possible.
- Floating search (Add k, remove l)

Principal Components Analysis (PCA)

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected there, information loss is minimized.
- The projection of $\boldsymbol{x}$ on the direction of $\boldsymbol{w}$ is: $z = \boldsymbol{w}^T\boldsymbol{x}$
- Find $\boldsymbol{w}$ such that Var(z) is maximized

$$\begin{aligned}\text{Var}(z) &= \text{Var}(\boldsymbol{w}^T\boldsymbol{x}) = \text{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})^2] \\ &= \text{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})] \\ &= \text{E}[\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{w}] \\ &= \boldsymbol{w}^\text{T}\,\text{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{\Sigma}\, \boldsymbol{w}\end{aligned}$$

where $\text{Var}(\boldsymbol{x}) = \text{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T] = \boldsymbol{\Sigma}$

- Maximize Var(z) subject to $||\boldsymbol{w}||=1$

$$\max_{\boldsymbol{w}_1} \boldsymbol{w}_1^T \Sigma \boldsymbol{w}_1 - \alpha\left(\boldsymbol{w}_1^T \boldsymbol{w}_1 - 1\right)$$

$\Sigma \boldsymbol{w}_1 = \alpha \boldsymbol{w}_1$ that is, $\boldsymbol{w}_1$ is an eigenvector of Σ

Choose the one with the largest eigenvalue for Var(z) to be max

- Second principal component: Max Var(z_2), s.t., $||\boldsymbol{w}_2||=1$ and orthogonal to $\boldsymbol{w}_1$

$$\max_{\boldsymbol{w}_2} \boldsymbol{w}_2^T \Sigma \boldsymbol{w}_2 - \alpha\left(\boldsymbol{w}_2^T \boldsymbol{w}_2 - 1\right) - \beta\left(\boldsymbol{w}_2^T \boldsymbol{w}_1 - 0\right)$$

$\Sigma \boldsymbol{w}_2 = \alpha \boldsymbol{w}_2$ that is, $\boldsymbol{w}_2$ is another eigenvector of Σ

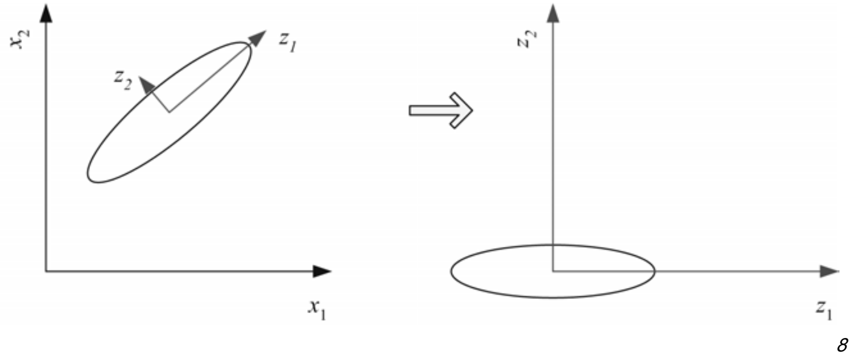
and so on.

What PCA does

$$\boldsymbol{z} = \mathbf{W}^T(\boldsymbol{x} - \boldsymbol{m})$$

where the columns of $\mathbf{W}$ are the eigenvectors of Σ, and $\boldsymbol{m}$ is sample mean

Centers the data at the origin and rotates the axes

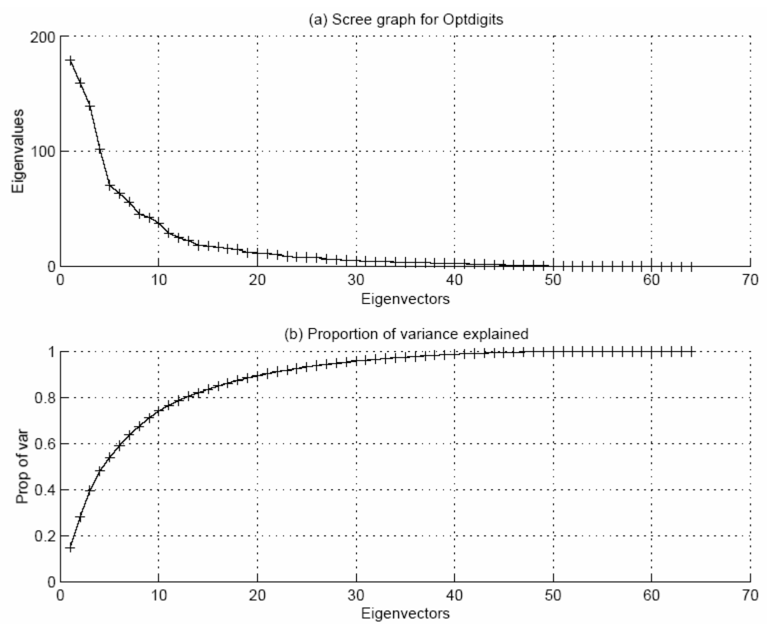


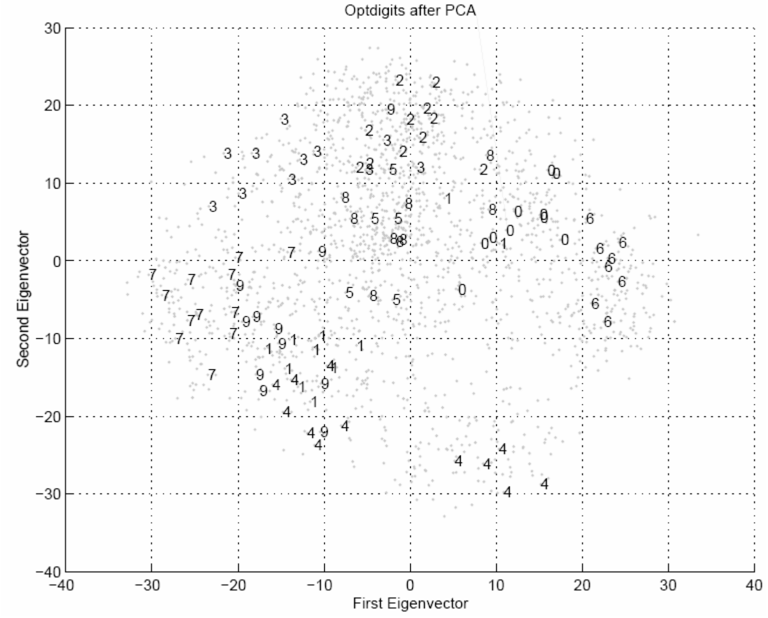
How to choose k ?

- Proportion of Variance (PoV) explained

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

  when λ_i are sorted in descending order
- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs k, stop at "elbow"



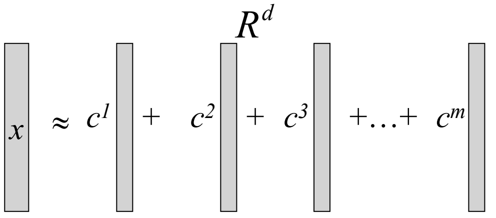


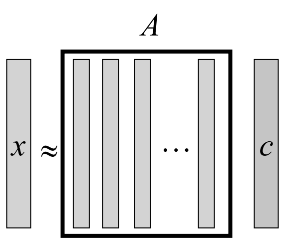
Principle Component Analysis PCA

- Given a set of points $\{x_1, x_2, \cdots, x_N\}, x_i \in R^d$
- We are looking for a linear projection: a linear combination of orthogonal basis vectors

$$x \approx A \cdot c$$

R^d $\quad$ $R^m, m << d$





What is the projection that minimizes the reconstruction error ? $E = \sum_i \|x_i - Ac_i\|$

Principle Component Analysis PCA

- Given a set of points

$$\{x_1, x_2, \cdots, x_N\}, x_i \in R^d$$

- Center the points: compute

$$\mu = \tfrac{1}{N}\sum_i x_i$$

$$P = [x_1 - \mu, x_2 - \mu, \cdots, x_N - \mu], x_i \in R^d$$

- Compute covariance matrix $Q = PP^T$
- Compute the eigen vectors for Q $\longrightarrow$ $Qe_k = \lambda_k e_k$
- *Eigenvectors are the orthogonal basis we are looking for*

Singular Value Decomposition

- SVD: If **A** is a real *m* by *n* matrix then there exist orthogonal matrices

U ($m \cdot m$) and **V** ($n \cdot n$) such that

$\mathbf{U^tAV} = \Sigma = \mathrm{diag}(\boldsymbol{\sigma 1, \sigma 2, \ldots, \sigma p})$ $\quad p = \min\{m,n\}$

$$\mathbf{U^tAV} = \Sigma \qquad \mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V^t}$$

- **Singular values**: Non negative square roots of the eigenvalues of $\mathbf{A^tA}$. Denoted σ_i, $i=1,\ldots,n$
- $\mathbf{A^tA}$ is symmetric $\Rightarrow$ eigenvalues and singular values are real.
- Singular values arranged in decreasing order.

$$A^t A = (U\textstyle\sum V^t)^t (U\sum V^t) = V\sum^t U^t U \sum V^t = V \sum^t \sum V^t = V\sum^2 V^{-1}$$

$$(A^t A)V = V\textstyle\sum^2$$

$$(A^t A)v = v\lambda$$

$$\underset{m\text{x}n}{A} = \underset{m\text{x}m}{\mathrm{U}}\ \underset{m\text{x}n}{\Sigma}\ \underset{n\text{x}n}{\mathrm{V}^{\mathrm{t}}}$$

SVD for PCA

- SVD can be used to efficiently compute the image basis

$$PP^t = (U \sum V^t)(U \sum V^t)^t = U \sum V^t V \sum^t U^t = U \sum^t \sum U^t = U \sum^2 U^{-1}$$

$$(PP^t)U = U \sum^2$$

$$(PP^t)v = v\lambda$$

- U are the eigen vectors (basis)
- Most important thing to notice: Distance in the eigen-space is an approximation to the correlation in the original space

$$\left\| x_i - x_j \right\| \approx \left\| c_i - c_j \right\|$$

PCA

$$x \approx Ac$$

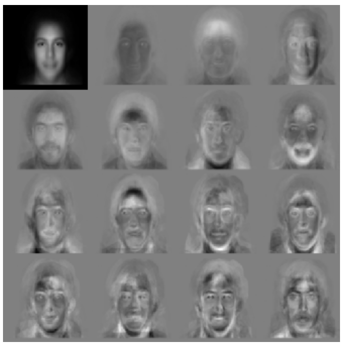
$$R^d \qquad c \approx A^T x \qquad R^m, m << d$$

- Most important thing to notice: Distance in the eigen-space is an approximation to the correlation in the original space

$$\left\| x_i - x_j \right\| \approx \left\| c_i - c_j \right\|$$

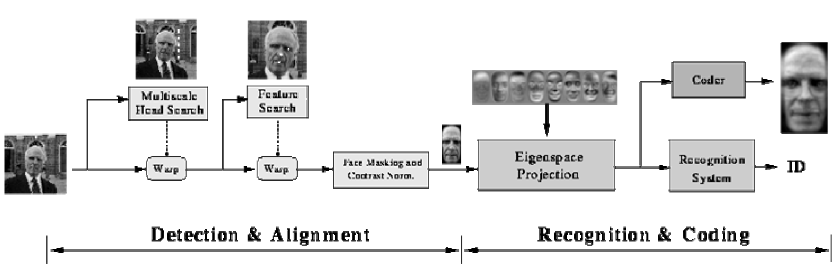
Eigenface

- Use PCA and subspace projection to perform face recognition
- How to describe a face as a linear combination of face basis
- Matthew Turk and Alex Pentland “Eigenfaces for Recognition” 1991



Face Recognition - Eigenface

- MIT Media Lab -Face Recognition demo page
  http://vismod.media.mit.edu/vismod/demos/facerec/



Lecture Notes for E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

Factor Analysis

- Find a small number of factors $\boldsymbol{z}$, which when combined generate $\boldsymbol{x}$:

$$x_i - \mu_i = v_{i1}z_1 + v_{i2}z_2 + \dots + v_{ik}z_k + \varepsilon_i$$

where z_j, $j = 1,...,k$ are the latent factors with

$$E[\,z_j\,]=0,\ \mathrm{Var}(z_j)=1,\ \mathrm{Cov}(z_i, z_j)=0,\ i \neq j\ ,$$

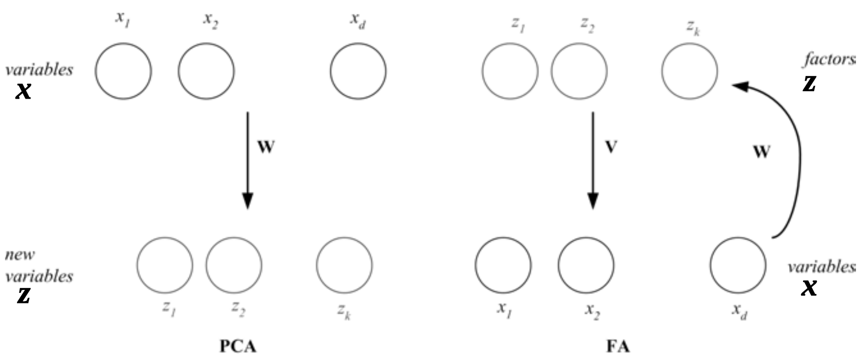
ε_i are the noise sources

$$E[\,\varepsilon_i\,]= \psi_i,\ \mathrm{Cov}(\varepsilon_i , \varepsilon_j) = 0,\ i \neq j,\ \mathrm{Cov}(\varepsilon_i , z_j) = 0\ ,$$

and v_{ij} are the factor loadings

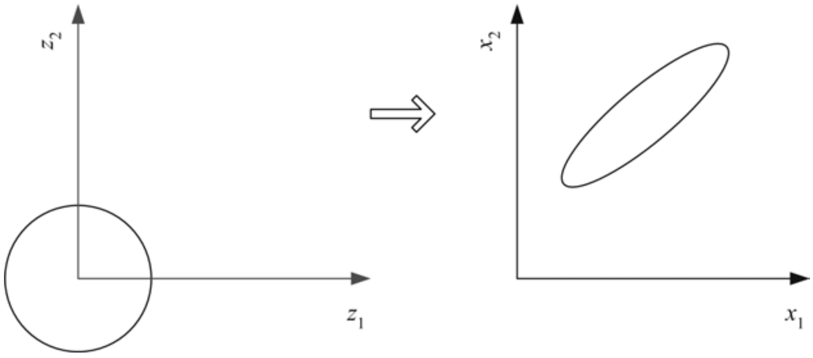
PCA vs FA

- PCA   From $\boldsymbol{x}$ to $\boldsymbol{z}$ $\boldsymbol{z} = \mathbf{W}^T(\boldsymbol{x} - \boldsymbol{\mu})$
- FA   From $\boldsymbol{z}$ to $\boldsymbol{x}$ $\boldsymbol{x} - \boldsymbol{\mu} = \mathbf{V}\boldsymbol{z} + \boldsymbol{\varepsilon}$



Factor Analysis

- In FA, factors z_j are stretched, rotated and translated to generate $\boldsymbol{x}$



Multidimensional Scaling

- Given pairwise distances between N points,

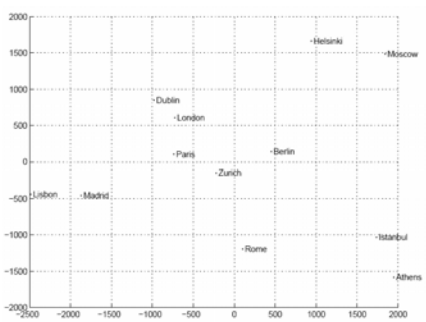
  d_{ij}, $i,j = 1,\ldots,N$

  place on a low-dim map s.t. distances are preserved.
- $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x} \mid \theta)$ Find θ that min Sammon stress

$$E(\theta \mid \mathcal{X}) = \sum_{r,s} \frac{\left(\left\|\boldsymbol{z}^r - \boldsymbol{z}^s\right\| - \left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|\right)^2}{\left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|^2}$$

$$= \sum_{r,s} \frac{\left(\left\|\boldsymbol{g}(\boldsymbol{x}^r \mid \theta) - \boldsymbol{g}(\boldsymbol{x}^s \mid \theta)\right\| - \left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|\right)^2}{\left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|^2}$$

Map of Europe by MDS

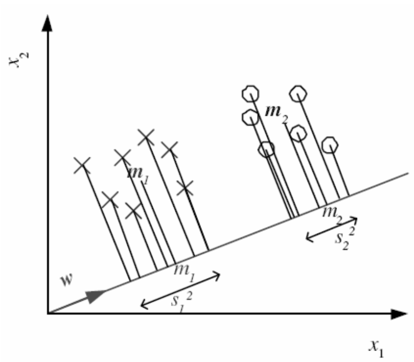


Map from CIA – The World Factbook: http://www.cia.gov/

Linear Discriminant Analysis

- Find a low-dimensional space such that when $\boldsymbol{x}$ is projected, classes are well-separated.
- Find $\boldsymbol{w}$ that maximizes

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$



$$m_1 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t r^t}{\sum_t r^t} \qquad s_1^2 = \sum_t \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_1\right)^2 r^t$$

- Between-class scatter:

$$(m_1 - m_2)^2 = \left(\boldsymbol{w}^T \boldsymbol{m}_1 - \boldsymbol{w}^T \boldsymbol{m}_2\right)^2$$
$$= \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}$$
$$= \boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w} \quad \text{where } \mathbf{S}_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$$

- Within-class scatter:

$$s_1^2 = \sum_t \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_1\right)^2 r^t$$
$$= \sum_t \boldsymbol{w}^T \left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)^T \boldsymbol{w} r^t = \boldsymbol{w}^T \mathbf{S}_1 \boldsymbol{w}$$
$$\text{where } \mathbf{S}_1 = \sum_t \left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)^T r^t$$
$$s_1^2 + s_1^2 = \boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w} \quad \text{where } \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Find $\boldsymbol{w}$ that max

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}} = \frac{\left|\boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)\right|^2}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}}$$

- LDA soln:

$$\boldsymbol{w} = c \cdot \mathbf{S}_W^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2)$$

- Parametric soln:

$$\boldsymbol{w} = \Sigma^{-1} (\mu_1 - \mu_2)$$
$$\text{when } p(\boldsymbol{x} \mid C_i) \sim \mathcal{N}(\mu_i, \Sigma)$$

$K>2$ Classes

- Within-class scatter:

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t \left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)^T$$

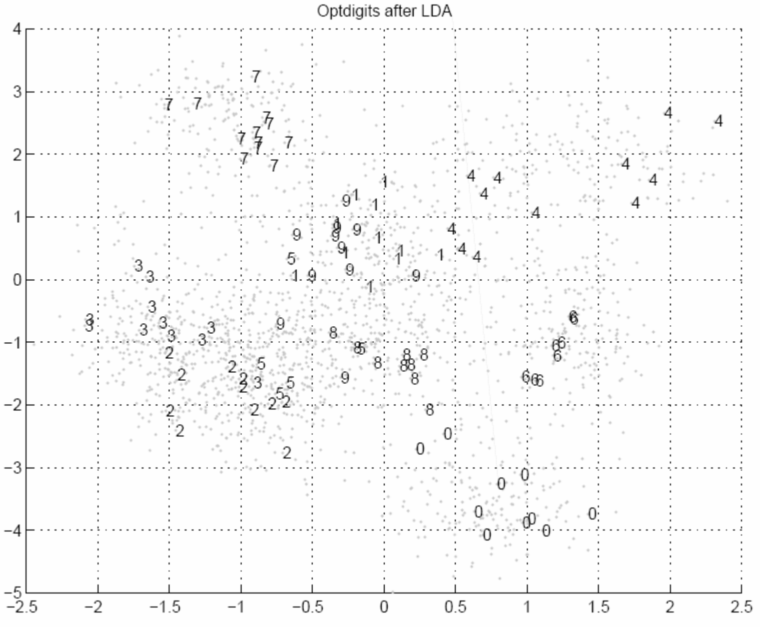
- Between-class scatter:

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i \left(\boldsymbol{m}_i - \boldsymbol{m}\right)\left(\boldsymbol{m}_i - \boldsymbol{m}\right)^T \qquad \boldsymbol{m} = \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{m}_i$$

- Find $\mathbf{W}$ that max

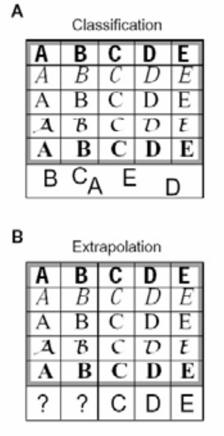
$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$
Maximum rank of K-1



Separating Style and Content

- Objective: Decomposing two factors using linear methods
  - Content: which character
  - Style : which font
- "Bilinear models"
- J. Tenenbaum and W. Freeman "Separating Style and Content with Bilinear Models" Neural computation 2000

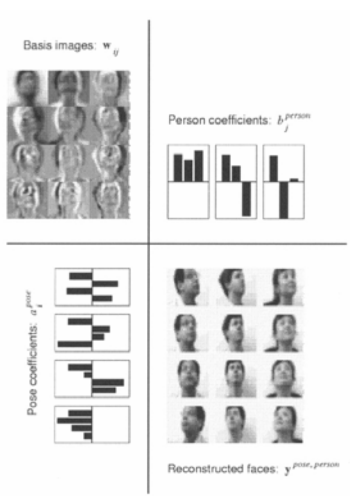


Figures from J. Tenenbaum and W. Freeman "Separating Style and Content with Bilinear Models" Neural computation 2000

Bilinear Models

- Symmetric bilinear model

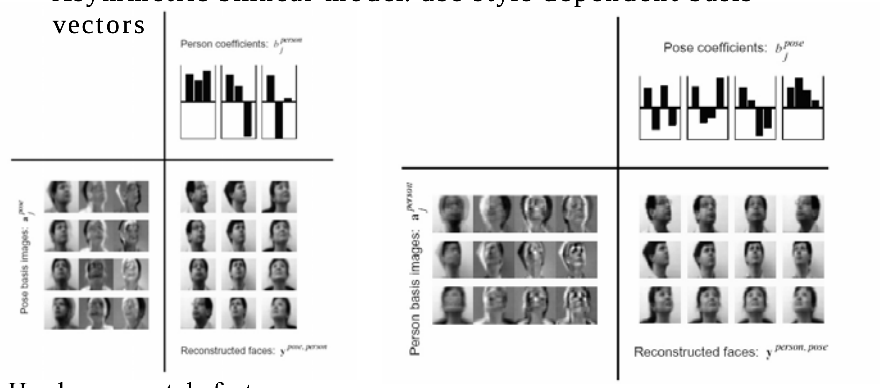
$$y^{sc} = \sum_{i,j} w_{ij} a_i^s b_j^c$$



Figures from J. Tenenbaum and W. Freeman "Separating Style and Content with Bilinear Models" Neural computation 2000

Bilinear models $y^{sc} = A^s b^c$

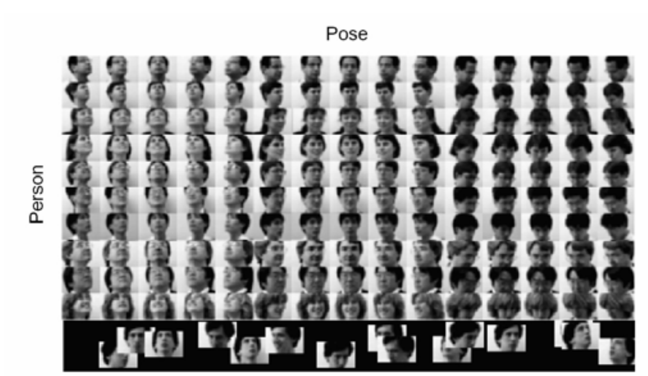
- Asymmetric bilinear model: use style dependent basis vectors



Head pose as style factor
person as content

Person as style factor
pose as content

Figures from J. Tenenbaum and W. Freeman “Separating Style and Content with Bilinear Models” Neural computation 2000



Figures from J. Tenenbaum and W. Freeman “Separating Style and Content with Bilinear Models” Neural computation 2000