Data Analysis and Dimension Reduction

- PCA, LDA and LSA

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

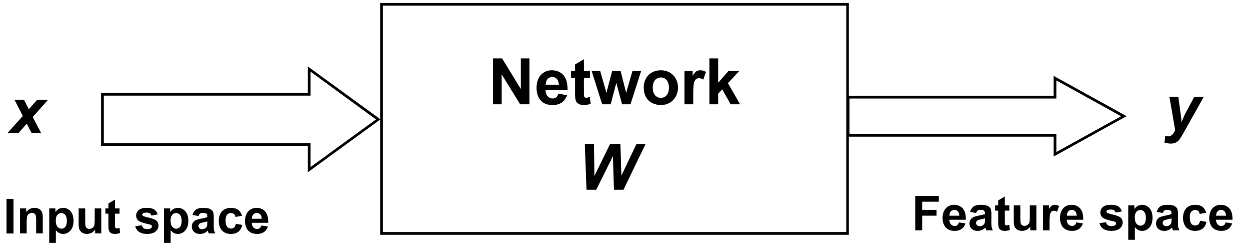
References:

1. *Introduction to Machine Learning* , Chapter 6

2. *Data Mining: Concepts, Models, Methods and Algorithms*, Chapter 3

Introduction (1/3)

- Goal: discover significant patterns or features from the input data
  - Salient feature selection or dimensionality reduction



  - Compute an input-output mapping based on some desirable properties

Introduction (2/3)

- Principal Component Analysis (PCA)

- Linear Discriminant Analysis (LDA)

- Latent Semantic Analysis (LSA)

Bivariate Random Variables

- If the random variables *X* and *Y* have a certain joint distribution that describes a bivariate random variable

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix}$$

bivariate random variable

$$\Rightarrow \mu_Z = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}$$

mean vector

$$\Rightarrow \Sigma_Z = \begin{bmatrix} \sigma_{X,X} & \sigma_{X,Y} \\ \sigma_{X,Y} & \sigma_{Y,Y} \end{bmatrix}$$

covariance matrix

$$\text{variance}\, \sigma_{X,X} = \sigma_X^2 = \mathbf{E}\left[(X - \mathbf{E}[X])^2\right] = \mathbf{E}\left[X^2\right] - (\mathbf{E}[X])^2$$

$$\text{covariance}\, \sigma_{X,Y} = \sigma_{Y,X} = \mathbf{E}[(X - \mathbf{E}[X])(Y - \mathbf{E}[Y])] = \mathbf{E}[XY] - \mathbf{E}[X]\mathbf{E}[Y]$$

Multivariate Random Variables

- If the random variables $X_1, X_2, \ldots, X_n$ have a certain joint distribution that describes a multivariate random variable

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$ multivariate random variable

$$\Rightarrow \mu_X = \begin{bmatrix} \mu_{X_1} \\ \vdots \\ \mu_{X_n} \end{bmatrix}$$ mean vector

$$\Rightarrow \Sigma_Z = \begin{bmatrix} \sigma_{X_1,X_1} & \cdots & \sigma_{X_1,X_n} \\ \vdots & \vdots & \vdots \\ \sigma_{X_n,X_1} & \cdots & \sigma_{X_n,X_n} \end{bmatrix}$$ covariance matrix

$$\text{variance}\, \sigma_{X_i,X_i} = \sigma_{X_i}^2 = \mathbf{E}\left[\left(X_i - \mathbf{E}[X_i]\right)^2\right] = \mathbf{E}\left[X_i^2\right] - \left(\mathbf{E}[X_i]\right)^2$$

$$\text{covariance}\, \sigma_{X_i,X_j} = \sigma_{X_j,X_i} = \mathbf{E}\left[\left(X_i - \mathbf{E}[X_i]\right)\left(X_j - \mathbf{E}\left[X_j\right]\right)\right] = \mathbf{E}\left[X_i X_j\right] - \mathbf{E}[X_i]\mathbf{E}\left[X_j\right]$$

Introduction (3/3)

- Formulation for feature extraction and dimension reduction
  - Model-free (nonparametric)
    - Without prior information: e.g., PCA
    - With prior information: e.g., LDA
  - Model-dependent (parametric), e.g.,
    - HLDA with Gaussian cluster distributions
    - PLSA with multinomial latent cluster distributions

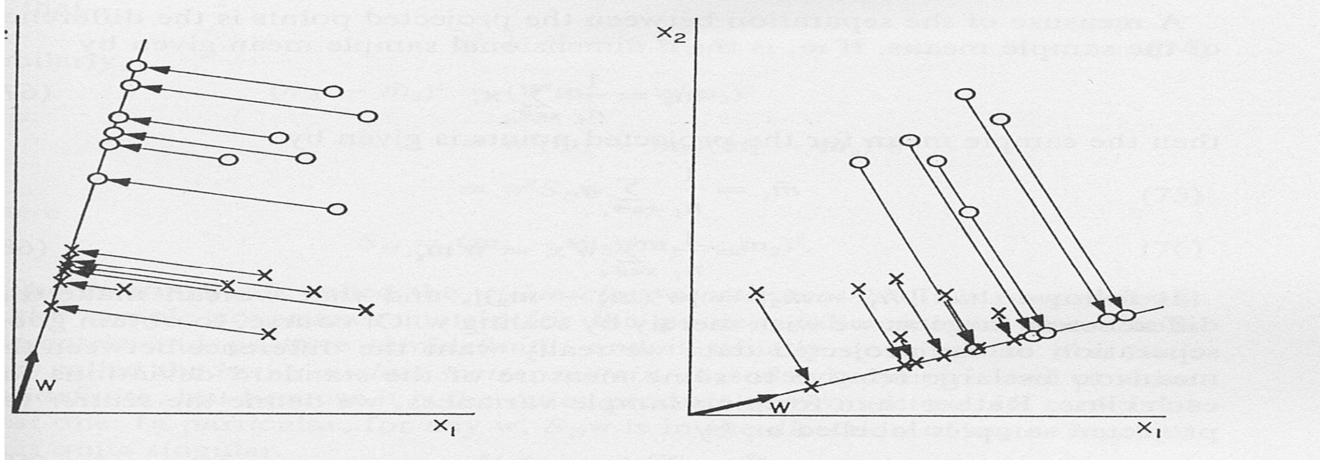


FIGURE 4.6. Projection of samples onto a line.

Principal Component Analysis (PCA) (1/2)

Pearson, 1901

- Known as Karhunen-Loẻve Transform (1947, 1963)
  - Or Hotelling Transform (1933)
- A standard technique commonly used for data reduction in statistical pattern recognition and signal processing
- A transform by which the data set can be represented by reduced number of effective features and still retain the most intrinsic information content
  - A small set of features to be found to represent the data samples accurately
- Also called "Subspace Decomposition", "Factor Analysis" ..

Principal Component Analysis (PCA) (2/2)

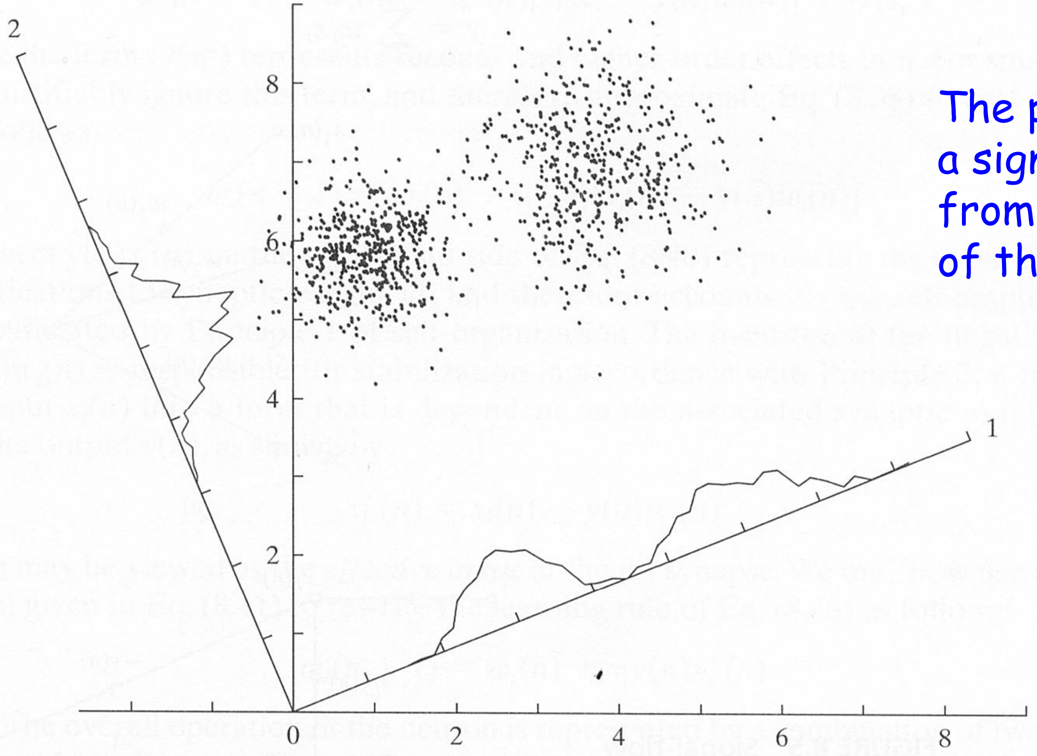
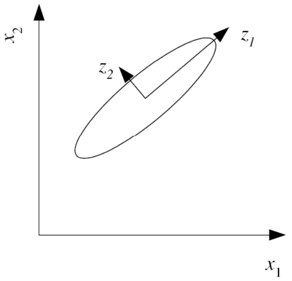


FIGURE 8.4 A cloud of data points is shown in two dimensions, and the density plots formed by projecting this cloud onto each of two axes, 1 and 2, are indicated. The projection onto axis 1 has maximum variance, and clearly shows the bimodal, or clustered character of the data.

The patterns show a significant difference from each other in one of the transformed axes

PCA Derivations (1/13)

- Suppose ***x*** is an *n*-dimensional zero mean random vector, $\boldsymbol{\mu} = \mathbf{E}\{\boldsymbol{x}\} = \boldsymbol{0}$
  - If ***x*** is not zero-mean, we can subtract the mean before processing the following analysis



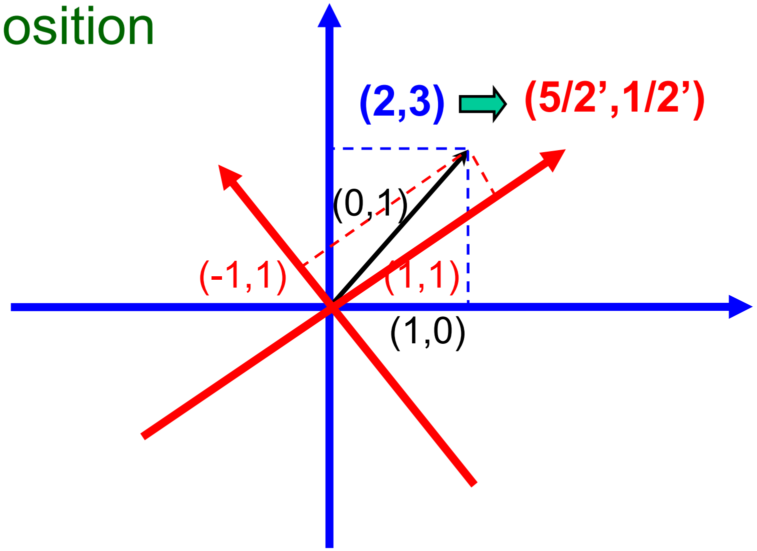
  - ***x*** can be represented without error by the summation of *n* linearly independent vectors

$$\boldsymbol{x} = \sum_{i=i}^{n} \underbrace{y_i}_{\text{The } i\text{-th component in the feature (mapped) space}} \boldsymbol{\varphi}_i = \boldsymbol{\Phi}\mathbf{y} \quad \text{where} \quad \mathbf{y} = \begin{bmatrix} y_1 & . & y_i & . & y_n \end{bmatrix}^T$$

$$\boldsymbol{\Phi} = \underbrace{\begin{bmatrix} \boldsymbol{\varphi}_1 & . & \boldsymbol{\varphi}_i & . & \boldsymbol{\varphi}_n \end{bmatrix}}_{\text{The basis vectors}}$$

PCA Derivations (2/13)

Subspace Decomposition



$$\begin{bmatrix}2\\3\end{bmatrix} = \underbrace{2\begin{bmatrix}1\\0\end{bmatrix} + 3\begin{bmatrix}0\\1\end{bmatrix} = \frac{5}{2}\begin{bmatrix}1\\1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}-1\\1\end{bmatrix}}_{\text{orthogonal basis sets}} = 1\begin{bmatrix}1\\2\end{bmatrix} + 1\begin{bmatrix}1\\1\end{bmatrix}$$

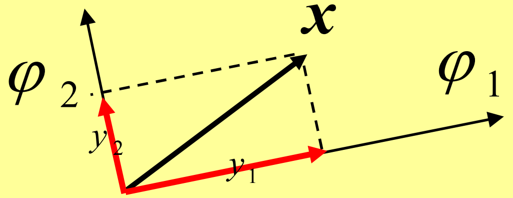
PCA Derivations (3/13)

- Further assume the column (basis) vectors of the matrix $\boldsymbol{\Phi}$ form an orthonormal set

$$\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

  - Such that y_i is equal to the projection of $\boldsymbol{x}$ on $\boldsymbol{\varphi}_i$

$$\forall_i \quad y_i = \boldsymbol{x}^T \varphi_i = \varphi_i^T \boldsymbol{x}$$



$$y_1 = \|\boldsymbol{x}\| \cos \theta_1 = \|\boldsymbol{x}\| \frac{\varphi_1^T \boldsymbol{x}}{\|\boldsymbol{x}\| \|\boldsymbol{\varphi}_1\|} = \varphi_1^T \boldsymbol{x}$$

, where $\|\boldsymbol{\varphi}_1\| = 1$

PCA Derivations (4/13)

- Further assume the column (basis) vectors of the matrix $\boldsymbol{\Phi}$ form an orthonormal set
  - y_i also has the following properties
    - Its mean is zero, too

$$\mathbf{E}\{y_i\} = \mathbf{E}\{\varphi_i^T \boldsymbol{x}\} = \varphi_i^T\, \mathbf{E}\{\boldsymbol{x}\} = \varphi_i^T \boldsymbol{0} = 0$$

$$\boldsymbol{\Sigma} = \mathbf{E}\left\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\right\}$$

$$\approx \left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T\right) \underbrace{- \boldsymbol{\mu}\boldsymbol{\mu}^T}_{\mathbf{0}}$$

$$\boldsymbol{R} = \mathbf{E}\{\boldsymbol{x}\boldsymbol{x}^T\} \approx \frac{1}{N}\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T$$

- Its variance is

$$\sigma_i^2 = \mathbf{E}\{y_i^2\} - [\mathbf{E}\{y\}]^2 = \mathbf{E}\{y_i^2\} = \mathbf{E}\{\varphi_i^T \boldsymbol{x}\boldsymbol{x}^T \varphi_i\} = \varphi_i^T\, \mathbf{E}\{\boldsymbol{x}\boldsymbol{x}^T\}\varphi_i$$

$$= \varphi_i^T \boldsymbol{R} \varphi_i \qquad [\boldsymbol{R} \text{ is the (auto-)correlation matrix of } \boldsymbol{x}]$$

- The correlation between two projections y_i and y_j is

$$\mathbf{E}\{y_i y_j\} = \mathbf{E}\left\{(\boldsymbol{\varphi}_i^T \boldsymbol{x})(\boldsymbol{\varphi}_j^T \boldsymbol{x})^T\right\} = \mathbf{E}\{\boldsymbol{\varphi}_i^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{\varphi}_j\}$$

$$= \boldsymbol{\varphi}_i^T\, \mathbf{E}\{\boldsymbol{x}\boldsymbol{x}^T\}\boldsymbol{\varphi}_j = \boldsymbol{\varphi}_i^T \boldsymbol{R} \boldsymbol{\varphi}_j$$

PCA Derivations (5/13)

- Minimum Mean-Squared Error Criterion
  - We want to choose only *m* of $\boldsymbol{\varphi}_i$'s that we still can approximate $\boldsymbol{x}$ well in **mean-squared error criterion**

original vector $$\boldsymbol{x} = \sum_{i=1}^{n} y_i \boldsymbol{\varphi}_i = \sum_{i=1}^{m} y_i \boldsymbol{\varphi}_i + \sum_{j=m+1}^{n} y_j \boldsymbol{\varphi}_j$$

reconstructed vector $$\hat{\boldsymbol{x}}(m) = \sum_{i=1}^{m} y_i \boldsymbol{\varphi}_i$$

$$\bar{\varepsilon}(m) = \mathbf{E}\left\{ \left\| \hat{\boldsymbol{x}}(m) - \boldsymbol{x} \right\|^2 \right\} = \mathbf{E}\left\{ \left(\sum_{j=m+1}^{n} y_j \boldsymbol{\varphi}_j^T \right) \left(\sum_{k=m+1}^{n} y_k \boldsymbol{\varphi}_k \right) \right\}$$

$$= \mathbf{E}\left\{ \sum_{j=m+1}^{n} \sum_{k=m+1}^{n} y_j y_k \boldsymbol{\varphi}_j^T \boldsymbol{\varphi}_k \right\}$$

$$= \sum_{j=m+1}^{n} \mathbf{E}\left\{ y_j^2 \right\}$$

$$\because \boldsymbol{\varphi}_j^T \boldsymbol{\varphi}_k = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}$$

$$E\{y_j\} = 0$$
$$\sigma_j^2 = E\{y_j^2\} - [E\{y_j\}]^2$$
$$= E\{y_j^2\}$$

$$= \sum_{j=m+1}^{n} \sigma_j^2 = \sum_{j=m+1}^{n} \boldsymbol{\varphi}_j^T \boldsymbol{R} \boldsymbol{\varphi}_j$$

We should discard the bases where the projections have lower variances

PCA Derivations (6/13)

- Minimum Mean-Squared Error Criterion
  - If the orthonormal (basis) set $\boldsymbol{\varphi}_i$'s is selected to be the eigenvectors of the correlation matrix $\boldsymbol{R}$, associated with eigenvalues λ_i's
    - They will have the property that:

$$\boldsymbol{R}\boldsymbol{\varphi}_j = \lambda_j \boldsymbol{\varphi}_j$$

$\boldsymbol{R}$ is real and symmetric, therefore its eigenvectors form a orthonormal set

$\boldsymbol{R}$ is positive definite ($\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} > 0$) => all eigenvalues are positive

  - Such that the mean-squared error mentioned above will be

$$\bar{\varepsilon}(m) = \sum_{j=m+1}^{n} \boxed{\sigma_j^2}$$

$$= \sum_{j=m+1}^{n} \boldsymbol{\varphi}_j^T \boldsymbol{R}\boldsymbol{\varphi}_j = \sum_{j=m+1}^{n} \boldsymbol{\varphi}_j^T \lambda_j \boldsymbol{\varphi}_j = \sum_{j=m+1}^{n} \boxed{\lambda_j}$$

PCA Derivations (7/13)

- Minimum Mean-Squared Error Criterion
  - If the eigenvectors are retained associated with the *m* largest eigenvalues, the mean-squared error will be

$$\bar{\varepsilon}_{eigen}(m) = \sum_{j=m+1}^{n} \lambda_j \quad \left(\text{where } \lambda_1 \geq \ldots \geq \lambda_m \geq \ldots \geq \lambda_n \geq 0\right)$$

  - Any two projections y_i and y_j will be mutually uncorrelated

$$E\{y_i y_j\} = E\left\{\left(\boldsymbol{\varphi}_i^T \boldsymbol{x}\right)\left(\boldsymbol{\varphi}_j^T \boldsymbol{x}\right)^T\right\} = E\left\{\boldsymbol{\varphi}_i^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{\varphi}_j\right\}$$

$$= \boldsymbol{\varphi}_i^T E\left\{\boldsymbol{x}\boldsymbol{x}^T\right\}\boldsymbol{\varphi}_j = \boldsymbol{\varphi}_i^T \boldsymbol{R}\boldsymbol{\varphi}_j = \lambda_{\mathrm{j}} \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j = 0$$

$$\boldsymbol{\Sigma} = E[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T] \approx \left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T\right) - \boldsymbol{\mu}\boldsymbol{\mu}^T$$

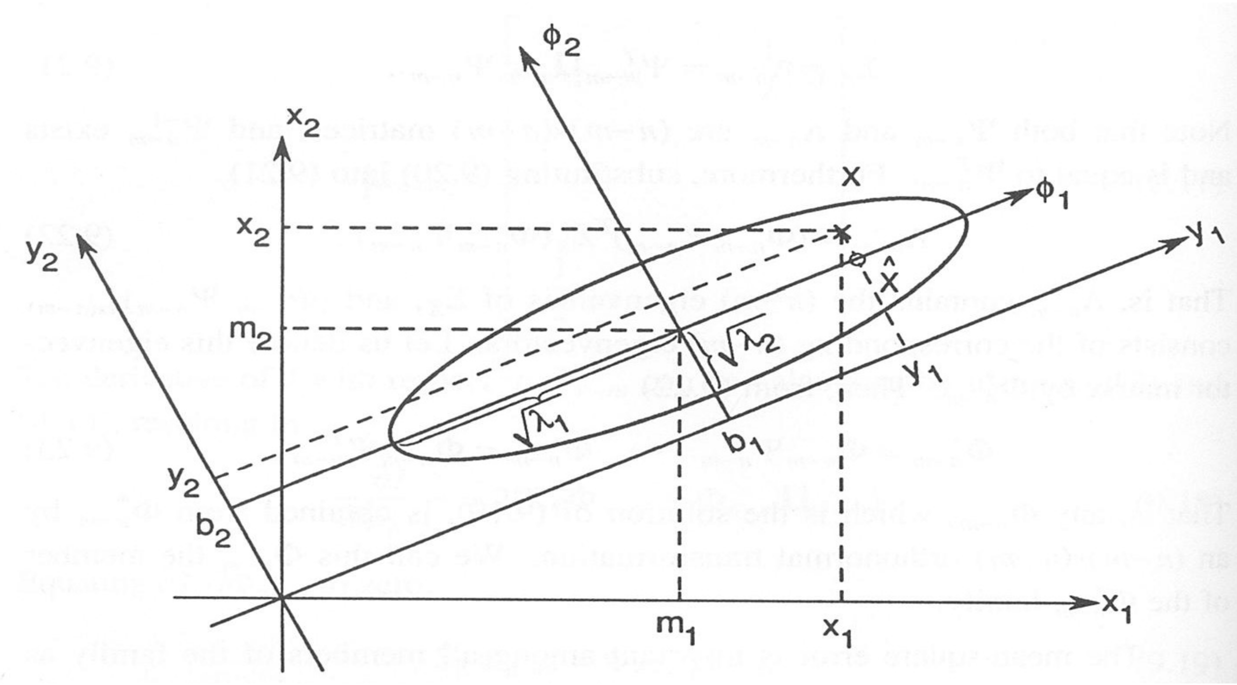
$$\boldsymbol{R} = E\left\{\boldsymbol{x}\boldsymbol{x}^T\right\} = \frac{1}{N}\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T$$

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdot & \cdot & \sigma_{1n} \\ \sigma_{22} & \sigma_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \cdot & \sigma_{nn} \end{bmatrix}$$

- Good news for most statistical modeling approaches
  - Gaussians and diagonal matrices

PCA Derivations (8/13)

- A Two-dimensional Example of Principle Component Analysis



PCA Derivations (9/13)

- Minimum Mean-Squared Error Criterion
  - It can be proved that $\bar{\varepsilon}_{eigen}(m)$ is the optimal solution under the mean-squared error criterion

$$\delta_{jk}=\begin{cases}1 & \text{if } j=k\\ 0 & \text{if } j\neq k\end{cases}$$

Objective function

To be minimized

Constraints

$$\text{Define: } J=\underbrace{\sum_{j=m+1}^{n}\boldsymbol{\varphi}_j^T\boldsymbol{R}\boldsymbol{\varphi}_j}_{\text{To be minimized}}-\underbrace{\sum_{j=m+1}^{n}\sum_{k=m+1}^{n}u_{jk}\left(\boldsymbol{\varphi}_j^T\boldsymbol{\varphi}_k-\delta_{jk}\right)}_{\text{Constraints}}$$

$$\frac{\partial\varphi^T\boldsymbol{R}\varphi}{\partial\varphi}=2\boldsymbol{R}\varphi$$

Partial Differentiation

$$\Rightarrow\forall_{m+1\le j\le n}\ \frac{\partial J}{\partial\boldsymbol{\varphi}_j}=2\boldsymbol{R}\boldsymbol{\varphi}_j-2\sum_{k=m+1}^{n}u_{jk}\boldsymbol{\varphi}_k=\boldsymbol{0}\quad\left(\text{where } \boldsymbol{u}_j^T=\left[u_{j\ m+1}\ldots\ldots u_{jn}\right]\right)$$

$$\Rightarrow\forall_{m+1\le j\le n}\ \boldsymbol{R}\boldsymbol{\varphi}_j=\boldsymbol{\Phi}_{n-m}\boldsymbol{u}_j\quad\left(\text{where } \boldsymbol{\Phi}_{n-m}=\left[\boldsymbol{\varphi}_{m+1}\ldots\ldots\boldsymbol{\varphi}_n\right]\right)$$

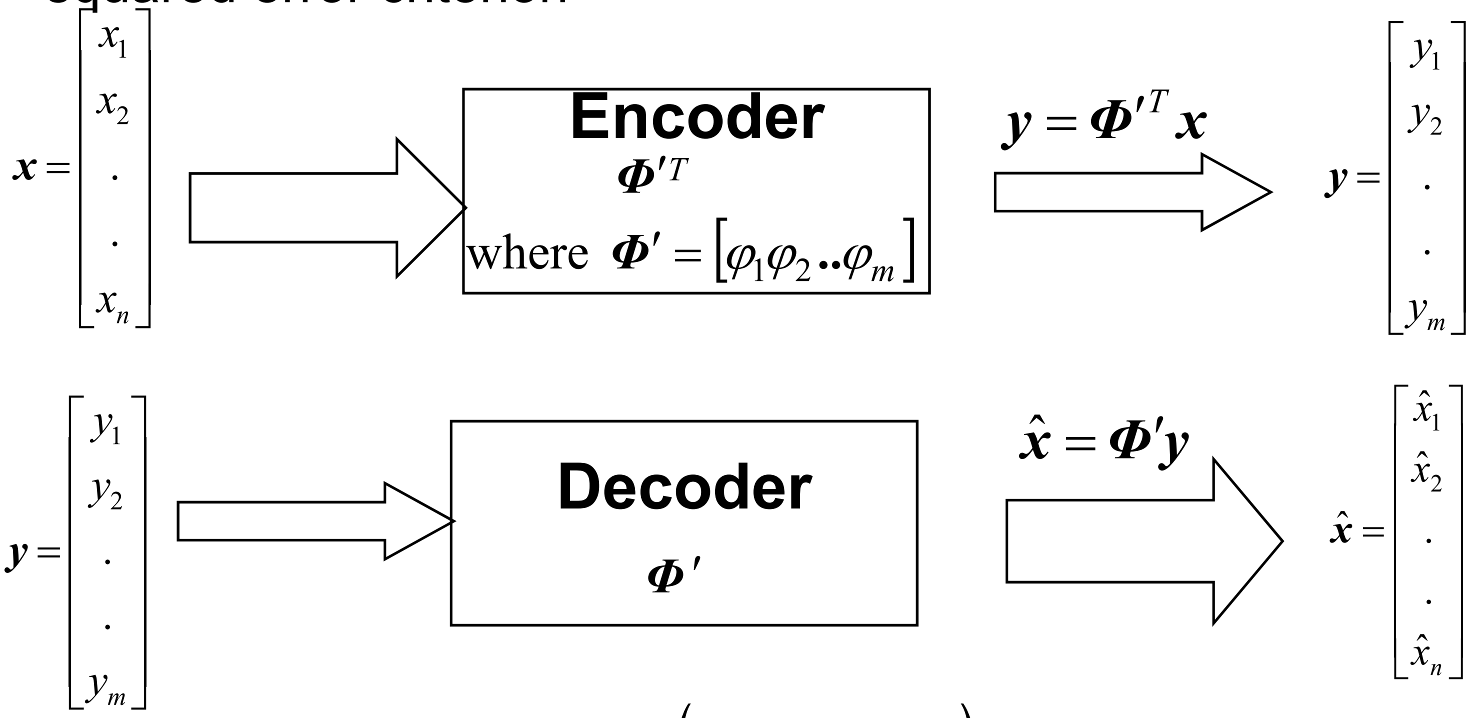
$$\Rightarrow\boldsymbol{R}\left[\boldsymbol{\varphi}_{m+1}\ldots\ldots\boldsymbol{\varphi}_n\right]=\boldsymbol{\Phi}_{n-m}\left[\boldsymbol{u}_{m+1}\ldots\ldots\boldsymbol{u}_n\right]$$

$$\Rightarrow\boldsymbol{R}\boldsymbol{\Phi}_{n-m}=\boldsymbol{\Phi}_{n-m}\boldsymbol{U}_{n-m}\quad\left(\text{where } \boldsymbol{U}_{n-m}=\left[\boldsymbol{u}_{m+1}\ldots\ldots\boldsymbol{u}_n\right]\right)$$

Have a particular solution if $\boldsymbol{U}_{n-m}$ is a diagonal matrix and its diagonal elements is the eigenvalues $\lambda_{m+1}\ldots\lambda_n$ of $\boldsymbol{R}$ and $\boldsymbol{\varphi}_{m+1}\cdots\boldsymbol{\varphi}_n$ is their corresponding eigenvectors

PCA Derivations (10/13)

- Given an input vector ***x*** with dimension *m*
  - Try to construct a linear transform ***Φ’*** (***Φ’*** is an *n*x*m* matrix *m*<*n*) such that the truncation result, ***Φ’***^{T}***x*,** is optimal in mean-squared error criterion



$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$$

Encoder

$\boldsymbol{\Phi}'^{T}$

where $\boldsymbol{\Phi}' = \left[\varphi_1\varphi_2..\varphi_m\right]$

$$\boldsymbol{y} = \boldsymbol{\Phi}'^{T}\boldsymbol{x}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix}$$

Decoder

$\boldsymbol{\Phi}'$

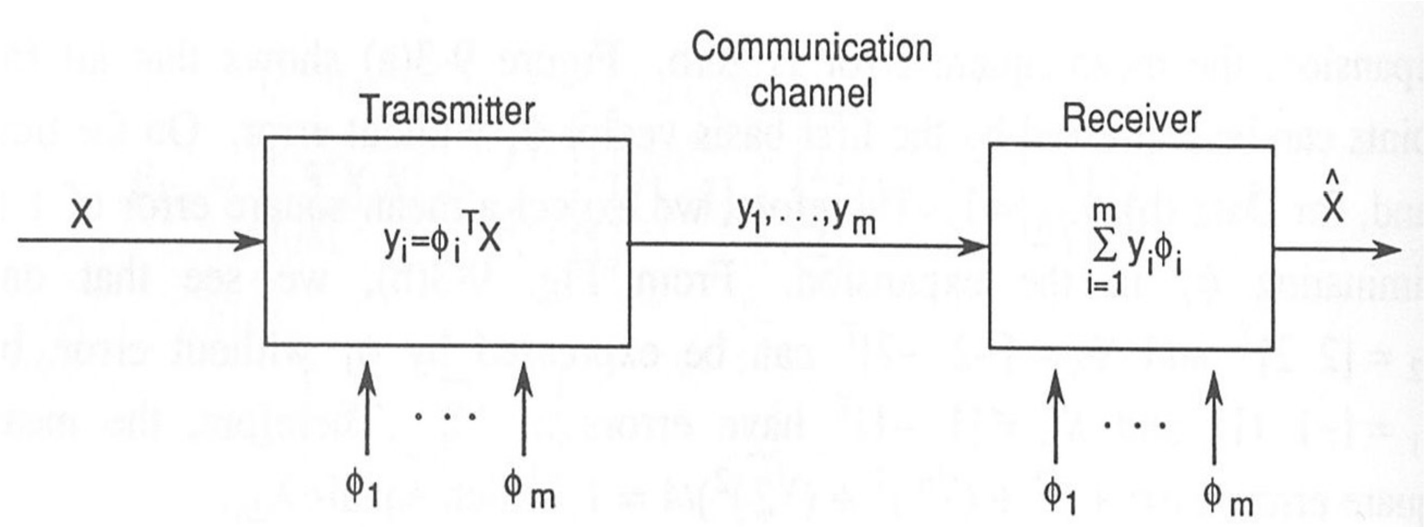
$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi}'\boldsymbol{y}$$

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ . \\ . \\ \hat{x}_n \end{bmatrix}$$

$$\text{minimize } E_x\left(\left(\hat{x}\text{-}x\right)^T\left(\hat{x}\text{-}x\right)\right)$$

PCA Derivations (11/13)

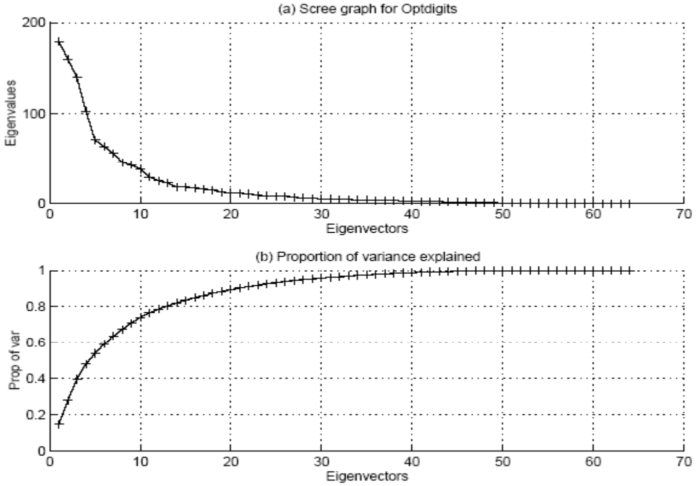
- Data compression in communication



  - PCA is an optimal transform for signal representation and dimensional reduction, but not necessary for classification tasks, such as speech recognition **? (To be discussed later on)**
  - PCA needs no prior information (e.g. class distributions of output information) of the sample patterns

PCA Derivations (12/13)

- Scree Graph
  - The plot of variance as a function of the number of eigenvectors kept
    - Select m such that $\frac{\lambda_1+\lambda_2+\cdots+\lambda_m}{\lambda_1+\lambda_2+\cdots+\lambda_m+\cdots+\lambda_n} \geq Threshold$



- Or select those eigenvectors with eigenvalues larger than the average input variance (average eivgenvalue)

$$\lambda_m \geq \frac{1}{n}\sum_{i=1}^{n}\lambda_i$$

PCA Derivations (13/13)

- PCA finds a linear transform ***W*** such that the sum of **average between-class variation** and **average within-class variation** is maximal

$$J(\boldsymbol{W}) = \left|\tilde{\boldsymbol{S}}\right| \overset{?}{=} \left|\tilde{\boldsymbol{S}}_w + \tilde{\boldsymbol{S}}_b\right| = \left|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W} + \boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}\right|$$

$$\boldsymbol{S} = \frac{1}{N}\sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

sample index

$$\boldsymbol{S}_w = \frac{1}{N}\sum_j N_j \boldsymbol{\Sigma}_j$$

class index

$$\boldsymbol{S}_b = \frac{1}{N}\sum_j N_j (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T$$

$$\tilde{\boldsymbol{S}}_w = \boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}$$

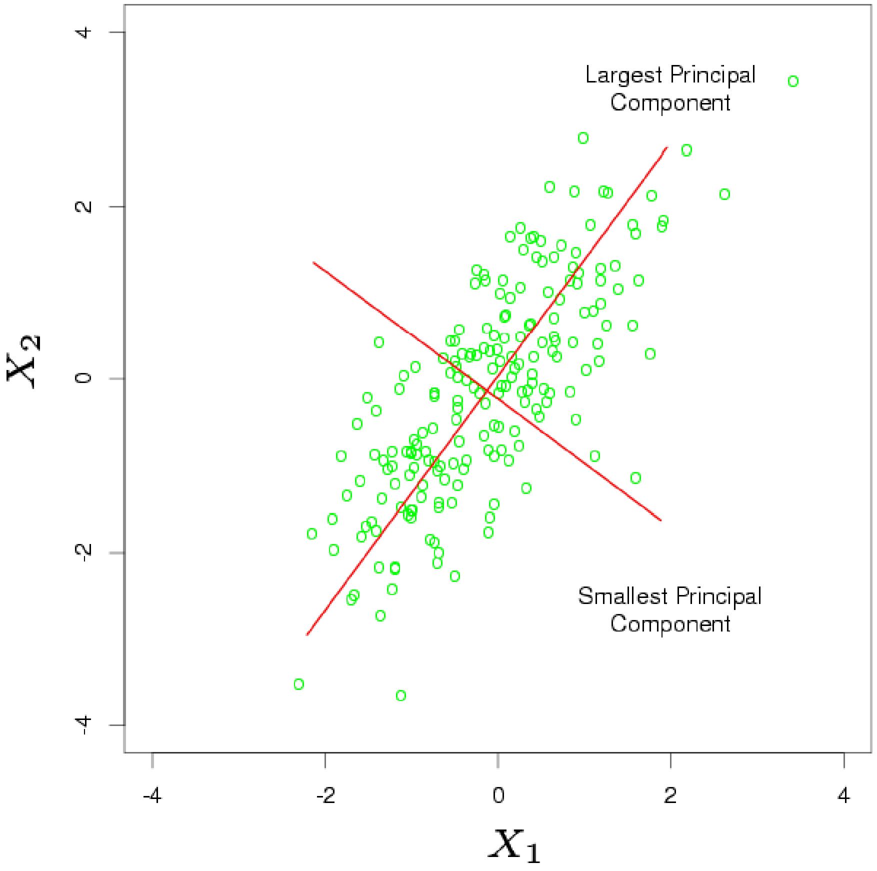
$$\tilde{\boldsymbol{S}}_b = \boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}$$

Try toshow that:

$\mathbf{S} = \mathbf{S}_w + \mathbf{S}_b$

PCA Examples: Data Analysis

- Example 1: principal components of some data points



PCA Examples: Feature Transformation

- Example 2: feature transformation and selection

Correlation matrix for old feature dimensions

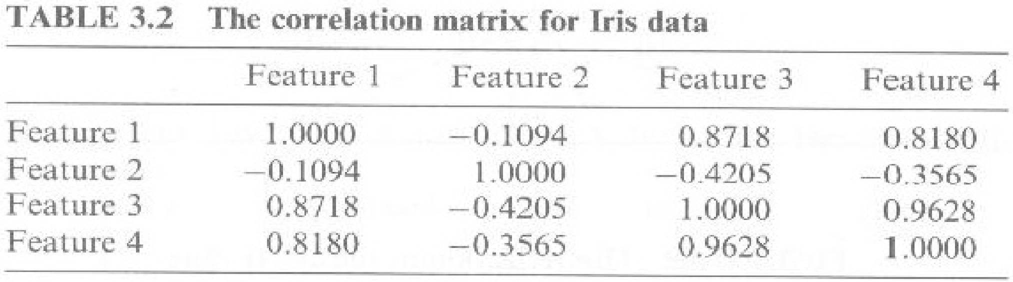


TABLE 3.2 The correlation matrix for Iris data

	Feature 1	Feature 2	Feature 3	Feature 4
Feature 1	1.0000	−0.1094	0.8718	0.8180
Feature 2	−0.1094	1.0000	−0.4205	−0.3565
Feature 3	0.8718	−0.4205	1.0000	0.9628
Feature 4	0.8180	−0.3565	0.9628	1.0000

New feature dimensions

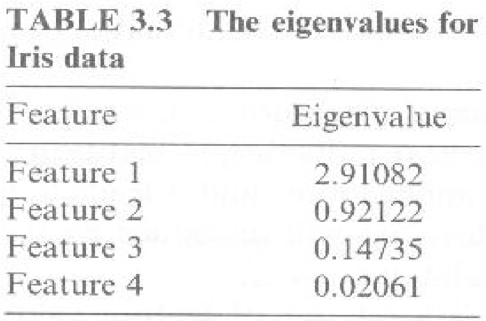
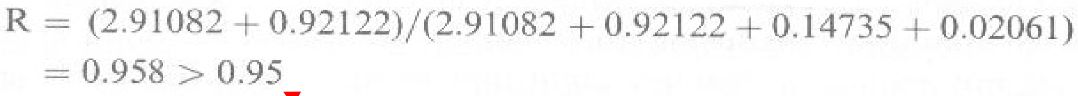


TABLE 3.3 The eigenvalues for Iris data

Feature	Eigenvalue
Feature 1	2.91082
Feature 2	0.92122
Feature 3	0.14735
Feature 4	0.02061

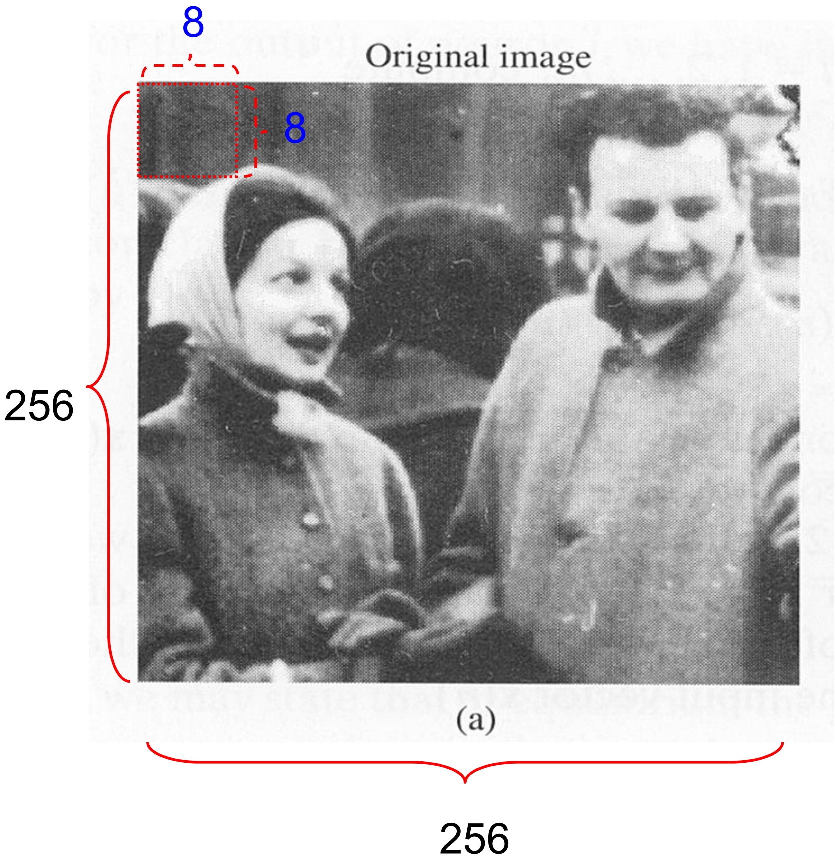


$$
\begin{aligned}
R &= (2.91082 + 0.92122)/(2.91082 + 0.92122 + 0.14735 + 0.02061) \\
&= 0.958 > 0.95
\end{aligned}
$$

threshold for information content reserved

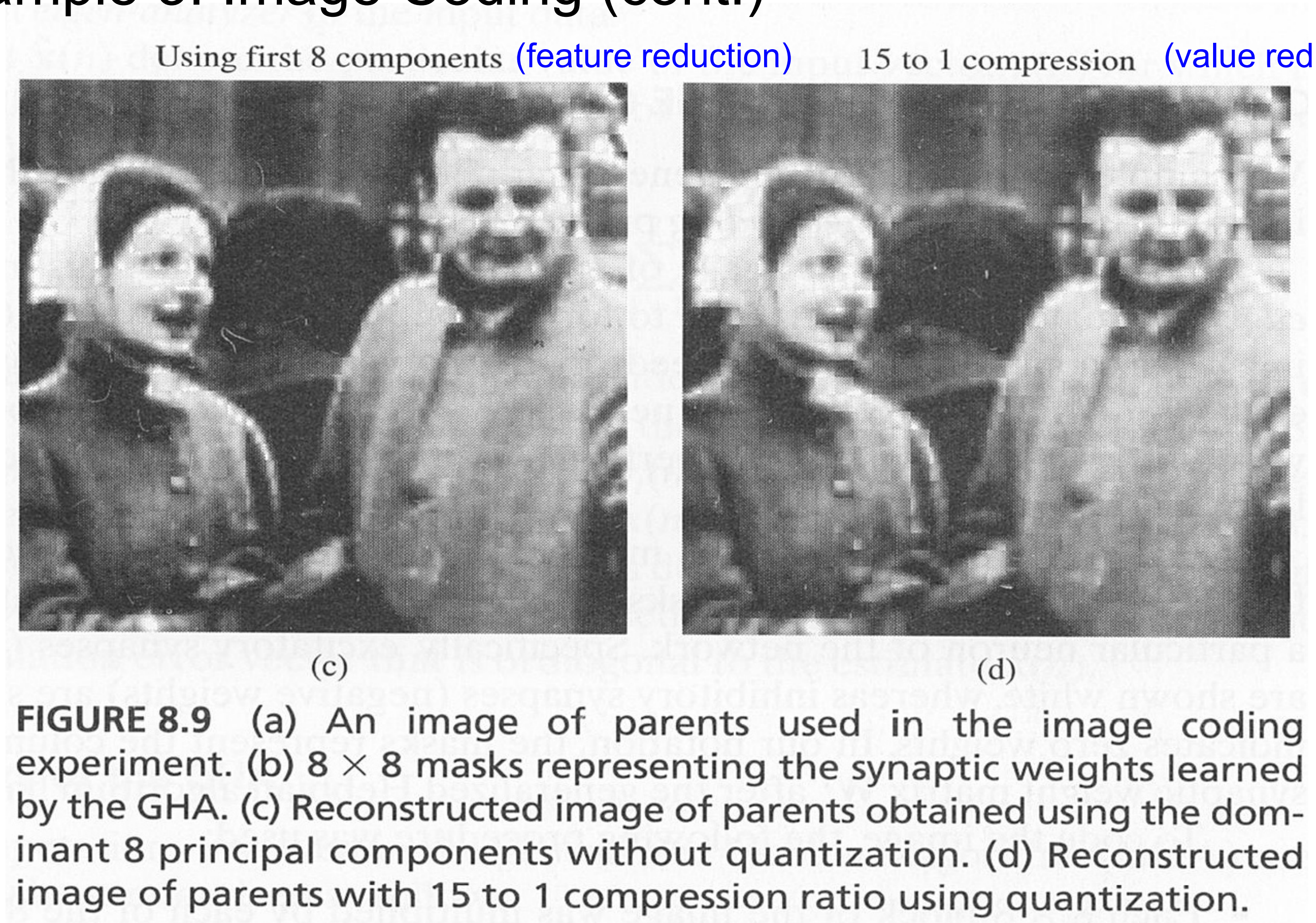
PCA Examples: Image Coding (1/2)

- Example 3: Image Coding



PCA Examples: Image Coding (2/2)

- Example 3: Image Coding (cont.)



Using first 8 components (feature reduction)

15 to 1 compression (value reduction)

(c)

(d)

FIGURE 8.9 (a) An image of parents used in the image coding experiment. (b) 8 × 8 masks representing the synaptic weights learned by the GHA. (c) Reconstructed image of parents obtained using the dominant 8 principal components without quantization. (d) Reconstructed image of parents with 15 to 1 compression ratio using quantization.

PCA Examples: Eigenface (1/4)

- Example 4: Eigenface in face recognition (Turk and Pentland, 1991)
  - Consider an individual image to be a linear combination of a small number of face components or “eigenfaces” derived from a set of reference images

$$\mathbf{x}_1 = \begin{bmatrix} x_{1,1} \\ x_{1.2} \\ . \\ . \\ x_{1.n} \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} x_{2,1} \\ x_{2.2} \\ . \\ . \\ x_{2.n} \end{bmatrix}, \ldots\ldots, \quad \mathbf{x}_L = \begin{bmatrix} x_{L,1} \\ x_{L.2} \\ . \\ . \\ x_{L.n} \end{bmatrix}$$

  - Steps
    - Convert each of the *L* reference images into a vector of floating point numbers representing light intensity in each pixel
    - Calculate the coverance/correlation matrix between these reference vectors
    - Apply Principal Component Analysis (PCA) find the eigenvectors of the matrix: the eigenfaces
    - Besides, the vector obtained by averaging all images are called “eigenface 0”. The other eigenfaces from “eigenface 1” onwards model the variations from this average face

PCA Examples: Eigenface (2/4)

- Example 4: Eigenface in face recognition (cont.)
  - Steps
    - Then the faces are then represented as eigenvoice 0 plus a linear combination of the remain *K* (*K* ≤ *L*) eigenfaces
  - The Eigenface approach persists the minimum mean-squared error criterion
  - Incidentally, the eigenfaces are not only themselves usually plausible faces, but also directions of variations between faces

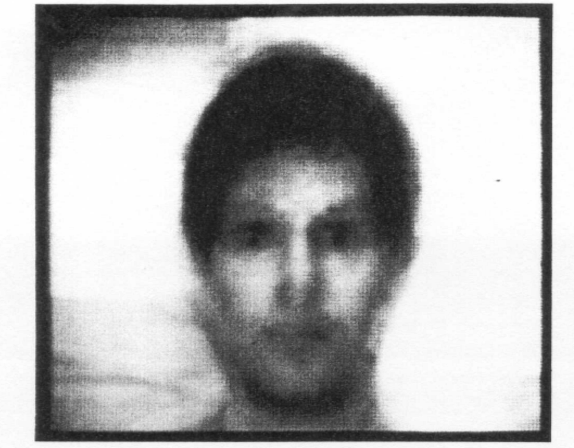
$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + w_{i,1}\mathbf{e}(1) + w_{i,2}\mathbf{e}(2) + \ldots.. + w_{i,K}\mathbf{e}(K)$$

$$\Rightarrow \boxed{\mathbf{y}_i = \left[1, w_{i,1}, w_{i,2}, \ldots, w_{i,K}\right]}$$

Feature vector of a person i

PCA Examples: Eigenface (3/4)

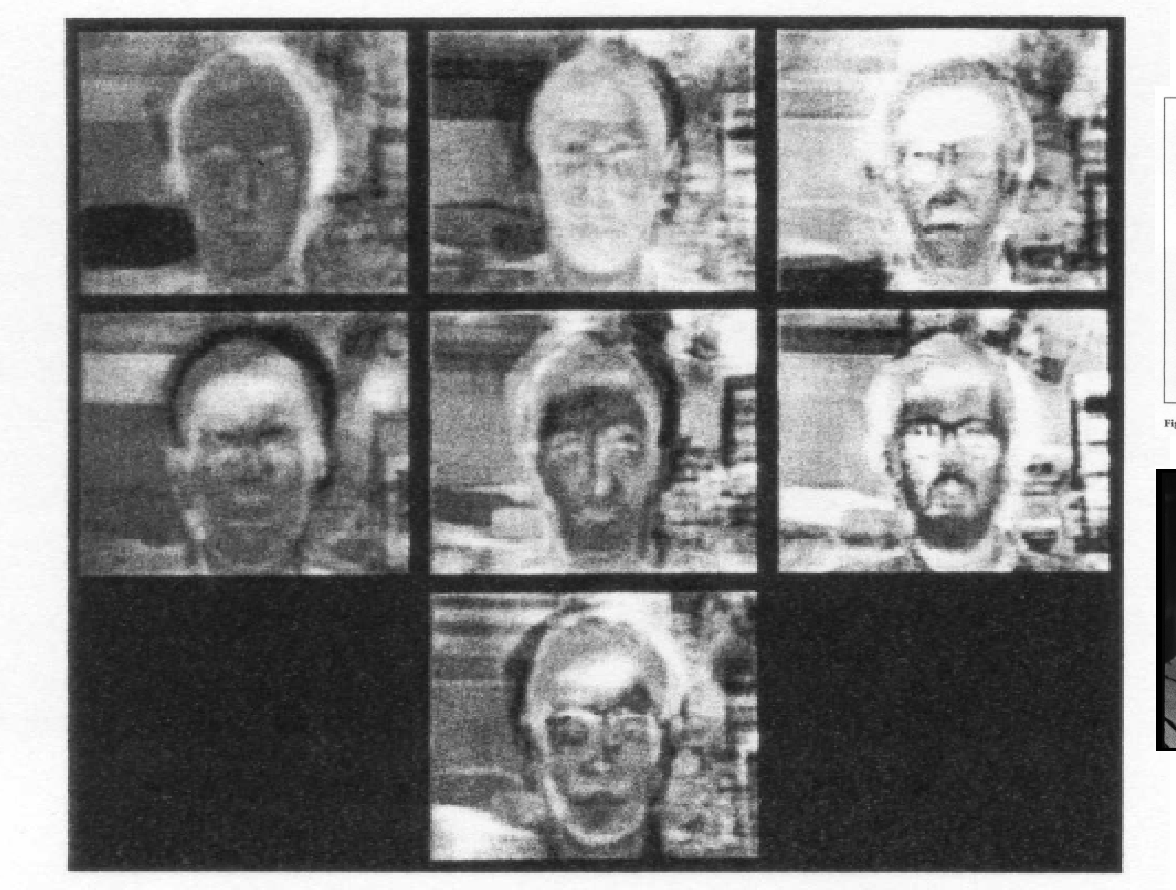
Face images as the training set



The averaged face

PCA Examples: Eigenface (4/4)

Seven eigenfaces derived from the training set



(Indicate directions of variations between faces)

A projected face image

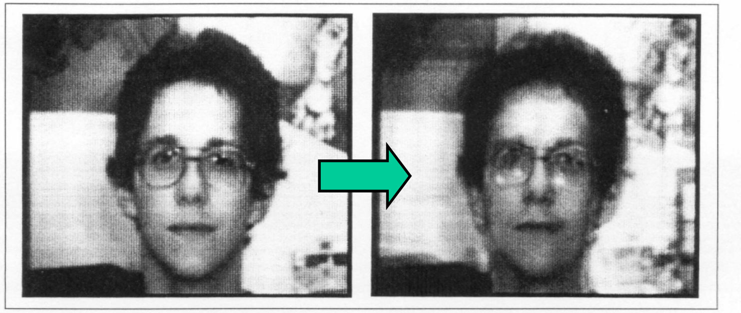
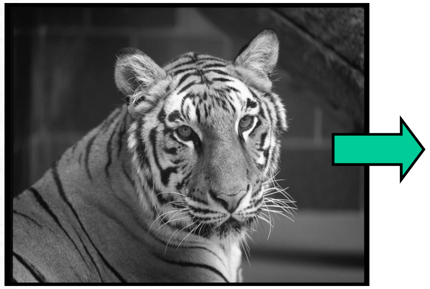


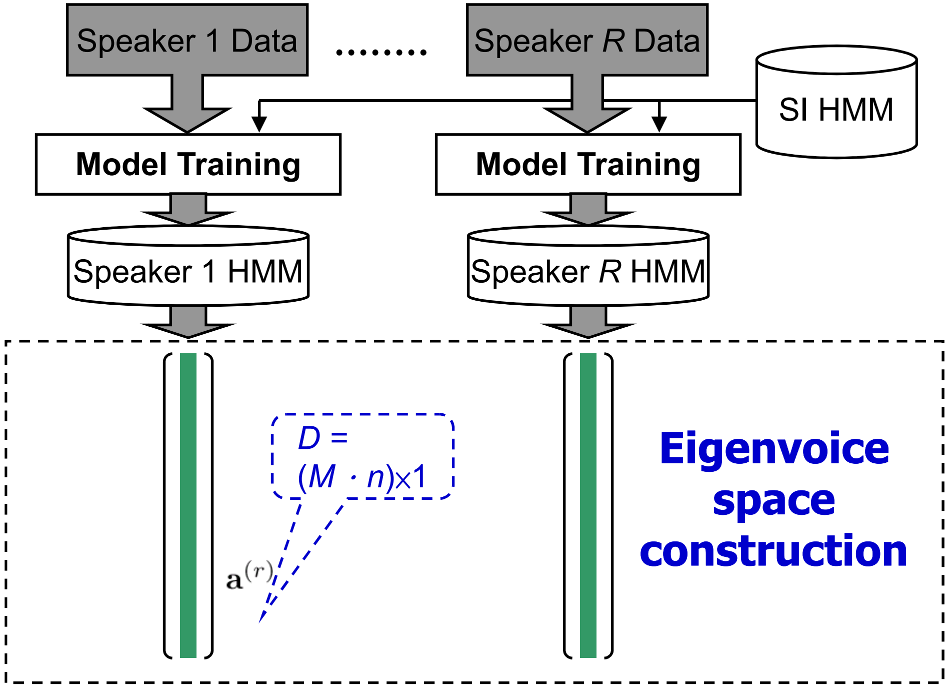
Figure 3. An original face image and its projection onto the face space defined by the eigenfaces of Figure 2.



?

PCA Examples: Eigenvoice (1/3)

- Example 5: Eigenvoice in speaker adaptation (PSTL, 2000)
  - Steps
    - Concatenating the regarded parameters for each speaker *r* to form a huge vector **a**$^{(r)}$ (a supervectors)
    - SD HMM model mean parameters (μ)



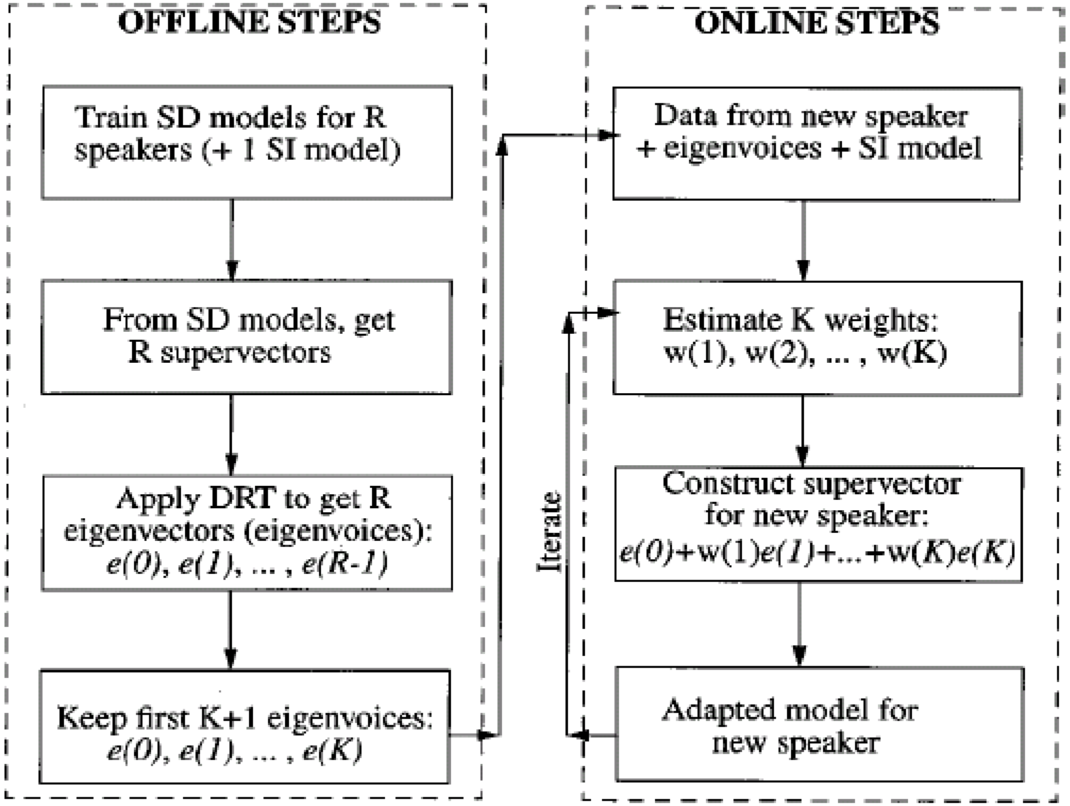
Each new speaker S is represented by a point *P* in *K*-space

$$\mathbf{P}_i = \underset{\overset{\uparrow}{\text{SI HMM model}}}{\mathbf{e}(0)} + w_{i,1}\mathbf{e}(1) + w_{i,2}\mathbf{e}(2) + ... + w_{i,K}\mathbf{e}(K)$$

⇨ **Principal Component Analysis**

PCA Examples: Eigenvoice (2/3)

- Example 4: Eigenvoice in speaker adaptation (cont.)



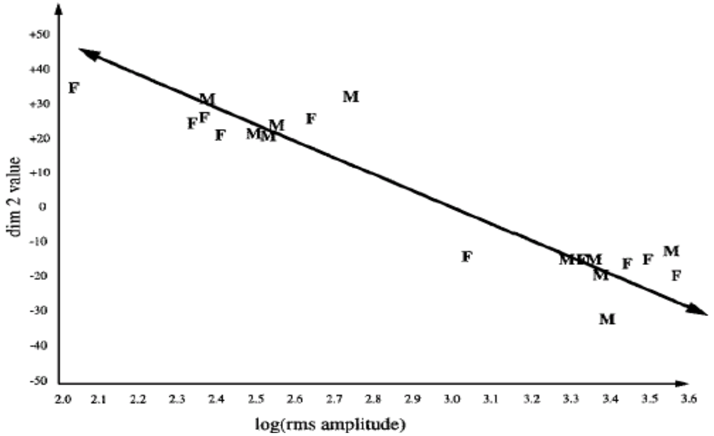
OFFLINE STEPS
Train SD models for R speakers (+ 1 SI model)
From SD models, get R supervectors
Apply DRT to get R eigenvectors (eigenvoices): e(0), e(1), ... , e(R-1)
Keep first K+1 eigenvoices: e(0), e(1), ... , e(K)
ONLINE STEPS
Data from new speaker + eigenvoices + SI model
Estimate K weights: w(1), w(2), ... , w(K)
Construct supervector for new speaker: e(0)+w(1)e(1)+...+w(K)e(K)
Adapted model for new speaker
Iterate

Fig. 1. Block diagram for eigenvoice speaker adaptation

PCA Examples: Eigenvoice (3/3)

- Example 5: Eigenvoice in speaker adaptation (cont.)
  - Dimension 1 (eigenvoice 1):
    - Correlate with pitch or sex
  - Dimension 2 (eigenvoice 2):
    - Correlate with amplitude
  - Dimension 3 (eigenvoice 3):
    - Correlate with second-formant movement

Note that:
Eigenface performs on feature space while eigenvoice performs on model space



F
e

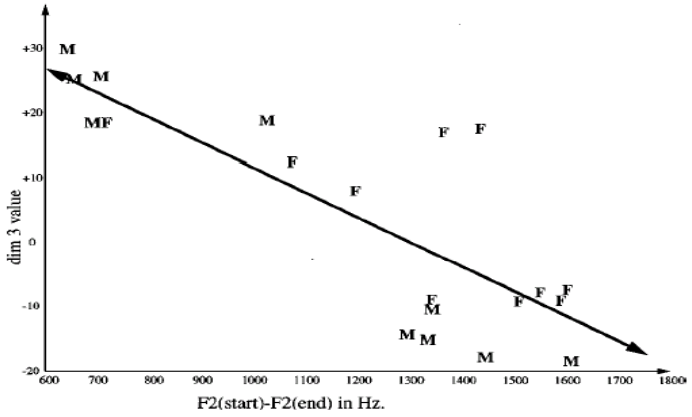


Fig. 4. Dimension 3 versus F2(start)–F2(end) for "U," extreme M and F in each speaker set

Linear Discriminant Analysis (LDA) (1/2)

- Also called
  - Fisher’s Linear Discriminant Analysis, Fisher-Rao Linear Discriminant Analysis
    - Fisher (1936): introduced it for two-class classification
    - Rao (1965): extended it to handle multiple-class classification

Linear Discriminant Analysis (LDA) (2/2)

- Given a set of sample vectors with labeled (class) information, try to find a linear transform ***W*** such that the ratio of **average between-class variation** over **average within-class variation** is maximal

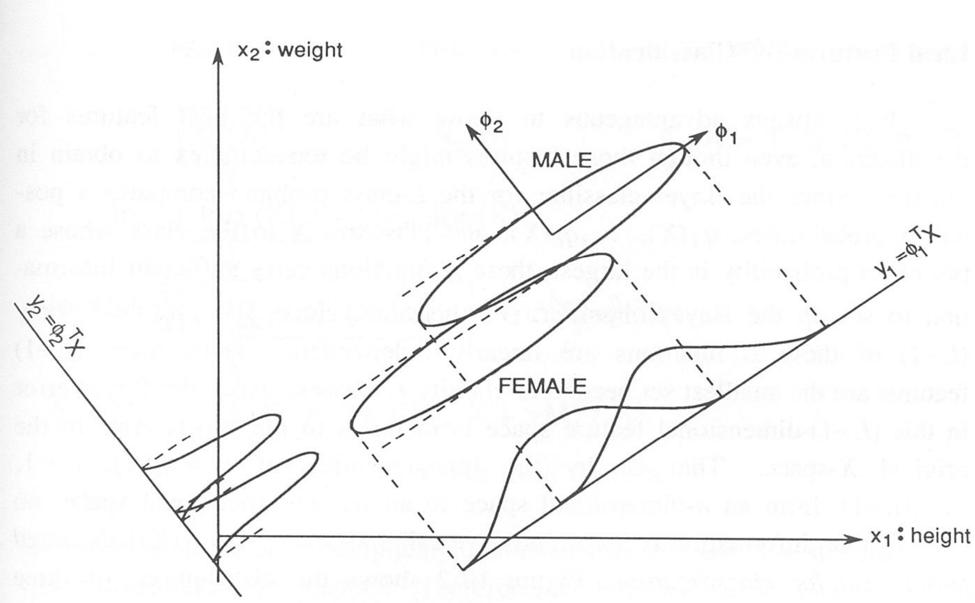


Fig. 10-1 An example of feature extraction for classification.

Within-class distributions are assumed here to be Gaussians With equal variance in the two-dimensional sample space

LDA Derivations (1/4)

- Suppose there are *N* sample vectors $\boldsymbol{x}_i$ with dimensionality *n*, each of them is belongs to one of the *J* classes $g(\boldsymbol{x}_i) = j,\quad j \in \{1,2,....,J\}, g(\cdot) \text{ is class index}$
  - The sample mean is: $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$
  - The class sample means are: $\bar{\boldsymbol{x}}_j = \frac{1}{N_j}\sum_{g(\boldsymbol{x}_i)=j} \boldsymbol{x}_i$
  - The class sample covariances are: $\boldsymbol{\Sigma}_j = \frac{1}{N_j}\sum_{g(\boldsymbol{x}_i)=j} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T$
  - The **average within-class variation** before transform

$$\boldsymbol{S}_w = \frac{1}{N}\sum_j N_j \boldsymbol{\Sigma}_j$$

  - The **average between-class variation** before transform

$$\boldsymbol{S}_b = \frac{1}{N}\sum_j N_j (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T$$

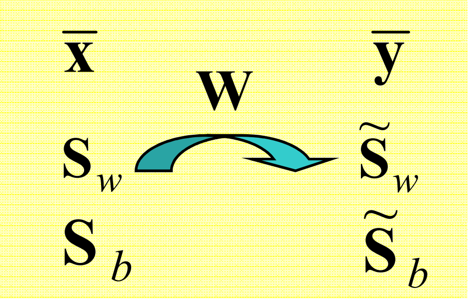
LDA Derivations (2/4)

- If the transform $\boldsymbol{W} = \left[\boldsymbol{w}_1 \boldsymbol{w}_2 \ldots . \boldsymbol{w}_m\right]$ is applied
  - The sample vectors will be $\boldsymbol{y}_i = \boldsymbol{W}^T \boldsymbol{x}_i$
  - The sample mean will be $\overline{\boldsymbol{y}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{W}^T \boldsymbol{x}_i = \boldsymbol{W}^T\left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i\right) = \boldsymbol{W}^T \overline{\boldsymbol{x}}$
  - The class sample means will be $\overline{\boldsymbol{y}}_j = \frac{1}{N_j}\sum_{g(\boldsymbol{x}_i)=j} \boldsymbol{W}^T \boldsymbol{x}_i = \boldsymbol{W}^T \overline{\boldsymbol{x}}_j$
  - The **average within-class variation** will be

$$\tilde{\boldsymbol{S}}_w = \frac{1}{N}\sum_j N_j \left\{ \frac{1}{N_j} \cdot \sum_{g(\boldsymbol{x}_i)=j} \left(\boldsymbol{W}^T \boldsymbol{x}_i - \frac{1}{N_j} \sum_{g(\boldsymbol{x}_i)=j} \left(\boldsymbol{W}^T \boldsymbol{x}_i \right) \right) \left(\boldsymbol{W}^T \boldsymbol{x}_i - \frac{1}{N_j} \sum_{g(\boldsymbol{x}_i)=j} \left(\boldsymbol{W}^T \boldsymbol{x}_i \right) \right)^T \right\}$$

$$= \boldsymbol{W}^T \left\{ \frac{1}{N} \sum_j N_j \boldsymbol{\Sigma}_j \right\} \boldsymbol{W}$$

$$= \boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}$$



LDA Derivations (3/4)

- If the transform $\boldsymbol{W} = [\boldsymbol{w}_1 \boldsymbol{w}_2 \ldots . \boldsymbol{w}_m]$ is applied
  - Similarly, the **average between-class variation** will be

$$\tilde{\boldsymbol{S}}_b = \boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}$$

  - Try to find optimal $\boldsymbol{W}$ such that the following objective function is maximized

$$J(\boldsymbol{W}) = \frac{\left|\tilde{\boldsymbol{S}}_b\right|}{\left|\tilde{\boldsymbol{S}}_w\right|} = \frac{\left|\boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}\right|}{\left|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}\right|}$$

    - A closed-form solution: the column vectors of an optimal matrix $\boldsymbol{W}$ are the generalized eigenvectors corresponding to the largest eigenvalues in

$$\boldsymbol{S}_b \boldsymbol{w}_i = \lambda_i \boldsymbol{S}_w \boldsymbol{w}_i$$

    - That is, $\boldsymbol{w}_i$'s are the eigenvectors corresponding to the largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$

$$\boxed{\boldsymbol{S}_w^{-1} \boldsymbol{S}_b} \boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$$

LDA Derivations (4/4)

- Proof:

determinant

$$\because \hat{\boldsymbol{W}} = \underset{\hat{\boldsymbol{W}}}{\arg\max}\, J(\boldsymbol{W}) = \underset{\hat{\boldsymbol{W}}}{\arg\max} \frac{\left|\tilde{\boldsymbol{S}}_b\right|}{\left|\tilde{\boldsymbol{S}}_w\right|} = \underset{\hat{\boldsymbol{W}}}{\arg\max} \frac{\left|\boldsymbol{W}^T \boldsymbol{S}_b \boldsymbol{W}\right|}{\left|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}\right|}$$

Or equivalently, for each column vector $\boldsymbol{w}_i$ of $\boldsymbol{W}$, we want to find that :

$$\left(\frac{\mathbf{F}}{\mathbf{G}}\right)' = \frac{\mathbf{F}'\mathbf{G} - \mathbf{G}'\mathbf{F}}{\mathbf{G}^2}$$

The qradtic form has optimal solution : $\lambda_i = \dfrac{\boldsymbol{w}_i^T \boldsymbol{S}_b \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i}$

$$\frac{d\ (\mathbf{x}^{\mathbf{T}}\mathbf{C}\mathbf{x})}{d\mathbf{x}} = \left(\mathbf{C} + \mathbf{C}^{\mathbf{T}}\right)\mathbf{x}$$

$$\Rightarrow \frac{\partial \lambda_i}{\partial \boldsymbol{w}_i} = \frac{2\boldsymbol{S}_b \boldsymbol{w}_i \left(\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i\right) - 2\boldsymbol{S}_w \boldsymbol{w}_i \left(\boldsymbol{w}_i^T \boldsymbol{S}_b \boldsymbol{w}_i\right)}{\left(\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i\right)^2} = 0$$

$$\Rightarrow \frac{\boldsymbol{S}_b \boldsymbol{w}_i \left(\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i\right)}{\left(\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i\right)^2} - \frac{\boldsymbol{S}_w \boldsymbol{w}_i \left(\boldsymbol{w}_i^T \boldsymbol{S}_b \boldsymbol{w}_i\right)}{\left(\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i\right)^2} = 0$$

$$\frac{\boldsymbol{S}_b \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i} - \frac{\boldsymbol{S}_w \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i} \lambda_i = 0 \quad \left(\because \lambda_i = \frac{\boldsymbol{w}_i^T \boldsymbol{S}_b \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_w \boldsymbol{w}_i}\right)$$

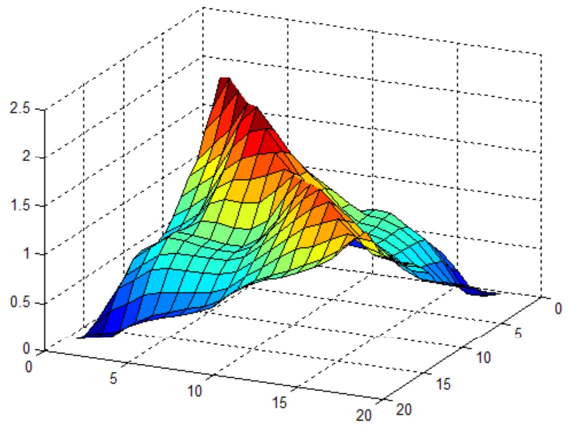
$$\Rightarrow \boldsymbol{S}_b \boldsymbol{w}_i - \lambda_i \boldsymbol{S}_w \boldsymbol{w}_i = 0 \Rightarrow \boldsymbol{S}_b \boldsymbol{w}_i = \lambda_i \boldsymbol{S}_w \boldsymbol{w}_i$$

$$\Rightarrow \boldsymbol{S}_w^{-1} \boldsymbol{S}_b \boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$$

LDA Examples: Feature Transformation (1/2)

- Example1: Experiments on Speech Signal Processing

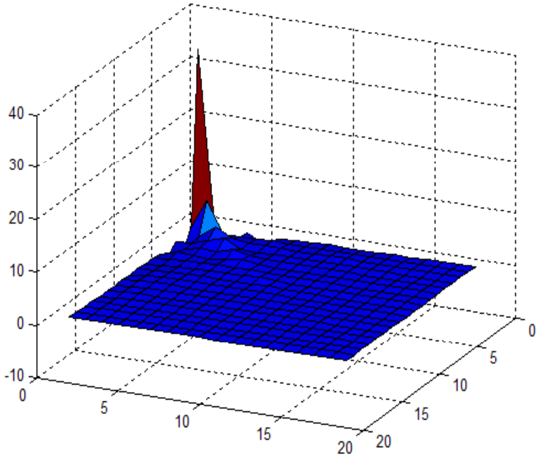
Covariance Matrix of the 18-Mel-filter-bank vectors



Calculated using Year-99’s 5471 files

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{\mathbf{x}_i}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$$

Covariance Matrix of the 18-cepstral vectors



Calculated using Year-99’s 5471 files

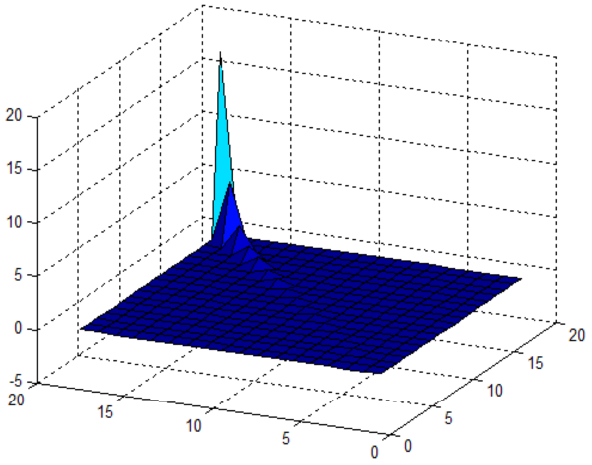
$$\mathbf{\Sigma}' = \frac{1}{N}\sum_{\mathbf{y}_i}(\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})^T$$

After Cosine Transform

LDA Examples: Feature Transformation (2/2)

- Example1: Experiments on Speech Signal Processing (cont.)

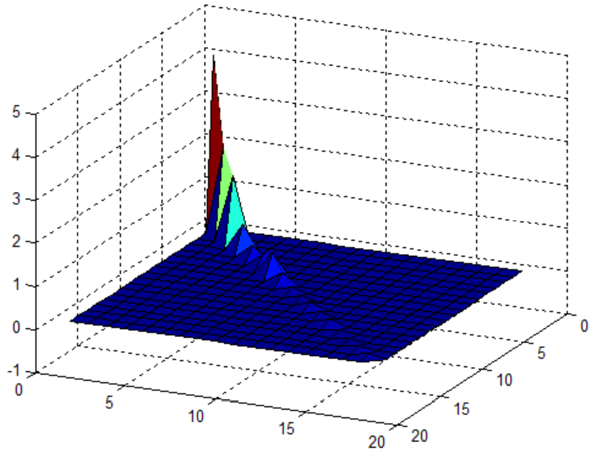
Covariance Matrix of the 18-PCA-cepstral vectors



Calculated using Year-99’s 5471 files

After PCA Transform

Covariance Matrix of the 18-LDA-cepstral vectors



Calculated using Year-99’s 5471 files

After LDA Transform

	Character Error Rate	
	TC	WG
MFCC	26.32	22.71
LDA-1	23.12	20.17
LDA-2	23.11	20.11

PCA vs. LDA (1/2)

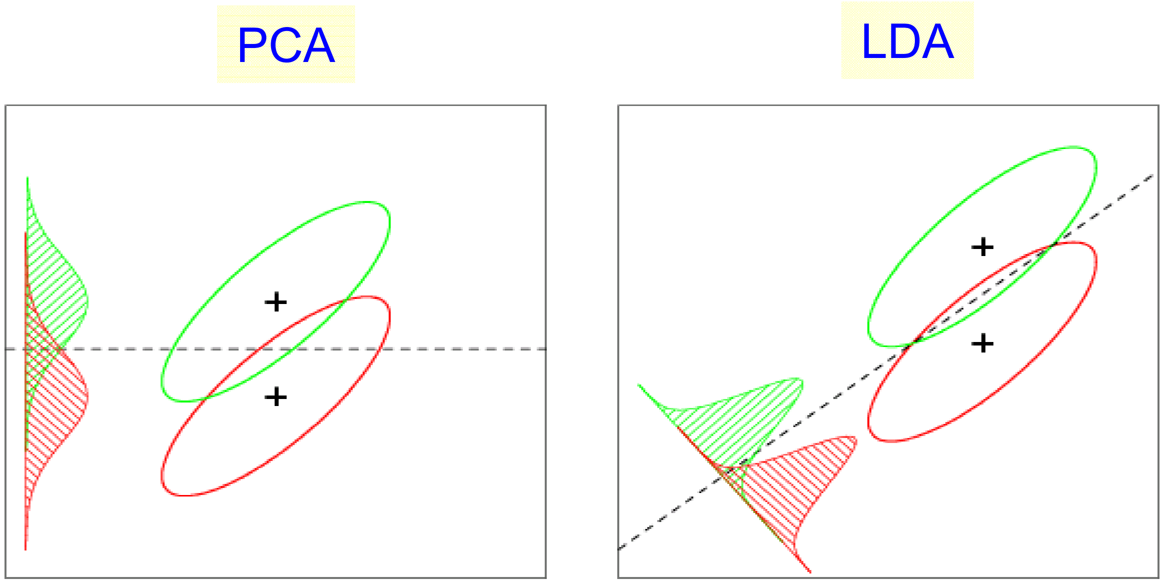


Figure 4.9: *Although the line joining the centroids defines the direction of greatest centroid spread, the projected data overlap because of the covariance (left panel). The discriminant direction minimizes this overlap for Gaussian data (right panel).*

Heteroscedastic Discriminant Analysis (HDA)

- HDA: Heteroscedastic Discriminant Analysis
  - The difference in the projections obtained from LDA and HDA for 2-class case

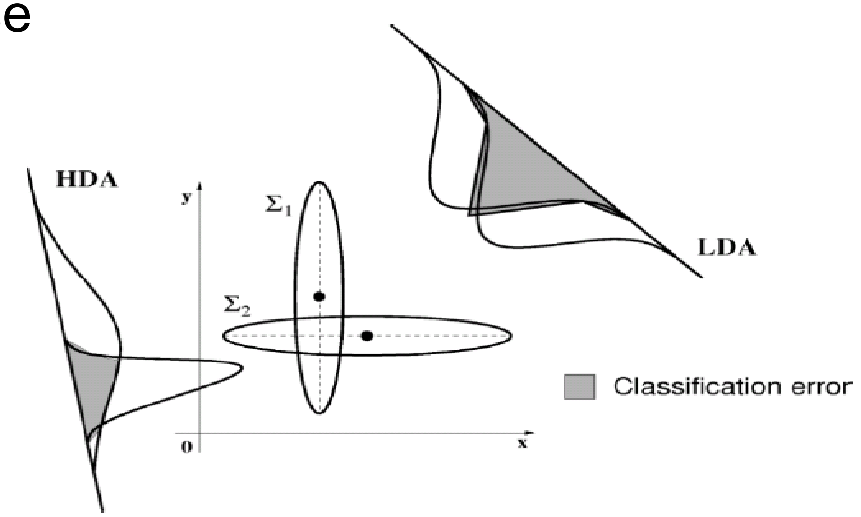


Fig. 1. Difference between LDA and HDA.

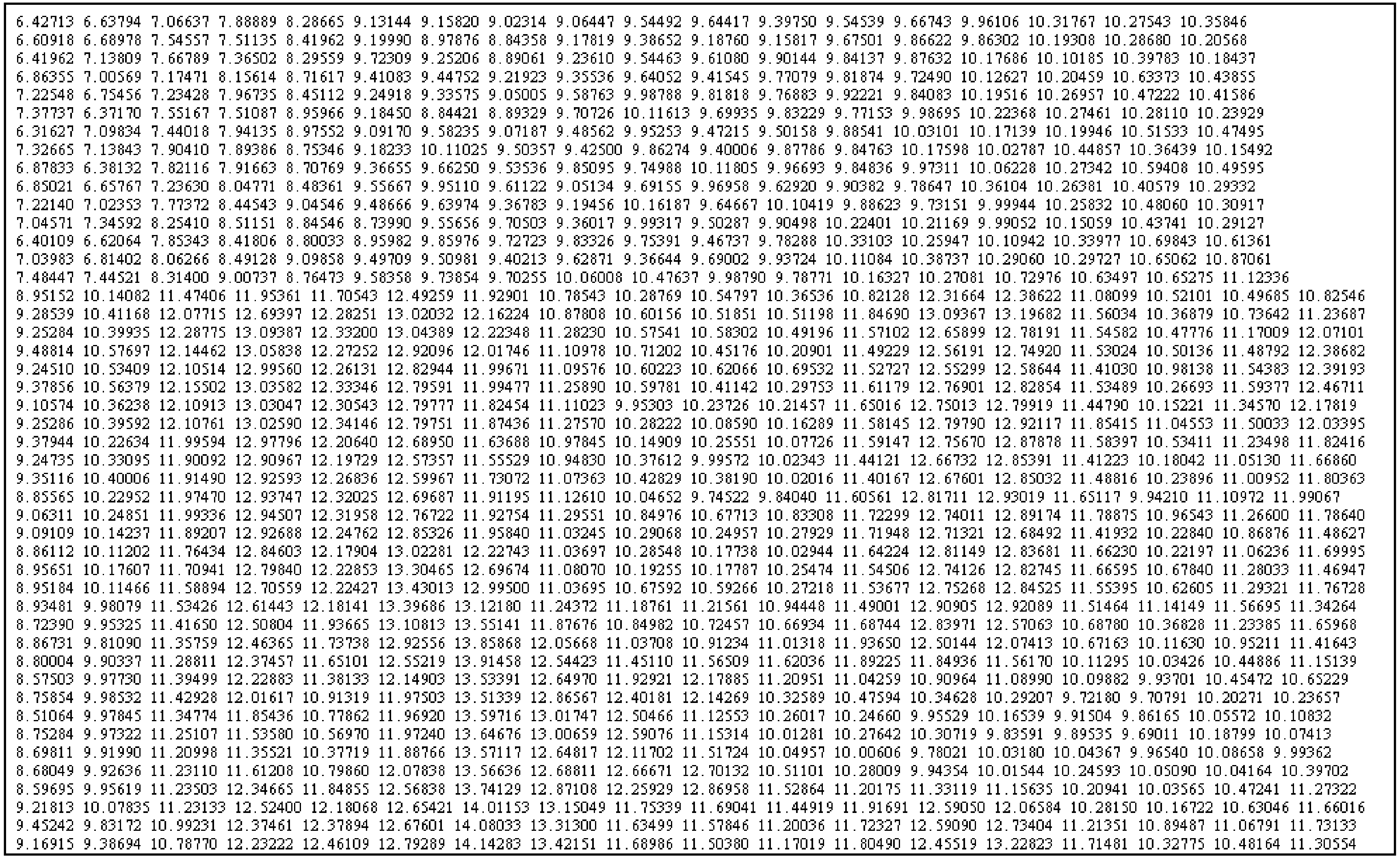
- Clearly, the HDA provides a much lower classification error than LDA theoretically
  - However, most statistical modeling approaches assume data samples are Gaussian and have **diagonal** covariance matrices

HW: Feature Transformation (1/4)

- Given two data sets (MaleData, Female Data) in which each row is a sample with 39 features, please perform the following operations:
  1. Merge these two data sets and find/plot the covariance matrix for the merged data set.
  2. Apply PCA and LDA transformations to the merged data set, respectively. Also, find/plot the covariance matrices for transformations, respectively. Describe the phenomena that you have observed.
  3. Use the first two principal components of PCA as well as the first two eigenvectors of LDA to represent the merged data set. Selectively plot portions of samples from MaleData and FemaleData, respectively. Describe the phenomena that you have observed.

http://berlin.csie.ntnu.edu.tw/PastCourses/2004S-MachineLearningandDataMining/Homework/HW-1/MaleData.txt

HW: Feature Transformation (2/4)



```
6.42713 6.63794 7.06637 7.88889 8.28665 9.13144 9.15820 9.02314 9.06447 9.54492 9.64417 9.39750 9.54539 9.66743 9.96106 10.31767 10.27543 10.35846
6.60918 6.68978 7.54557 7.51135 8.41962 9.19990 8.97876 8.84358 9.17819 9.38652 9.18760 9.15817 9.67501 9.86622 9.86302 10.19308 10.28680 10.20568
6.41962 7.13809 7.66789 7.36502 8.29559 9.72309 9.25206 8.89061 9.23610 9.54463 9.61080 9.90144 9.84137 9.87632 10.17686 10.10185 10.39783 10.18437
6.86355 7.00569 7.17471 8.15614 8.71617 9.41083 9.44752 9.21923 9.35536 9.64052 9.41545 9.77079 9.81874 9.72490 10.12627 10.20459 10.63373 10.43855
7.22548 6.75456 7.23428 7.96735 8.45112 9.24918 9.33575 9.05005 9.58763 9.98788 9.81818 9.76883 9.92221 9.84083 10.19516 10.26957 10.47222 10.41586
7.37737 6.37170 7.55167 7.51087 8.95966 9.18450 8.84421 8.89329 9.70726 10.11613 9.69935 9.83229 9.77153 9.98695 10.22368 10.27461 10.28110 10.23929
6.31627 7.09834 7.44018 7.94135 8.97552 9.09170 9.58235 9.07187 9.48562 9.95253 9.47215 9.50158 9.88541 10.03101 10.17139 10.19946 10.51533 10.47495
7.32665 7.13843 7.90410 7.89386 8.75346 9.18233 10.11025 9.50357 9.42500 9.86274 9.40006 9.87786 9.84763 10.17598 10.02787 10.44857 10.36439 10.15492
6.87833 6.38132 7.82116 7.91663 8.70769 9.36655 9.66250 9.53536 9.85095 9.74988 10.11805 9.96693 9.84836 9.97311 10.06228 10.27342 10.59408 10.49595
6.85021 6.65767 7.23630 8.04771 8.48361 9.55667 9.95110 9.61122 9.05134 9.69155 9.96958 9.62920 9.90382 9.78647 10.36104 10.26381 10.40579 10.29332
7.22140 7.02353 7.77372 8.44543 9.04546 9.48666 9.63974 9.36783 9.19456 10.16187 9.64667 10.10419 9.88623 9.73151 9.99944 10.25832 10.48060 10.30917
7.04571 7.34592 8.25410 8.51151 8.84546 8.73990 9.55656 9.70503 9.36017 9.99317 9.50287 9.90498 10.22401 10.21169 9.99052 10.15059 10.43741 10.29127
6.40109 6.62064 7.85343 8.41806 8.80033 8.95982 9.85976 9.72723 9.83326 9.75391 9.46737 9.78288 10.33103 10.25947 10.10942 10.33977 10.69843 10.61361
7.03983 6.81402 8.06266 8.49128 9.09858 9.49709 9.50981 9.40213 9.62871 9.36644 9.69002 9.93724 10.11084 10.38737 10.29060 10.29727 10.65062 10.87061
7.48447 7.44521 8.31400 9.00737 8.76473 9.58358 9.73854 9.70255 10.06008 10.47637 9.98790 9.78771 10.16327 10.27081 10.72976 10.63497 10.65275 11.12336
8.95152 10.14082 11.47406 11.95361 11.70543 12.49259 11.92901 10.78543 10.28769 10.54797 10.36536 10.82128 12.31664 12.38622 11.08099 10.52101 10.49685 10.82546
9.28539 10.41168 12.07715 12.69397 12.28251 13.02032 12.16224 10.87808 10.60156 10.51851 10.51198 11.84690 13.09367 13.19682 11.56034 10.36879 10.73642 11.23687
9.25284 10.39935 12.28775 13.09387 12.33200 13.04389 12.22348 11.28230 10.57541 10.58302 10.49196 11.57102 12.65899 12.78191 11.54582 10.47776 11.17009 12.07101
9.48814 10.57697 12.14462 13.05838 12.27252 12.92096 12.01746 11.10978 10.71202 10.45176 10.20901 11.49229 12.56191 12.74920 11.53024 10.50136 11.48792 12.38682
9.24510 10.53409 12.10514 12.99560 12.26131 12.82944 11.99671 11.09576 10.60223 10.62066 10.69532 11.52727 12.55299 12.58644 11.41030 10.98138 11.54383 12.39193
9.37856 10.56379 12.15502 13.03582 12.33346 12.79591 11.99477 11.25890 10.59781 10.41142 10.29753 11.61179 12.76901 12.82854 11.53489 10.26693 11.59377 12.46711
9.10574 10.36238 12.10913 13.03047 12.30543 12.79777 11.82454 11.11023 9.95303 10.23726 10.21457 11.65016 12.75013 12.79919 11.44790 10.15221 11.34570 12.17819
9.25286 10.39592 12.10761 13.02590 12.34146 12.79751 11.87436 11.27570 10.28222 10.08590 10.16289 11.58145 12.79790 12.92117 11.85415 11.04553 11.50033 12.03395
9.37944 10.22634 11.99594 12.97796 12.20640 12.68950 11.63688 10.97845 10.14909 10.25551 10.07726 11.59147 12.75670 12.87878 11.58397 10.53411 11.23498 11.82416
9.24735 10.33095 11.90092 12.90967 12.19729 12.57357 11.55529 10.94830 10.37612 9.99572 10.02343 11.44121 12.66732 12.85391 11.41223 10.18042 11.05130 11.66860
9.35116 10.40006 11.91490 12.92593 12.26836 12.59967 11.73072 11.07363 10.42829 10.38190 10.02016 11.40167 12.67601 12.85032 11.48816 10.23896 11.00952 11.80363
8.85565 10.22952 11.97470 12.93747 12.32025 12.69687 11.91195 11.12610 10.04652 9.74522 9.84040 11.60561 12.81711 12.93019 11.65117 9.94210 11.10972 11.99067
9.06311 10.24851 11.99336 12.94507 12.31958 12.76722 11.92754 11.29551 10.84976 10.67713 10.83308 11.72299 12.74011 12.89174 11.78875 10.96543 11.26600 11.78640
9.09109 10.14237 11.89207 12.92688 12.24762 12.85326 11.95840 11.03245 10.29068 10.24957 10.27929 11.71948 12.71321 12.68492 11.41932 10.22840 10.86876 11.48627
8.86112 10.11202 11.76434 12.84603 12.17904 13.02281 12.22743 11.03697 10.28548 10.17738 10.02944 11.64224 12.81149 12.83681 11.66230 10.22197 11.06236 11.69995
8.95651 10.17607 11.70941 12.79840 12.22853 13.30465 12.69674 11.08070 10.19255 10.17787 10.25474 11.54506 12.74126 12.82745 11.66595 10.67840 11.28033 11.46947
8.95184 10.11466 11.58894 12.70559 12.22427 13.43013 12.99500 11.03695 10.67592 10.59266 10.27218 11.53677 12.75268 12.84525 11.55395 10.62605 11.29321 11.76728
8.93481 9.98079 11.53426 12.61443 12.18141 13.39686 13.12180 11.24372 11.18761 11.21561 10.94448 11.49001 12.90905 12.92089 11.51464 11.14149 11.56695 11.34264
8.72390 9.95325 11.41650 12.50804 11.93665 13.10813 13.55141 11.87676 10.84982 10.72457 10.66934 11.68744 12.83971 12.57063 10.68780 10.36828 11.23385 11.65968
8.86731 9.81090 11.35759 12.46365 11.73738 12.92556 13.85868 12.05668 11.03708 10.91234 11.01318 11.93650 12.50144 12.07413 10.67163 10.11630 10.95211 11.41643
8.80004 9.90337 11.28811 12.37457 11.65101 12.55219 13.91458 12.54423 11.45110 11.56509 11.62036 11.89225 11.84936 11.56170 10.11295 10.03426 10.44886 11.15139
8.57503 9.97730 11.39499 12.22883 11.38133 12.14903 13.53391 12.64970 11.92921 12.17885 11.20951 11.04259 10.90964 11.08990 10.09882 9.93701 10.45472 10.65229
8.75854 9.98532 11.42928 12.01617 10.91319 11.97503 13.51339 12.86567 12.40181 12.14269 10.32589 10.47594 10.34628 10.29207 9.72180 9.70791 10.20271 10.23657
8.51064 9.97845 11.34774 11.85436 10.77862 11.96920 13.59716 13.01747 12.50466 11.12553 10.26017 10.24660 9.95529 10.16539 9.91504 9.86165 10.05572 10.10832
8.75284 9.97322 11.25107 11.53580 10.56970 11.97240 13.64676 13.00659 12.59076 11.15314 10.01281 10.27642 10.30719 9.83591 9.89535 9.69011 10.18799 10.07413
8.69811 9.91990 11.20998 11.35521 10.37719 11.88766 13.57117 12.64817 12.11702 11.51724 10.04957 10.00606 9.78021 10.03180 10.04367 9.96540 10.08658 9.99362
8.68049 9.92636 11.23110 11.61208 10.79860 12.07838 13.56636 12.68811 12.66671 12.70132 10.51101 10.28009 9.94354 10.01544 10.24593 10.05090 10.04164 10.39702
8.59695 9.95619 11.23503 12.34665 11.84855 12.56838 13.74129 12.87108 12.25929 12.86958 11.52864 11.20175 11.33119 11.15635 10.20941 10.03565 10.47241 11.27322
9.21813 10.07835 11.23133 12.52400 12.18068 12.65421 14.01153 13.15049 11.75339 11.69041 11.44919 11.91691 12.59050 12.06584 10.28150 10.16722 10.63046 11.66016
9.45242 9.83172 10.99231 12.37461 12.37894 12.67601 14.08033 13.31300 11.63499 11.57846 11.20036 11.72327 12.59090 12.73404 11.21351 10.89487 11.06791 11.73133
9.16915 9.38694 10.78770 12.23222 12.46109 12.79289 14.14283 13.42151 11.68986 11.50380 11.17019 11.80490 12.45519 13.22823 11.71481 10.32775 10.48164 11.30554
```

HW: Feature Transformation (3/4)

- Plot Covariance Matrix

```
CoVar=[
            3.0      0.5     0.4;
            0.9      6.3     0.2;
            0.4      0.4     4.2;
];
colormap('default');
surf(CoVar);
```

- Eigen Decomposition

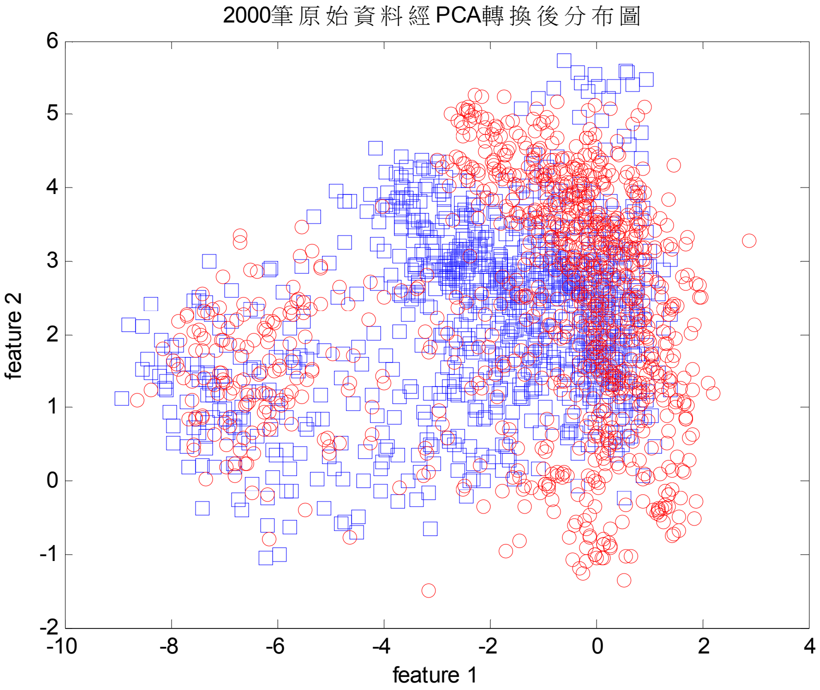
```
BE=[
            3.0      3.5     1.4;
            1.9      6.3     2.2;
            2.4      0.4     4.2;
];

WI=[
            4.0      4.1     2.1;
            2.9      8.7     3.5;
            4.4      3.2     4.3;
];
```

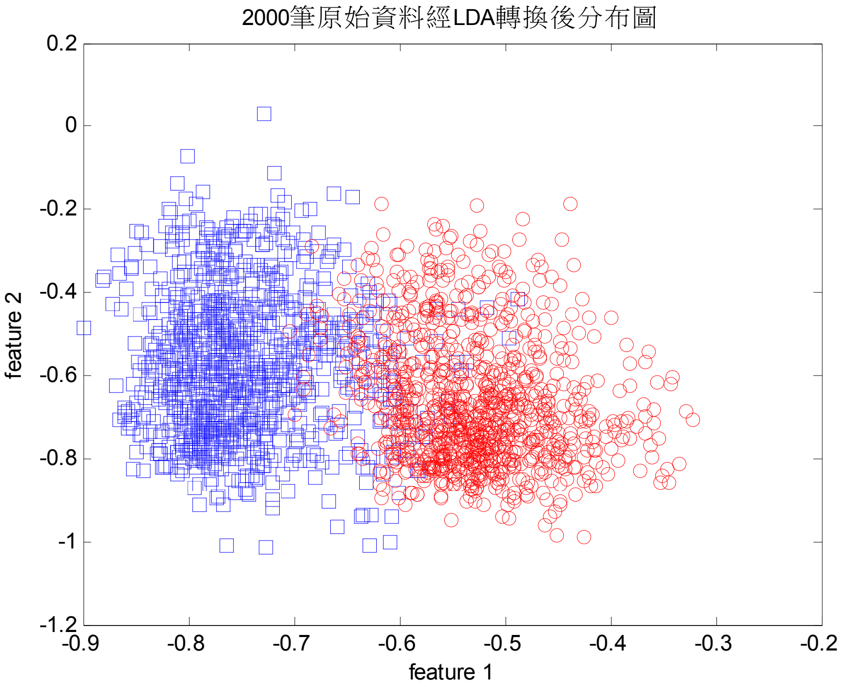
⇨

```
%LDA
IWI=inv(WI);
A=IWI*BE;
%PCA
A=BE+WI; % why ??  ( Prove it! )

[V,D]=eig(A);
[V,D]=eigs(A,3);

fid=fopen('Basis','w');
for i=1:3 % feature vector length
  for j=1:3  % basis number
   fprintf(fid,'%10.10f ',V(i,j));
  end
  fprintf(fid,'\n');
end
fclose(fid);
```

HW: Feature Transformation (4/4)

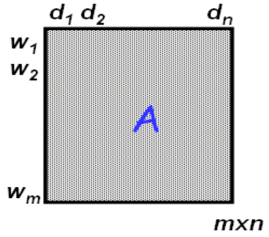
- Examples





Latent Semantic Analysis (LSA) (1/7)

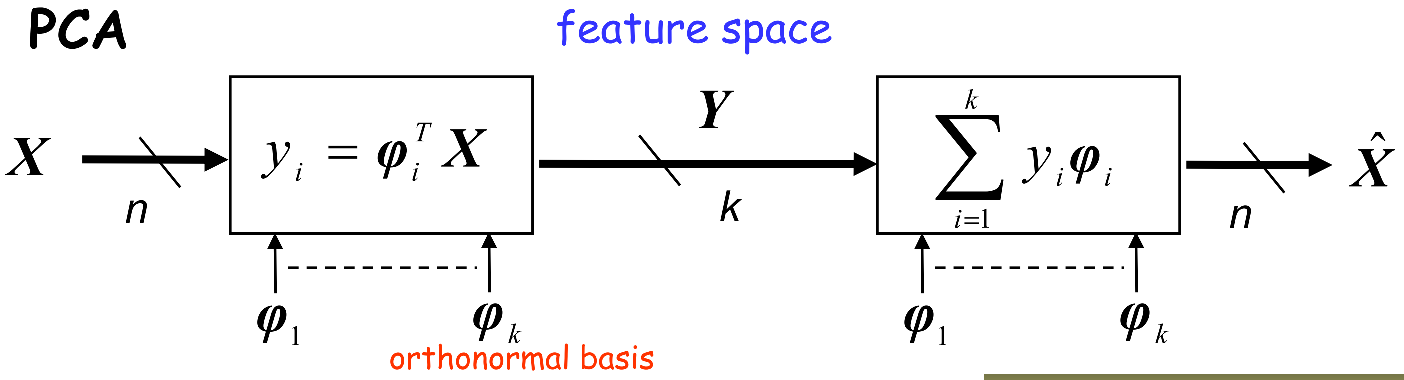
- Also called Latent Semantic Indexing (LSI), Latent Semantic Mapping (LSM)
- A technique originally proposed for Information Retrieval (IR), which projects queries and docs into a space with "latent" semantic dimensions
  - Co-occurring terms are projected onto the same dimensions



  - In the latent semantic space (with fewer dimensions), a query and doc can have high cosine similarity even if they do not share any terms
  - Dimensions of the reduced space correspond to the axes of greatest variation
    - Closely related to Principal Component Analysis (PCA)

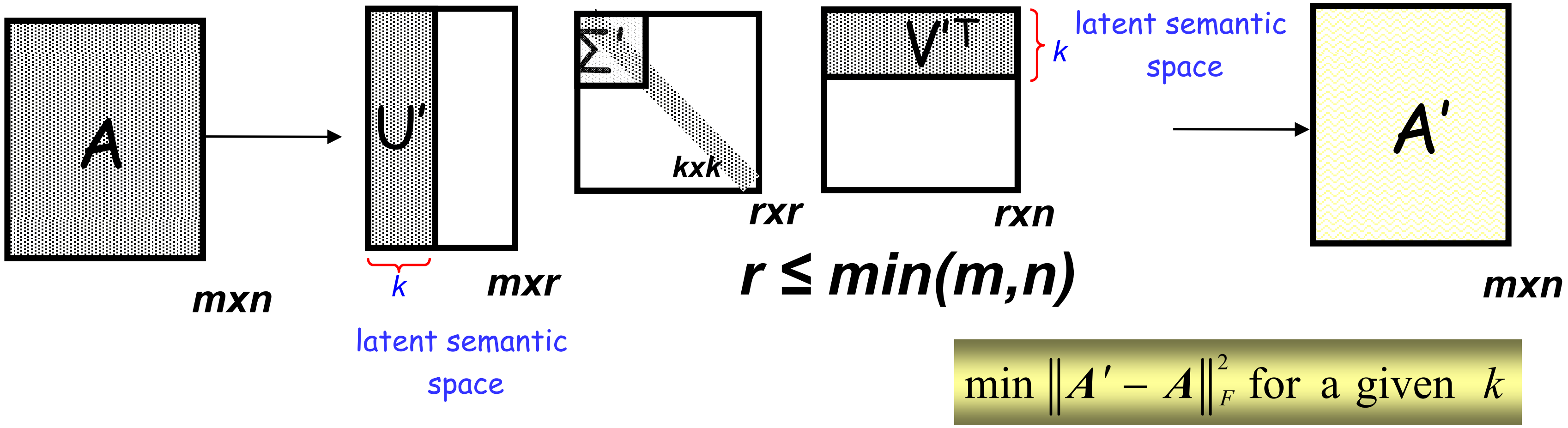
LSA (2/7)

- Dimension Reduction and Feature Extraction
  - PCA



$$\min \left\| \hat{X} - X \right\|^2 \quad \text{for a given } k$$

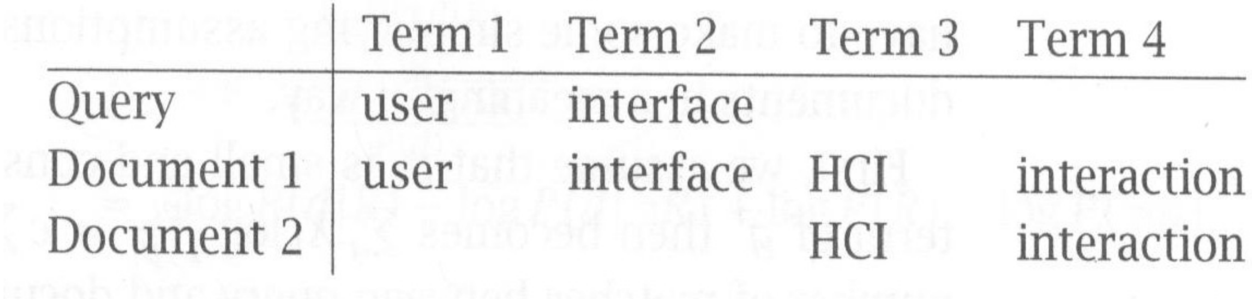
  - SVD (in LSA)



$$\min \left\| A' - A \right\|_F^2 \quad \text{for a given } k$$

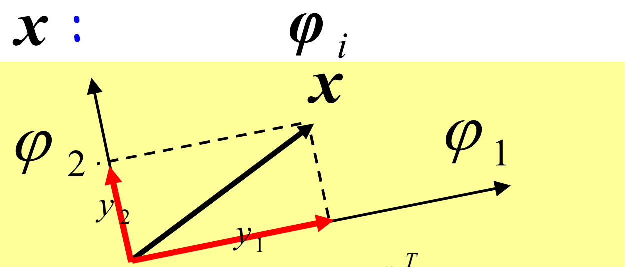
LSA (3/7)

- Singular Value Decomposition (SVD) used for the word-document matrix
  - *A least-squares method for dimension reduction*



	Term 1	Term 2	Term 3	Term 4
Query	user	interface		
Document 1	user	interface	HCI	interaction
Document 2			HCI	interaction

Projection of a Vector $\boldsymbol{x}$:

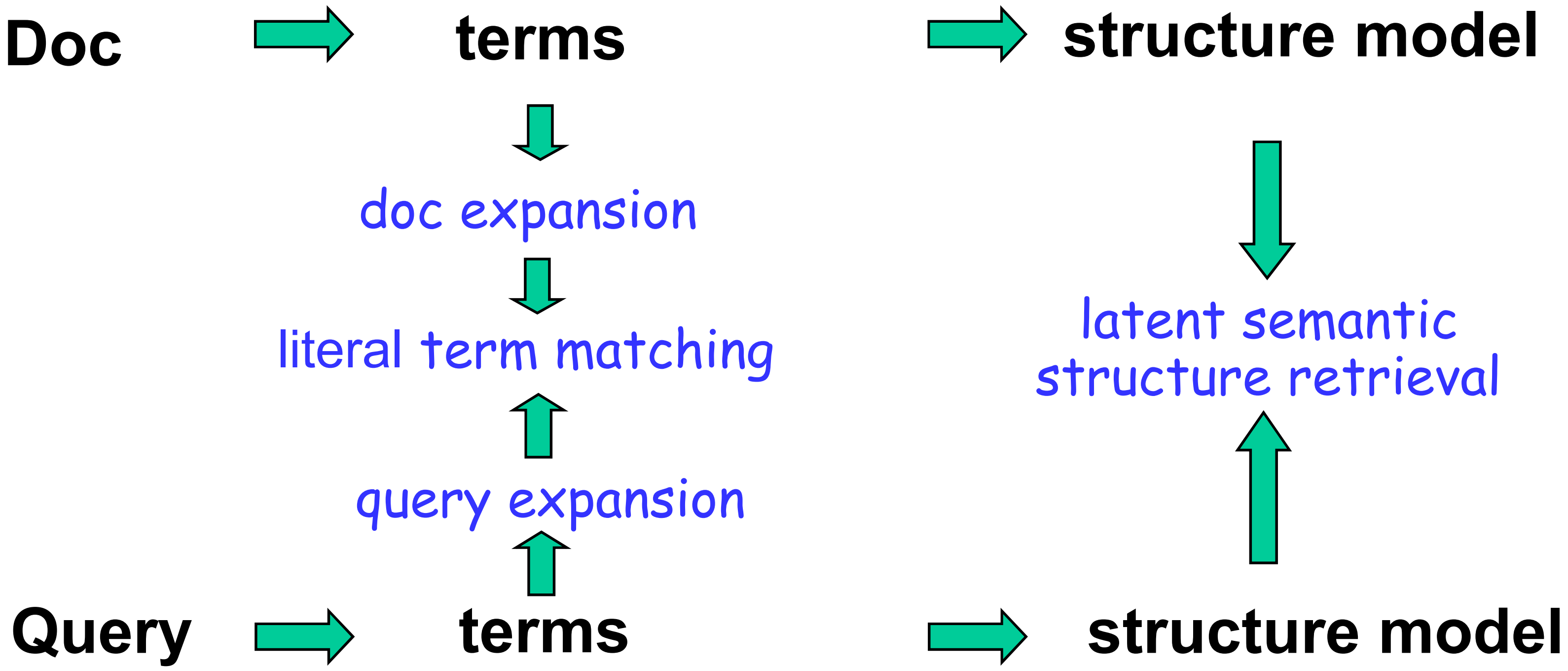


$$y_1 = \|\boldsymbol{x}\| \cos \theta_1 = \|\boldsymbol{x}\| \frac{\varphi_1^T \boldsymbol{x}}{\|\boldsymbol{x}\| \|\boldsymbol{\varphi}_1\|} = \varphi_1^T \boldsymbol{x}$$

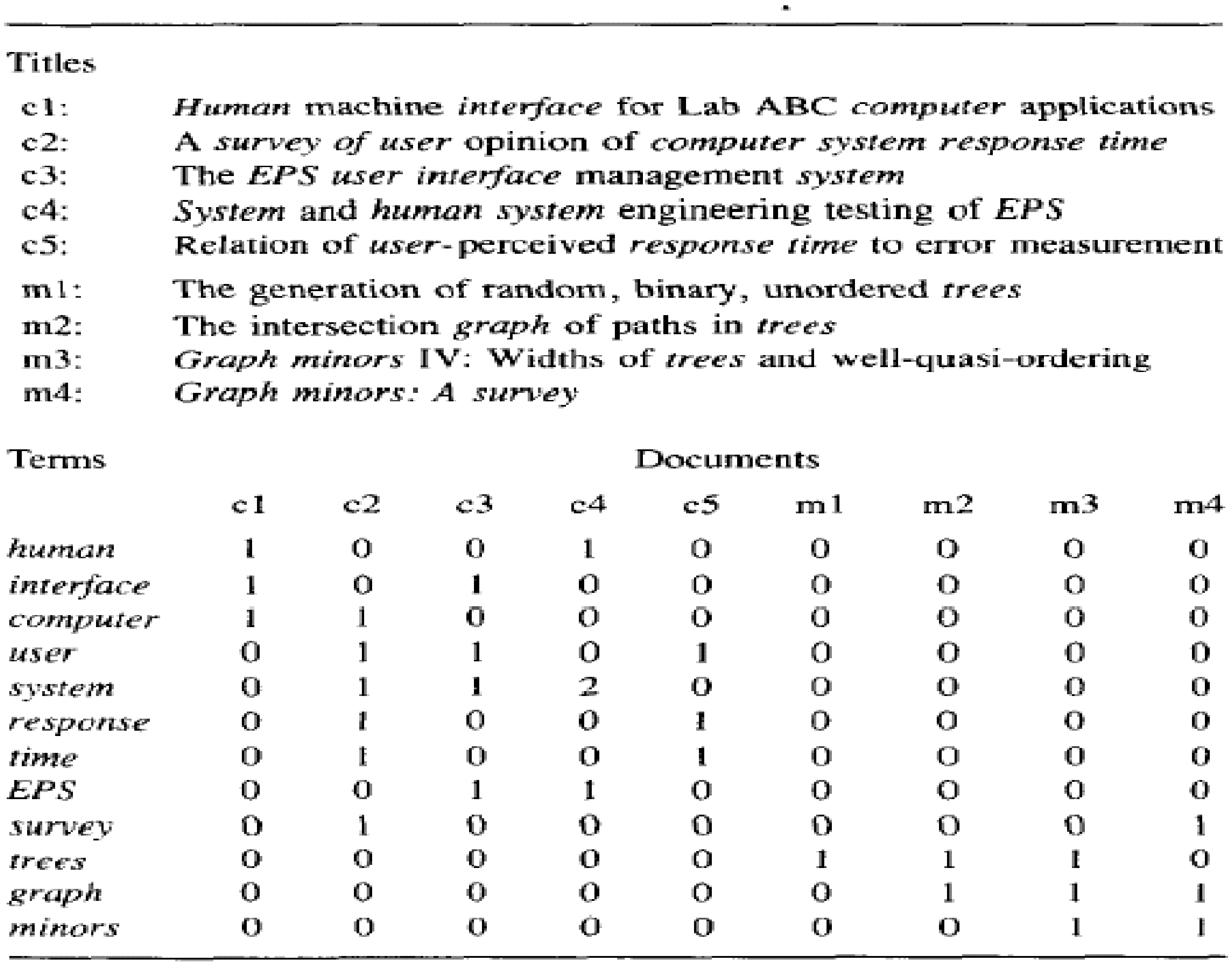
, where $\|\boldsymbol{\varphi}_1\| = 1$

LSA (4/7)

- Frameworks to circumvent vocabulary mismatch



LSA (5/7)



Titles

c1: *Human* machine *interface* for Lab ABC *computer* applications

c2: A *survey of user* opinion of *computer system response time*

c3: The *EPS user interface* management *system*

c4: *System* and *human system* engineering testing of *EPS*

c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*

m2: The intersection *graph* of paths in *trees*

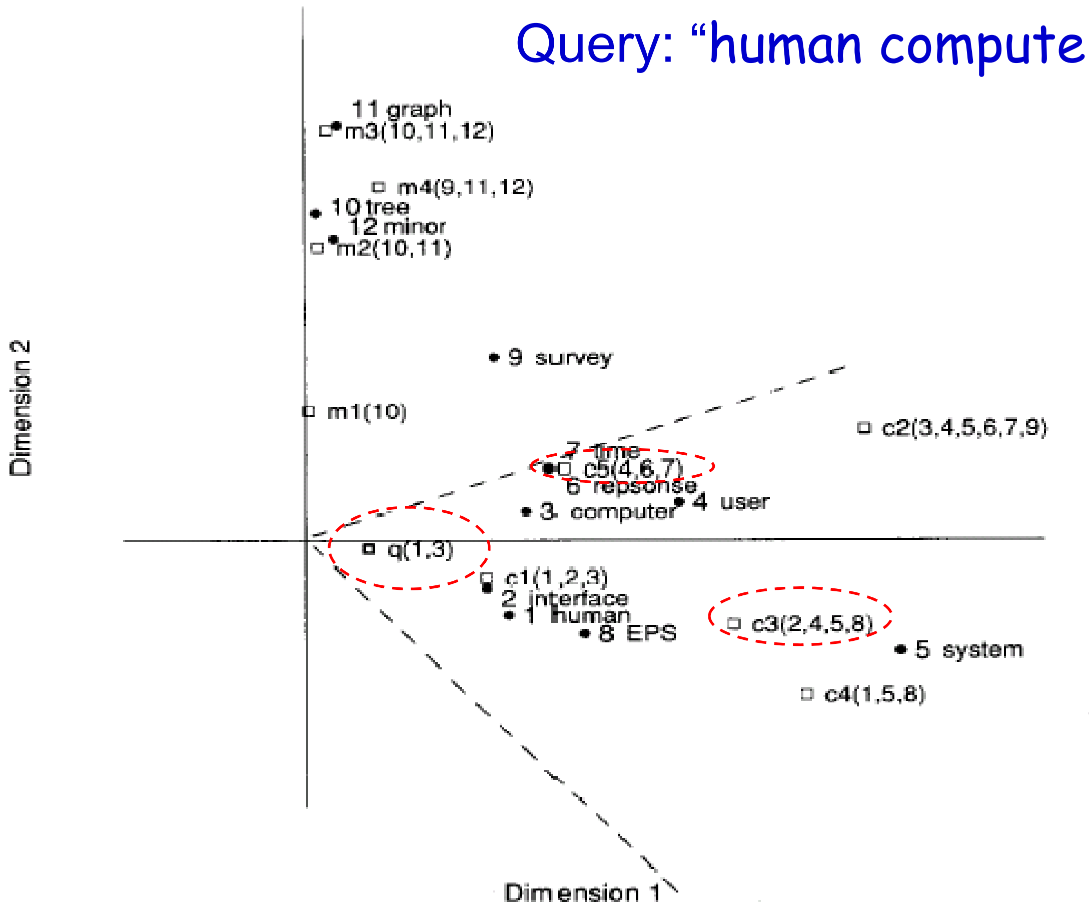
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

m4: *Graph minors: A survey*

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

LSA (6/7)

2-D Plot of Terms and Docs from Example



Query: “human computer interaction”

An OOV word

FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sampe TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query q. All documents about human-computer (c1–c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1–m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.

LSA (7/7)

- Singular Value Decomposition (SVD)

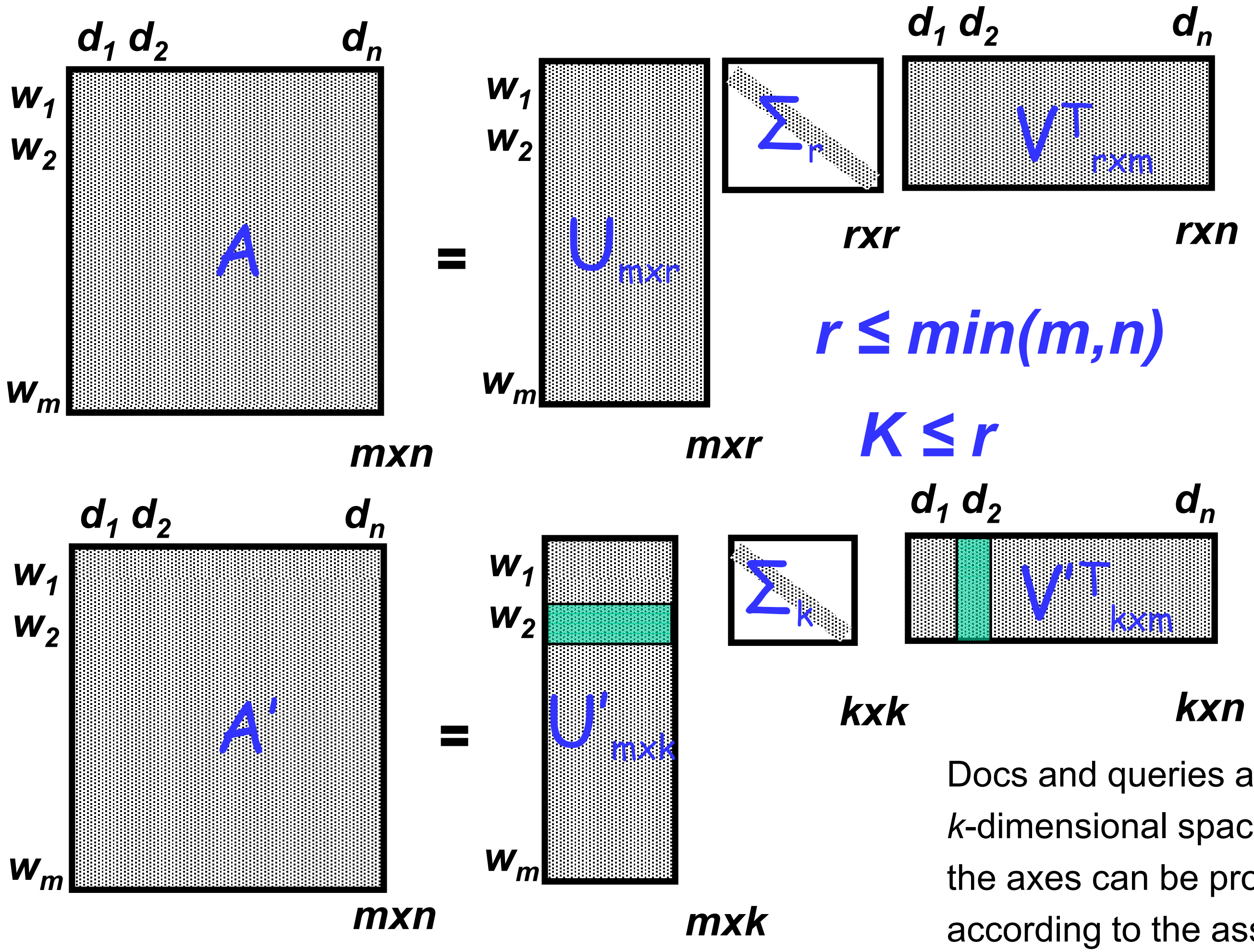
Row A $\in R^n$

Col A $\in R^m$

Both U and V has orthonormal column vectors

$U^TU=I_{rXr}$

$V^TV=I_{rXr}$



$r \le min(m,n)$

$K \le r$

$$||A||_F^2 \geq ||A'||_F^2$$

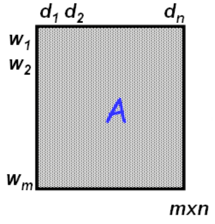
$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$$

Docs and queries are represented in a *k*-dimensional space. The quantities of the axes can be properly weighted according to the associated diagonal values of Σ_k

LSA Derivations (1/7)

- Singular Value Decomposition (SVD)
  - $A^T A$ is symmetric nxn matrix
    - All eigenvalues λ_j are nonnegative real numbers

$$\lambda_1 \ge \lambda_2 \ge .... \ge \lambda_n \ge 0 \quad \Sigma^2 = diag(\lambda_1, \lambda_1, ..., \lambda_n)$$



- All eigenvectors v_j are orthonormal ($\in R^n$)

$$V_{n \times n} = [v_1 v_2 ... v_n] \quad v_j^T v_j = 1 \quad (V^T V = I_{nxn})$$

sigma $\sigma_j = \sqrt{\lambda_j},\ j = 1, ..., n$

- Define **singular values:**
  - As the square roots of the eigenvalues of $A^T A$
  - As the lengths of the vectors Av_1, Av_2 , …., Av_n

For $\lambda_i \neq 0$, $i=1,\ldots r$,
$\{Av_1, Av_2, \ldots, Av_r\}$ is an orthogonal basis of Col A

$$\sigma_1 = \|Av_1\|$$
$$\sigma_2 = \|Av_2\|$$
.....

$$\|Av_i\|^2 = v_i^T A^T A v_i = v_i^T \lambda_i v_i = \lambda_i$$
$$\Rightarrow \|Av_i\| = \sigma_i$$

LSA Derivations (2/7)

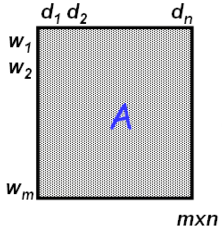
- $\{Av_1, Av_2, \ldots, Av_r\}$ is an orthogonal basis of Col A

$$Av_i \bullet Av_j = (Av_i)^T Av_j = v_i^T A^T Av_j = \lambda_j v_i^T v_j = 0$$

  - Suppose that A (or A^TA) has rank $r \leq n$

$$\lambda_1 \geq \lambda_2 \geq \ldots. \geq \lambda_r > 0, \quad \lambda_{r+1} = \lambda_{r+2} = \ldots. = \lambda_n = 0$$

  - Define an orthonormal basis $\{u_1, u_2, \ldots., u_r\}$ for Col A



$$u_i = \frac{1}{\|Av_i\|} Av_i = \frac{1}{\sigma_i} Av_i \Rightarrow \sigma_i u_i = Av_i$$

U: also an orthonormal matrix (mxr)

V: an orthonormal matrix (nxr)

$$\Rightarrow \boxed{[u_1\ u_2 \ldots u_r]}\Sigma_r = A\boxed{[v_1\ v_2 \quad v_r]}$$

Known in advance

- Extend to an orthonormal basis $\{u_1, u_2, \ldots, u_m\}$ of R^m

$$\Rightarrow [u_1\ u_2 \ldots u_r \ldots u_m]\Sigma = A\overbrace{[v_1\ v_2 \ldots v_r \ldots v_n]}$$

$$\Rightarrow \boxed{U\Sigma = AV} \Rightarrow U\Sigma V^T = A\underbrace{VV^T}_{I_{nxn}}$$

$$\Rightarrow A = U\Sigma V^T \qquad \boldsymbol{\Sigma}_{m\times n} = \begin{pmatrix} \boldsymbol{\Sigma}_r & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(n-r)} \end{pmatrix}$$

?

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$$

$$\|A\|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2$$

?

LSA Derivations (3/7)

$\boldsymbol{v}_i$ spans the row space of $\boldsymbol{A}$

$\mathbf{u}_i$ spans the row space of $\mathbf{A}^T$

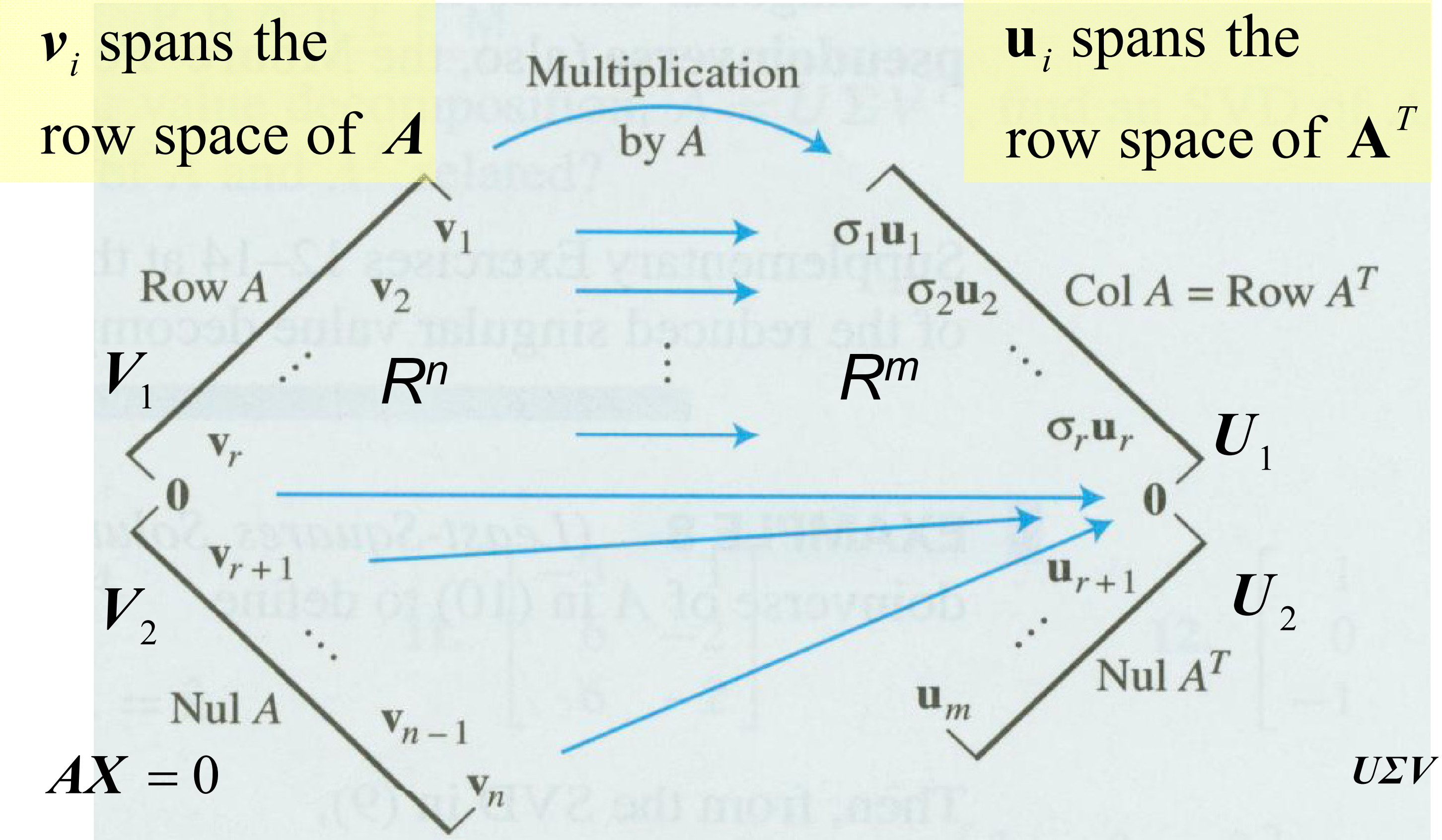
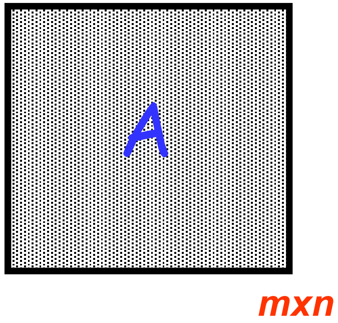


FIGURE 4 The four fundamental subspaces and the action of A.



$$U\Sigma V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix}\begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

$$= U_1\Sigma_1V_1^T$$

$$= AV_1V_1^T$$

$$= A$$

$U\Sigma = AV$

LSA Derivations (4/7)

- Additional Explanations
  - Each row of U is related to the projection of a corresponding row of A onto the basis formed by columns of V

    $$A = U\Sigma V^T$$

    $$\Rightarrow AV = U\Sigma V^T V = U\Sigma \quad \Rightarrow \quad U\Sigma = AV$$

    - the *i*-th entry of a row of U is related to the projection of a corresponding row of A onto the *i*-th column of V

  - Each row of V is related to the projection of a corresponding row of A^T onto the basis formed by U

    $$A = U\Sigma V^T$$

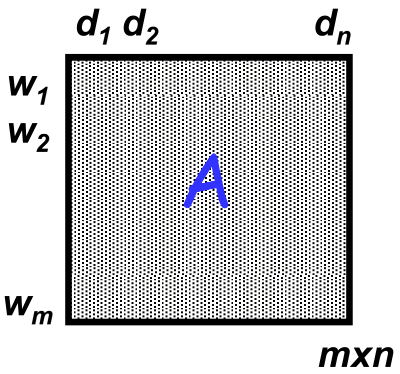
    $$\Rightarrow A^T U = \left(U\Sigma V^T\right)^T U = V\Sigma U^T U = V\Sigma$$

    $$\Rightarrow V\Sigma = A^T U$$

    - the *i*-th entry of a row of V is related to the projection of a corresponding row of A^T onto the *i*-th column of U

LSA Derivations (5/7)

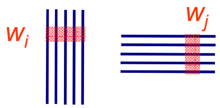
- Fundamental comparisons based on SVD
  - The original word-document matrix (A)



- compare two terms → dot product of two rows of A
  - or an entry in AA^T
- compare two docs → dot product of two columns of A
  - or an entry in A^TA
- compare a term and a doc → each individual entry of A

  - The new word-document matrix (A')

$U'=U_{m\times k}$
$\Sigma'=\Sigma_k$
$V'=V_{n\times k}$



- compare two terms
  → dot product of two rows of $U'\Sigma'$

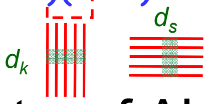
$$A'A'^T=(U'\Sigma'V'^T)\,(U'\Sigma'V'^T)^T=U'\Sigma'V'^TV'\Sigma'^TU'^T=(U'\Sigma')(U'\Sigma')^T$$

$I_{r\times r}$

For stretching or shrinking

- compare two docs
  → dot product of two rows of $V'\Sigma'$

$$A'^TA'=(U'\Sigma'V'^T)^T{}'(U'\Sigma'V'^T)=V'\Sigma'^T{}'U^TU'\Sigma'V'^T=(V'\Sigma')(V'\Sigma')^T$$



- compare a query word and a doc → each individual entry of A'

LSA Derivations (6/7)

- **Fold-in**: find representations for pesudo-docs *q*
  - For objects (new queries or docs) that did not appear in the original analysis
    - Fold-in a new *m*x1 query (or doc) vector

$$\hat{q}_{1\times k}=\boxed{\left(q^{T}\right)_{1\times m}U_{m\times k}}\boxed{\Sigma_{k\times k}^{-1}}$$

Just like a row of V

Query represented by the weighted sum of it constituent term vectors

The separate dimensions are differentially weighted

  - Cosine measure between the query and doc vectors in the latent semantic space

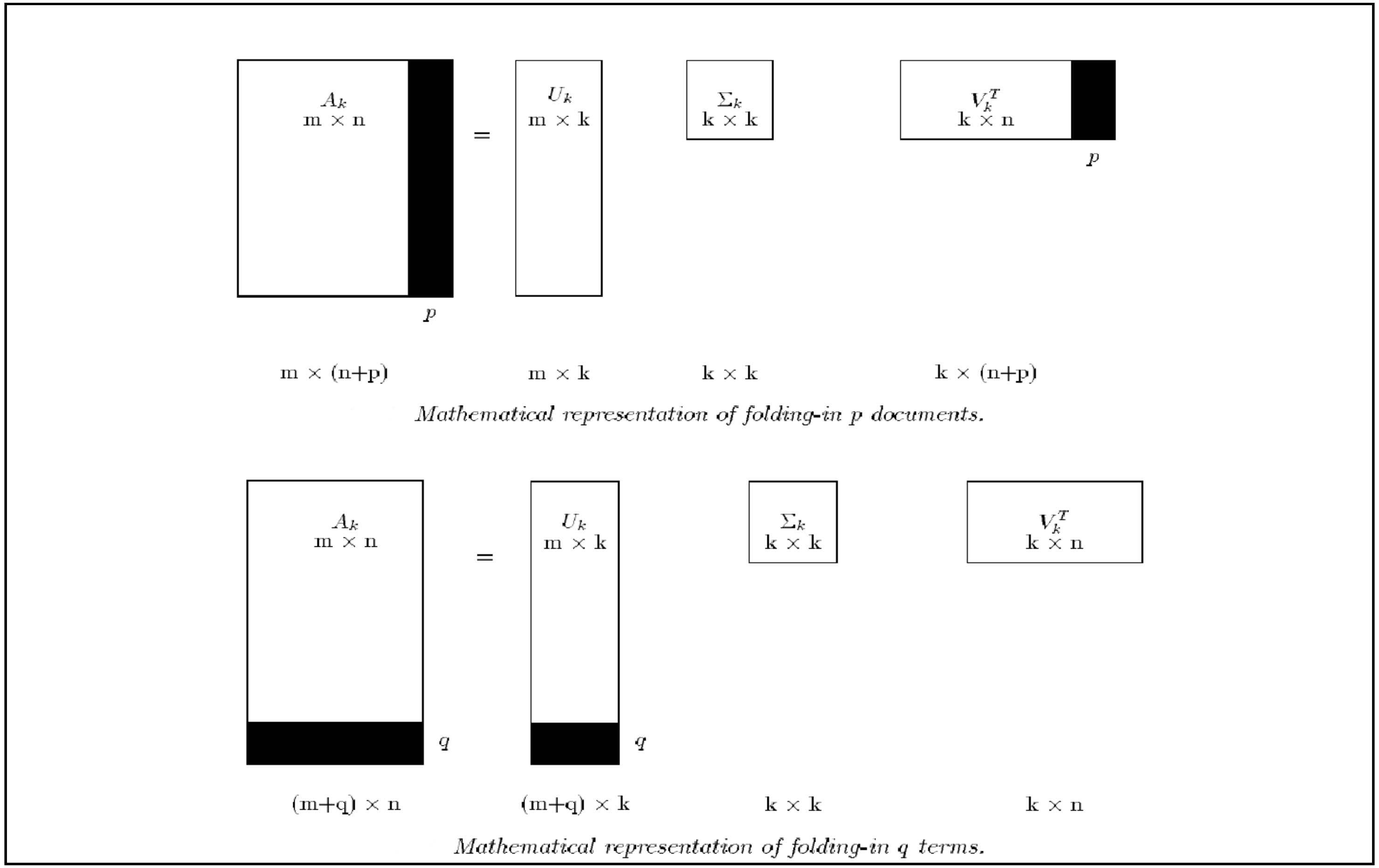
$$sim\left(\hat{q},\hat{d}\right)=coine\left(\hat{q}\Sigma,\hat{d}\Sigma\right)=\frac{\hat{q}\Sigma^{2}\hat{d}^{T}}{\left|\hat{q}\Sigma\right|\left|\hat{d}\Sigma\right|}$$

row vectors

LSA Derivations (7/7)

- Fold-in a new 1 x n term vector

$$\hat{t}_{1\times k} = t_{1\times n} V_{n\times k} \Sigma^{-1}_{k\times k}$$

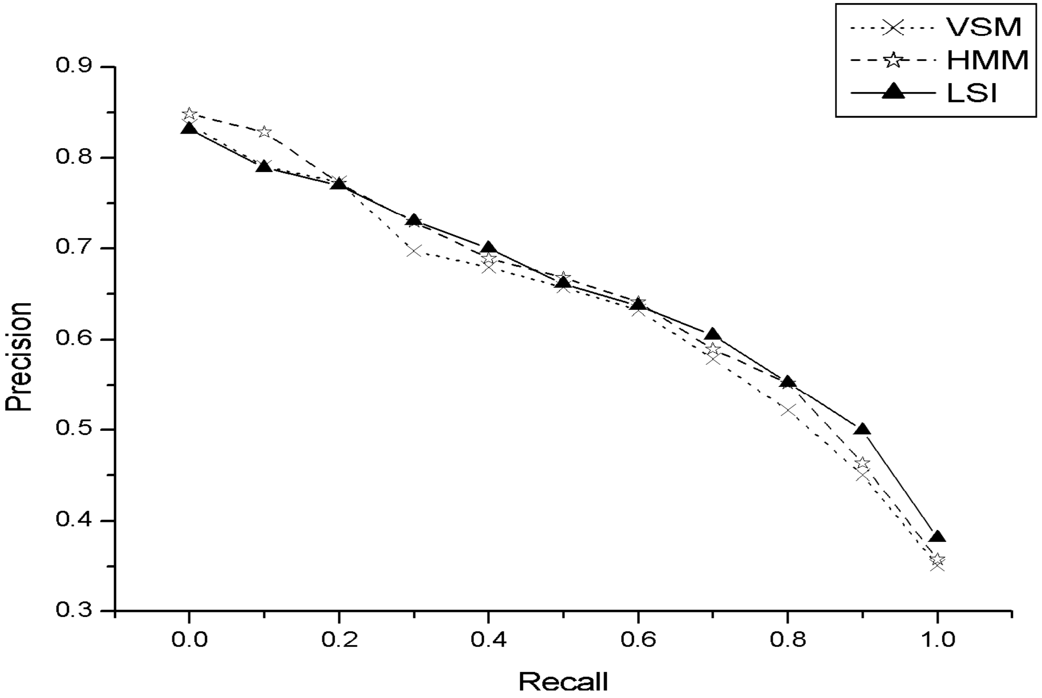


Mathematical representation of folding-in p documents.

Mathematical representation of folding-in q terms.

LSA Example

- Experimental results
  - HMM is consistently better than VSM at all recall levels
  - LSA is better than VSM at higher recall levels



Recall-Precision curve at 11 standard recall levels evaluated on TDT-3 SD collection. (Using word-level indexing terms)

LSA: Conclusions

- Advantages
  - A clean formal framework and a clearly defined optimization criterion (least-squares)
    - Conceptual simplicity and clarity
  - Handle synonymy problems ("heterogeneous vocabulary")
  - Good results for high-recall search
    - Take term co-occurrence into account
- Disadvantages
  - High computational complexity
  - LSA offers only a partial solution to polysemy
    - E.g. bank, bass,…

LSA Toolkit: SVDLIBC (1/5)

- Doug Rohde's SVD C Library version 1.3 is based on the SVDPACKC library
- Download it at http://tedlab.mit.edu/~dr/

LSA Toolkit: SVDLIBC (2/5)

- Given a sparse term-doc matrix
  - E.g., 4 terms and 3 docs

$$\text{Term}\left\{\overbrace{\begin{matrix} 2.3 & 0.0 & 4.2 \\ 0.0 & 1.3 & 2.2 \\ 3.8 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{matrix}}^{\text{Doc}}\right.$$

Row #Tem	Col. # Doc	Nonzero entries	
4	3	6	
2			2 nonzero entries at **Col 0**
0	2.3		Col 0, Row 0
2	3.8		Col 0, Row 2
1			1 nonzero entry at **Col 1**
1	1.3		Col 1, Row 1
3			3 nonzero entry at **Col 2**
0	4.2		Col 2, Row 0
1	2.2		Col 2, Row 1
2	0.5		Col 2, Row 2

  - Each entry is weighted by *TF*x*IDF* score

- Perform SVD to obtain corresponding term and doc vectors represented in the latent semantic space
- Evaluate the information retrieval capability of the LSA approach by using varying sizes (e.g., 100, 200, ..,600 etc.) of LSA dimensionality

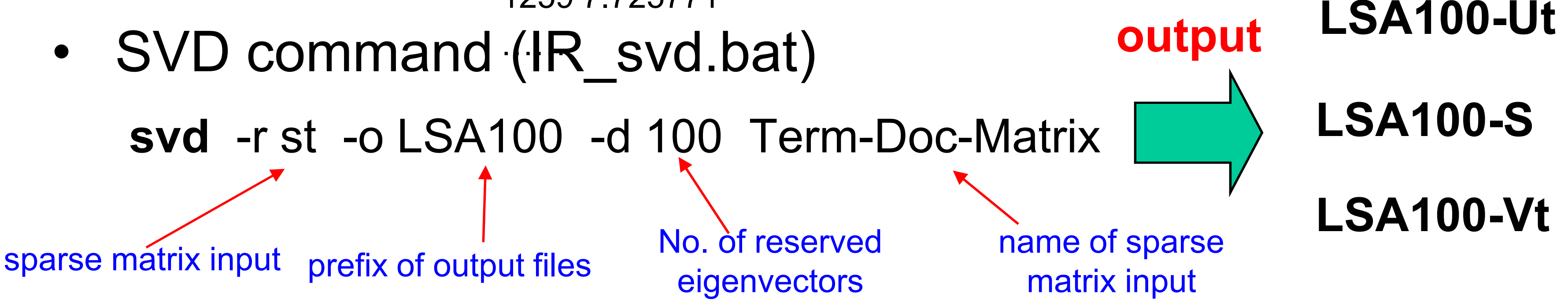
LSA Toolkit: SVDLIBC (3/5)

- Example: term-docmatrix

Indexing Term no. Doc no. Nonzero entries

```
51253 2265 218852
77
508 7.725771
596 16.213399
612 13.080868
709 7.725771
713 7.725771
744 7.725771
1190 7.725771
1200 16.213399
1259 7.725771
```

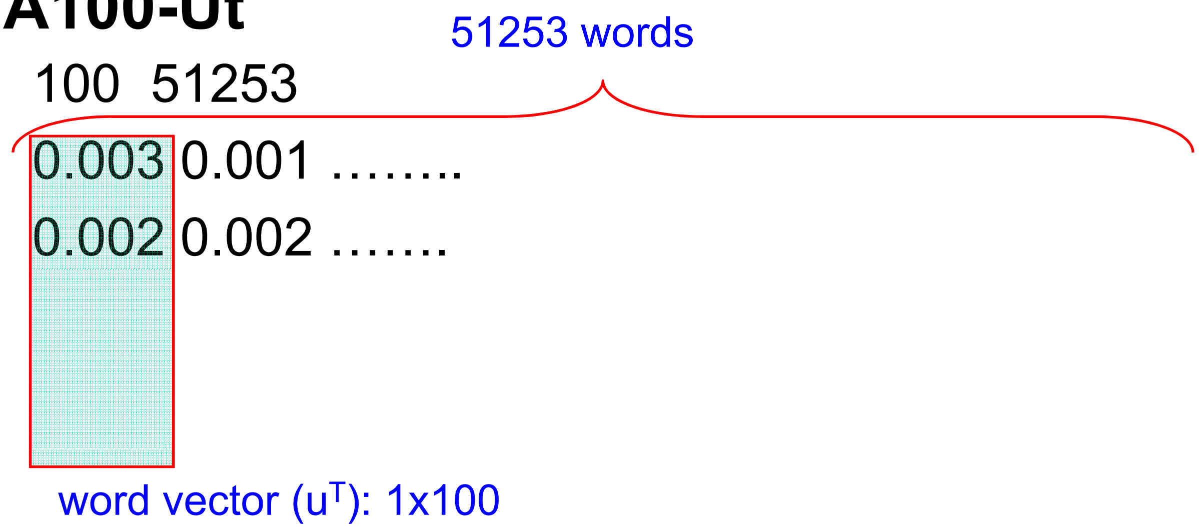
- SVD command (IR_svd.bat)



```
svd  -r st  -o LSA100  -d 100  Term-Doc-Matrix
```

LSA Toolkit: SVDLIBC (4/5)

- **LSA100-Ut**



51253 words

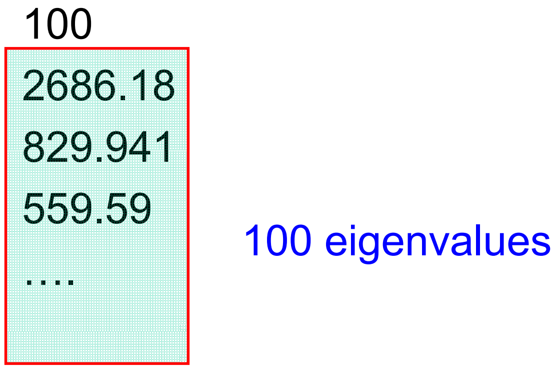
100 51253

0.003 0.001 ……..

0.002 0.002 …….

word vector (u^T): 1x100

- **LSA100-S**



100

2686.18

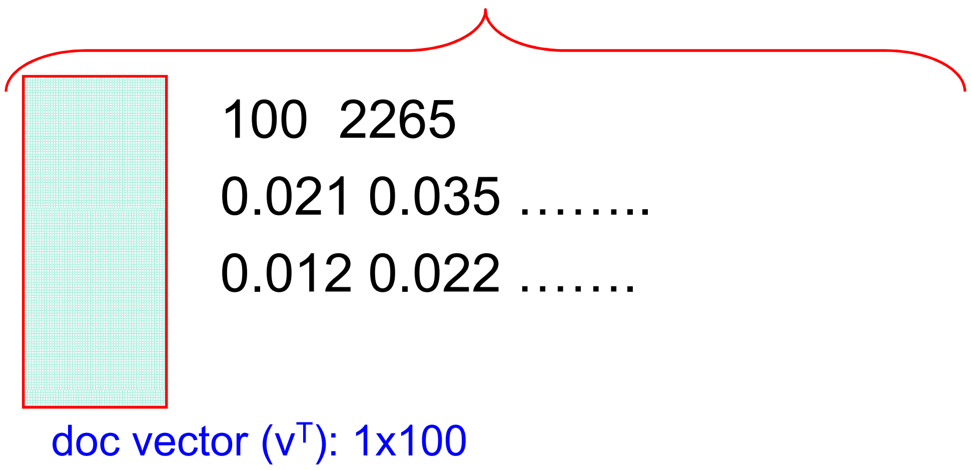
829.941

559.59

….

100 eigenvalues

- **LSA100-Vt**



2265 docs

100 2265

0.021 0.035 ……..

0.012 0.022 …….

doc vector (v^T): 1x100

LSA Toolkit: SVDLIBC (5/5)

- Fold-in a new *m*x1 query vector

$$\hat{q}_{1\times k} = \boxed{\left(q^{T}\right)_{1\times m} U_{m\times k}}\boxed{\Sigma^{-1}_{k\times k}}$$

The separate dimensions are differentially weighted

Just like a row of V

Query represented by the weighted sum of it constituent term vectors

TFxIDF weighted beforehand

- Cosine measure between the query and doc vectors in the latent semantic space

$$sim\left(\hat{q},\hat{d}\right) = coine\left(\hat{q}\Sigma,\hat{d}\Sigma\right) = \frac{\hat{q}\Sigma^{2}\hat{d}^{T}}{\left|\hat{q}\Sigma\right|\left|\hat{d}\Sigma\right|}$$