CH. 6: Dimensionality Reduction

- **Objectives:** Reduces space complexity,
  Reduces time complexity,
  Data visualization

- **Two main methods of dimensionality reduction:**

  1) Feature selection: Choosing $k < d$ important features,

  2) Feature extraction: Mapping data points from d-D to k-D space, where $k < d$, while preserving as many properties of the data as possible.

6.1 Feature Selection

i) **Forward search**: Add the best feature at each step

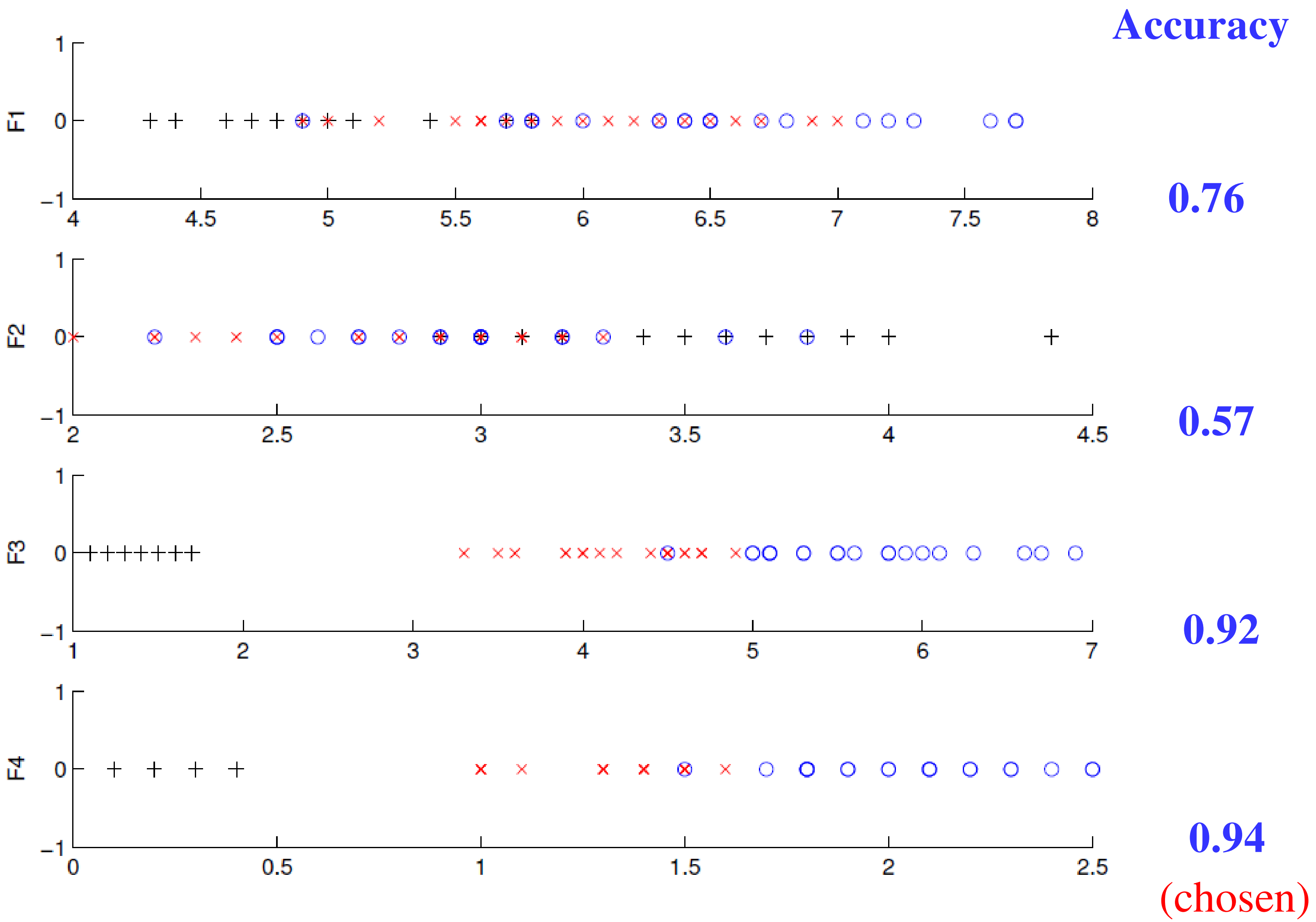
- Initially, $F=\phi$ (F: feature set)
- At each iteration, find the best new feature using a sample $j=\arg\max_i P(F\cup x_i)$, where

  $P()$: performance function
- Add x_j to F if $P(F\cup x_i)>P(F)$

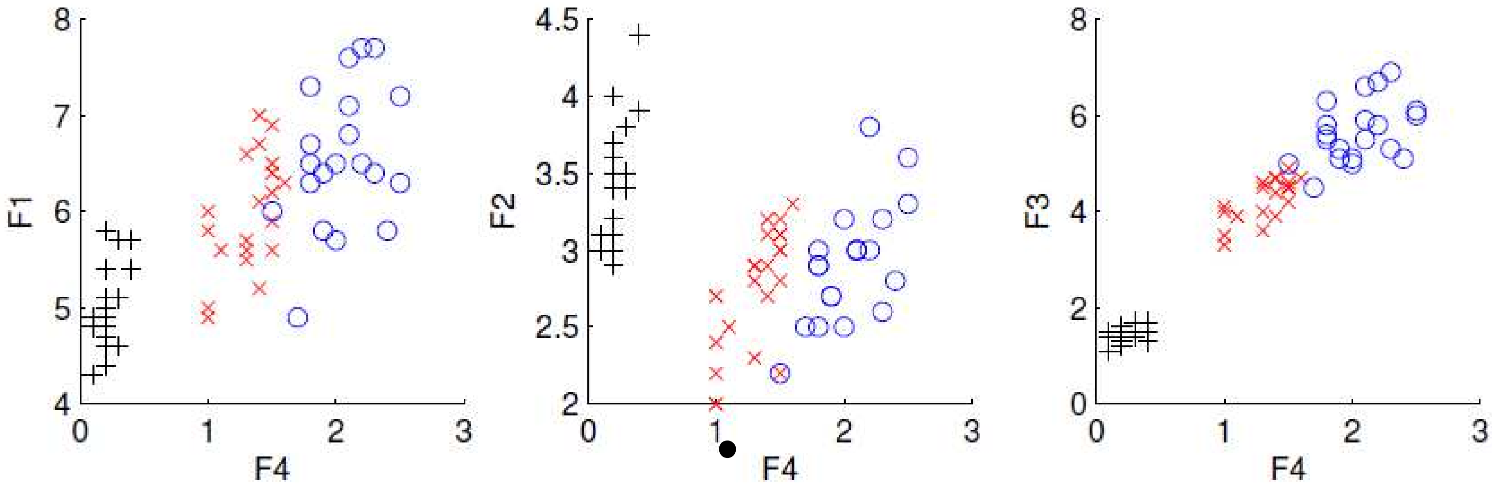
Example: Iris data (3 classes: (+, ×, ○),

4 features: (F1,F2,F3,F4))

Single feature



Add one more feature to F4



Accuracy **0.87** **0.92** **0.96**

(chosen)

Since the accuracies of (F1,F3,F4) and (F2,F3,F4) are both 0.94 smaller than 0.96.

Stop the feature selection process at (F3,F4) .

ii) Backward search: Start with all features and remove one at a time.

Remove x_j from F if $j = \arg\min_i E(F - \{x_i\})$

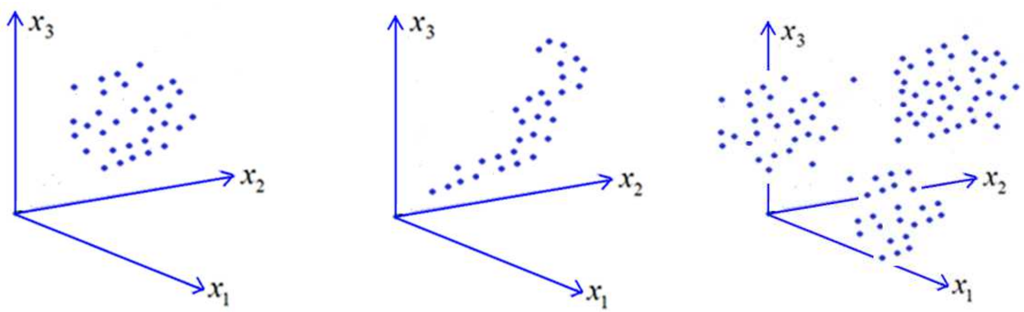
$E()$: error function

iii) Floating search: The numbers of added and removed features can change at each step.

6.2 Feature Extraction

- Graphical representation (visualization)

  A data set $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ can be represented as a set of points in a space.



The data set may possess certain properties. Feature extraction (FE) attempts to preserve or even improve the properties during an FE process.

i) Principal Components Analysis (PCA)

Data points: $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T, \quad \boldsymbol{i} = 1, 2, \cdots, N$

Matrix representation:

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)^T = \begin{pmatrix} x_{11} & x_{12} & \cdot\cdot & x_{1d} \\ x_{21} & x_{22} & \cdot\cdot & x_{2d} \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ x_{N1} & x_{N2} & \cdot\cdot & x_{Nd} \end{pmatrix}$$

Mean vector: $\boldsymbol{\mu}_x = \dfrac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i,$

Covariance matrix: $C_x = \dfrac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_x)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^T$

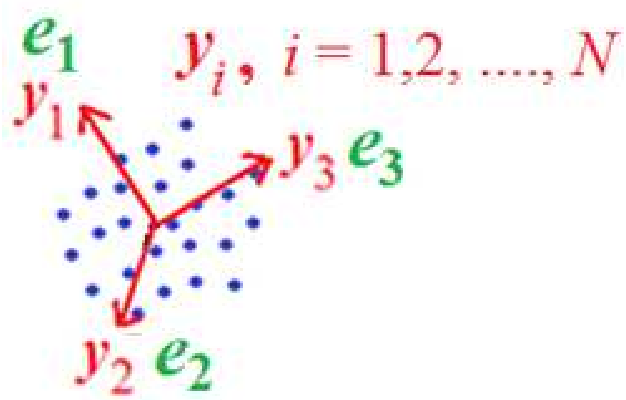
Let λ_i and $\boldsymbol{e}_i$, $i = 1, \cdots, d$, be the eigenvalues and eigenvectors of C_x, i.e., $C_x \boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i$.

Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

Let $A_{d \times d} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_d \end{bmatrix}^T$.

Compute $\boldsymbol{y}_i = A(\boldsymbol{x}_i - \boldsymbol{\mu}_x)$, $i = 1, 2, \cdots, N$.

y-axes corresponding to eigenvectors $\boldsymbol{e}$'s are orthogonal, i.e., uncorrelated.



The variances over y-axes $\approx$ eigenvalues

For dimensionality reduction,

Let $A_k = [\boldsymbol{e}_1 \cdots \boldsymbol{e}_k]^T, \; k < d.$

$$\hat{\boldsymbol{y}} = A_k(\boldsymbol{x} - \boldsymbol{\mu}_x)$$

$$\hat{\boldsymbol{y}}_{k\times 1} = (A_k)_{k\times d}(\boldsymbol{x} - \boldsymbol{\mu}_x)_{d\times 1}$$

Let $\hat{\boldsymbol{x}}$ be the reconstruction of $\boldsymbol{x}$ from $\hat{\boldsymbol{y}}$.

The reconstruction error $\left\| \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i \right\|^2$ depends on

$\sum_{i=k+1}^{d} \lambda_i$, which are relatively smaller than $\sum_{i=1}^{k} \lambda_i$.

Eigen faces for face recognition

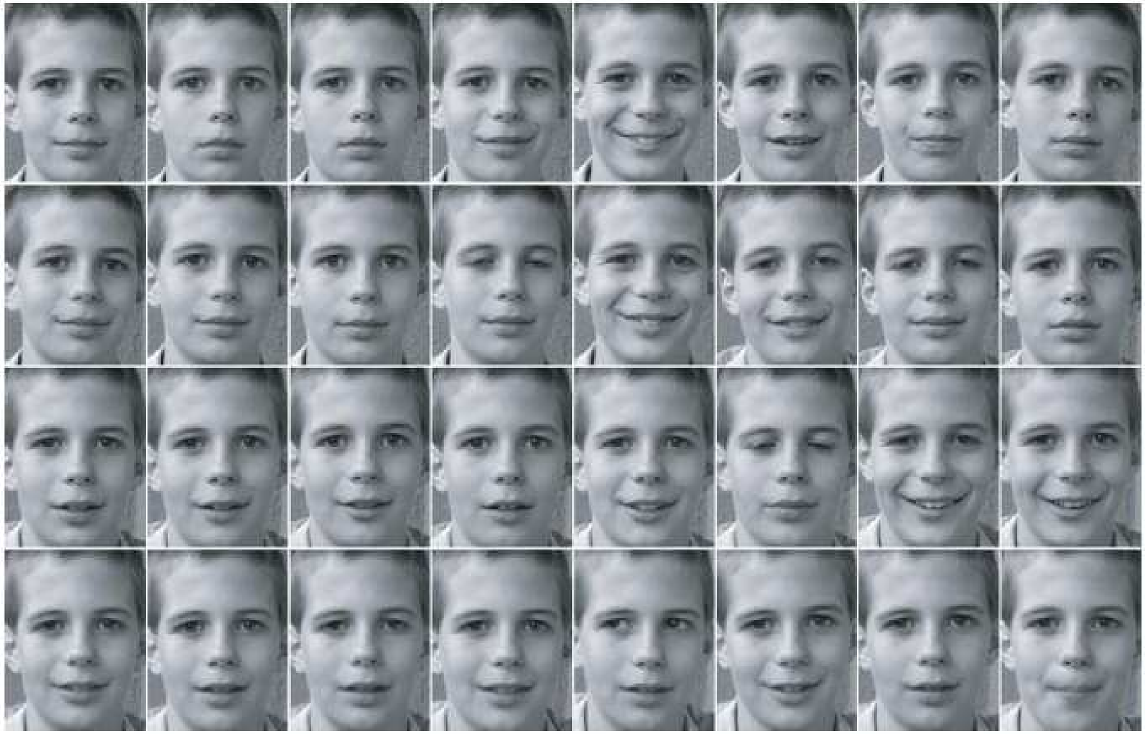
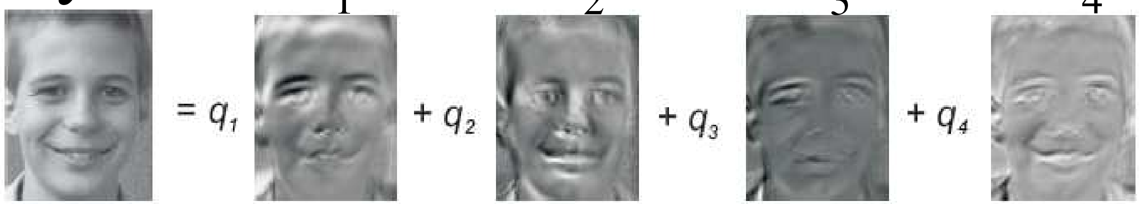


Figure 3.23: 32 original images of a boy's face, each 321 × 261 pixels.

$\boldsymbol{i}$ $\boldsymbol{e}_1$ $\boldsymbol{e}_2$ $\boldsymbol{e}_3$ $\boldsymbol{e}_4$



$$q_i = \boldsymbol{i}^T \boldsymbol{e}_i,$$
$$i = 1, 2, 3, 4$$

Figure 3.24: Reconstruction of the image from four basis vectors $\mathbf{b}_i$, $i = 1, \ldots, 4$ which can be displayed as images. The linear combination was computed as $q_1\mathbf{b}_1 + q_2\mathbf{b}_2 + q_3\mathbf{b}_3 + q_4\mathbf{b}_4 = 0.078\,\mathbf{b}_1 + 0.062\,\mathbf{b}_2 - 0.182\,\mathbf{b}_3 + 0.179\,\mathbf{b}_4$.

ii) Feature Embedding (FE)

FE places d-D data points in a k-D space ($k < d$) such that pairwise similarities in the new space respect the original pairwise similarities.

Let $X_{N\times d}$ be data matrix, λ_i and $\boldsymbol{w}_i$ be the eigenvalues and eigenvectors of correlation matrix $(X^T X)_{d\times d}$ of features, i.e., $(X^T X)\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$. Multiply both sides X, $X(X^T X)\boldsymbol{w}_i = (XX^T)X\boldsymbol{w}_i = \lambda_i X\boldsymbol{w}_i$, i.e., λ_i, $X\boldsymbol{w}_i$ are eigenvalues and eigenvectors of similarity matrix $(XX^T)_{N\times N}$ of instances.

Let $\boldsymbol{v}_i = X\boldsymbol{w}_i,\ i = 1,\cdots,k\ (< d)$ corresponding to k leading eigenvalues, which form the coordinates of the new space.

Since $\boldsymbol{v}_i$ are the eigenvectors of similarity matrix $(XX^T)_{N\times N}$. Pairwise similarities between instances will be preserved in the new space.

iii) Factor Analysis (FA)

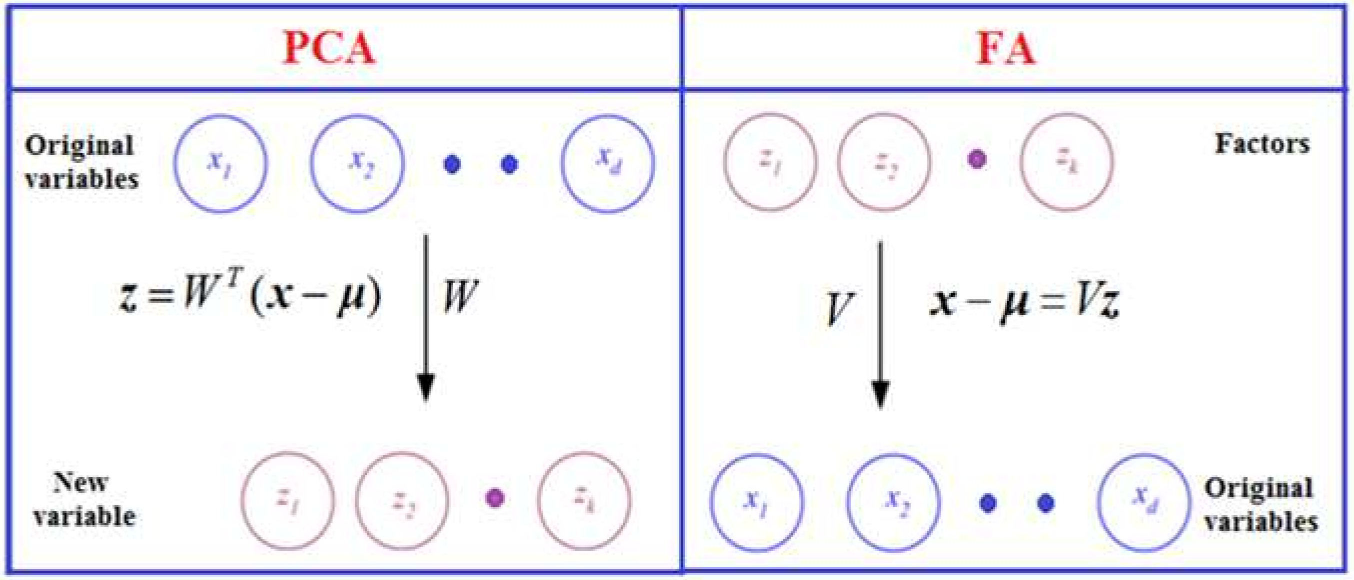
- In PCA, from the original features $x_i,\ i=1,\cdots,d$ to form a new set of features $z_j = \sum_{i=1}^{d} w_{ji} x_i,\ j=1,\cdots,k$ For dimension reduction, $k < d$. Mathematically,

  $\boldsymbol{z} = W^T(\boldsymbol{x} - \boldsymbol{\mu}).$

- In FA, a set of unobservable (latent) factors z_j, $j=1,\cdots,k$ that combine to generate $x_i,\ i=1,\cdots,d$.

  $x_i = \sum_{j=1}^{k} v_{ij} z_j,\quad i=1,\cdots,d.$ Mathematically,

  $\boldsymbol{x} - \boldsymbol{\mu} = V\boldsymbol{z}.$



- Given a sample $X = \{\boldsymbol{x}^t\}_{t=1}^N$, $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$, find a small number of factors z_i, $i = 1, \cdots, k$ $(k < d)$, s.t. each x_i can be written as a weighted sum of z_i,

$$x_i - \mu_i = v_{i1} z_1 + v_{i2} z_2 + \cdots + v_{ik} z_k + \varepsilon_i, \ i = 1, \cdots, d$$

In vector-matrix form, $\boldsymbol{x} - \boldsymbol{\mu} = V\boldsymbol{z} + \boldsymbol{\varepsilon}$.

where z_i : latent factors $(\sim N(0,1),\ \mathrm{Cov}(z_i, z_j) = 0,\ i \neq j)$

v_{ij} : factor loadings

ε_i : errors $(E[\varepsilon_i] = 0,\ \mathrm{Var}(\varepsilon_i) = \psi_i,$

$\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0,\ i \neq j).$

$\mathrm{Cov}(\varepsilon_i, z_j) = 0,\ \forall i, j$

- **Example:** Let s_c, s_e, s_m, s_p and s_{ch} be the score variables of Chinese(c), English(e), Mathematics(m), Physics(p), and Chemistry(ch), respectively, which are observable. Let z_m, z_i and z_o be the talent variables of memory(m), inference(i), organization(o), which are latent.

Specifically, given the scores of a student

$s_c = 78,\ s_e = 82,\ s_m = 94,\ s_p = 89,\ s_{ch} = 92,$

what are **loadings** $v_{ij}\ (i = c, e, m, p, ch;\ \ j = m, i, o)$

of **factors** $z_m,\ z_i$ and z_o of the student?

$$\boldsymbol{s} = (s_c\ \ s_e\ \ s_m\ \ s_p\ \ s_{ch}),\quad s_i = \sum_{j \in \{m,i.o\}} v_{ij} z_j\ \ (i = c, e, m, p, ch)$$

Two uses of factor analysis:

i) Knowledge extraction,

ii) Dimensionality reduction

Knowledge Extraction – Given X and Z, find V

From $\boldsymbol{x} - \boldsymbol{\mu} = V\boldsymbol{z} + \boldsymbol{\varepsilon}$, for simplicity, let $\boldsymbol{\mu} = \boldsymbol{0}$,

$$\Rightarrow \boldsymbol{x} = V\boldsymbol{z} + \boldsymbol{\varepsilon}.$$

$$\Sigma = \mathrm{Cov}(\boldsymbol{x}) = \mathrm{Cov}(V\boldsymbol{z} + \boldsymbol{\varepsilon}) = \mathrm{Cov}(V\boldsymbol{z}) + \mathrm{Cov}(\boldsymbol{\varepsilon})$$

$$= V\mathrm{Cov}(\boldsymbol{z})V^T + \boldsymbol{\psi} = VIV^T + \boldsymbol{\psi} = VV^T + \boldsymbol{\psi}$$

$$(\because\ z_i \sim N(0,1),\ \mathrm{Cov}(z_i, z_j) = 0,\ i \neq j,\ \therefore\ \mathrm{Cov}(\boldsymbol{z}) = I)$$

$\boldsymbol{\psi}$: diagonal matrix with ψ_i on the diagonals.

Ignoring $\boldsymbol{\psi}$, $\Sigma = VV^T$.

Let S be the estimator Σ of sample X, $S = VV^T$.

Spectral decomposition of S:

$$S = E\Lambda E^T = (E\Lambda^{1/2})(E\Lambda^{1/2})^T = VV^T,$$

$$\therefore \; V = E\Lambda^{1/2}$$

Dimensionality Reduction – Given X, find Z

Let $z_j = \sum_{i=1}^{d} w_{ji} x_i, \quad j = 1, \cdots, k$

$$z_1 = \sum_{i=1}^{d} w_{1i} x_i = w_{11} x_1 + w_{12} x_2 + \cdots + w_{1d} x_d$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$z_k = \sum_{i=1}^{d} w_{ki} x_i = w_{k1} x_1 + w_{k2} x_2 + \cdots + w_{kd} x_d$$

$$\begin{pmatrix} z_1 \\ z_2 \\ \cdot\cdot \\ z_k \end{pmatrix}_{k\times 1} = \begin{bmatrix} w_{11} & w_{12} & \cdot\cdot & w_{1d} \\ w_{21} & w_{22} & \cdot\cdot & w_{2d} \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ w_{k1} & w_{k2} & \cdot\cdot & w_{kd} \end{bmatrix}_{k\times d} \begin{pmatrix} x_1 \\ x_2 \\ \cdot\cdot \\ x_d \end{pmatrix}_{d\times 1}$$

In vector-matrix form, $\boldsymbol{z} = W^T \boldsymbol{x}$.

Given a sample $X = \{\boldsymbol{x}^t\}_{t=1}^N, \boldsymbol{z}^i = W^T \boldsymbol{x}^i, \quad \boldsymbol{i} = 1, \cdots, N$

In matrix form, $Z_{N\times k} = X_{N\times d} W_{d\times k}$.

Solve for $W = (X^T X)^{-1} X^T Z$

$S = X^T X$ is the estimated covariance matrix Σ of sample X.

$S = VV^T, \;\; V = E\Lambda^{1/2}.$

$\boldsymbol{x} - \boldsymbol{\mu} = V\boldsymbol{z} + \boldsymbol{\varepsilon}.$ Ignore $\boldsymbol{\mu}$ and $\boldsymbol{\varepsilon}$, $\Rightarrow \boldsymbol{x} = V\boldsymbol{z}.$

$\boldsymbol{x}\boldsymbol{z}^T = V\boldsymbol{z}\boldsymbol{z}^T.$ Given a sample $X = \{\boldsymbol{x}^t\}_{t=1}^N$,

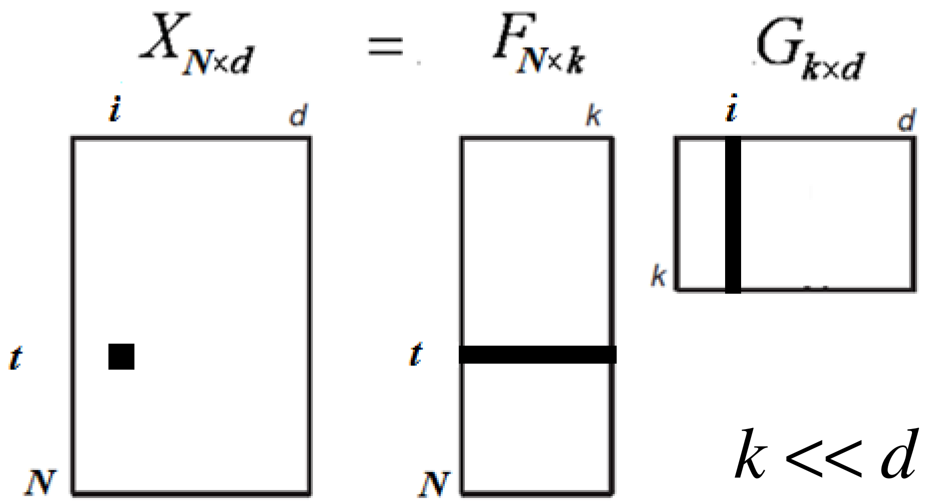
$\boldsymbol{x}^i \boldsymbol{z}^T = V\boldsymbol{z}^i(\boldsymbol{z}^i)^T = V, \;\; \boldsymbol{i} = 1, \cdots, N.$

In matrix form, $X^T Z = V.$

$W = (X^T X)^{-1} X^T Z = S^{-1}V = (VV^T)^{-1}V, \;\; V = E\Lambda^{1/2}$

$Z = XW.$

iv) Matrix Factorization (MF)



$k << d$

G defines k new factors in terms of the attributes of data X.

F defines instances in terms of the new factors in G.

Objective: solve $X = FG$ for $F = G^{+}X$

v) Linear Discriminant Analysis (LDA)

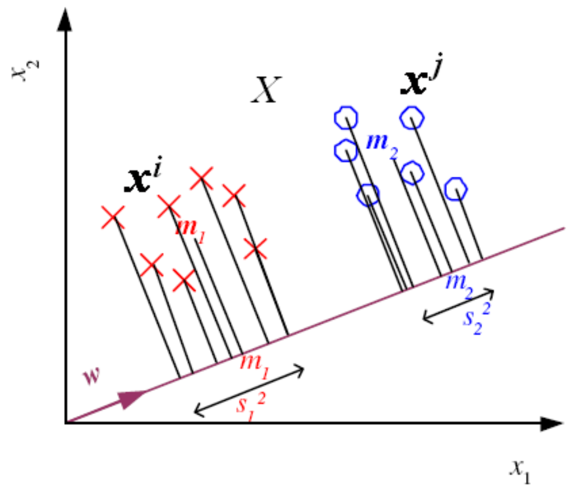
-- Find a low dimension space such that when data are pojected onto it, the examples of different classes are as well separated as possible.

- In 2-Class (d-D to 1-D) case, find a direction $\boldsymbol{w}$, such that when data are projected onto $\boldsymbol{w}$, the examples of different classes are well-separated.

Given a sample

$$X = \{\boldsymbol{x}^t, r^t\}_{t=1}^{N} \quad \text{s.t.}$$

$$r^t = \begin{cases} 1 & \text{if } \boldsymbol{x}^t \in C_1 \\ 0 & \text{if } \boldsymbol{x}^t \in C_2 \end{cases}$$



Means:

$$m_1 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t r^t}{\sum_t r^t} = \boldsymbol{w}^T \boldsymbol{m}_1,\ m_2 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t (1-r^t)}{\sum_t (1-r^t)} = \boldsymbol{w}^T \boldsymbol{m}_2$$

Scatters:

$$s_1^2 = \sum_t \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_1\right)^2 r^t,\ s_2^2 = \sum_t \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_2\right)^2 (1-r^t)$$

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

$$\begin{aligned}(m_1 - m_2)^2 &= \left(\boldsymbol{w}^T \boldsymbol{m}_1 - \boldsymbol{w}^T \boldsymbol{m}_2\right)^2 \\ &= \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} = \boldsymbol{w}^T S_B \boldsymbol{w}\end{aligned}$$

where $S_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$

(Between-class scatter matrix)

$$s_1^2 = \sum_{t=1}^{N} \left(\boldsymbol{w}^T \boldsymbol{x}^t - m_1\right)^2 r^t = \sum_{t=1}^{N} \boldsymbol{w}^T r^t \left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)^T \boldsymbol{w}$$

$$= \boldsymbol{w}^T S_1 \boldsymbol{w}, \quad \text{where} \quad S_1 = \sum\nolimits_t r^t \left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_1\right)^T$$

Similarly, $s_2^2 = \boldsymbol{w}^T S_2 \boldsymbol{w}$

where $S_2 = \sum\nolimits_t (1 - r^t)\left(\boldsymbol{x}^t - \boldsymbol{m}_2\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_2\right)^T$

$$s_1^2 + s_1^2 = \boldsymbol{w}^T S_1 \boldsymbol{w} + \boldsymbol{w}^T S_2 \boldsymbol{w} = \boldsymbol{w}^T (S_1 + S_2) \boldsymbol{w} = \boldsymbol{w}^T S_W \boldsymbol{w}$$

where $S_W = S_1 + S_2$ (Within-class scatter matrix)

$$\text{(A)} \Rightarrow J\left(\boldsymbol{w}\right) = \frac{\left(m_1 - m_2\right)^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

$$= \frac{\boldsymbol{w}^T \left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)^T \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = \frac{d}{d\boldsymbol{w}}\left(\frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)^T\boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}\right) \quad \left(d\left(\frac{f}{g}\right) = \frac{gdf-fdg}{g^2}\right)$$

$$= \boldsymbol{w}^T S_W \boldsymbol{w} \frac{2\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\left(\boldsymbol{w}^T S_W \boldsymbol{w}\right)^2} - \frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)\boldsymbol{w}}{\left(\boldsymbol{w}^T S_W \boldsymbol{w}\right)^2} 2\boldsymbol{w}^T S_W$$

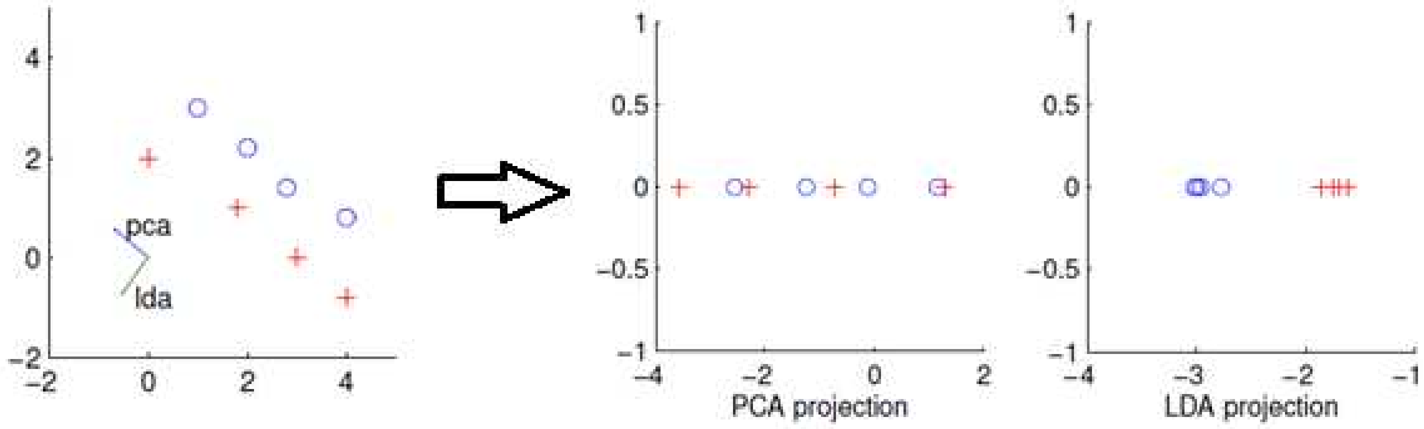
$$= \frac{2\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^T S_W \boldsymbol{w}} - \frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)\boldsymbol{w}}{\left(\boldsymbol{w}^T S_W \boldsymbol{w}\right)^2} 2\boldsymbol{w}^T S_W$$

$$= 2\frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^T S_W \boldsymbol{w}}\left((\boldsymbol{m}_1-\boldsymbol{m}_2) - \frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^T S_W \boldsymbol{w}} S_W \boldsymbol{w}\right) = 0 \text{ -------- (B)}$$

Let $\frac{\boldsymbol{w}^T(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^T S_W \boldsymbol{w}} = c.$ (B) $\Rightarrow c\left((\boldsymbol{m}_1-\boldsymbol{m}_2) - cS_W\boldsymbol{w}\right) = 0$

$$\Rightarrow \boldsymbol{w} = cS_W^{-1}(\boldsymbol{m}_1-\boldsymbol{m}_2).$$

Example:



- In $n > 2$ Class (d-D to k-D) case,

Within-class scatter matrix:

$$S_W = \sum_{i=1}^{n} S_i, \text{ where } S_i = \sum_t r_i^t \left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)^T$$

$$r_i^t = 1 \text{ if } \boldsymbol{x}^t \in C_i \text{ and } 0 \text{ otherwise}$$

Between-class scatter matrix:

$$S_B = \sum_{i=1}^{n} N_i \left(\boldsymbol{m}_i - \boldsymbol{m}\right)\left(\boldsymbol{m}_i - \boldsymbol{m}\right)^T, \ \ \boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{m}_i, \ N_i = \sum_t r_i^t$$

Let $W_{d\times k}$ be the projection matrix from the *d*-D space to the *k*-D space ($k < d$), then

$(W^T S_W W)_{k\times k}$, $(W^T S_B W)_{k\times k}$: projections of $(S_W)_{d\times d}$, $(S_B)_{d\times d}$

A spread measure of a scatter matrix is its determinant.

Find W such that $J\left(W\right) = \dfrac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$ is maximized.

The determinant of matrix $A_{n\times n}$ is the product of its eigenvalues, i.e., $|A| = \lambda_1 \cdot \lambda_2 \cdots \cdot \lambda_n$.

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} = \frac{|\lambda_1^B \cdot \lambda_2^B \cdots \cdot \lambda_k^B|}{|\lambda_1^W \cdot \lambda_2^W \cdots \cdot \lambda_k^W|} = \left|\frac{\lambda_1^B \cdot \lambda_2^B \cdots \cdot \lambda_k^B}{\lambda_1^W \cdot \lambda_2^W \cdots \cdot \lambda_k^W}\right|$$

$$= \left|(\lambda_1^W \cdot \lambda_2^W \cdots \cdot \lambda_k^W)^{-1}(\lambda_1^B \cdot \lambda_2^B \cdots \cdot \lambda_k^B)\right|$$

$$= \left\| W^T S_W W \right|^{-1} \left| W^T S_B W \right\| \underset{\substack{\uparrow \\ |AB|=|A||B|}}{=} \left|(W^T S_W W)^{-1}(W^T S_B W)\right|$$

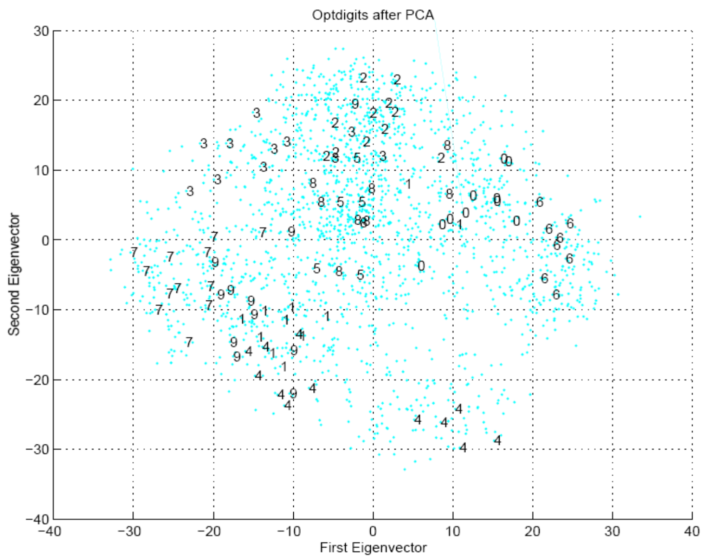
Let $\dfrac{dJ(W)}{dW} = \dfrac{d}{dW}\left|(W^T S_W W)^{-1}(W^T S_B W)\right|$

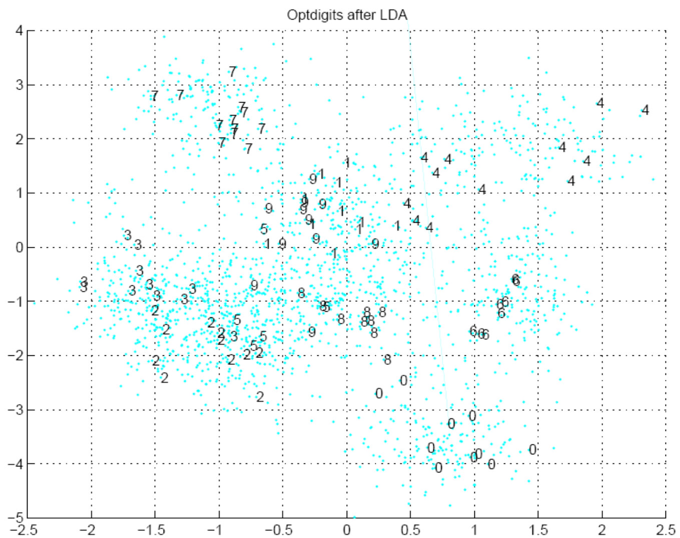
$$= \frac{d}{dW}\left|(W^{-1} S_W^{-1} (W^T)^{-1} W^T S_B W\right| = \frac{d}{dW}\left|(W^{-1} S_W^{-1} S_B W)\right| = 0.$$

The solution of W is formed by the k largest eigenvectors of $S_W^{-1} S_B$.

PCA

LDA





Fisher Discriminant Analysis with Kernels, S. Mika, G. Ratsch, J. Weaton, B. Scholkopf, and K.R. Muller, IEEE, 1999.

vi) Laplacian Eigenmaps (LE)

Let $\boldsymbol{x}^r$ and $\boldsymbol{x}^s$ be any two out of N data instances and b_{rs} is their similarity. Find $\boldsymbol{y}^r$ and $\boldsymbol{y}^s$ that

$\min \sum_{r,s} \left\| \boldsymbol{y}^r - \boldsymbol{y}^s \right\|^2 b_{rs}$, i.e., two similar instances (large b_{rs}) should be close in the new space (small $\left\| \boldsymbol{y}^r - \boldsymbol{y}^s \right\|$). Define $b_{rs} = \exp\left[-\frac{\left\| \boldsymbol{x}^r - \boldsymbol{x}^s \right\|^2}{2\sigma^2} \right]$

if $\boldsymbol{x}^r$ and $\boldsymbol{x}^s$ are in the predefined neighborhood, and 0 otherwise, i.e., only local similarities are cared.

Consider the 1-D new space

$$\sum_{r,s}\left\|\boldsymbol{y}^r-\boldsymbol{y}^s\right\|^2 b_{rs}=\frac{1}{2}\sum_{r,s}(y_r-y_s)^2 b_{rs}$$

$$=\frac{1}{2}\left(\sum_{r,s}b_{rs}y_r^2-2\sum_{r,s}b_{rs}y_r y_s+\sum_{r,s}b_{rs}y_s^2\right)$$

$$=\frac{1}{2}\left(\sum_{r}d_r y_r^2-2\sum_{r,s}b_{rs}y_r y_s+\sum_{s}d_s y_s^2\right) \quad \begin{matrix}(d_r=\sum_s b_{rs}, \\ d_s=\sum_r b_{rs})\end{matrix}$$

$$=\sum_r d_r y_r^2-\sum_r\sum_s b_{rs}y_r y_s=\boldsymbol{y}^T D\boldsymbol{y}-\boldsymbol{y}^T B\boldsymbol{y}$$

$$=\boldsymbol{y}^T(D-B)\,\boldsymbol{y}=\boldsymbol{y}^T L\boldsymbol{y},$$

where $B = [b_{rs}]$, $D = \text{diag}[d_r]$, L : Laplacian matrix

$\boldsymbol{y}$: N-D vector, y_r : the new coordinate of $\boldsymbol{x}^r$.

The solution to $\min\{\boldsymbol{y}^T L \boldsymbol{y}\}$ subject to $\|\boldsymbol{y}\| = 1$

$$\approx \frac{d(\boldsymbol{y}^T L \boldsymbol{y})}{d\boldsymbol{y}} = 0 \text{ subject to } \|\boldsymbol{y}\| = 1$$

$$\approx L\boldsymbol{y} = 0 \text{ subject to } \|\boldsymbol{y}\| = 1$$

- The method of Lagrange multipliers

Error: $E = \|L\boldsymbol{y} - \boldsymbol{0}\|^2 = \|L\boldsymbol{y}\|^2 = \boldsymbol{y}^T (L^T L)\, \boldsymbol{y}$

Constraint: $\|\boldsymbol{y}\| = 1$

Minimize $F(\boldsymbol{y}) = \boldsymbol{y}^T (L^T L) \boldsymbol{y} + \lambda(\|\boldsymbol{y}\|^2 - 1)$

where λ : Lagrange multiplier

Let $\dfrac{dF(\boldsymbol{y})}{d\boldsymbol{y}} = 2(L^T L)\boldsymbol{y} + 2\lambda \boldsymbol{y} = \boldsymbol{0}$

We obtain $(L^T L)\boldsymbol{y} = -\lambda \boldsymbol{y}$

The solution ***y*** is an eigenvector of $L^T L$ with eigenvalue $-\lambda$

The associated error $E = \boldsymbol{y}^T (L^T L)\boldsymbol{y} = -\lambda \boldsymbol{y}^T \boldsymbol{y} = -\lambda$

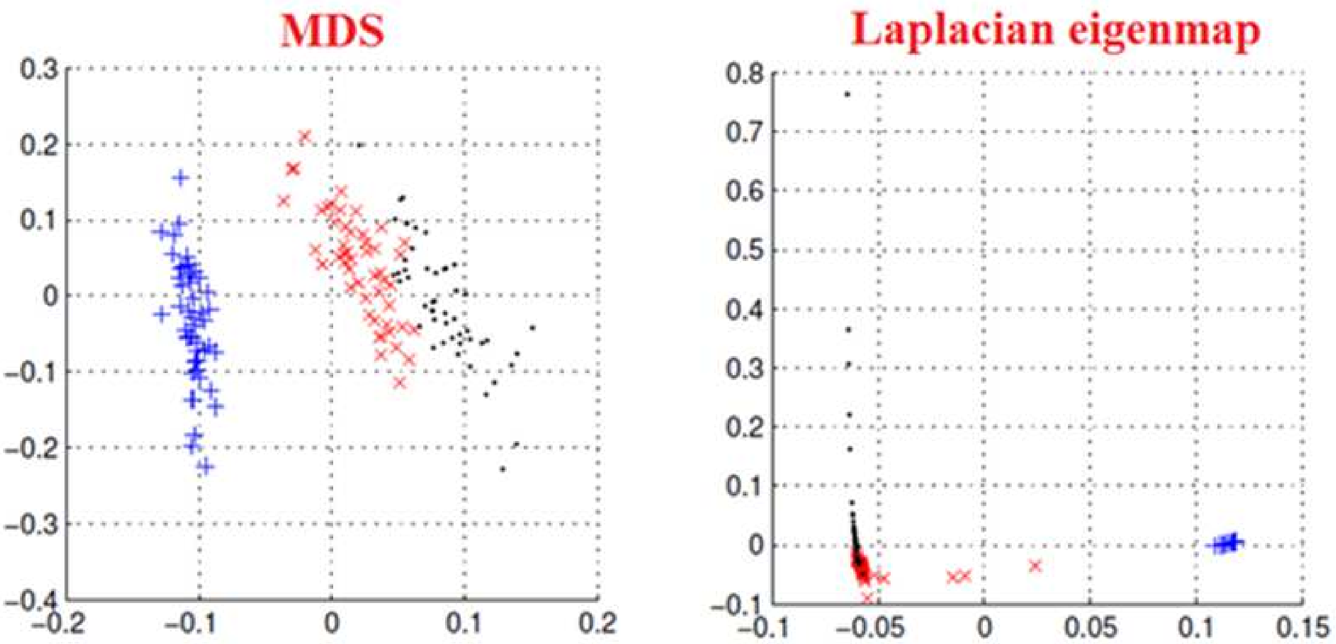
The ***y*** with the smallest λ is the least square error solution of $L\boldsymbol{y} = \boldsymbol{0}$.

LE Algorithm: Given N points $\boldsymbol{x}_i \in R^d$, $i = 1, \cdots, N$

1. Put an edge e_{ij} between nodes n_i and n_j if $\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\| < \varepsilon$.
2. Weight the edge by $b_{ij} = \exp\left(-\dfrac{\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2}{t} \right)$
3. For each connected component of G, compute eigenvalues λ and eigenvectors $\boldsymbol{y}$ of L, i.e., $L\boldsymbol{y} = \lambda \boldsymbol{y}$, where $L = D - B$, D: diagonal matrix, $d_{ii} = \sum_j b_{ji}$. Let $\boldsymbol{y}_i$, $i = 0, \cdots, k-1$ be the solutions of (A), ordered from small to large eigenvalues.

Example: Iris data

LE lead to denser data than MDS.



Laplace Eigenmaps for Dimensionality Reduction and Data Representation, M. Belkin, Neural Computing, 15, pp. 1373-1396, 2003.

ix) Canonical Correlation Analysis (CCA)

- Given a sample $X = \{\boldsymbol{x}^t, \boldsymbol{y}^t\}_{t=1}^N$, both x and y are inputs, e.g., (1) acoustic information and visual information in speech recognition, (2) image data and text annotations in image retrieval application.
- Take the correlation of $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ into account while reducing dimensionality to a joint space, i.e., find two vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ s.t. when $\boldsymbol{x}$ is projected along $\boldsymbol{w}$ and $\boldsymbol{y}$ is projected along $\boldsymbol{v}$, their correlation ρ is maximized, where

$$\rho = \frac{\mathrm{Cov}(\boldsymbol{w}^T\boldsymbol{x},\ \boldsymbol{v}^T\boldsymbol{y})}{\sqrt{\mathrm{Var}(\boldsymbol{w}^T\boldsymbol{x})}\sqrt{\mathrm{Var}(\boldsymbol{v}^T\boldsymbol{y})}}$$

$$= \frac{\boldsymbol{w}^T\mathrm{Cov}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{v}}{\sqrt{\boldsymbol{w}^T\mathrm{Var}(\boldsymbol{x})\boldsymbol{w}}\sqrt{\boldsymbol{v}^T\mathrm{Var}(\boldsymbol{y})\boldsymbol{v}}} = \frac{\boldsymbol{w}^T S_{xy}\boldsymbol{v}}{\sqrt{\boldsymbol{w}^T S_{xx}\boldsymbol{w}}\sqrt{\boldsymbol{v}^T S_{yy}\boldsymbol{v}}}$$

where Covariance matrices:

$$S_{xx} = \mathrm{Var}(\boldsymbol{x}) = E[(\boldsymbol{x}-\boldsymbol{\mu}_x)^2]$$

$$S_{yy} = \mathrm{Var}(\boldsymbol{y}) = E[(\boldsymbol{y}-\boldsymbol{\mu}_y)^2]$$

Cross-covariance matrices:

$$S_{xy} = \mathrm{Cov}(\boldsymbol{x},\boldsymbol{y}) = E[(\boldsymbol{x}-\boldsymbol{\mu}_x)(\boldsymbol{y}-\boldsymbol{\mu}_y)^T]$$

$$S_{yx} = \mathrm{Cov}(\boldsymbol{y},\boldsymbol{x}) = E[(\boldsymbol{y}-\boldsymbol{\mu}_y)(\boldsymbol{x}-\boldsymbol{\mu}_x)^T]$$

Let $\frac{\partial \rho}{\partial \boldsymbol{w}} = 0$ and $\frac{\partial \rho}{\partial \boldsymbol{v}} = 0$

Solutions: $\boldsymbol{w}$ is an eigenvector of $S_{xx}^{-1} S_{xy} S_{yy}^{-1} S_{yx}$;

$\boldsymbol{v}$ is an eigenvector of $S_{yy}^{-1} S_{yx} S_{xx}^{-1} S_{xy}$.

Choose ($\boldsymbol{w}$, $\boldsymbol{v}$) with largest eigenvalues as the solution.

$\because$ $\rho \propto$ shared eigenvalue of λ_w and λ_v

- Look for k pairs $(\boldsymbol{w}_i, \boldsymbol{v}_i)$, $i = 1, \cdots, k$

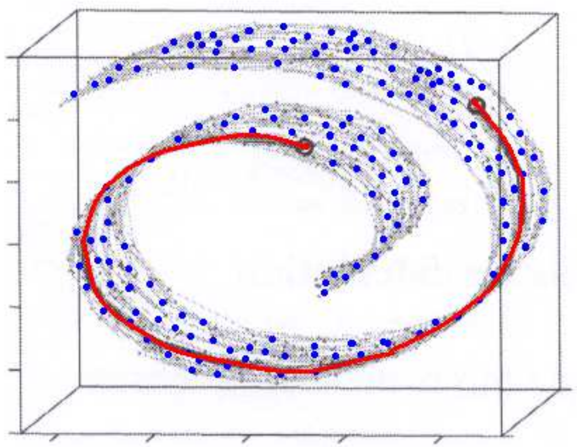
Let $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k]$, $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k]$

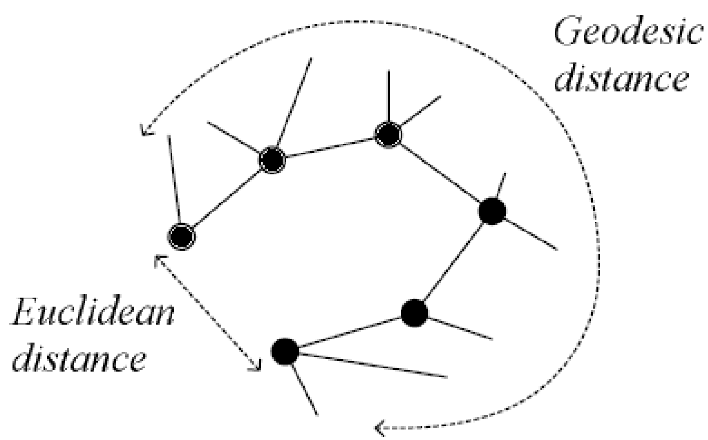
$\boldsymbol{r}^t = W^T \boldsymbol{x}^t$, $\boldsymbol{s}^t = V^T \boldsymbol{y}^t$.

$(\boldsymbol{r}^t, \boldsymbol{s}^t)$: lower-dimensional representation of $(\boldsymbol{x}^t, \boldsymbol{y}^t)$.

Canonical Correlation Analysis: An Overview with Application to Learning Methods, D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, Neural Computation, 16, 2004.

vi) Isometric Feature Mapping (Isomap)





-- Estimates the geodesic distance and applies multidimensional scaling for dimensionality reduction.

$$
D = \begin{array}{cc}
 & \begin{array}{cccc} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdot\cdot & \boldsymbol{x}_N \end{array} \\
\begin{array}{c} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \cdot\cdot \\ \boldsymbol{x}_N \end{array} &
\begin{pmatrix}
d_{11} & d_{12} & \cdot\cdot & d_{1N} \\
d_{21} & d_{22} & \cdot\cdot & d_{2N} \\
\cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\
d_{N1} & d_{N2} & \cdot\cdot & d_{NN}
\end{pmatrix}
\end{array}
$$

Multidimensional Scaling (MDS):

Given distance matrix $D=[d_{rs}]_{N\times N}$,

1. Calculate $B=\left[b_{rs}\right]_{N\times N}$, where

$$b_{rs}=\frac{1}{2}(d_{r\bullet}^2+d_{\bullet s}^2-d_{\bullet\bullet}^2-d_{rs}^2),\ d_{r\bullet}^2=\frac{1}{N}\sum_s d_{rs}^2,$$

$$d_{\bullet s}^2=\frac{1}{N}\sum_r d_{rs}^2,\ \ d_{\bullet\bullet}^2=\frac{1}{N^2}\sum_r\sum_s d_{rs}^2$$

2. Find the spectral decomposition of B,

$$B=E\Lambda E^T.$$

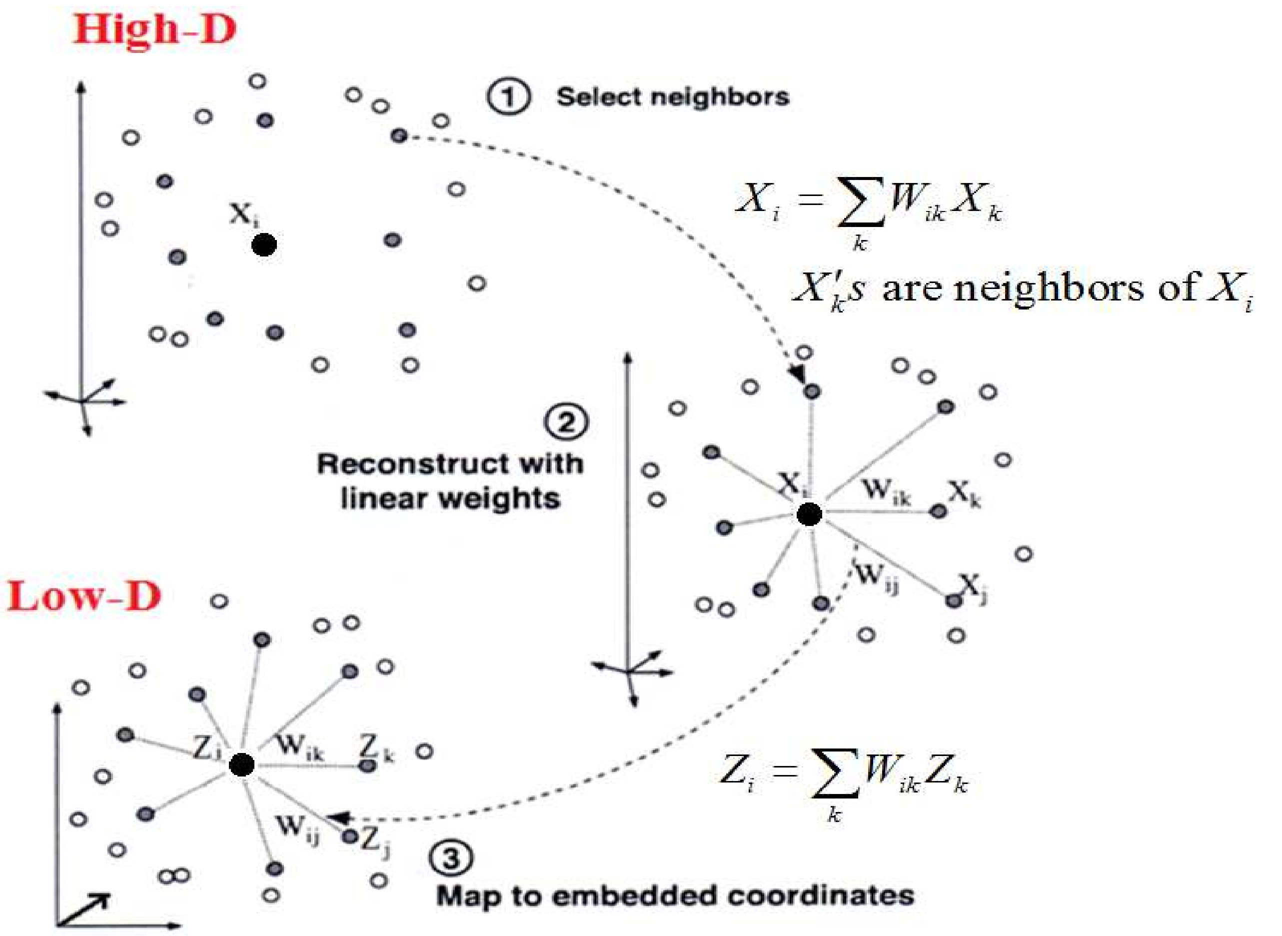
3. For dimensionality reduction, discard from Λ the p small eigenvalues and from E the corresponding eigenvectors to form Λ' and E', respectively.

4. Find $Z_{(N-p)\times(N-p)} = E'\Lambda'^{1/2}$.

* The coordinates of the points are the rows of Z.

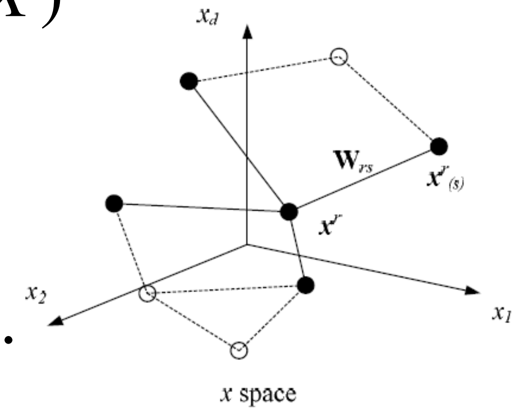
vii) Locally Linear Embedding (LLE)

-- Recovers global nonlinear structure from locally linear fit.



1. Given $\boldsymbol{x}^r$ and its neighbors $\boldsymbol{x}^s_{(r)}$, find W_{rs} by

minimizing the error function $E(W \mid X)$

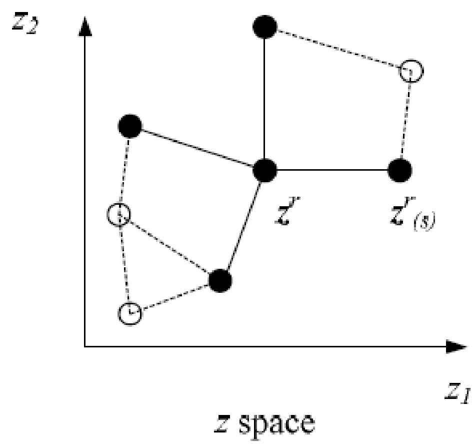


$$\min_{W_{rs}} E(W \mid X) = \sum_r \left\| \boldsymbol{x}^r - \sum_s W_{rs} \boldsymbol{x}^s_{(r)} \right\|^2$$

subject to $W_{rr} = 0,\ \forall r$ and $\sum_r W_{rs} = 1.$

2. Find new coordinates $\boldsymbol{z}^r$ that respect the constraints

given by W_{rs} ,



$$\min_{\boldsymbol{z}^r} E(Z \mid W) = \sum_r \left\| \boldsymbol{z}^r - \sum_s W_{rs} \boldsymbol{z}^s_{(r)} \right\|^2$$

subject to $E[\boldsymbol{z}] = 0,\ \mathrm{Cov}(\boldsymbol{z}) = I$

Dim($\boldsymbol{z}$-space) < Dim ($\boldsymbol{x}$-space)

Nonlinear Dimensionality Reduction by Locally Linear Embedding, S.T. Roweis and L.K. Saul, Science, 290, 2000.

viii) *t*-Distributed Stochastic Neighbor Embedding

-- Position the data points in the new space such that local neighborhood statistics are as similar as possible

The probability that $\boldsymbol{x}^s$ is a neighbor of $\boldsymbol{x}^r$

$$p_{s|r} = \frac{\exp[-\left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|^2 / 2\sigma_r^2]}{\sum_{l \neq r} \exp[-\left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|^2 / 2\sigma_r^2]}$$

t-SNE is the symmetrized version of SNE by defining $p_{rs} = \frac{p_{s|r} + p_{r|s}}{2N}$

The probability in the lower-dimensional space is calculated as

$$q_{rs} = \frac{(1+\left\| \boldsymbol{z}^l - \boldsymbol{z}^m \right\|^2)^{-1}}{\sum_{l \neq r} \sum_{m \neq l} (1+\left\| \boldsymbol{z}^l - \boldsymbol{z}^m \right\|^2)^{-1}} (t-\text{distribution})$$

The aim is to learn $\boldsymbol{z}^r$ so that for all pairs (r,s) , q_{rs} can be as close as possible to p_{rs}

Gradient descent for finding optimal $\boldsymbol{z}^r$:

1. Start from small random $\boldsymbol{z}^r$
2. Update iteratively in the direction that decreases the KL distance in small steps

KL distance between the probability distributions P and Q, from which p_{rs} and q_{rs} are drawn

$$\mathrm{KL}(P \,\|\, Q) = \sum_r \sum_s p_{rs} \log \frac{p_{rs}}{q_{rs}}$$