CHAPTER 6:

DİMENSİONALİTY REDUCTİON

Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Fewer parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- 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

Subset Selection

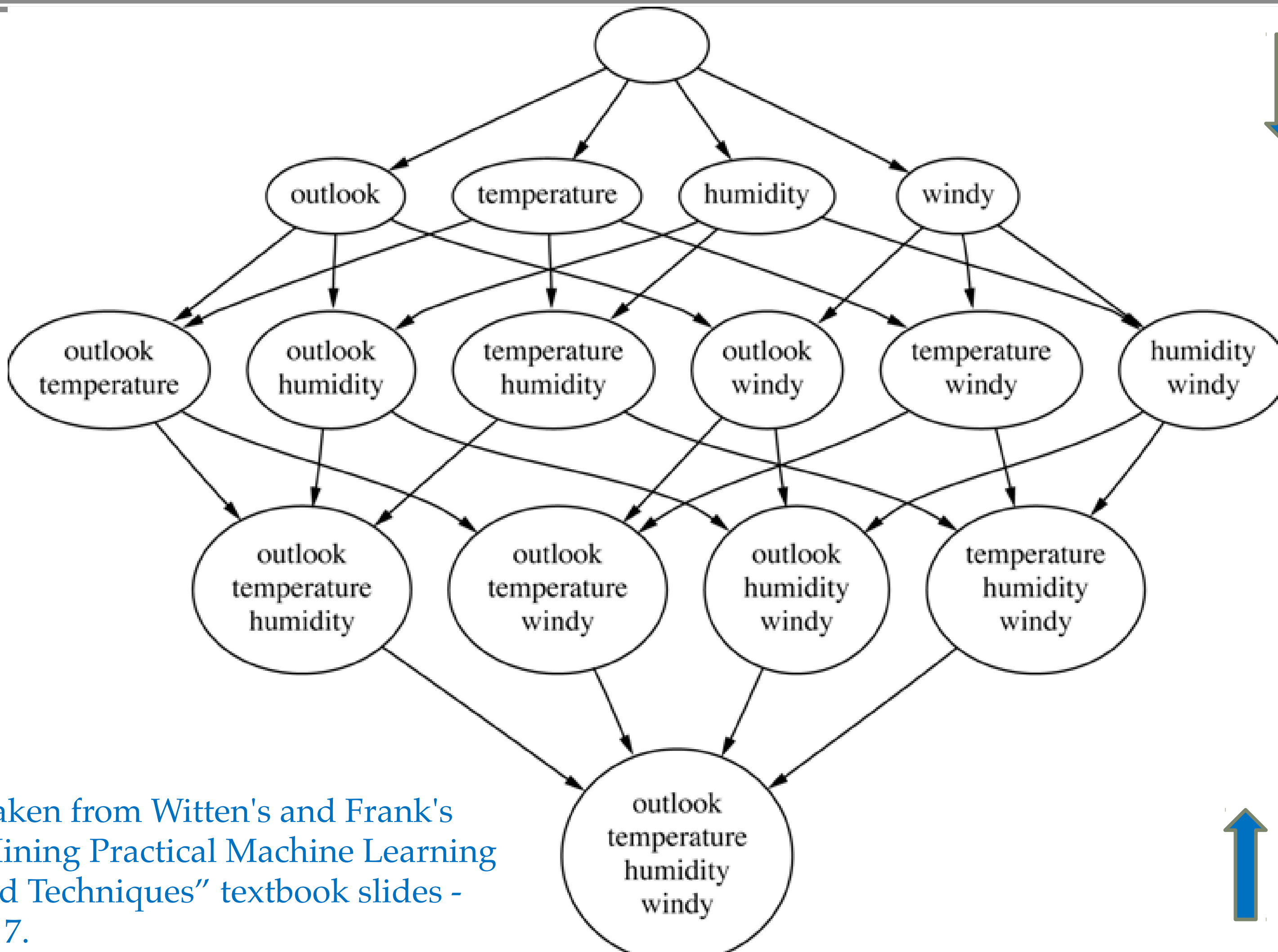
- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially Ø.
  - At each iteration, find the best new feature

    $j = \text{argmin}_i \; E (F \cup x_i)$ where E(.) is the error on the validation set
  - 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)

Example: weather dataset

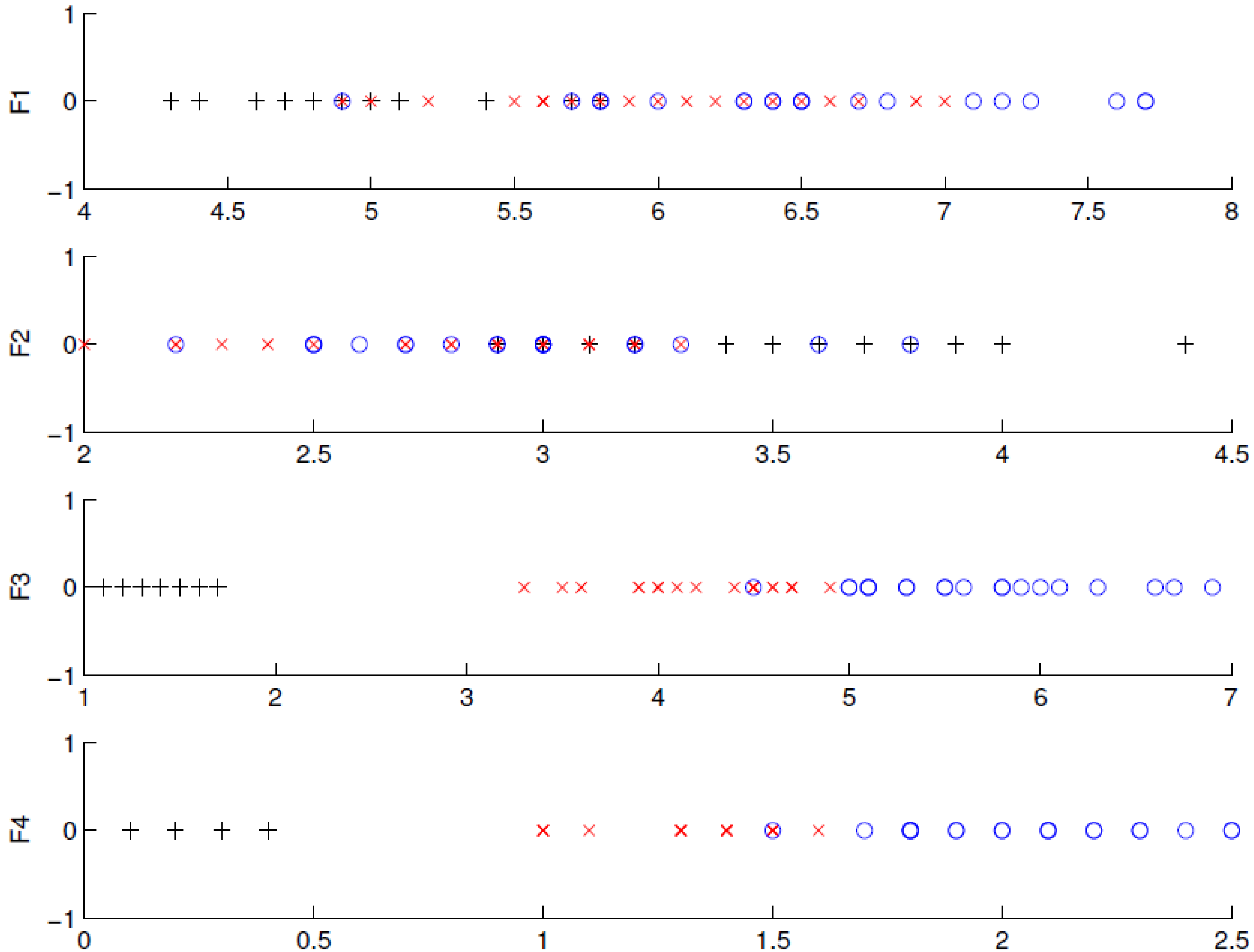
forward



backward

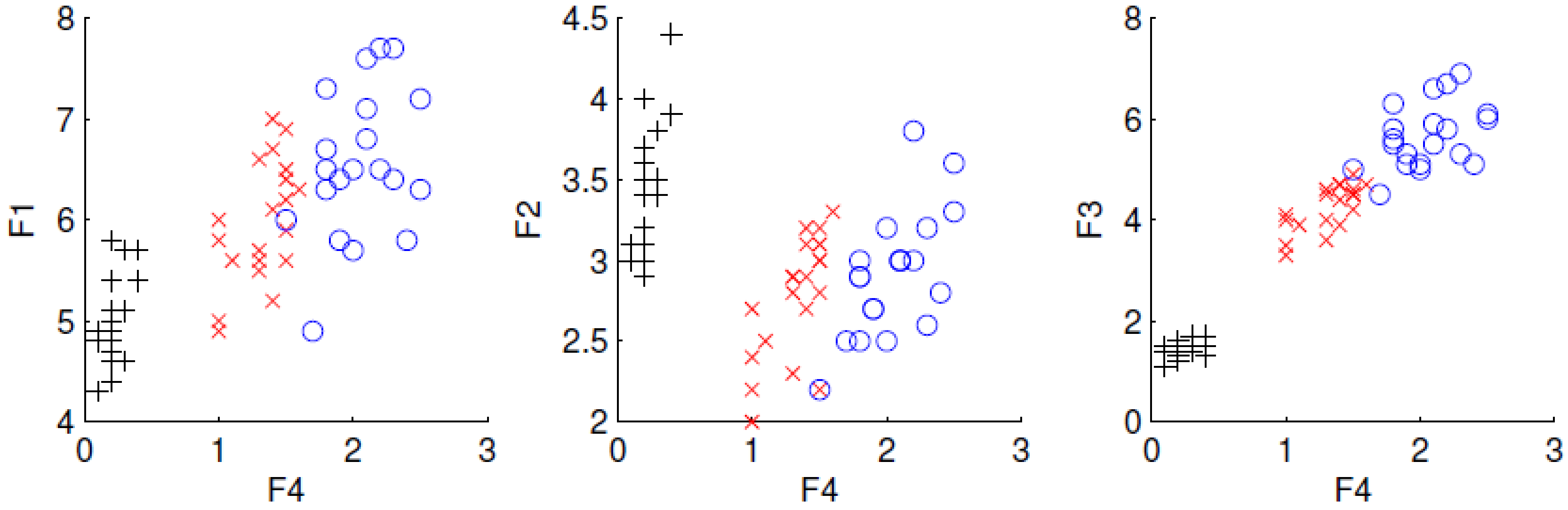
Figure taken from Witten's and Frank's "Data Mining Practical Machine Learning Tools and Techniques" textbook slides - Chapter 7.

Another Example Iris data: Single feature Training da



Chosen

Iris data: Add one more feature to F4



Chosen

Principal Components Analysis

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

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

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

- Maximize Var(z) subject to $||w||=1$ (i.e, $w_1^T w_1 = 1$)

  using Lagrange formulation: $$\max_{\mathbf{w}_1} \mathbf{w}_1^T \Sigma \mathbf{w}_1 - \alpha\left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right)$$

  taking derivative w.r.t. w_1 and making it equal to 0:

  $\sum w_1 = \alpha w_1$ that is, w_1 is an eigenvector of $\sum$ and α its eigenvalue

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

- Second principal component: Max Var(z_2), s.t., $||w_2||$ =1 and orthogonal to w_1

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

$\sum w_2 = \alpha\, w_2$ that is, w_2 is another eigenvector of $\sum$

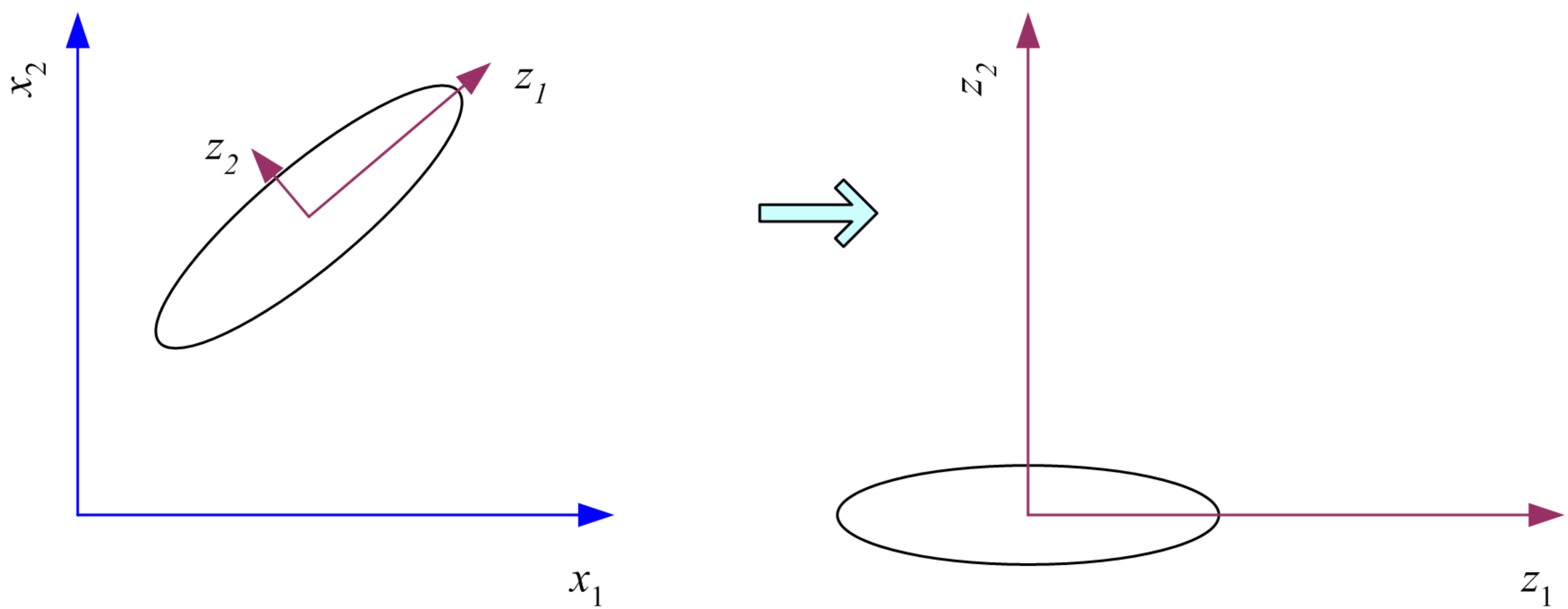
and so on.

What PCA does

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

where the columns of **W** are the eigenvectors of $\sum$ and $\boldsymbol{m}$ is sample mean

Centers the data at the origin and rotates the

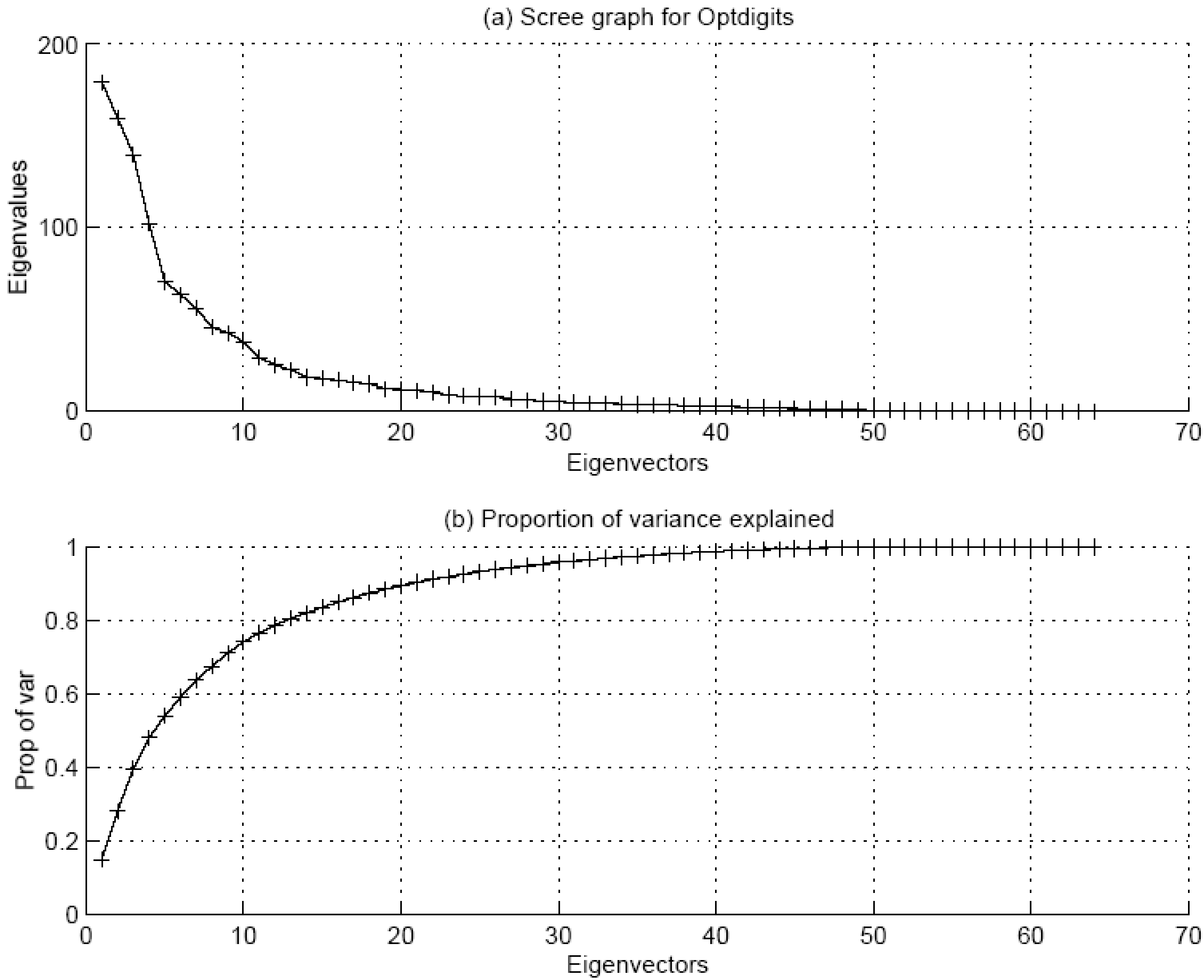


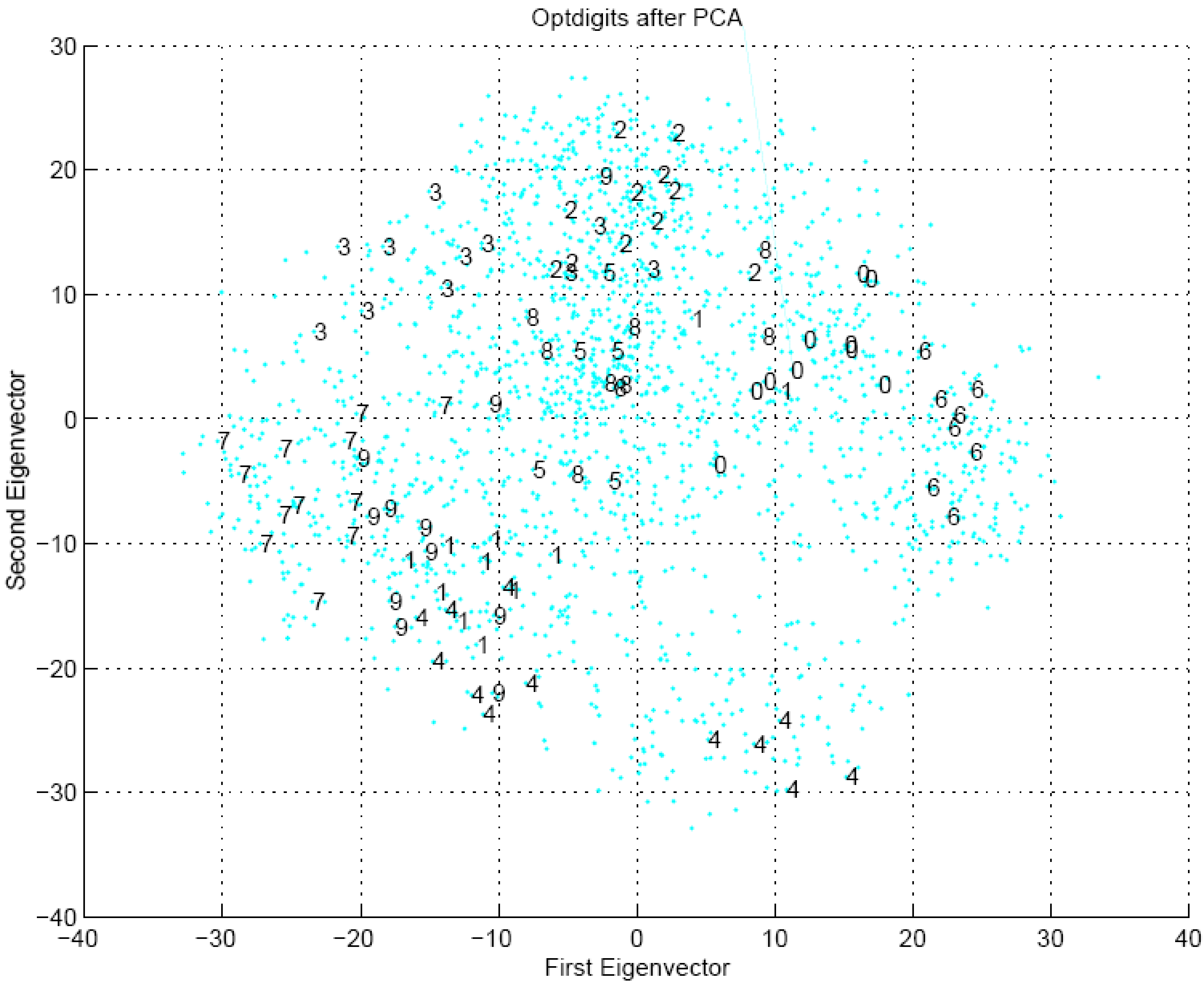
How to choose k ?

- 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”


(a) Scree graph for Optdigits
Eigenvalues
200
100
0
0
10
20
30
40
50
60
70
Eigenvectors
(b) Proportion of variance explained
Prop of var
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
70
Eigenvectors


Optdigits after PCA
First Eigenvector
Second Eigenvector

Feature Embedding

- When $\boldsymbol{X}$ is the Nxd data matrix,

$\boldsymbol{X}^T\boldsymbol{X}$ is the dxd matrix (covariance of features, if mean-centered)

$\boldsymbol{X}\boldsymbol{X}^T$ is the NxN matrix (pairwise similarities of instances)

- PCA uses the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ which are d-dim and can be used for projection
- Feature embedding uses the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ which are N-dim and which give directly the coordinates after projection
- Sometimes, we can define pairwise similarities (or distances) between instances, then we can use feature embedding without needing to represent instances as vectors.

Factor Analysis

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

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

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

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

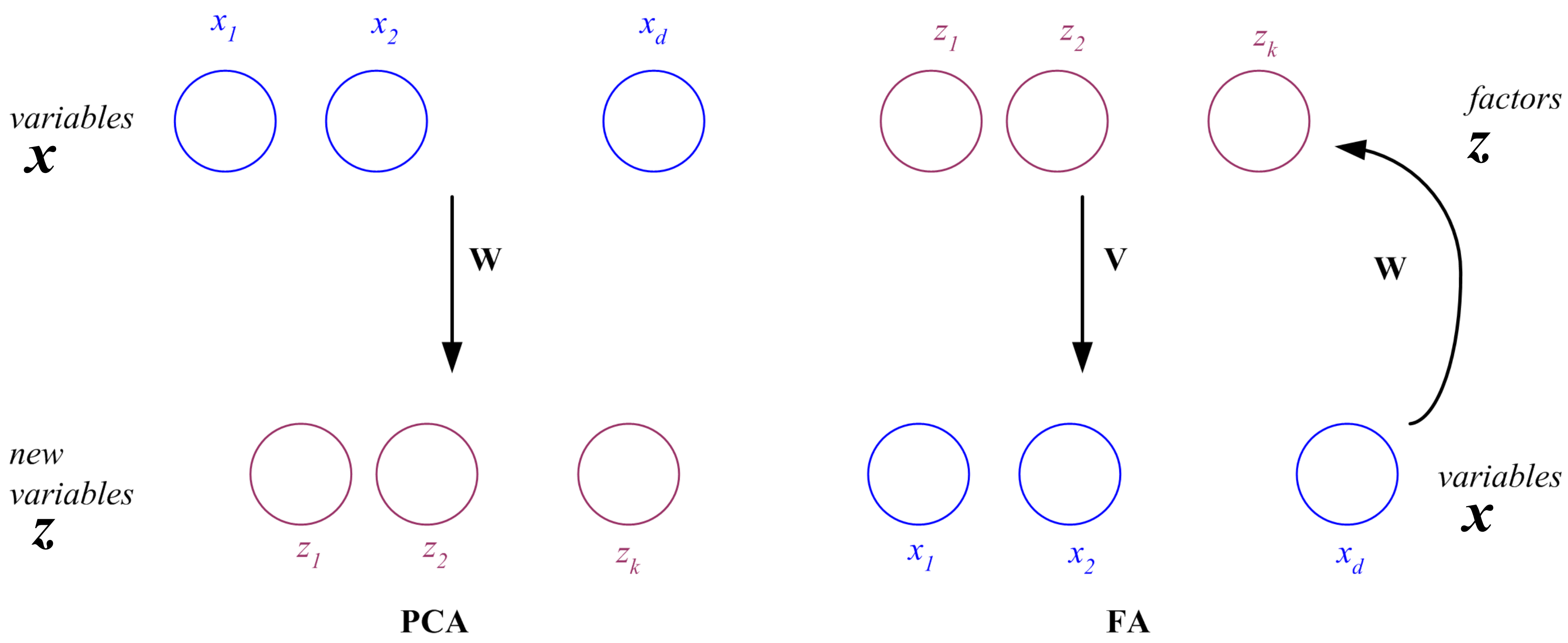
  ε_i are the noise sources

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

  and v_{ij} are the factor loadings

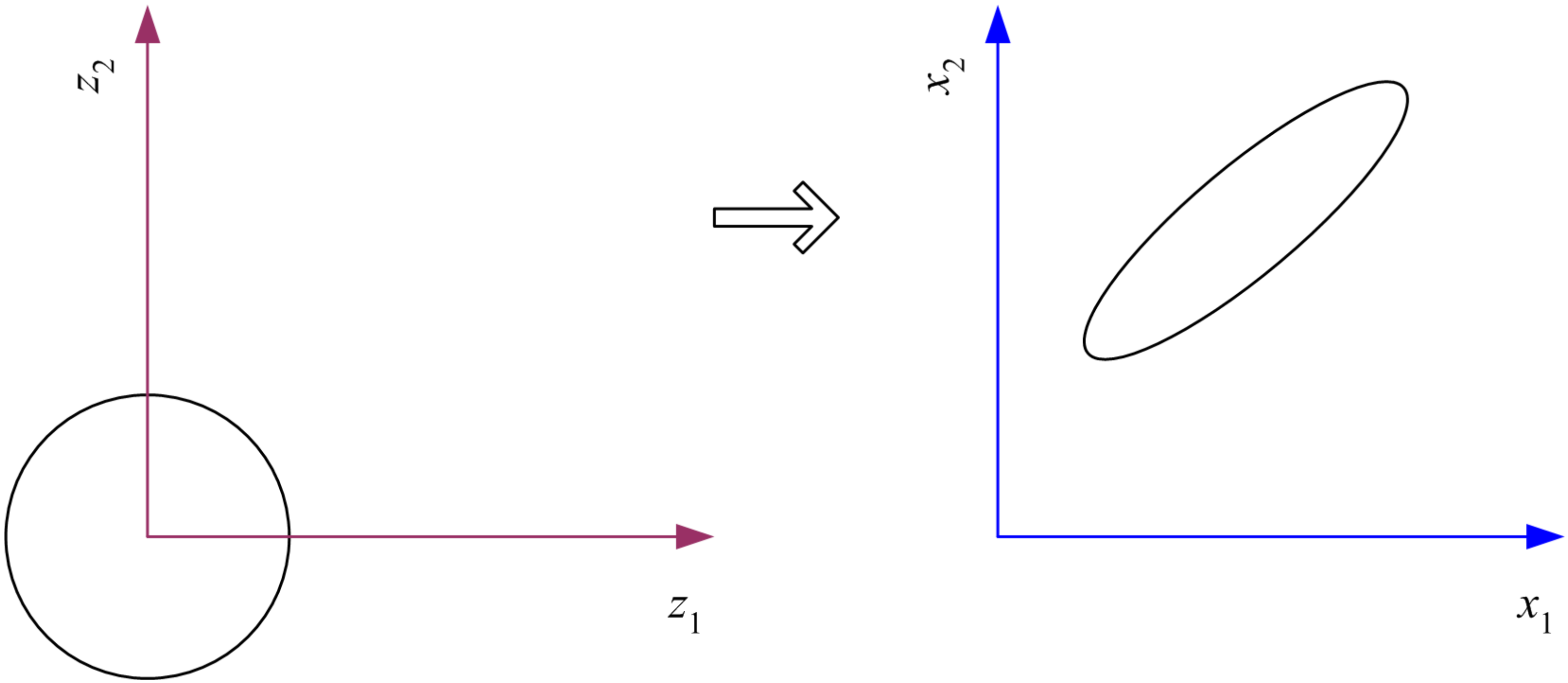
PCA vs FA

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



Factor Analysis

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



Singular Value Decomposition and Matrix Factorization

- Singular value decomposition: $\boldsymbol{X}=\boldsymbol{VAW}^T$

  $\boldsymbol{V}$ is NxN and contains the eigenvectors of $\boldsymbol{XX}^T$

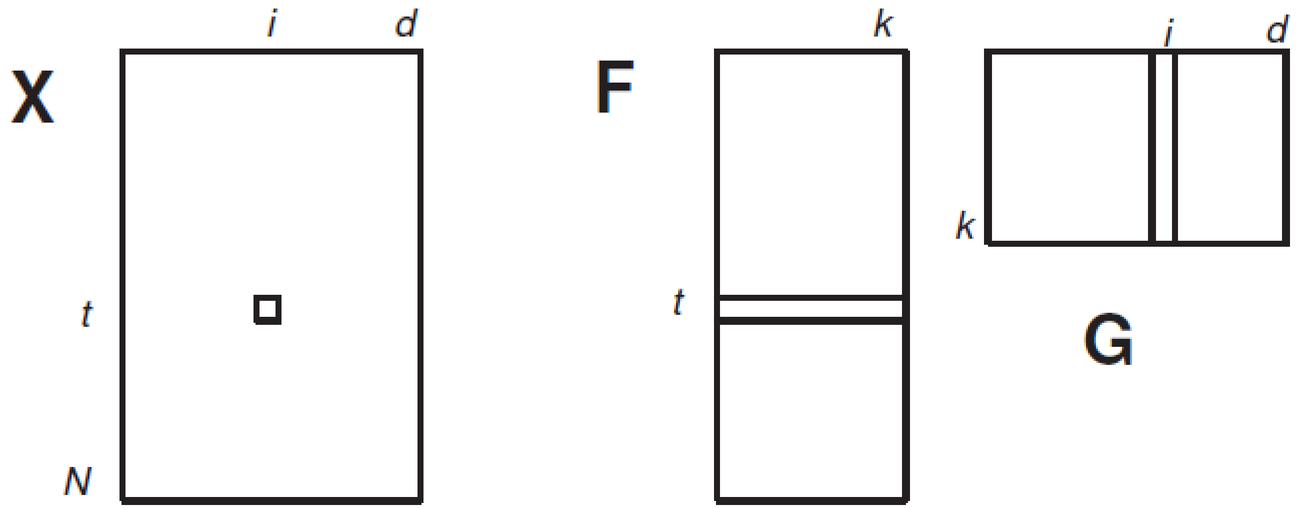
  $\boldsymbol{W}$ is dxd and contains the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$

  and **A** is Nxd and contains singular values on its first k diagonal
- $\boldsymbol{X}=\boldsymbol{u}_1 a_1 \boldsymbol{v}_1^T+\ldots+\boldsymbol{u}_k a_k \boldsymbol{v}_k^T$ where k is the rank of $\boldsymbol{X}$

Matrix Factorization

- Matrix factorization: $\boldsymbol{X}=\boldsymbol{FG}$

  $\boldsymbol{F}$ is Nxk and $\boldsymbol{G}$ is kxd



$$\mathbf{X}_{ti} = \mathbf{F}_t^T \mathbf{G}_i = \sum_{j=1}^{k} \mathbf{F}_{tj} \mathbf{G}_{ji}$$

Latent semantic indexing

Multidimensional Scaling

- Given pairwise distances between N points,

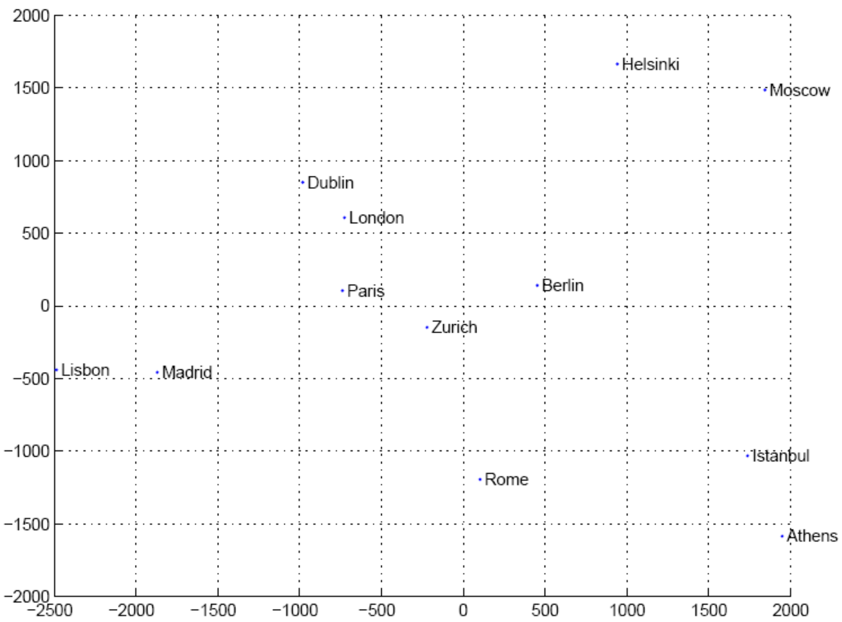
  d_{ij}, $i,j = 1,\ldots,N$

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

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

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

Map of Europe by MDS

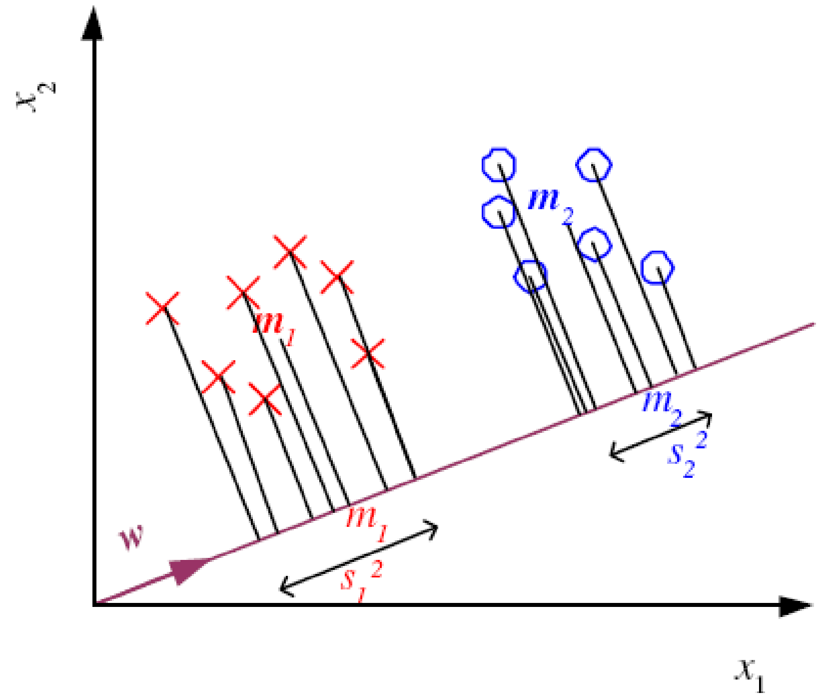


Map from CIA – The World Factbook: http://www.cia.gov/

Linear Discriminant Analysis

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

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



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

- Between-class scatter:

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

- Within-class scatter:

$$\begin{aligned}s_1^2 &= \sum_t \left(\mathbf{w}^T\mathbf{x}^t - m_1\right)^2 r^t\\ &= \sum_t \mathbf{w}^T(\mathbf{x}^t - \mathbf{m}_1)(\mathbf{x}^t - \mathbf{m}_1)^T\mathbf{w}r^t = \mathbf{w}^T\mathbf{S}_1\mathbf{w}\end{aligned}$$

$$\text{where } \mathbf{S}_1 = \sum_t (\mathbf{x}^t - \mathbf{m}_1)(\mathbf{x}^t - \mathbf{m}_1)^T r^t$$

$$s_1^2 + s_1^2 = \mathbf{w}^T\mathbf{S}_W\mathbf{w} \text{ where } \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Find *w* that max

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

- LDA soln:

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

- Parametric soln:

$$\mathbf{w} = \Sigma^{-1} (\mu_1 - \mu_2)$$

$$\text{when } p(\mathbf{x} \mid C_i) \sim \mathcal{N}\,(\mu_i, \Sigma)$$

K>2 Classes

- Within-class scatter:

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

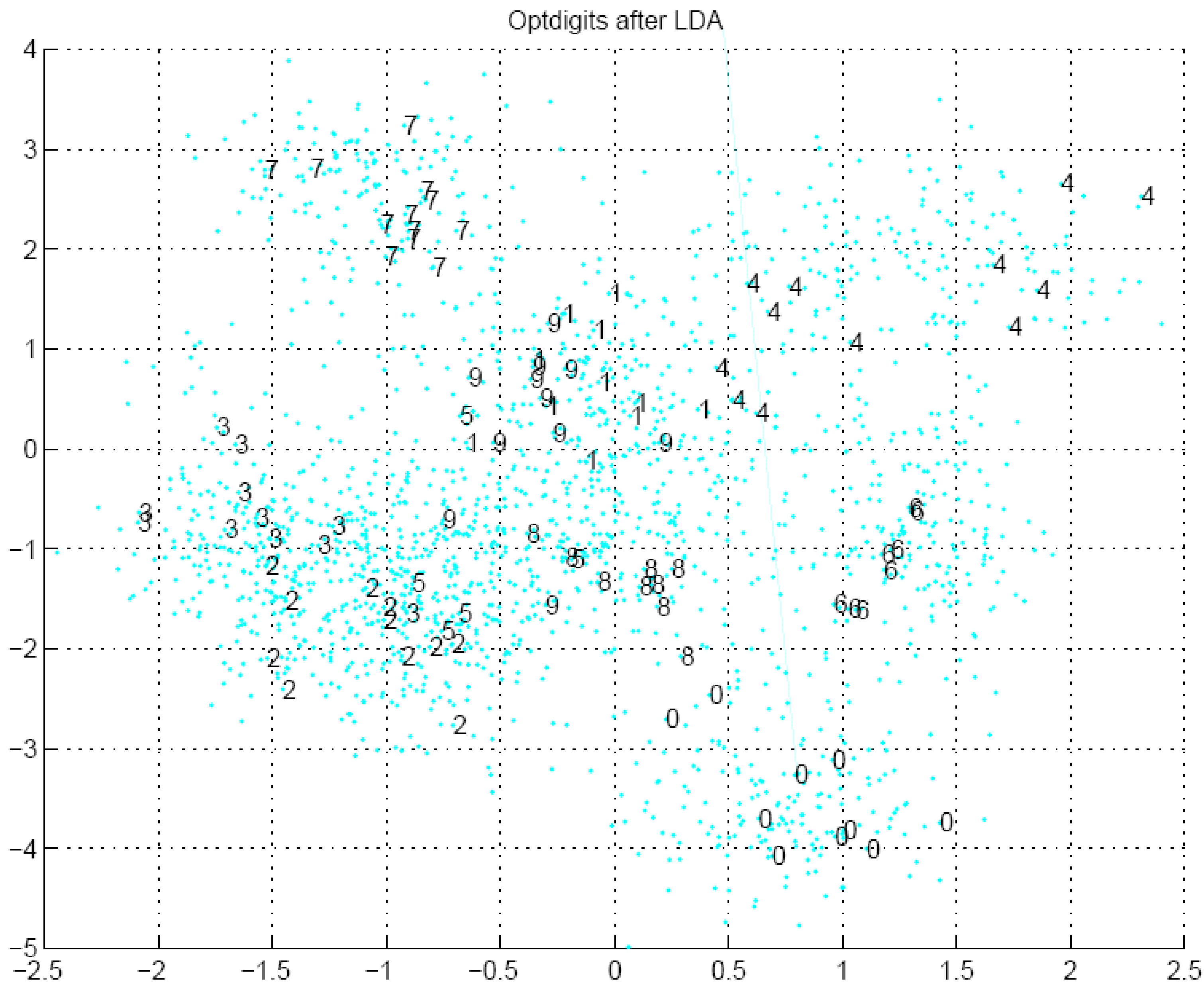
- Between-class scatter:

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

- Find **W** that max

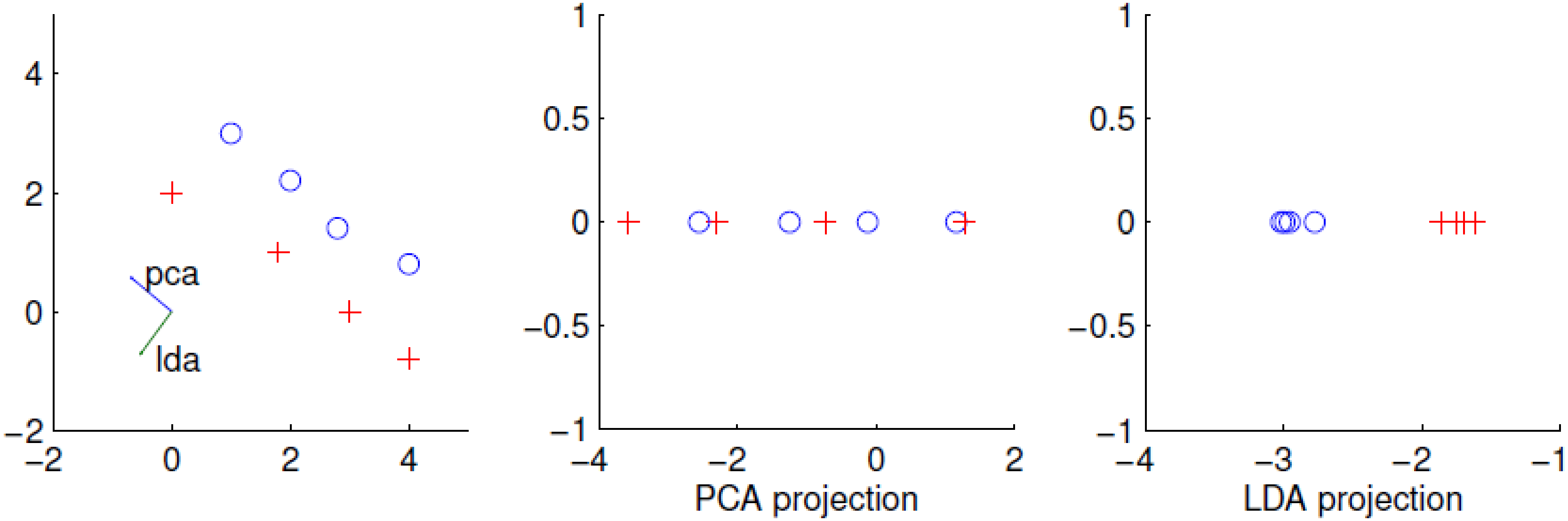
$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_{B;}$ maximum rank of K-1



Optdigits after LDA
4
3
2
1
0
−1
−2
−3
−4
−5
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5

PCA vs LDA



4
2
0
−2
−2
0
2
4
pca
lda
1
0.5
0
−0.5
−1
−4
−2
0
2
PCA projection
1
0.5
0
−0.5
−1
−4
−3
−2
−1
LDA projection

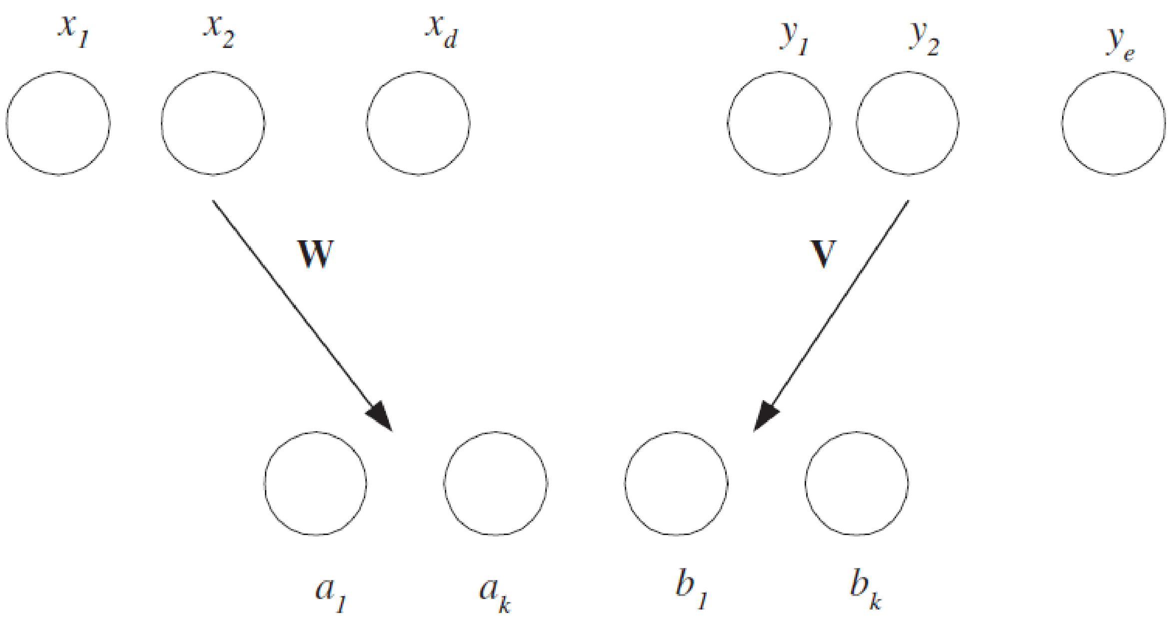
Canonical Correlation Analysis

- $\boldsymbol{X}=\{x^t,\mathbf{y}^t\}_t$; two sets of variables $\boldsymbol{x}$ and $\boldsymbol{y}$ x
- We want to find two projections $\boldsymbol{w}$ and $\boldsymbol{v}$ st when $\boldsymbol{x}$ is projected along $\boldsymbol{w}$ and $\boldsymbol{y}$ is projected along $\boldsymbol{v}$, the correlation is maximized:

$$
\begin{aligned}
\rho &= \mathrm{Corr}(\boldsymbol{w}^T\boldsymbol{x}, \boldsymbol{v}^T\boldsymbol{y}) = \frac{\mathrm{Cov}(\boldsymbol{w}^T\boldsymbol{x}, \boldsymbol{v}^T\boldsymbol{y})}{\sqrt{\mathrm{Var}(\boldsymbol{w}^T\boldsymbol{x})}\sqrt{\mathrm{Var}(\boldsymbol{v}^T\boldsymbol{y})}} \\
&= \frac{\boldsymbol{w}^T\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{v}}{\sqrt{\boldsymbol{w}^T\mathrm{Var}(\boldsymbol{x})\boldsymbol{w}}\sqrt{\boldsymbol{v}^T\mathrm{Var}(\boldsymbol{y})\boldsymbol{v}}} = \frac{\boldsymbol{w}^T\mathbf{S}_{xy}\boldsymbol{v}}{\sqrt{\boldsymbol{w}^T\mathbf{S}_{xx}\boldsymbol{w}}\sqrt{\boldsymbol{v}^T\mathbf{S}_{yy}\boldsymbol{v}}}
\end{aligned}
$$

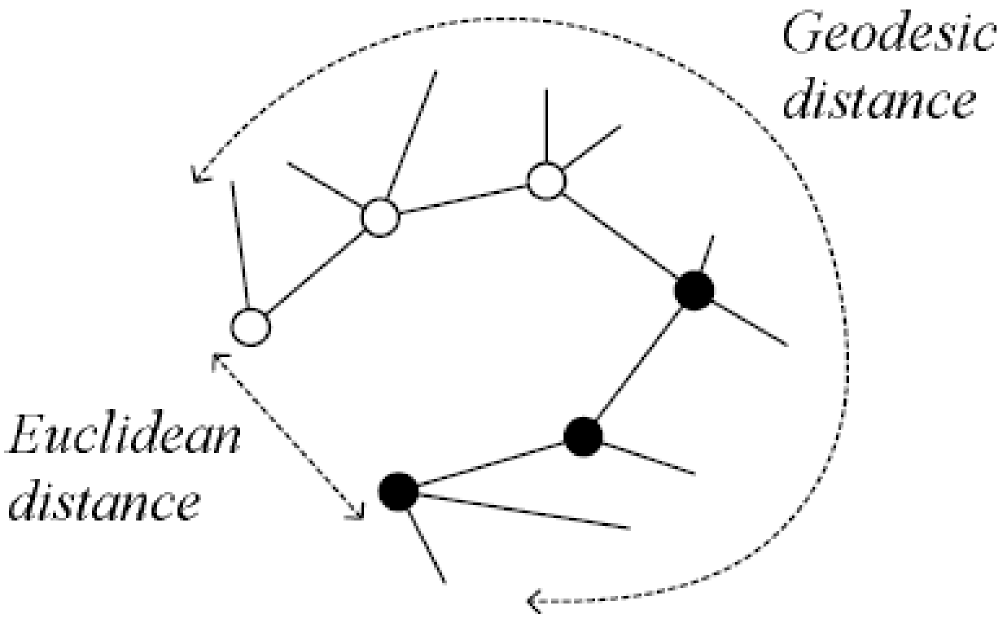
CCA

- *x* and *y* may be two different views or modalities; e.g., image and word tags, and CCA does a joint mapping



Isomap

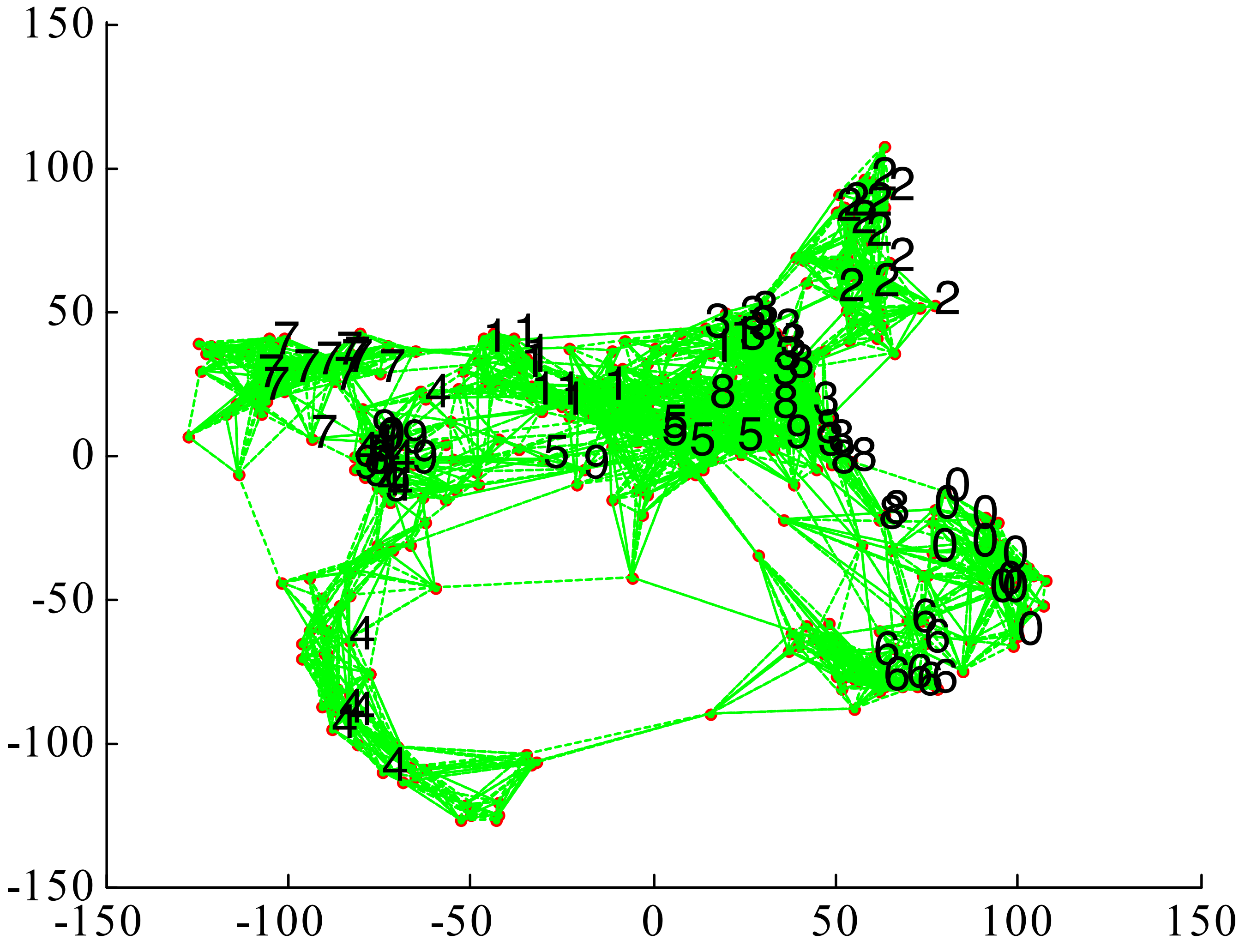
- Geodesic distance is the distance along the manifold that the data lies in, as opposed to the Euclidean distance in the input space



Isomap

- Instances r and s are connected in the graph if $\|\boldsymbol{x}^r-\boldsymbol{x}^s\|<\varepsilon$ or if $\boldsymbol{x}^s$ is one of the k neighbors of $\boldsymbol{x}^r$
  The edge length is $\|\boldsymbol{x}^r-\boldsymbol{x}^s\|$
- For two nodes r and s not connected, the distance is equal to the shortest path between them
- Once the NxN distance matrix is thus formed, use MDS to find a lower-dimensional mapping

Optdigits after Isomap (with neighborhood graph).



Matlab source from
http://web.mit.edu/cocosci/isomap/isomap.html

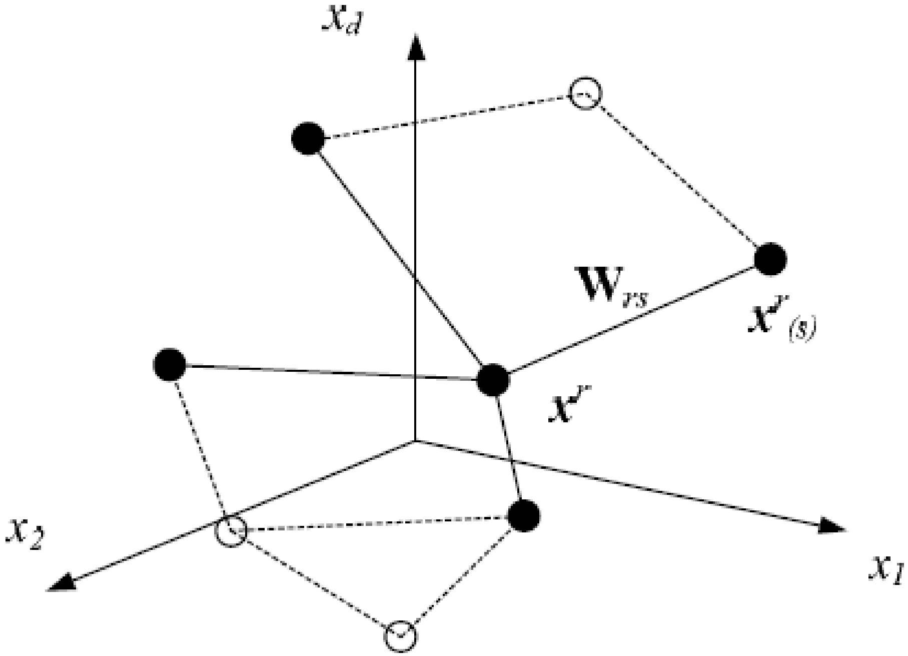
Locally Linear Embedding

1. Given $\boldsymbol{x}^r$ find its neighbors $\boldsymbol{x}^s_{(r)}$
2. Find $\mathbf{W}_{rs}$ that minimize

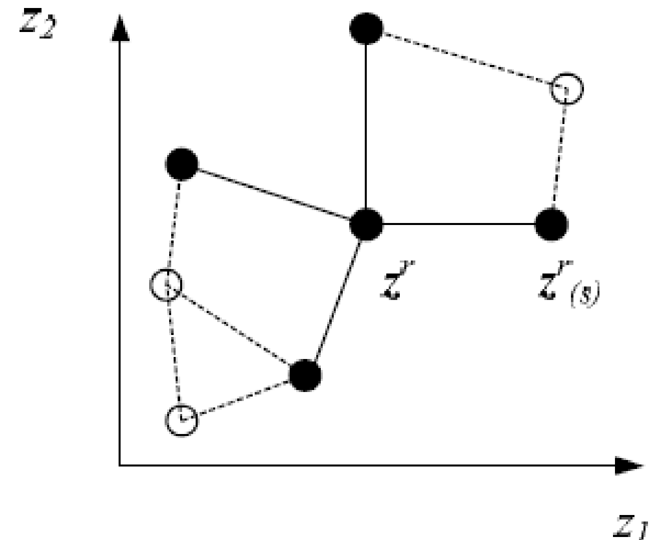
$$E(\mathbf{W} \mid X) = \sum_r \left\| \mathbf{x}^r - \sum_s \mathbf{W}_{rs} \mathbf{x}^s_{(r)} \right\|^2$$

3. Find the new coordinates $\boldsymbol{z}^r$ that minimize

$$E(\mathbf{z} \mid \mathbf{W}) = \sum_r \left\| z^r - \sum_s \mathbf{W}_{rs} z^s_{(r)} \right\|^2$$

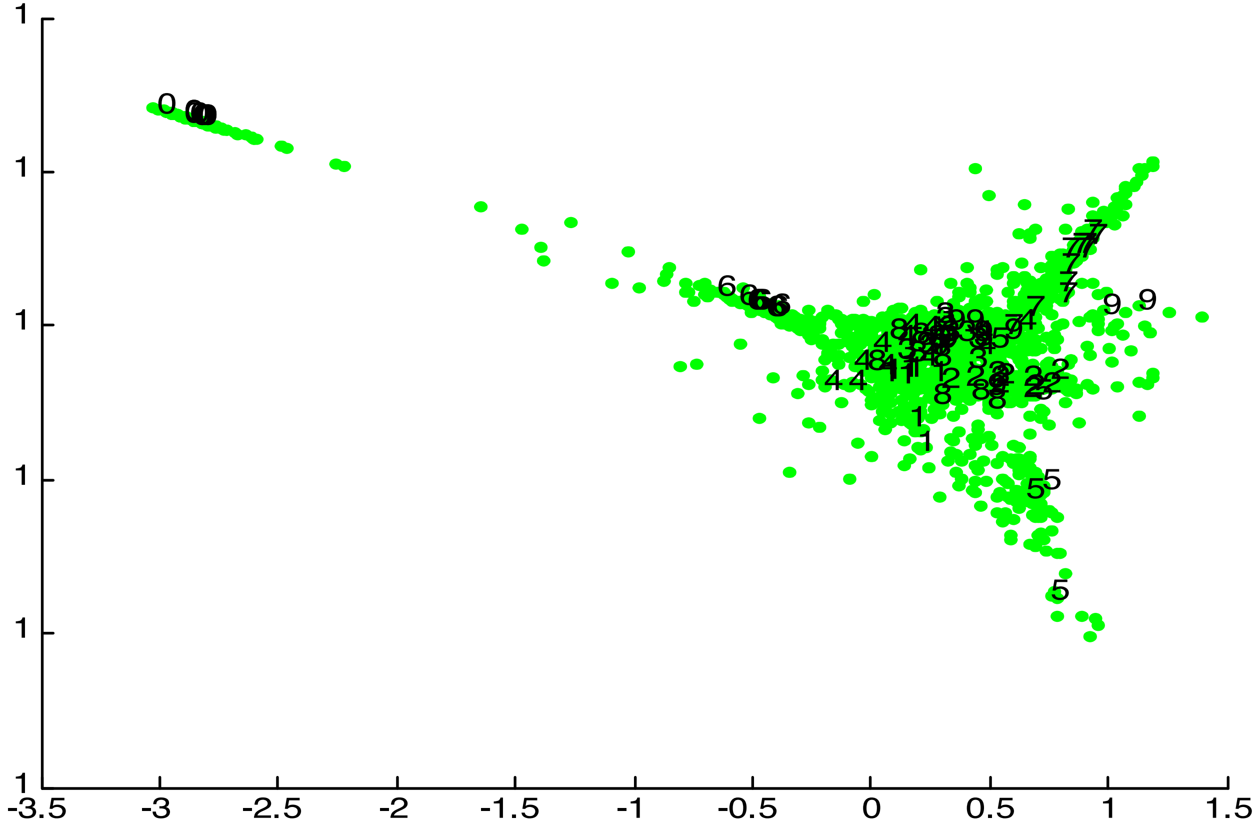


x space



z space

LLE on Optdigits



Matlab source from
http://www.cs.toronto.edu/~roweis/lle/code.html

Laplacian Eigenmaps

- Let r and s be two instances and B_{rs} is their similarity, we want to find $\boldsymbol{z}^r$ and $\boldsymbol{z}^s$ that

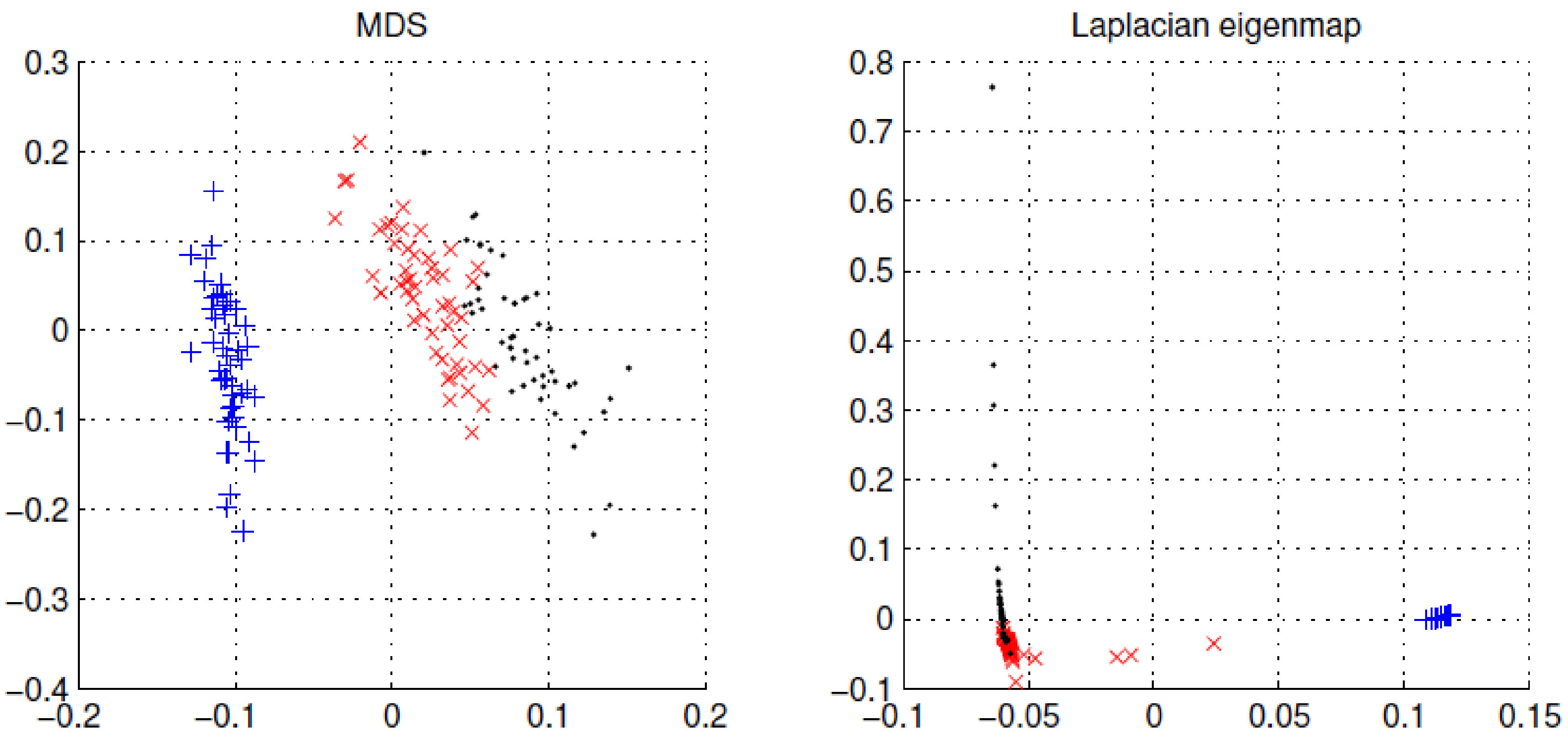
$$\min \sum_{r,s} \| \mathbf{z}^r - \mathbf{z}^s \|^2 B_{rs}$$

- B_{rs} can be defined in terms of similarity in an original space: 0 if $\boldsymbol{x}^r$ and $\boldsymbol{x}^s$ are too far, otherwise

$$B_{rs} = \exp\left[-\frac{\| \mathbf{x}^r - \mathbf{x}^s \|^2}{2\sigma^2} \right]$$

- Defines a graph Laplacian, and feature embedding returns $\boldsymbol{z}^r$

Laplacian Eigenmaps on Iris



Spectral clustering (chapter 7)