Introduction to Machine Learning and Data Mining Lecture-13: Dimensionality Reduction

Prof. Eugene Chang

Today

- Finish Lecture-12
  - Inverted Index
  - NLTK resources
    - http://www.nltk.org/book/
    - http://textminingonline.com/dive-into-nltk-part-i-getting-started-with-nltk
- Dimensionality Reduction
- Slides are partly based on materials by Prof Ethem Alpaydin, MIT

Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Less 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

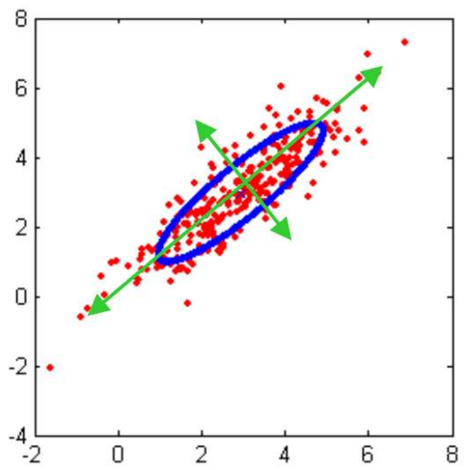
    original x_i , i =1,...,d dimensions to

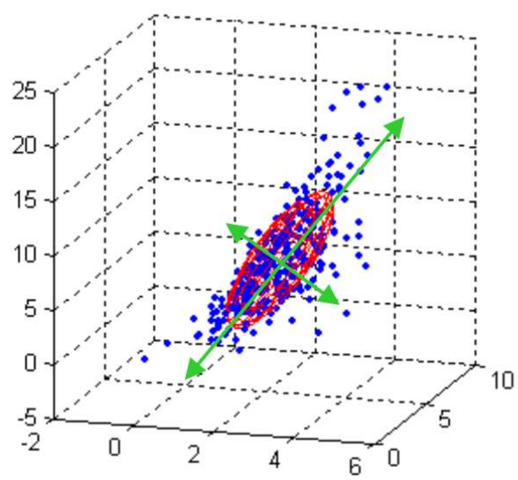
    new $k<d$ dimensions, z_j , j =1,...,k

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

Principal Component Analysis

- Principal Component Analysis (PCA) identifies the dimensions of greatest variance of a set of data.





Principal Components Analysis (PCA)

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

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

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

Principal Components Analysis (PCA)

- Maximize Var(z) subject to $||w||=1$

$$\max_{\mathbf{w}_1} \mathbf{w}_1^T \Sigma \mathbf{w}_1 - \lambda_1 \left(\mathbf{w}_1^T \mathbf{w}_1 - 1 \right)$$

$\sum w_1 = \alpha w_1$ that is, w_1 is an eigenvector of $\sum$ (a symmetric matrix)
Choose the one with the largest eigenvalue for Var(z) to be max

λ_1

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

$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \Sigma \mathbf{w}_2 - \lambda_1 \left(\mathbf{w}_2^T \mathbf{w}_2 - 1 \right) - \lambda_2 \left(\mathbf{w}_2^T \mathbf{w}_1 - 0 \right)$$

$\sum w_2 = \lambda_2 w_2$ that is, w_2 is another eigenvector of $\sum$

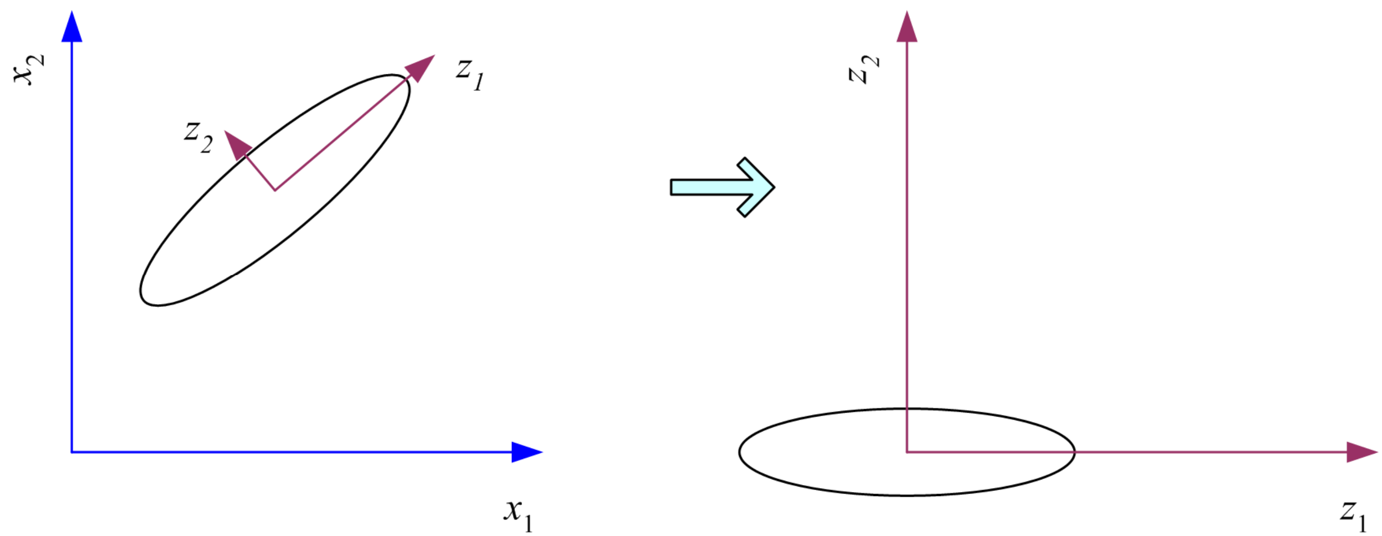
and so on.

Principal Component Analysis

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

where the columns of W are the eigenvectors of ∑, and $\boldsymbol{\mu}$ is sample mean

Centers the data at the origin and rotates the axes



Eigenvectors

- Eigenvectors are orthogonal vectors that define a space, the eigenspace.
- Any data point can be described as a linear combination of eigenvectors.
- Eigenvectors of a square matrix have the following property.

$$A\vec{v} = \lambda\vec{v}$$

- The associated lambda is the **eigenvalue**.

Eigenvectors of the Covariance Matrix

$$\begin{bmatrix} \Sigma(1,1) & \Sigma(1,2) & \Sigma(1,3) \\ \Sigma(1,2) & \Sigma(2,2) & \Sigma(2,3) \\ \Sigma(1,3) & \Sigma(2,3) & \Sigma(3,3) \end{bmatrix} = \begin{bmatrix} [\vec{v}_1] & [\vec{v}_2] & [\vec{v}_3] \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} [\vec{v}_1] & [\vec{v}_2] & [\vec{v}_3] \end{bmatrix}$$

- Eigenvectors are orthonormal $\Sigma = V \Lambda V^T$
- In the eigenspace, the Gaussian is diagonal – zero covariance.
- All eigen values are non-negative.
- Eigenvalues are sorted. $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots$
- Larger eigenvalues, higher variance

Dimensionality Reduction with PCA

- To convert from an original data point to PCA

$$c_{ij} = (\vec{x_i} - \vec{\mu})^T \vec{v_j}$$

- To reconstruct a point

$$\left\{ \left(x'_1 = \mu + \sum_{j=1}^{C} c_{1j} \vec{v_j} \right), \ldots, \left(x'_n = \mu + \sum_{j=1}^{C} c_{nj} \vec{v_j} \right) \right\}$$

Identifying Eigenvectors

- PCA is easy once we have eigenvectors and the mean.
- Identifying the mean is easy.
- Eigenvectors of the covariance matrix, represent a set of direction of variance.
- Eigenvalues represent the degree of the variance.

$$\vec{x_i} \approx \vec{\mu} + \sum_{j=1}^{C} c_{ij}\vec{v_j}$$

Dimensionality reduction with PCA

- Write each data point in this new space

$$\vec{x_i} \approx \vec{\mu} + \sum_{j=1}^{C} c_{ij}\vec{v_j}$$

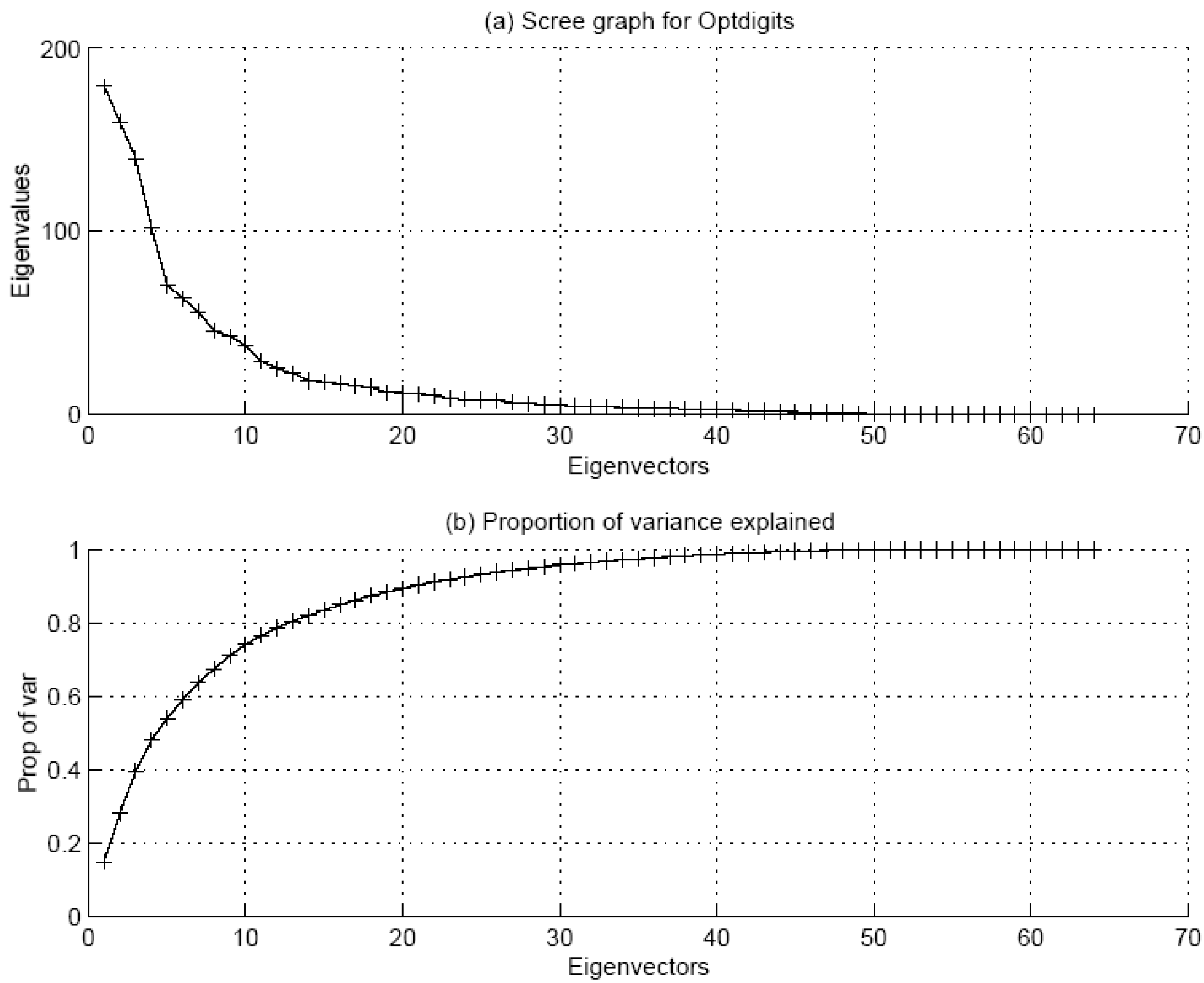
- To do the dimensionality reduction, keep C < D dimensions.
- Each data point is now represented as a vector of c's.

How to choose c ?

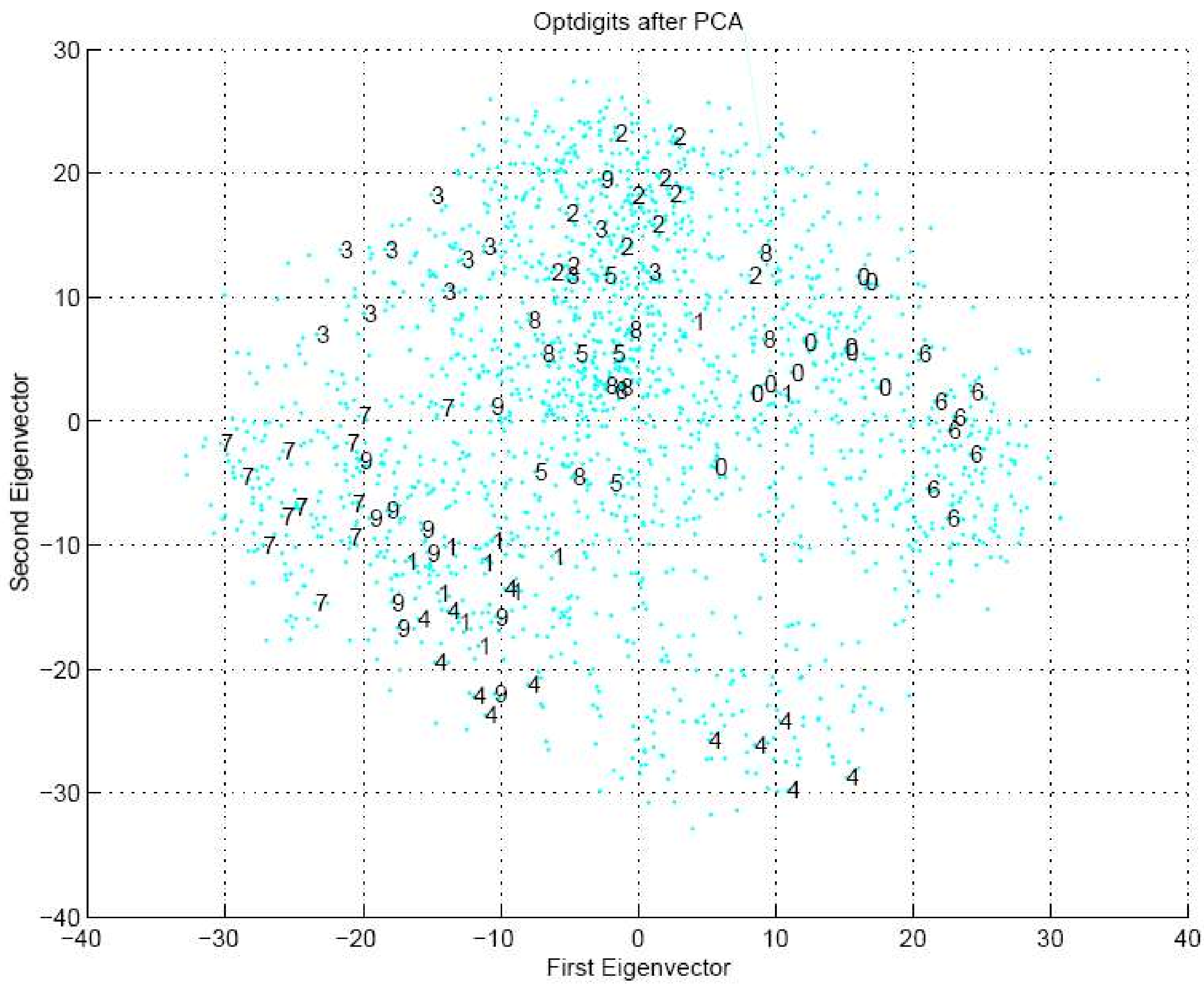
- Proportion of Variance (PoV) explained

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

  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

Example: Eigenfaces

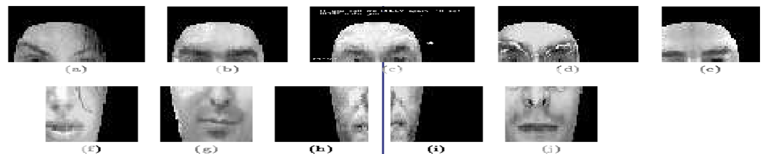
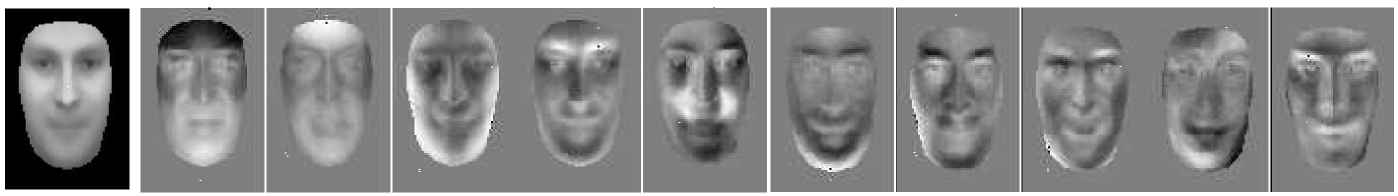


Figure 4.5: A gallery of face normalization results. (a) (b) (c) (d) (e) The original faces. (f) (g) (h) (i) (j) The corresponding synthesized mug-shots.

$$\{\vec{x_1}, \ldots, \vec{x_n}\}$$



$\vec{\mu}$ $\vec{v_1}$ $\vec{v_C}$

$$c_{ij} = (\vec{x_i} - \vec{\mu})^T \vec{v_j}$$

$$\left\{ \left(x_1' = \mu + \sum_{j=1}^{C} c_{1j} \vec{v_j} \right), \ldots, \left(x_n' = \mu + \sum_{j=1}^{C} c_{nj} \vec{v_j} \right) \right\}$$

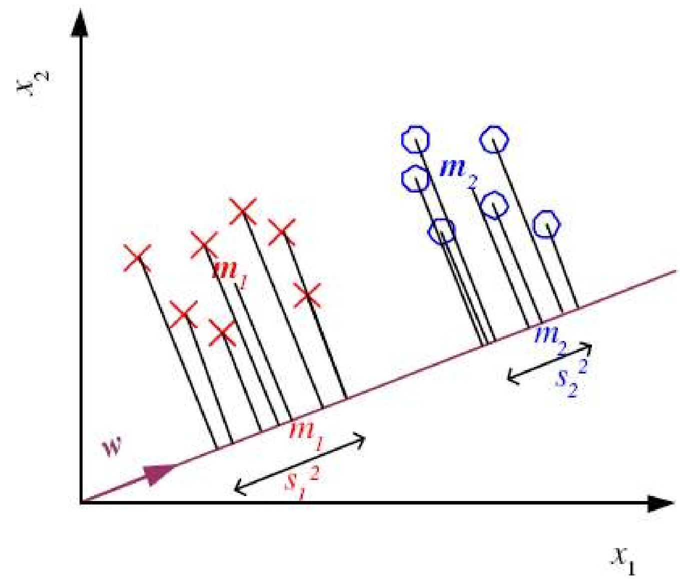
Encoded then Decoded.

Efficiency can be evaluated with Absolute or Squared error

Figure 4.7: Re-approximating the gallery's mug-shot images after KL-encoding

Linear Discriminant Analysis

- Find a low-dimensional space such that when *x* is projected, classes are well-separated.
- Find *w* that maximizes



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

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

- 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:

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

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

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

Fisher's Linear Discriminant

- Find *w* that max $J(\mathbf{w}) = \dfrac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \dfrac{\left|\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$

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

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

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

K>2 Classes

- Within-class scatter:

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

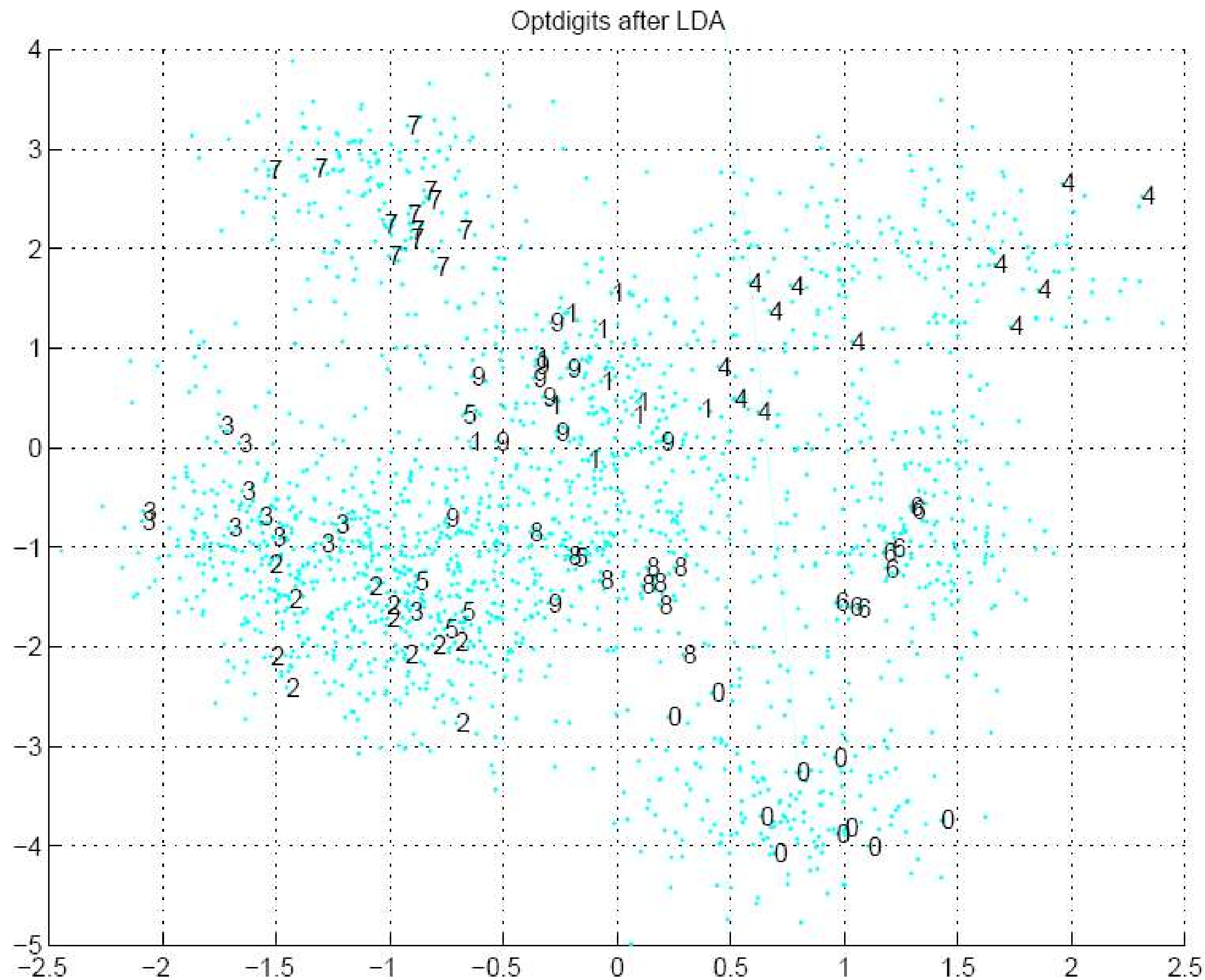
- Between-class scatter:

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

- 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 $S_W^{-1} S_B$
Maximum rank of K-1


Optdigits after LDA

Factor Analysis

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

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

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

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

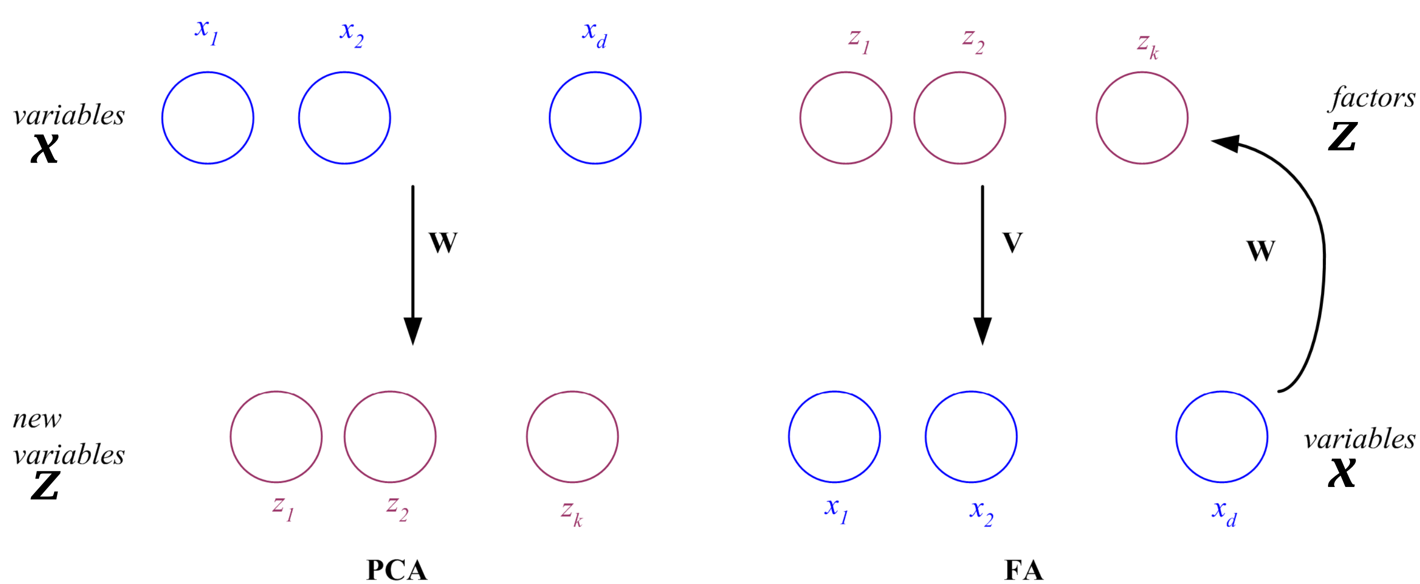
ε_i are the noise sources

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

and v_{ij} are the factor loadings

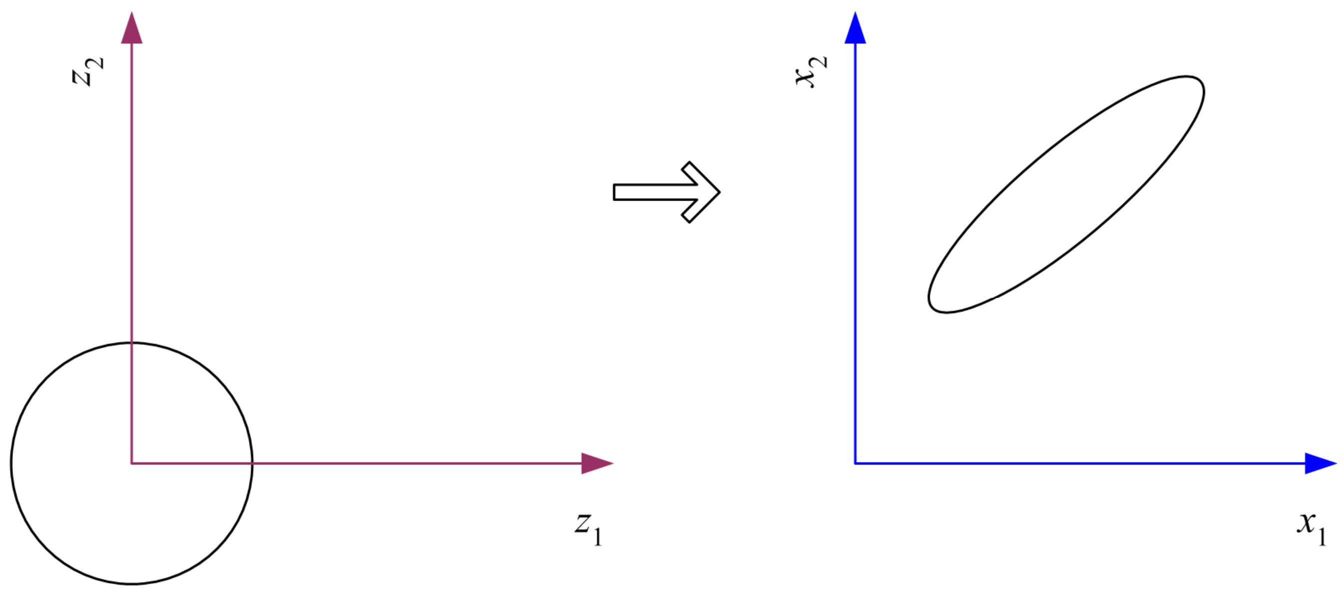
PCA vs FA

- PCA From x to z $\quad z = W^T(x - \mu)$
- FA From z to x $\quad x - \mu = Vz + \varepsilon$



Factor Analysis

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



Python Examples

- Inverted Index
  - inverted_index1.py
  - inverted_index2.py
- NLTK
  - nltk_example.py
  - spam_detector.py
- PCA & LDA
  - plot_pca_vs_lda.py