Pattern Classification

06. Feature Selection & Extraction

AbdElMoniem Bayoumi, PhD

Fall 2021

Recap: Feature Vector

- Let $\underline{X}(m) = \begin{bmatrix} X_1(m) \\ X_2(m) \\ \vdots \\ X_N(m) \end{bmatrix}$ be the feature vector of the m^{th} training pattern

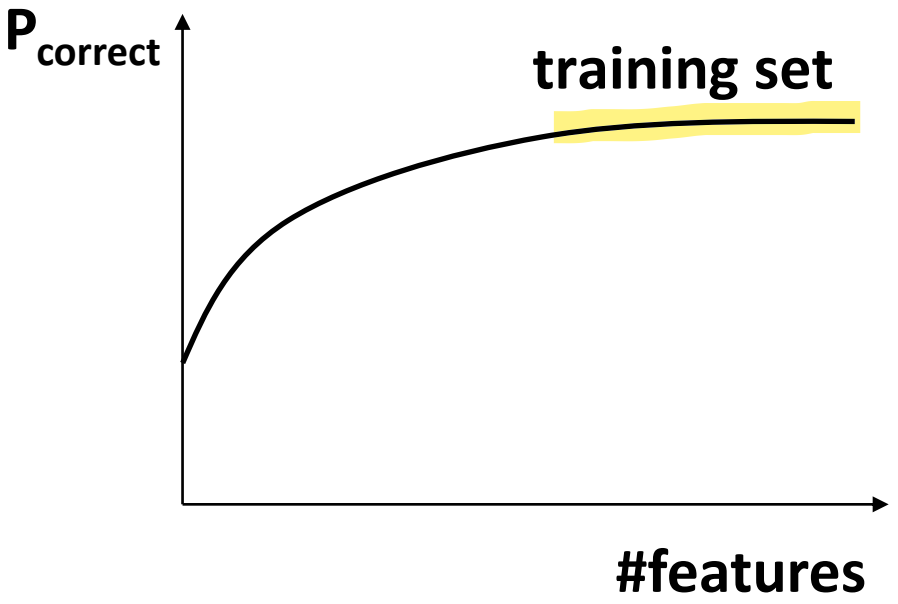
- N is the number of features, i.e., the dimension of $\underline{X}(m)$

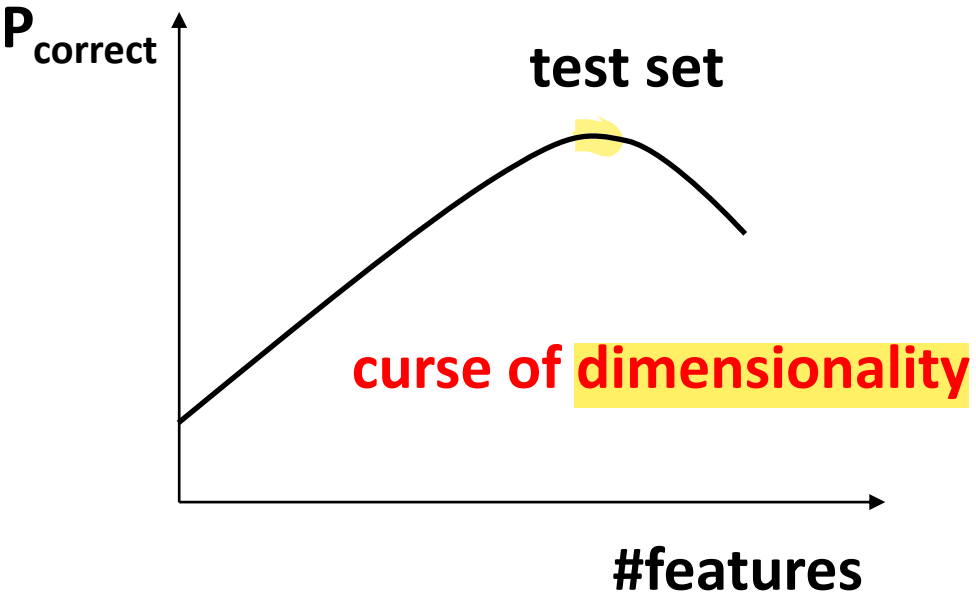
- M is the number of the training patterns

Feature Selection Problem

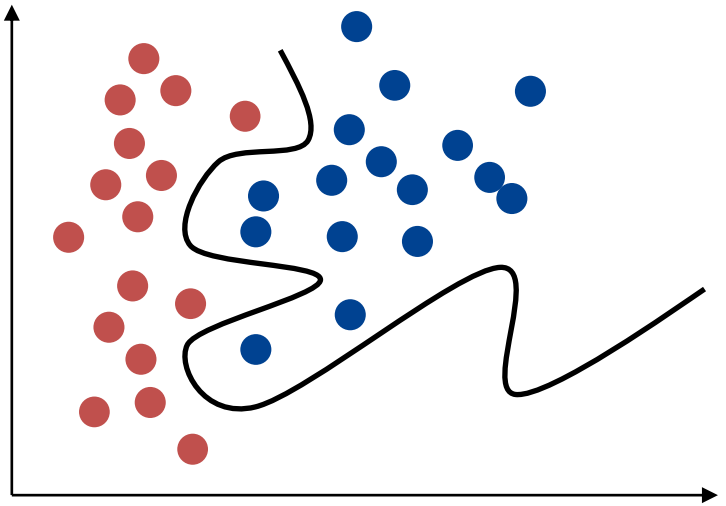
- Choosing good features affects the classification performance
- In many applications, we can think of a large number of possible features
- But if we use too many features we will suffer from curse of dimensionality

Feature Selection Problem

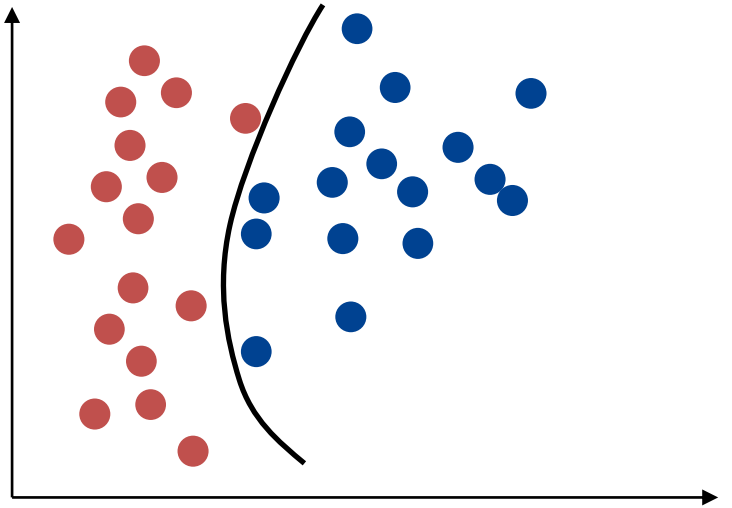




Example:



A bad classifier with bad generalization ability



A good classifier with good generalization ability

Feature Selection Problem

- The generalization ability is the ability of the classifier to generalize well for data it had not seen before, i.e., test data

Curse of Dimensionality

- The parameters' estimates, e.g., mean vector and covariance matrix, will not be very accurate

- The data points will become scattered and clustering of classes will not be clear

- It is therefore best to reduce the no. of features and only retain the few best ones

Feature Selection & Extraction

- **Feature Selection:** From the available N features choose a subset of L («N)

- **Feature Extraction:** Transform the available N features into a smaller no. of L features through certain transformations

$$u_1 = f_1(x_1, x_2 \cdots x_N) \qquad \underline{X} \equiv \text{original feature vector}$$

$$u_L = f_L(x_1, x_2 \cdots x_N) \qquad \underline{U} \equiv \text{transformed feature vector}$$

usually $f_1, f_2 \cdots f_L$ are linear fn's, e.g.,
$u_j = \sum_{i=1}^{N} V_{ij} X_i$, $V_{ij} \equiv$ constant

  - Transform original features to a more compact set of dimensions, while retaining as much information as possible.
  - Transformed features may have no physical meaning → explaining the model is problematic
  - May not be suitable to every domain

Types of Features to be Removed

- Irrelevant Features:
  - Features that do not help in discriminating between the classes

- Correlated Features:
  - Features that vary very closely with other features (i.e., redundant features)
  - Hence can be removed without much loss of information
  - E.g., #corners & #lines

Sequential Forward Selection Algorithm (SFS)

1. Examine each feature x_i, taken alone, design the classifier & select the best feature

2. From the remaining features, select the one that together with the selected group, gives best performance

3. Continue in this manner until you have selected L features

Example on Sequential Forward Selection (SFS)

1. Take x_1 → design the classifier only on x_1
   Get P(correct) ≡ P_1
   Take x_2 → design the classifier only on x_2 → P_2
   ⋮
   Take x_N → design the classifier only on x_N → P_N

   Find highest Pi. Assume P5 was highest → choose X5 and fix it among the chosen features in next iterations

2. Test X_1, X_5 → P_1
   Test X_2, X_5 → P_2
   ⋮
   Test X_N, X_5 → P_N

   Find highest P_i. Assume P_2 was highest

3. Test X_1, X_2, X_5 → P_1
   ⋮

Sequential Backward Selection Algorithm SBS

Top down

- Similar to SFS, but done backward:

bngeb el classification lw khadna kol el features, w b3den nfdl nshel wahed wara wahed w nshof ahsan combination hatb2a anhy.

1. Start with all features selected. Remove one feature at a time, design the classifier and compute P_c
2. The feature we end up removing is the one that results in the least reduction of Pc
3. Keep removing till reaching the no. L selected features

usually SBS and SFS remove the same features, but it is not a must.

Example on Sequential Backward Selection

- $X_1, X_2, X_3, X_4 \rightarrow P_c = 0.87$
  - $X_2, X_3, X_4 \rightarrow P_c = 0.86$ ← BEST
  - $X_1, X_3, X_4 \rightarrow P_c = 0.82$
  - $X_1, X_2, X_4 \rightarrow P_c = 0.81$
  - $X_1, X_2, X_3 \rightarrow P_c = 0.84$

  Remove X_1

- $X_2, X_3, X_4 \rightarrow P_c = 0.86$
  - $X_2, X_3 \rightarrow P_c = 0.79$
  - $X_2, X_4 \rightarrow P_c = 0.80$
  - $X_3, X_4 \rightarrow P_c = 0.84$

  Remove X_2

Assuming L = 2 then stop
& Select X_3, X_4

Methods of Feature Selection

- Methods of feature selection can be categorized into:

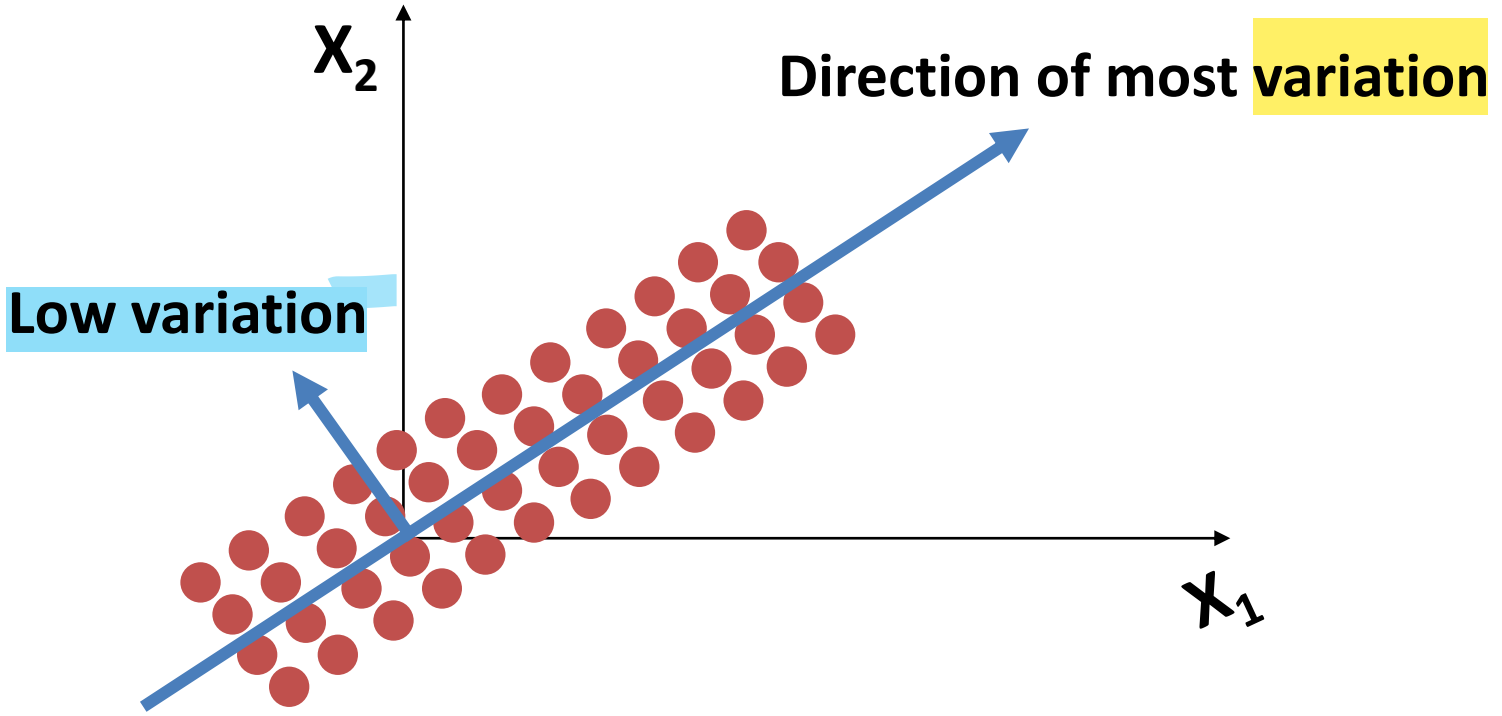
  - **Filter type:** select features without looking into the classifier you are going to use, e.g., based on correlations of the features, distances between class means, … etc
    - Adv: fast execution, generality
    - Disadv: tendency to select large subsets

  - **Wrapper type:** select features by taking into consideration the classifier you will use, e.g. SFS, SBS
    - Adv: accuracy
    - Disadv: slow execution, lack of generality

Feature Extraction

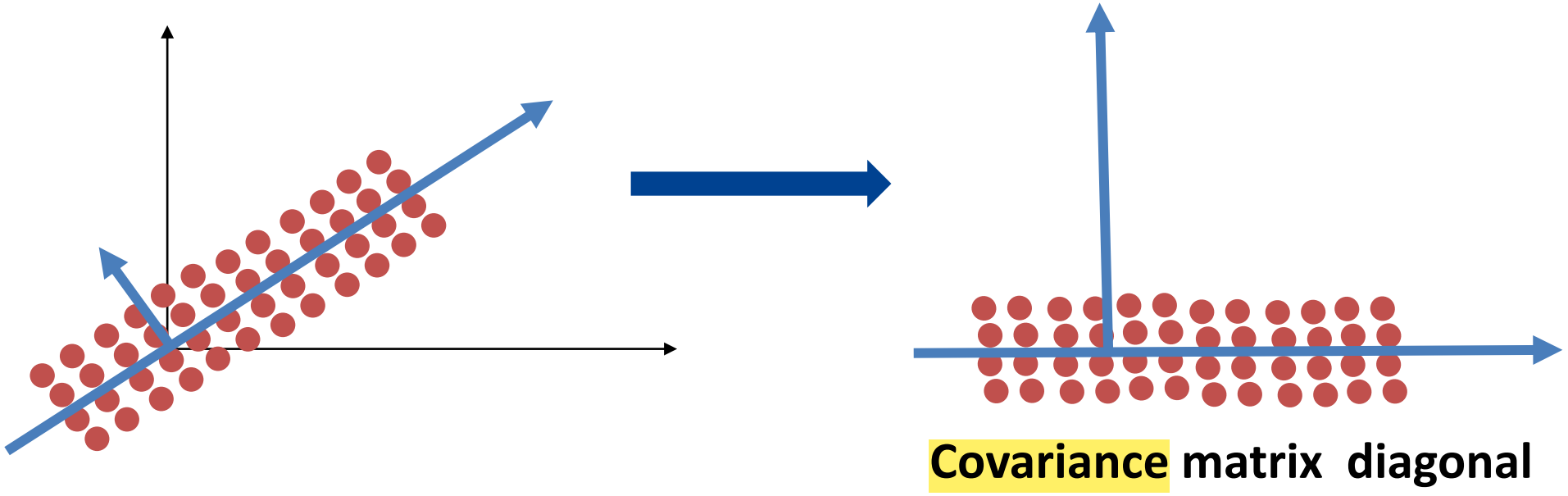
- Principal Component Analysis (PCA)

- In PCA, the most interesting directions are those having large variations (large variance)



Principal Component Analysis (PCA)

- Consider the most interesting directions, i.e., transformed features show an impact on data variations



PCA Summary Steps

- Transform data to zero mean, i.e., $\underline{Y} = \underline{X} - \underline{\mu}$
- Estimate covariance matrix from data, i.e.,

$$\hat{\Sigma} = \frac{1}{M} \sum_{m=1}^{M} \underline{Y}(m)\underline{Y}(m)^T$$

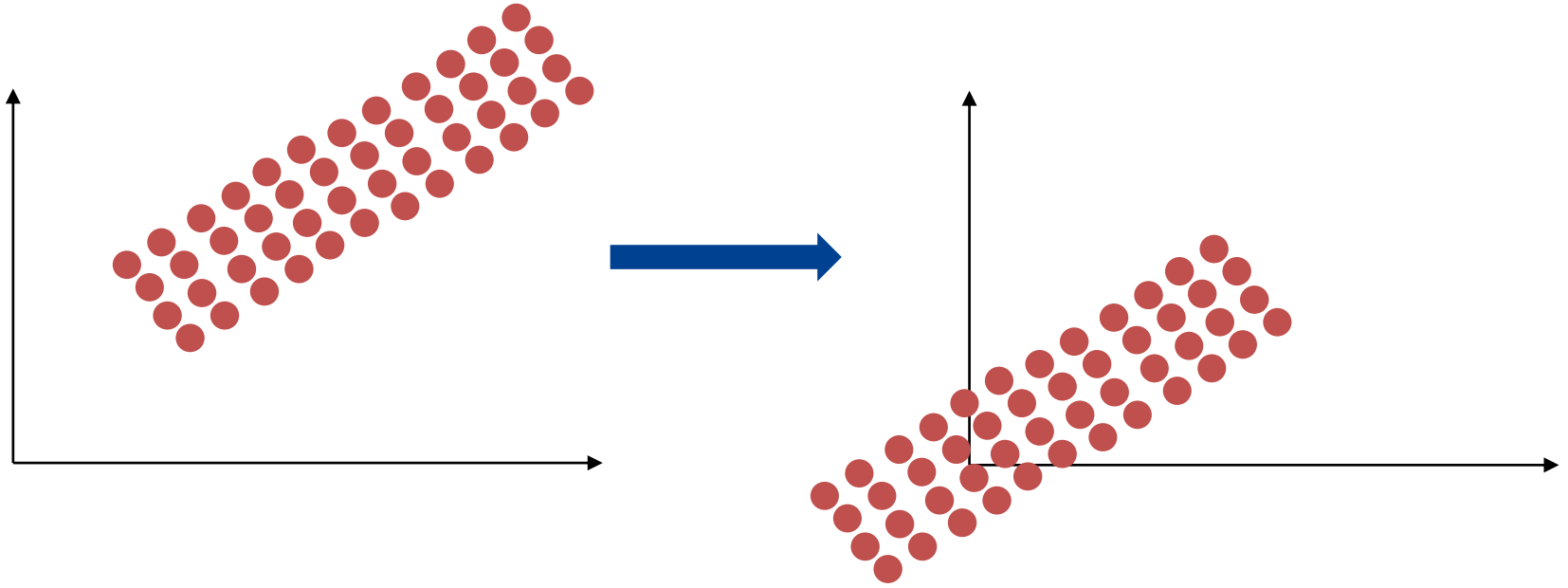
- Compute eigen values and vectors of $\hat{\Sigma}$
- Choose the eigen vectors corresponding to big eigen values
- Perform transformation of data to a new space

$$\underline{Z} = \begin{bmatrix} \underline{u}_1^T \\ \underline{u}_2^T \\ \vdots \\ \underline{u}_L^T \end{bmatrix} \underline{Y}$$

Principal Component Analysis (PCA)

- First, transform the problem to zero mean

$$\underline{Y} = \underline{X} - \underline{\hat{\mu}}$$



Principal Component Analysis (PCA)

- Consider the covariance matrix:

$$\Sigma = E\left[\underline{Y}\underline{Y}^T\right]$$

or its estimate from the data:

$$\hat{\Sigma} = \frac{1}{M-1}\sum_{m=1}^{M}\underline{Y}(m)\underline{Y}(m)^T$$

where

$$\underline{Y}(m) = \underline{X}(m) - \underline{\hat{\mu}}$$

Principal Component Analysis (PCA)

- Linear Transformation (i.e., Rotations):

$$\underline{Z} = A\underline{Y}$$

- So covariance matrix of $\underline{Z}$:

hwa da el covariance el geded el ana 3auzo

$$\begin{aligned} cov(\underline{Z}) &= E\left[(A\underline{Y})(A\underline{Y})^T\right] \\ &= E\left[A\underline{Y}\underline{Y}^T A^T\right] \\ &= AE\left[\underline{Y}\underline{Y}^T\right]A^T \\ &= A\Sigma A^T \end{aligned}$$

$(\boldsymbol{AB})^{\boldsymbol{T}} = \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{A}^{\boldsymbol{T}}$

A is constant matrix

Eigenvectors & Eigenvalues

- An eigenvector $\underline{\boldsymbol{u}}$ of a linear transformation matrix is a non-zero vector

- The eigenvector changes by only a scalar factor (called eigenvalue) when that linear transformation is applied to it:

$$\boldsymbol{B}\underline{\boldsymbol{u}} = \boldsymbol{\lambda}\underline{\boldsymbol{u}}$$

Eigenvectors & Eigenvalues

$$B\underline{u} = \lambda\underline{u}$$

- There are generally N eigenvectors for a **B** (NxN matrix) if **B** is diagonalizable
- Symmetric matrix is always diagonalizable

$$B\left[\underline{u}_1|\underline{u}_2|\cdots|\underline{u}_N\right] = \left[\lambda_1\underline{u}_1|\lambda_2\underline{u}_2|\cdots|\lambda_N\underline{u}_N\right]$$

Normalized eigen vectors

$$= \left[\underline{u}_1|\underline{u}_2|\cdots|\underline{u}_N\right]\begin{bmatrix}\lambda_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \lambda_N\end{bmatrix}$$

$$\mathbf{BU} = U\Omega$$

Eigenvectors & Eigenvalues

- Eigenvectors of symmetric matrix is guaranteed to be orthogonal (i.e., perpendicular to each other)

- An orthogonal matrix is a square matrix whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors)

$$\boldsymbol{Q^T Q = I}$$

$$\boldsymbol{Q^T = Q^{-1}}$$

Principal Component Analysis (PCA)

to get omega only, we need to multiply by U^-1
but using the property that if the eigenvectors where orthonormal we can say that
U^-1 = U^T
then after we substitute we can get this equation

$$\mathbf{BU} = \boldsymbol{U\Omega}$$

$$\Sigma\, U = U\Omega$$

$$U^T \Sigma\, U = \Omega$$

omega is the selection criteria

Take $B = \Sigma$

- Suppose we take $\boldsymbol{A} = \boldsymbol{U^T}$ to be the transformation, so:

ehna msh 3arfen A, fa hanftrd enha U^t
w han7sb aknna 3arfenha, fa hytl3lna el omega fa keda el denya zabatet.

$$\boldsymbol{Z} = \boldsymbol{U^T Y}$$

Recall: $cov(\underline{Z}) = A\Sigma A^T$

$$cov(\underline{Z}) = A\Sigma A^T$$
$$= U^T \Sigma\, U$$
$$= \Omega$$

covariance matrix is diagonal !
So we find which transformed features that cause difference

- We took the transformation matrix A as $\begin{bmatrix} \underline{u_1}^T \\ \underline{u_2}^T \\ \vdots \\ \underline{u_N}^T \end{bmatrix}$ where $\underline{u_i}$ is

  the eigenvector of Σ corresponding to eigenvalue λ_i

Principal Component Analysis (PCA)

- Assume we reordered variables such that $\lambda_1 \gg \lambda_2 \gg \cdots \gg \lambda_N$
- We take only $L \ll N$ directions.
- These are the directions corresponding to the largest L eigenvalues of Σ
- Since the eigenvalue corresponds to the variance of the transformed variable Z_i , this means we took L directions with largest variance & thrown out the rest of directions

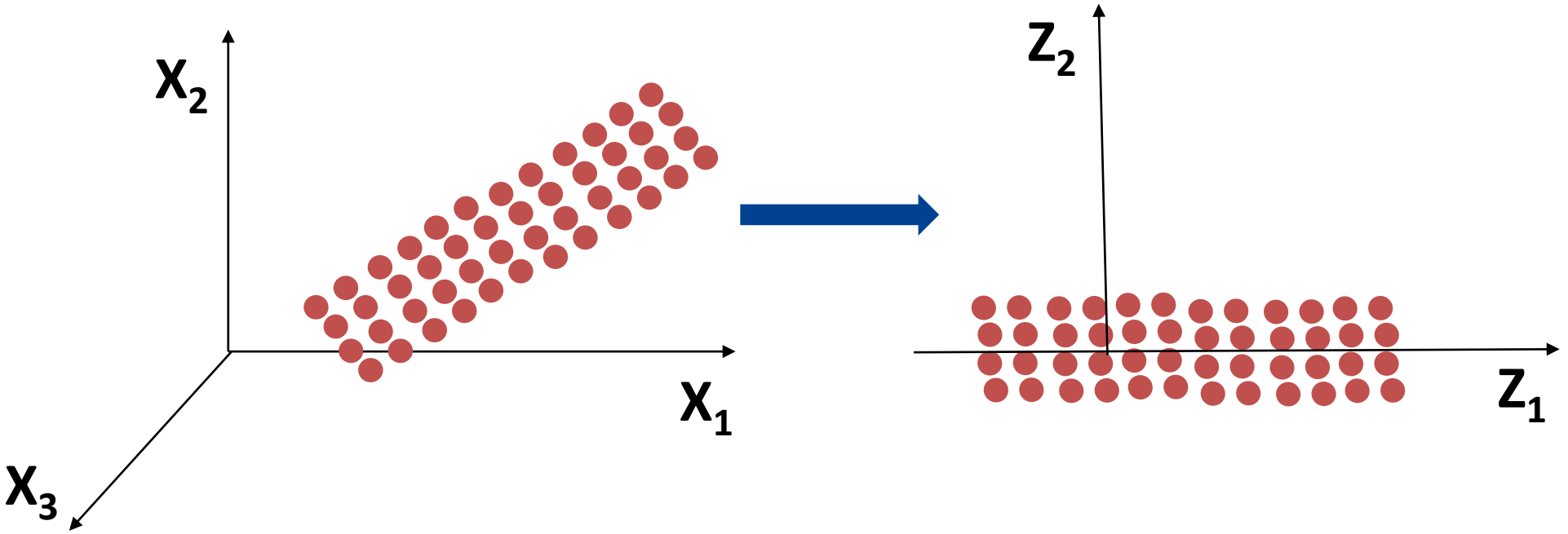
Principal Component Analysis (PCA)

- The transformation can be re-written as follows:

$$\underline{Z} = \begin{bmatrix} \underline{u}_1^T \\ \underline{u}_2^T \\ \vdots \\ \underline{u}_L^T \end{bmatrix} \underline{Y}$$

$\underline{Z}$ is the extracted (transformed) feature vector

Principal Component Analysis (PCA)



Principal Component Analysis (PCA)

- Example:
  - 20 features problem
  - Eigen values: 12
    5
    2
    0.3
    0.01
    0.002
    0.0005
    ⋮

Take 4 PC's, it explains:

Note: scale affects PCA!

$$\%var = \frac{12 + 5 + 2 + 0.3}{12 + 5 + 2 + 0.3 + 0.01 + 0.002 + \cdots}$$

$$\simeq 99.9\%$$

PCA Summary Steps

- Transform data to zero mean, i.e., $\underline{Y} = \underline{X} - \underline{\hat{\mu}}$
- Estimate covariance matrix from data, i.e.,

$$\hat{\Sigma} = \frac{1}{M-1} \sum_{m=1}^{M} \underline{Y}(m)\underline{Y}(m)^T$$

- Compute eigen values and vectors of $\hat{\Sigma}$
- Choose the eigen vectors corresponding to big eigen values
- Perform transformation of data to a new space

$$\underline{Z} = \begin{bmatrix} \underline{u}_1^T \\ \underline{u}_2^T \\ \vdots \\ \underline{u}_L^T \end{bmatrix} \underline{Y}$$

very interesting lecture :))))

Acknowledgment

- These slides have been created relying on lecture notes of Prof. Dr. Amir Atiya