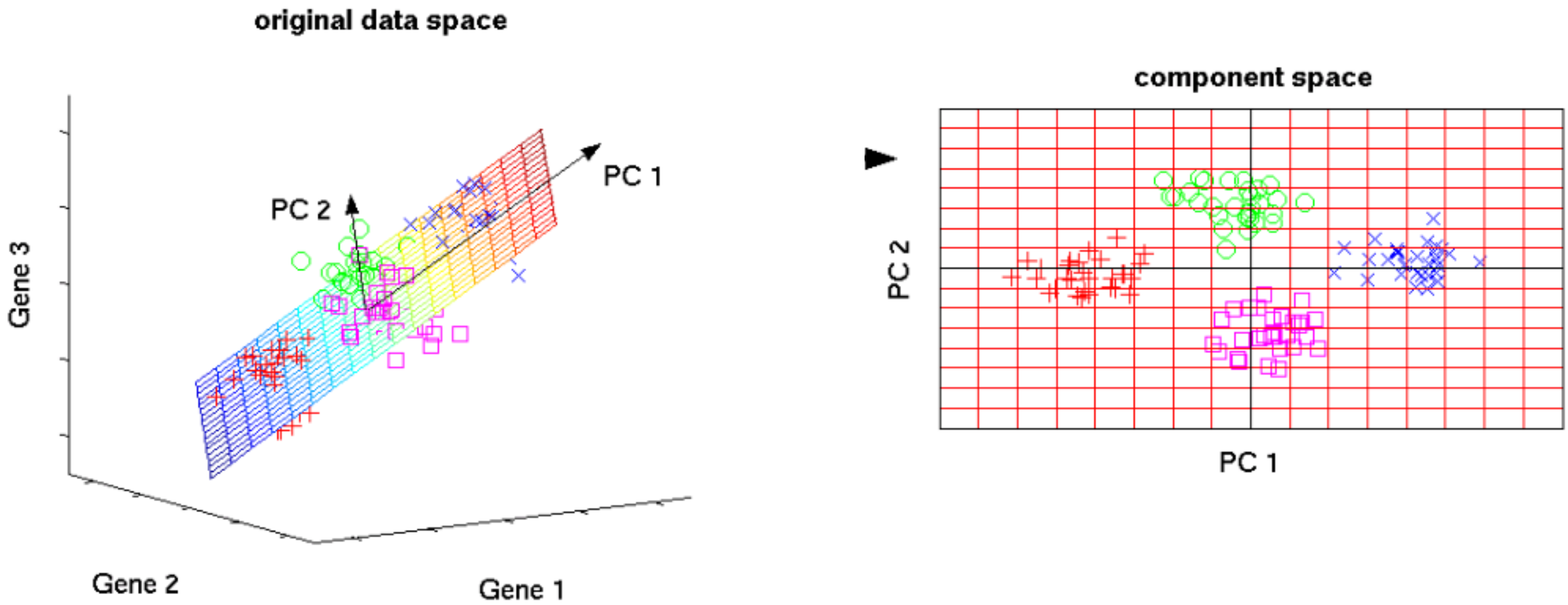
機器學習

Lecture 11 Dimension reduction

Dimension Reduction



Source: https://hackernoon.com/a-laymans-introduction-to-principal-components-2fca55c19fa0

Dimension Reduction

- Principal Component Analysis (PCA，主成分分析): 對「非監督式數據」壓縮
- Linear Discriminant Analysis (LDA，線性判別分析) ：對「監督式數據」降維來最大化類別分離性

Subspace

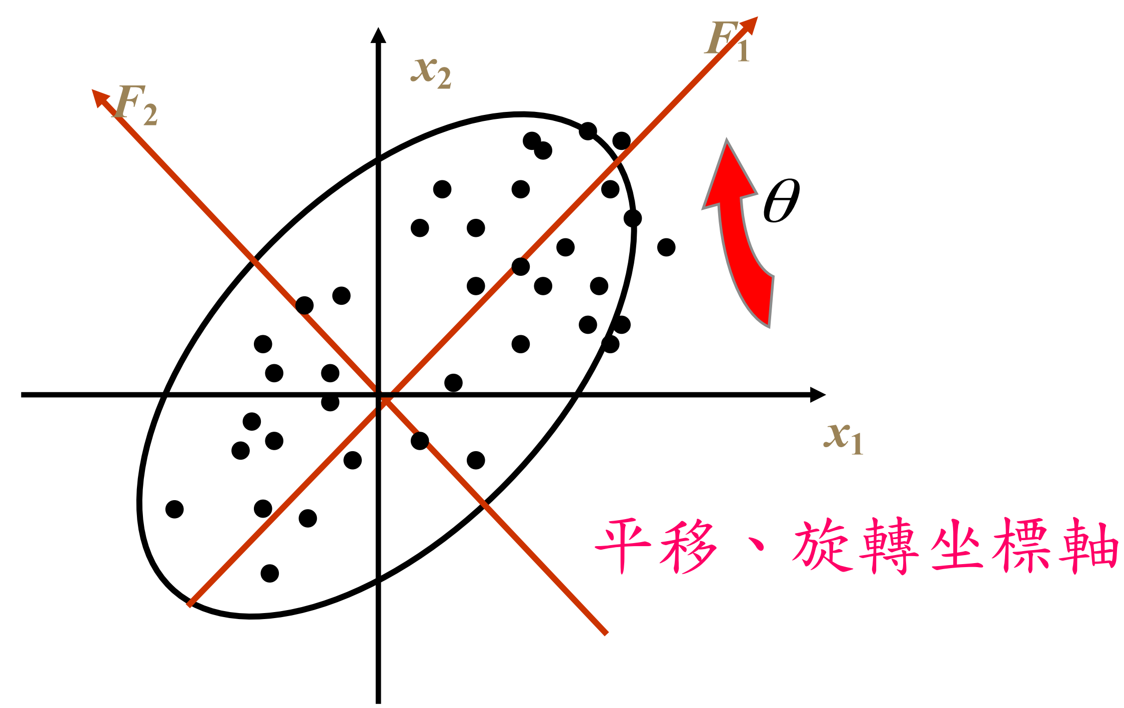
$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{(standard basis)}$$

$$x_v = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 3v_1 + 3v_2 + 3v_3$$

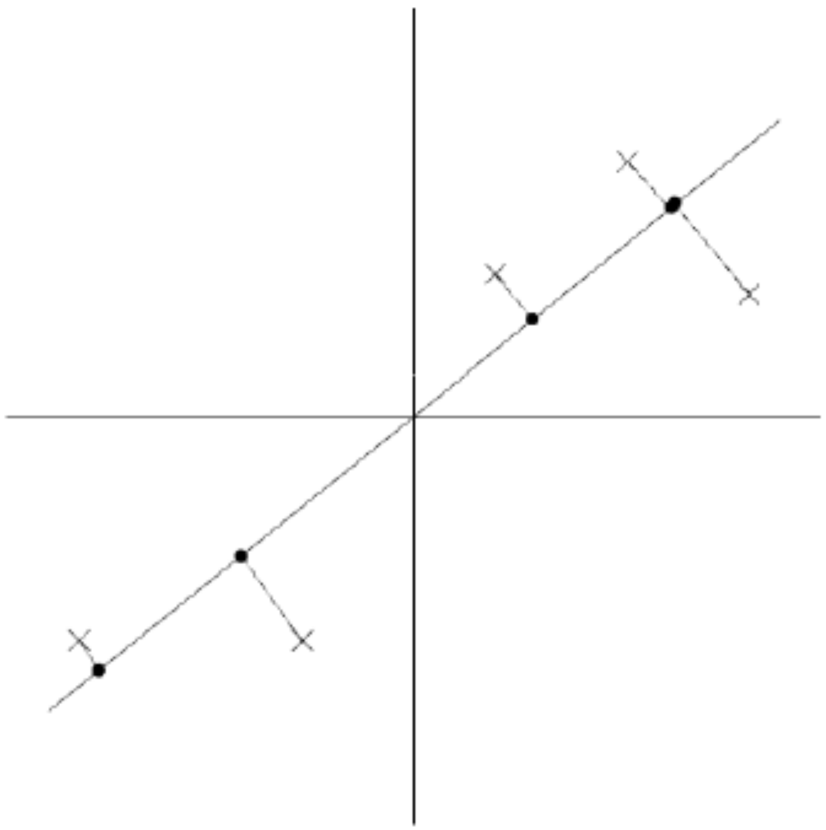
$$u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, u_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{(some other basis)}$$

$$x_u = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 0u_1 + 0u_2 + 3u_3$$

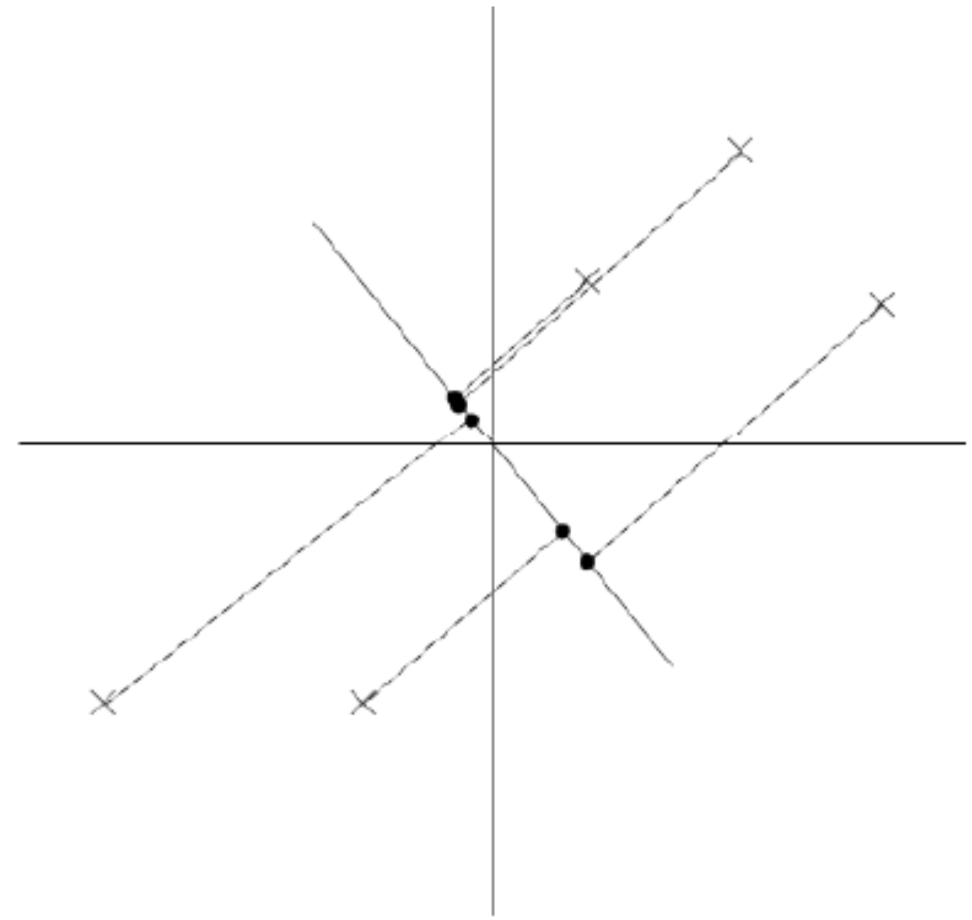
$$\text{thus, } x_v = x_u$$



What is good projection?



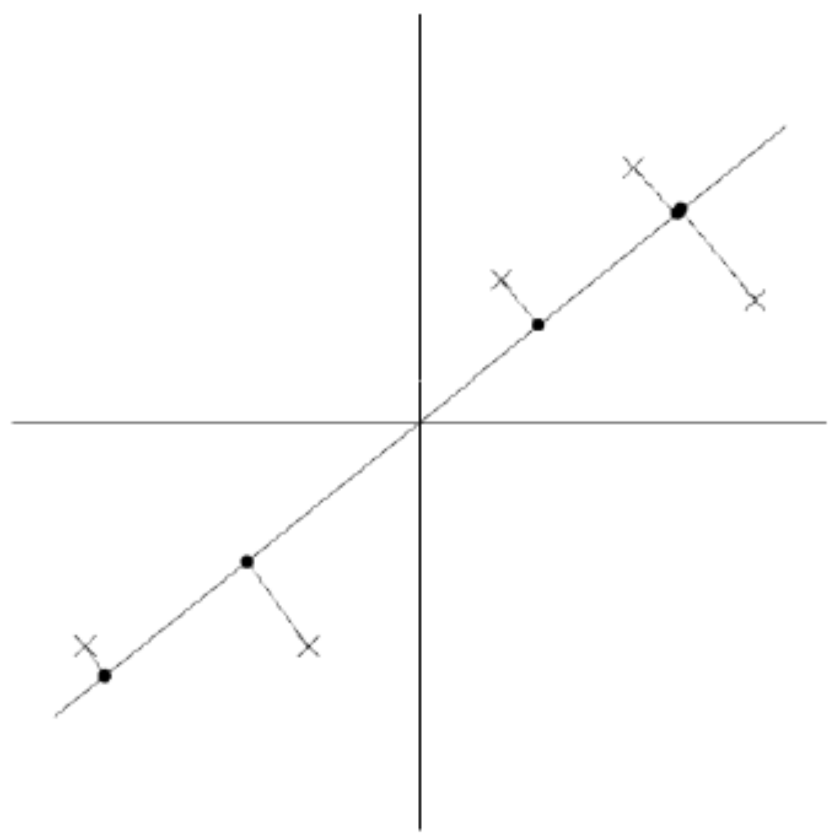
the projected data has a fairly large variance, and the points tend to be far from zero.



the projections have a significantly smaller variance, and are much closer to the origin.

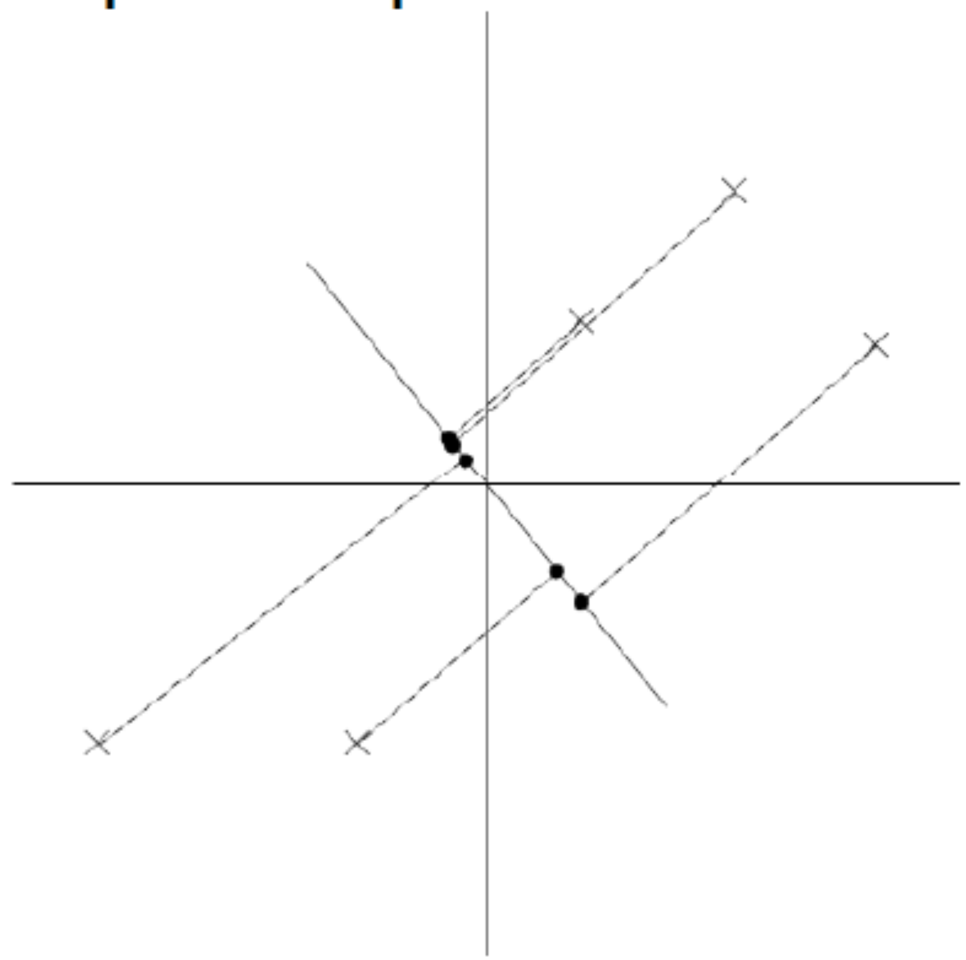
What is good projection?

good representation



the projected data has a fairly large variance, and the points tend to be far from zero.

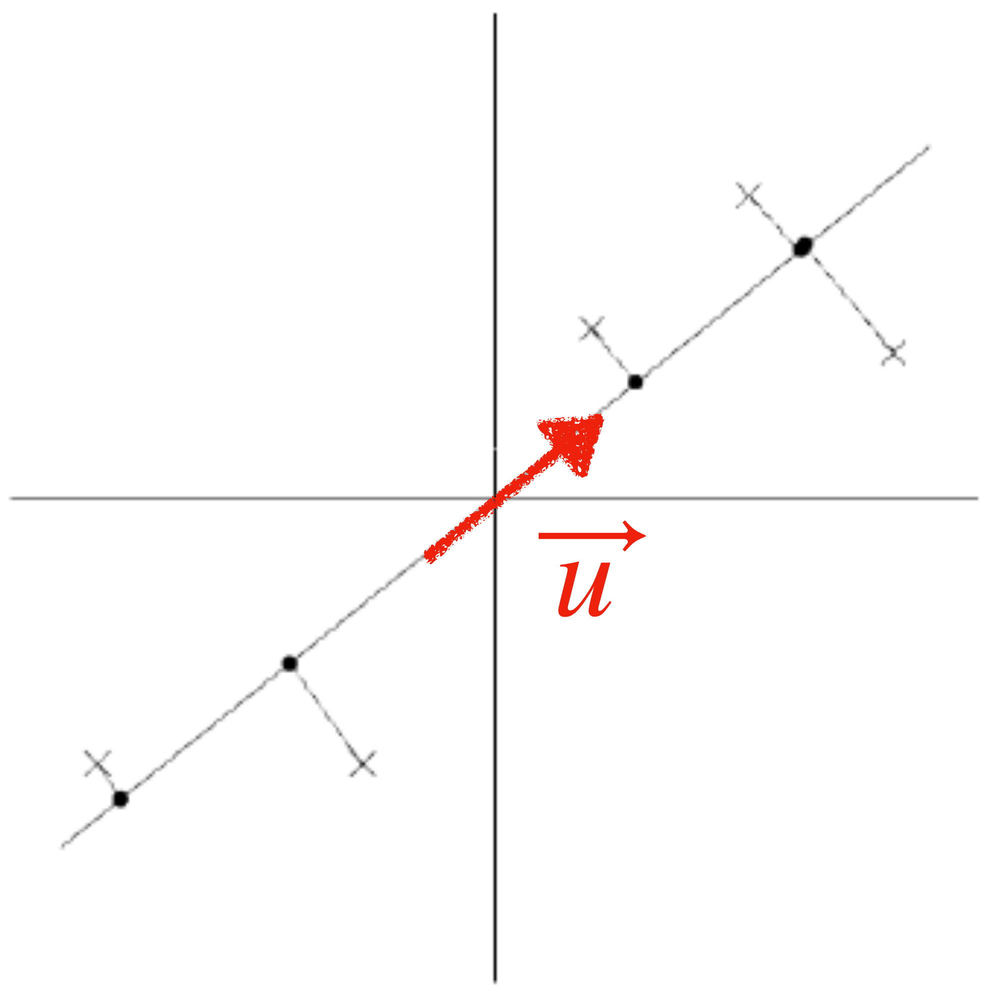
poor representation



the projections have a significantly smaller variance, and are much closer to the origin.

目標

- 目標
  - 找到一個 unit vector $\overrightarrow{u}$ with $\|\overrightarrow{u}\| = 1$，使得投影的變異數越大越好



資料預處理

- 給定 m 筆 n-維的資料 $\left\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\right\}, x^{(i)} \in \mathbb{R}^n, \forall i$
- 先做資料預處理

1. Let $\mu = \dfrac{1}{m}\sum_{i=1}^{m} x^{(i)}$
2. For each i, replace $x^{(i)}$ with $x^{(i)} - \mu$
3. Let $\sigma_j^2 = \dfrac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right)^2$
4. For each i, replace $x_j^{(i)}$ with $\dfrac{x_j^{(i)}}{\sigma_j}$

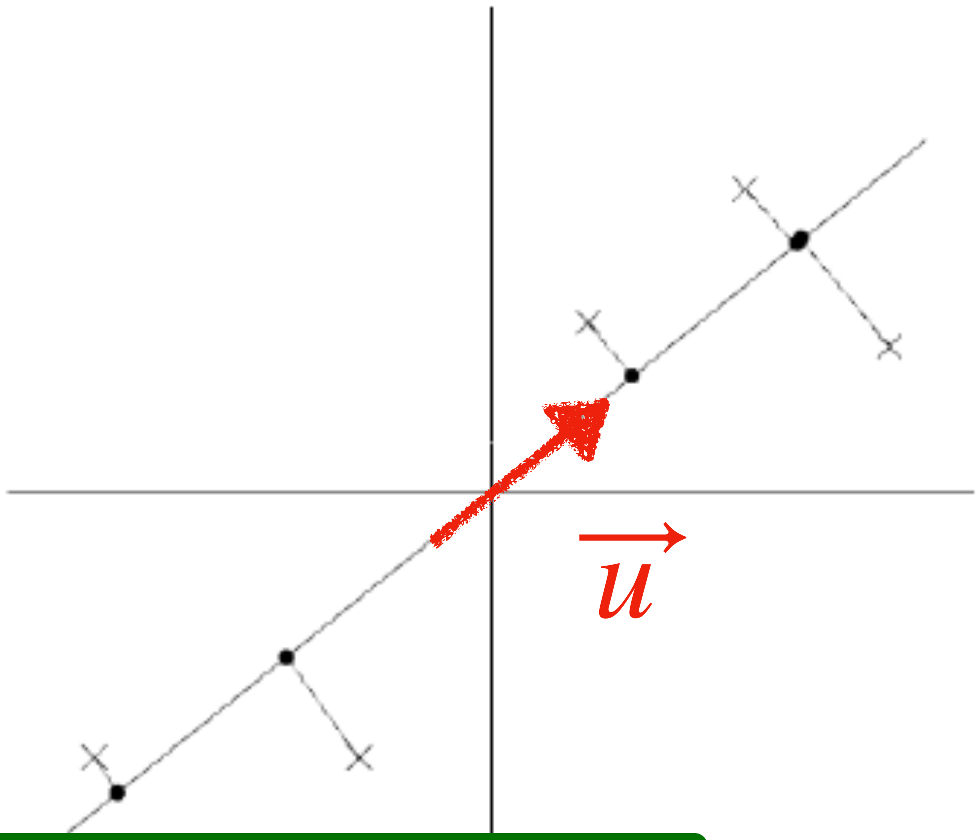
目標

- 目標
  - 找到一個 unit vector $\vec{u}$ with $\|\vec{u}\| = 1$，使得投影的變異數越大越好

$$\frac{1}{m}\sum_{i=1}^{m}\left(u^T x^{(i)}\right)^2$$

$$= \frac{1}{m}\sum_{i=1}^{m} u^T x^{(i)}(x^{(i)})^T u$$

$$= u^T \boxed{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}(x^{(i)})^T\right)} u$$



A:「共變異數矩陣」(covariance matrix)

目標

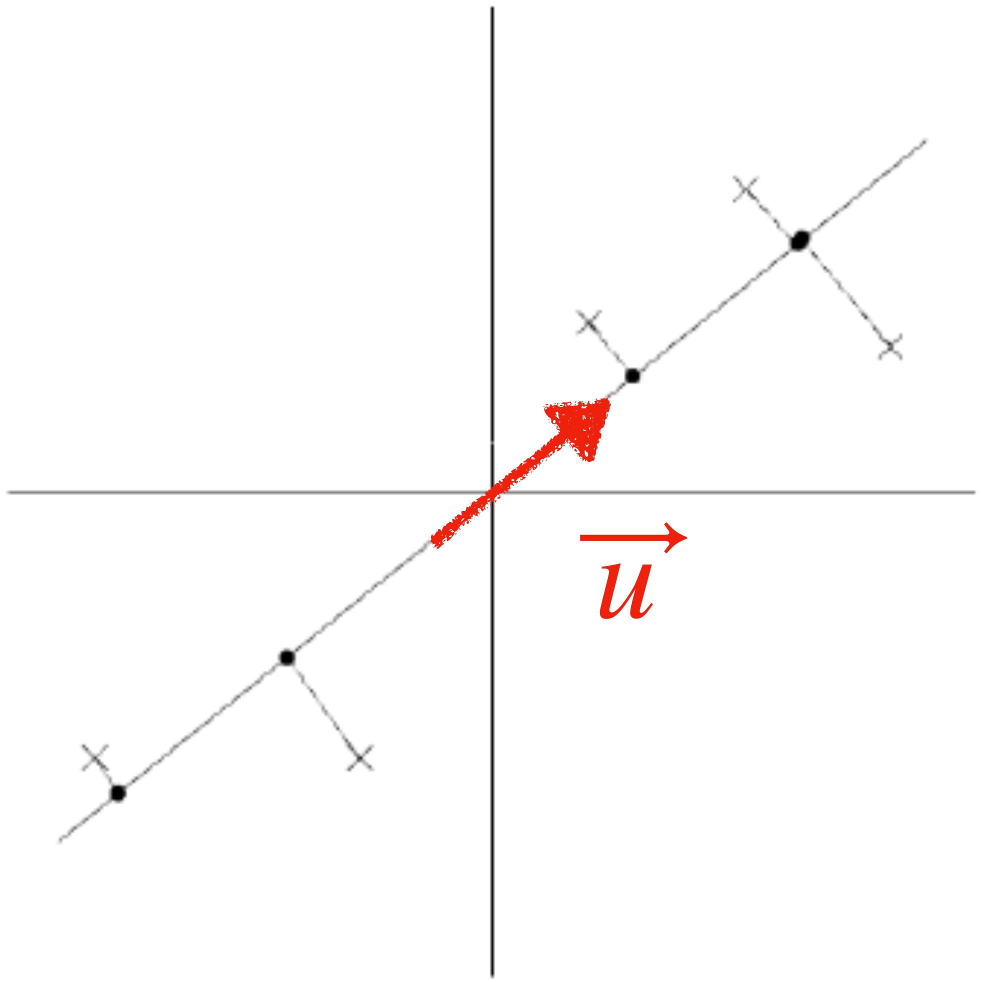
- 解下列的最佳化問題

$$\textbf{max}\ u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}(x^{(i)})^T\right)u$$

$$\textbf{subject to}\ \|\overrightarrow{u}\| = 1$$

By Lagrange multipliers:

$\overrightarrow{u}$ is an eigenvector of the covariance matrix



Eigenvalues and eigenvectors

設$\mathbf{A}$是n階矩陣，如果數λ和n維非零列向量$\boldsymbol{x}$使關係式

$$\mathbf{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

成立，則稱**數λ為方陣$\mathbf{A}$的特徵值**，非零**向量$\boldsymbol{x}$稱為$\mathbf{A}$的對應於特徵值λ的特徵向量。**

特徵值

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix} = 7 \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

特徵向量

Eigenvalues and eigenvectors

$$A = \begin{pmatrix} .52 & .36 \\ .36 & .73 \end{pmatrix}$$

$$Det(A-\lambda I) = \begin{vmatrix} .52-\lambda & .36 \\ .36 & .73-\lambda \end{vmatrix}$$

$$= (.52-\lambda)(.73-\lambda) - .36^2$$

$$= \lambda^2 - (.52+.73)\lambda + (.52 \cdot .73 - .36^2)$$

對應於 λ1=1的特徵向量為

$$\begin{pmatrix} 0.52 & 0.36 \\ 0.36 & 0.73 \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = 1 \cdot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

特徵向量為

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

對應於 λ2=1.25的特徵向量為

$$\begin{pmatrix} 0.52 & 0.36 \\ 0.36 & 0.73 \end{pmatrix}\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = 1.25 \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

特徵向量為

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix}$$

PCA

1. 標準化數據集
2. 計算**「共變異數矩陣」(covariance matrix)** A

$$\sigma^2_{x,y} = \mathbb{E}[(x-\mu_x)(y-\mu_y)] = \frac{1}{n}\sum_{i=1}^{n}(x_i-\nu_x)(y_i-\mu_y)$$

3. 計算A的特徵值和特徵向量
4. 選取 k 個最大的特徵值
5. 用此 **k 個特徵值對應的特徵向量**建立「投影矩陣」(project matrix) W
6. 利用 W 轉換數據集

Principal Components (主成份)

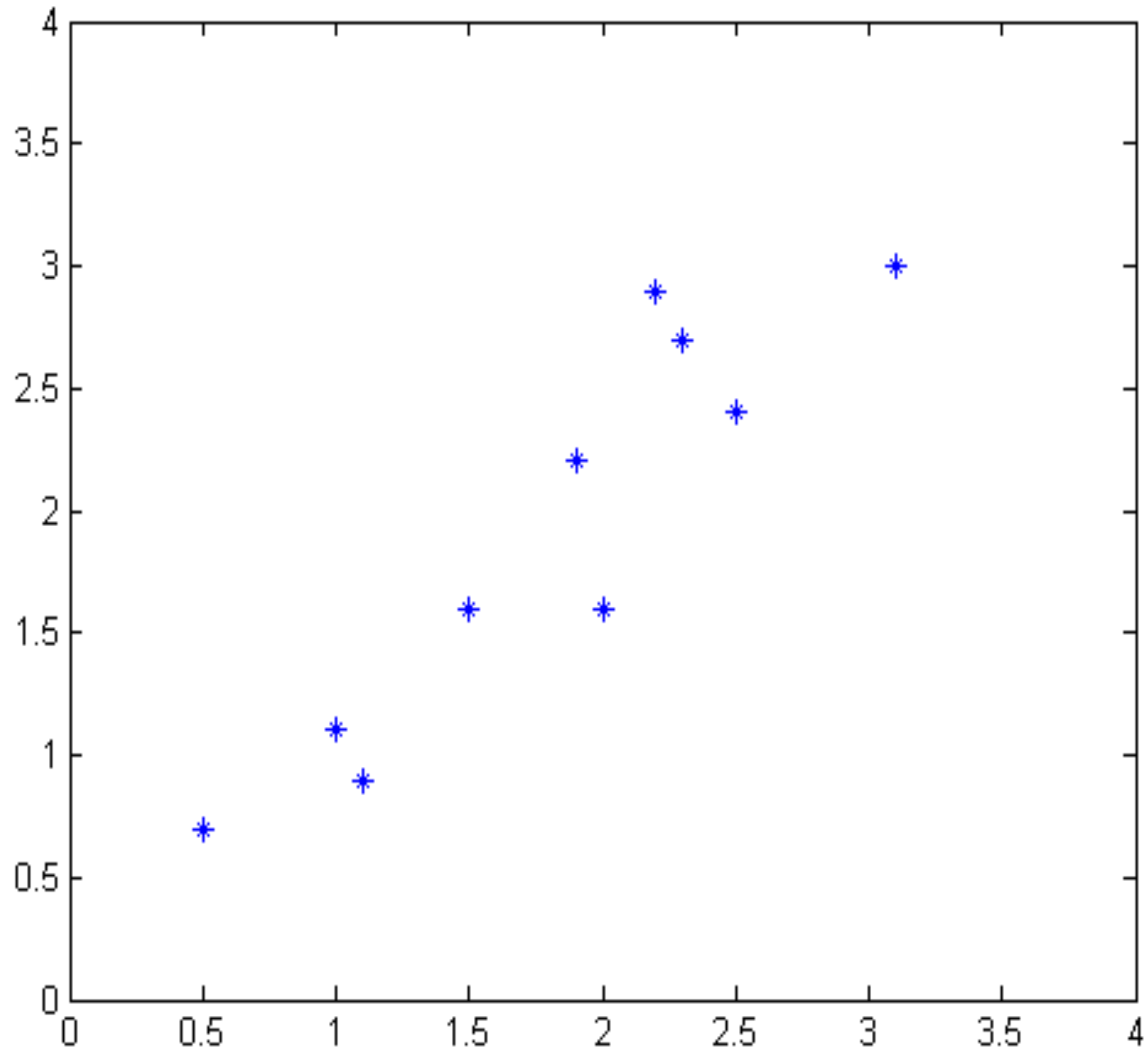
$$x = [x_1, x_2, \cdots, x_n] \rightarrow z = xW = [z_1, \cdots, z_k]$$

$$z_i = x^T v_i$$

Example

Original data

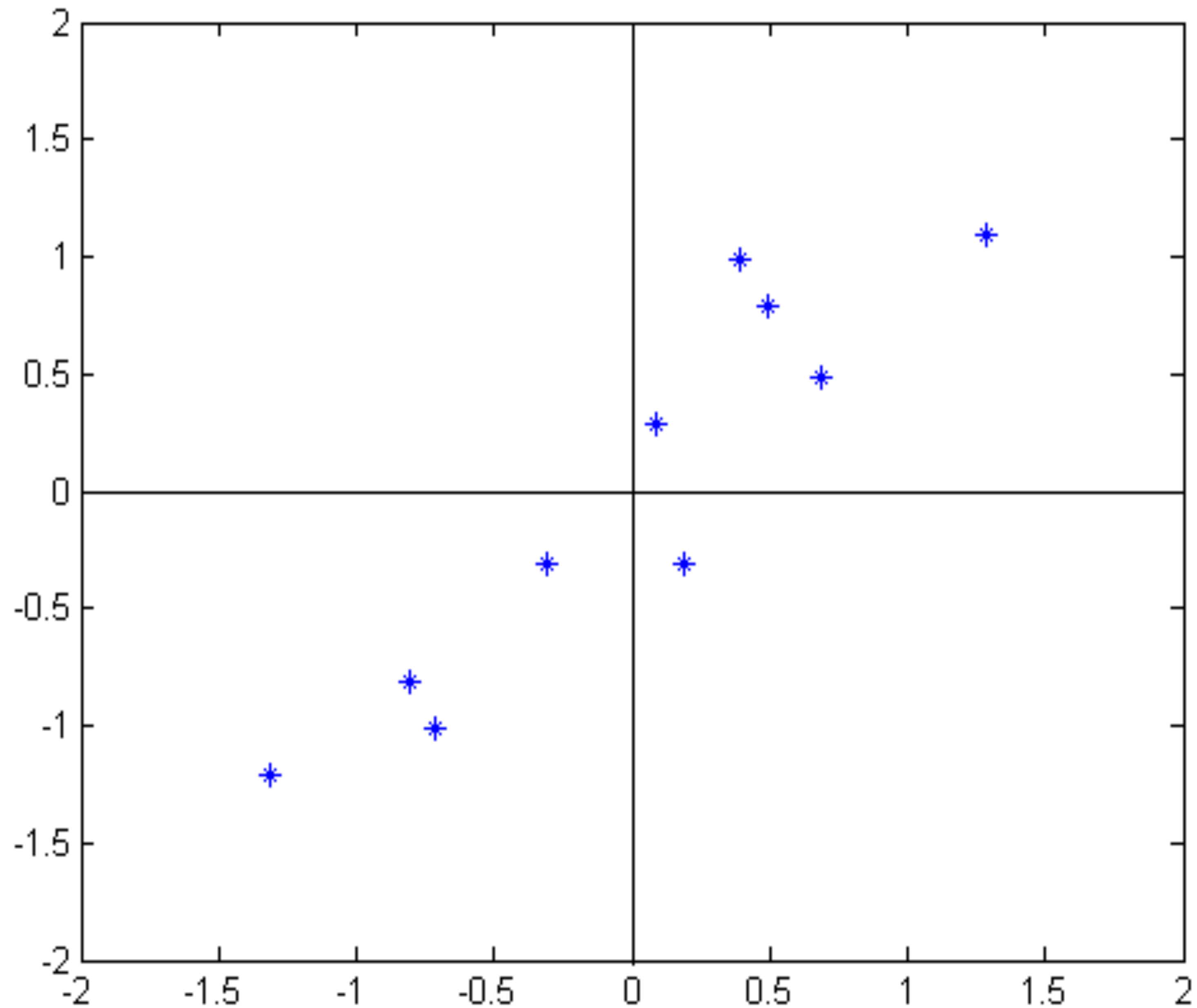
X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.1	0.9



Example

(1)Get some data and subtract the mean

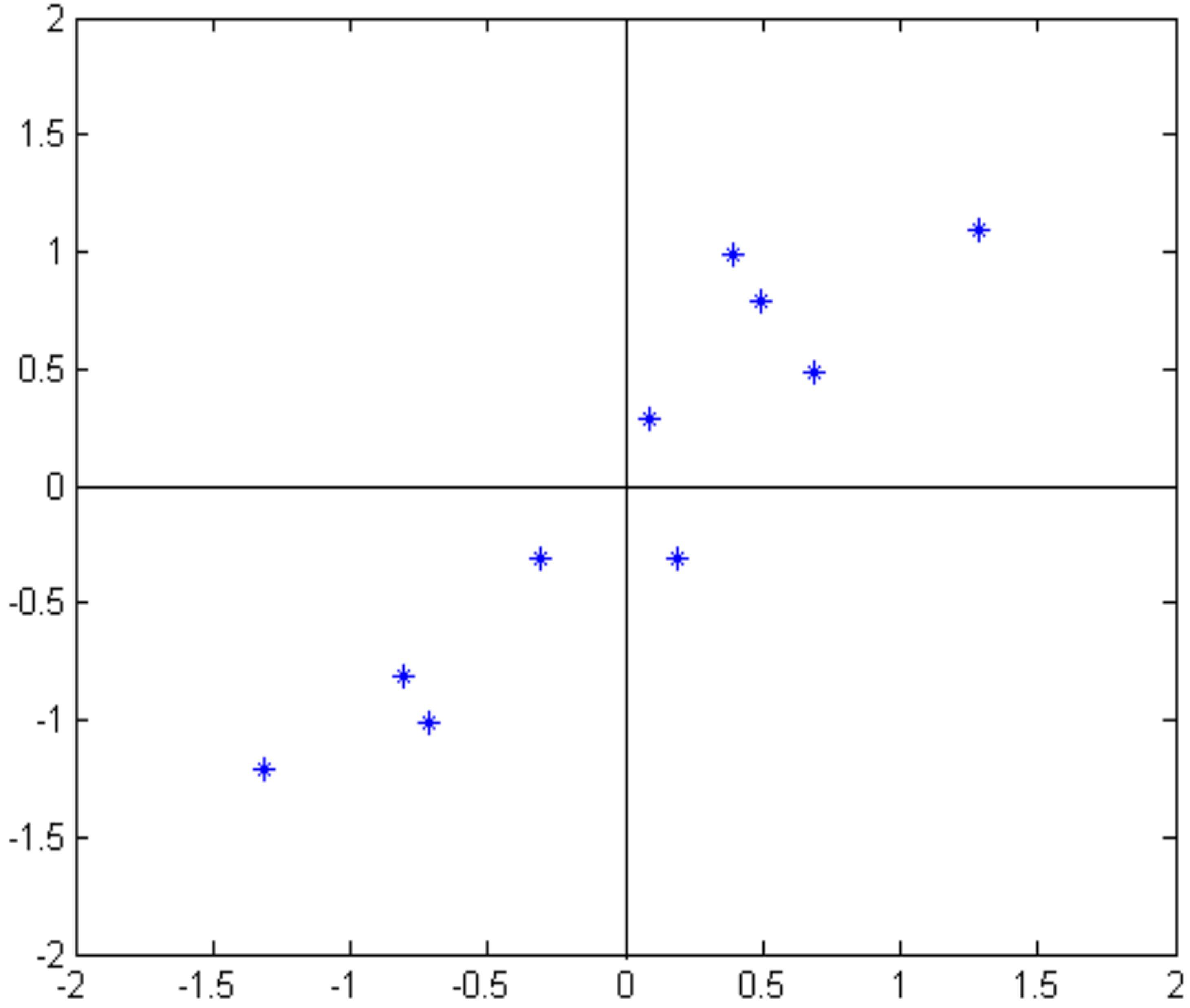
X	Y
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01



Example

(2)Get the **covariance matrix**

$$A = \begin{bmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{bmatrix}$$



Example

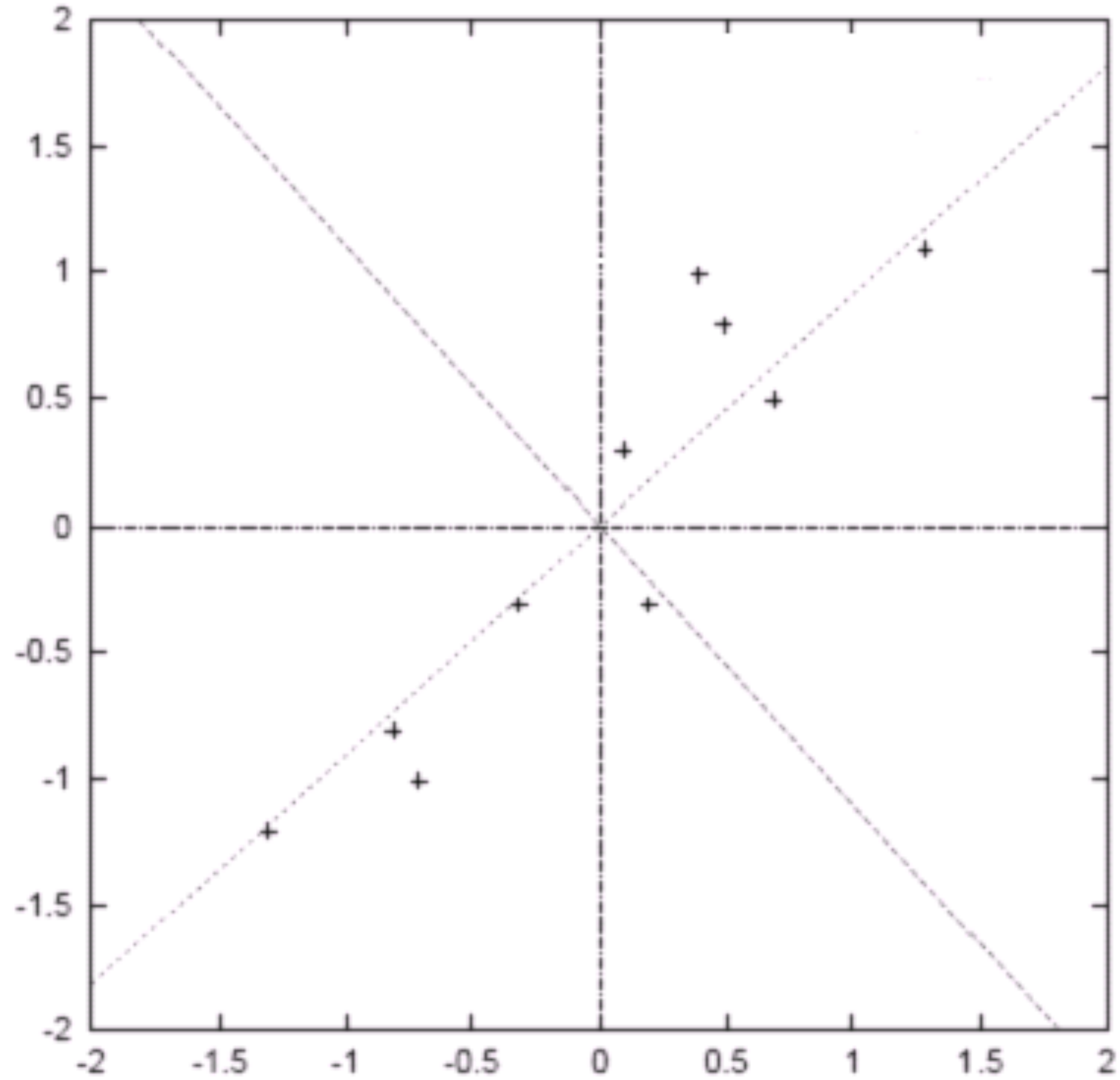
(3)Get their **eigenvectors & eigenvalues**

$$\lambda_1 = 1.2840$$

$$v_1 = \begin{bmatrix} 0.6779 \\ 0.7352 \end{bmatrix}$$

$$\lambda_2 = 0.0491$$

$$v_2 = \begin{bmatrix} -0.7352 \\ 0.6779 \end{bmatrix}$$



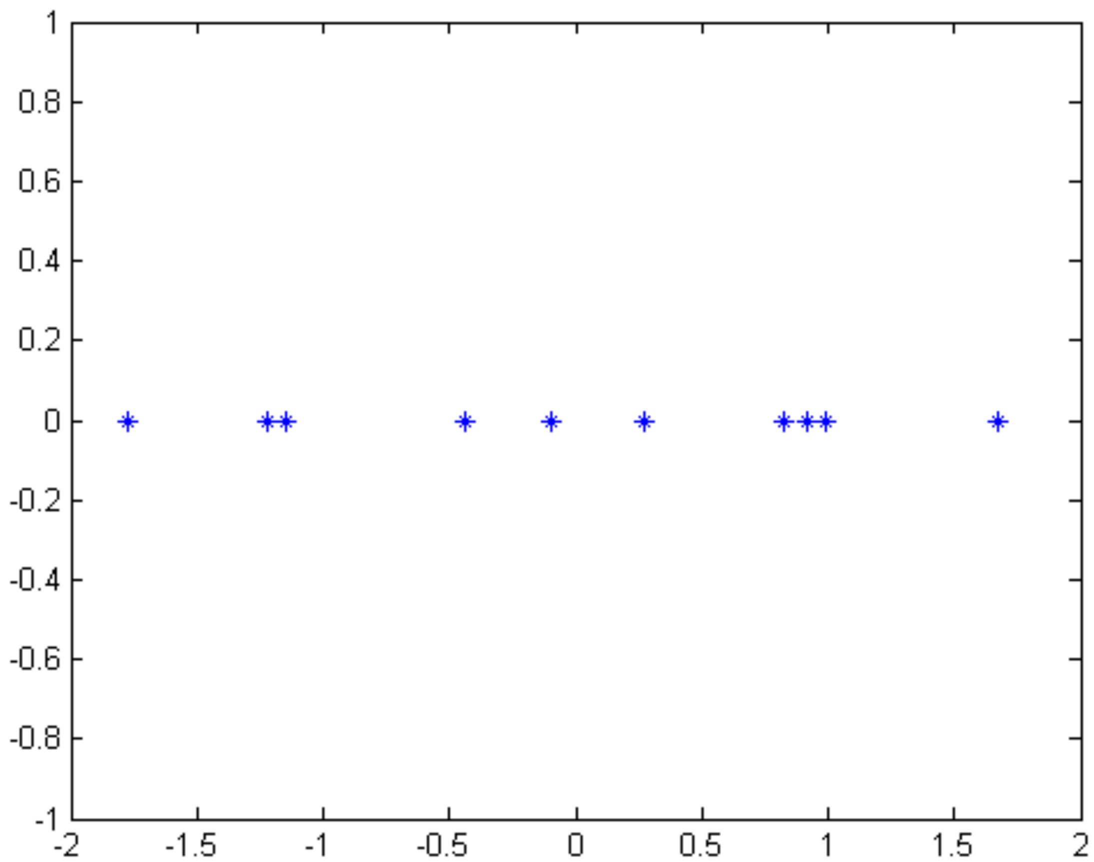
Example

(5) feature $v_1 = \begin{bmatrix} 0.6779 \\ 0.7352 \end{bmatrix}$ = **W**

$z_i = x^T v_i$

$x = [x_1, x_2, \cdots, x_n] \rightarrow z = xW = [z_1, \cdots, z_k]$

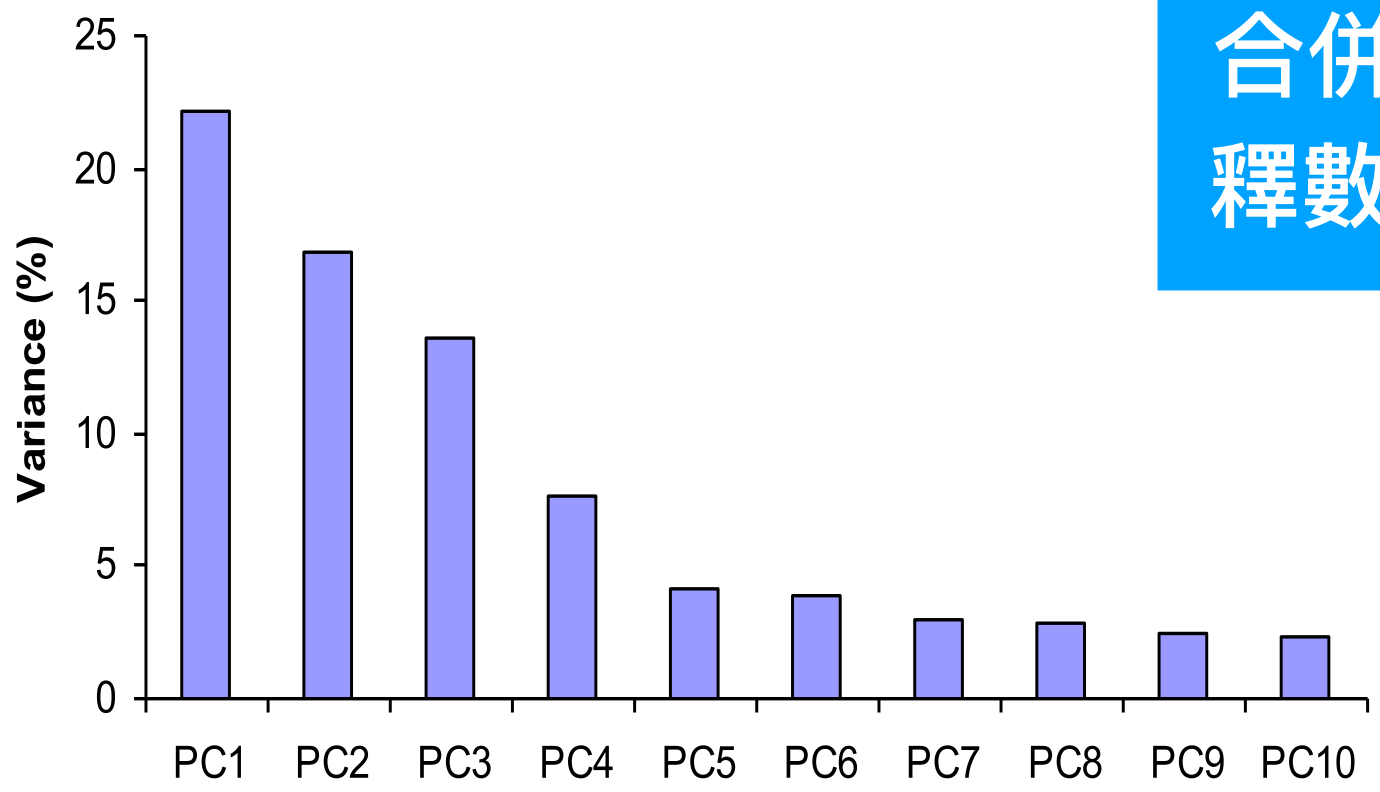
$$\begin{bmatrix} 0.69 & 0.49 \\ -1.31 & -1.21 \\ 0.39 & 0.99 \\ 0.09 & 0.29 \\ 1.29 & 1.09 \\ 0.49 & 0.79 \\ 0.19 & -0.31 \\ -0.81 & -0.81 \\ -0.31 & -0.31 \\ -0.71 & -1.01 \end{bmatrix} \begin{bmatrix} 0.6779 \\ 0.7352 \end{bmatrix} = \begin{bmatrix} 0.8280 \\ -1.7776 \\ 0.9922 \\ 0.2742 \\ 1.6758 \\ 0.9129 \\ -0.0991 \\ -1.1446 \\ -0.4380 \\ -1.2238 \end{bmatrix}$$



How Many PCs?

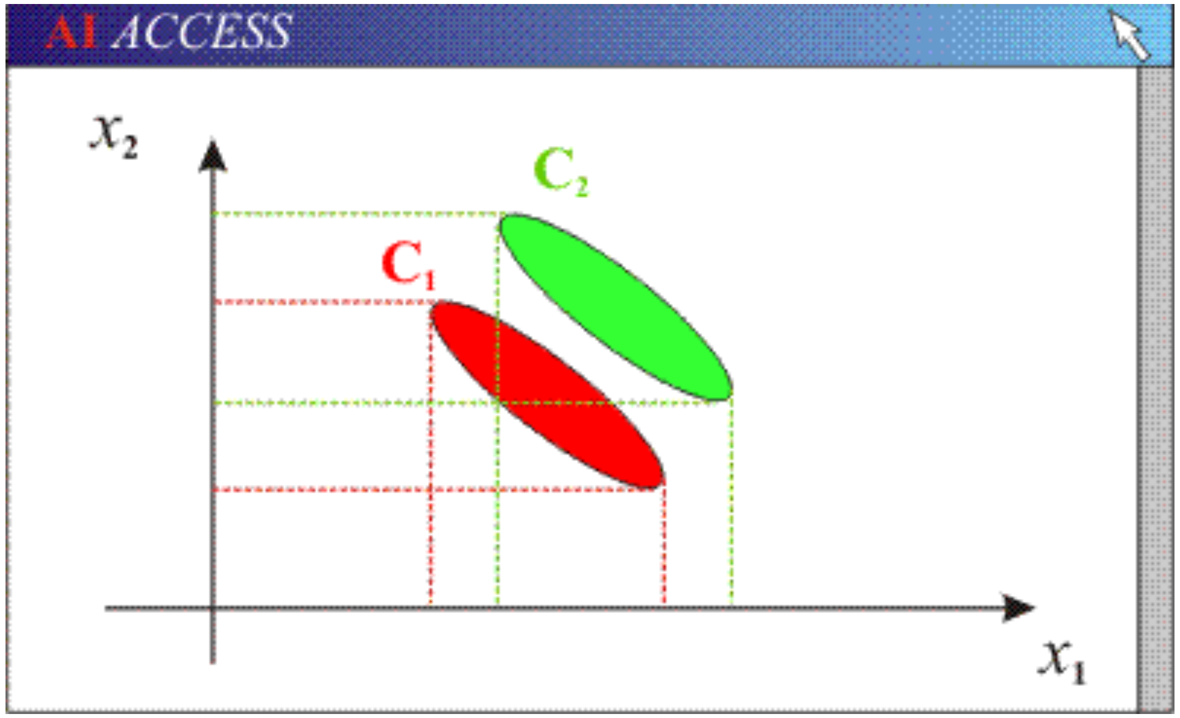
解釋變異數比率 (variance explained ratios) [% of total variance] (V_j) for each component j:

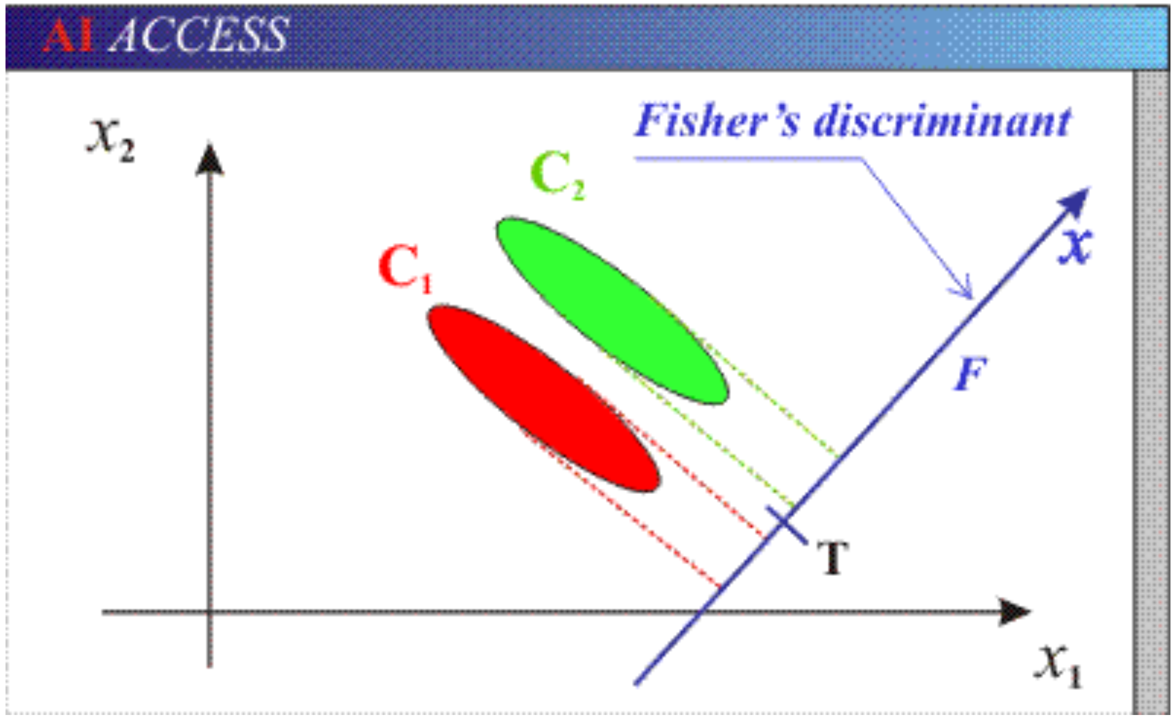
$$V_j = 100 \cdot \frac{\lambda_j}{\sum_{x=1}^{n} \lambda_x}$$



合併前兩個「主成份」可以解釋數據集約40%的「變異數」

Fisher's Linear Discriminant



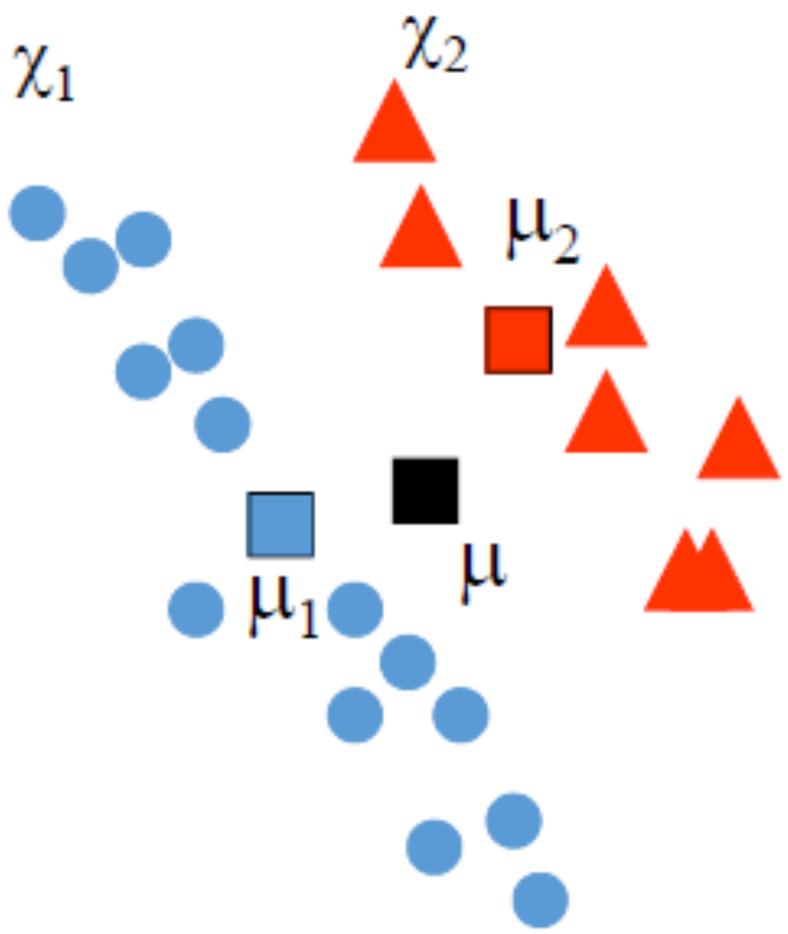


可以看到兩個類別，一個綠色類別，一個紅色類別。

左圖是兩個類別的原始數據，將數據從二維降維到一維，直接投影到x_1軸或者x_2軸，不同類別之間會有重複，導致分類效果下降。

右圖映射到的直線就是用LDA方法計算得到的，紅色類別和綠色類別在映射之後之間的距離是最大的，而且每個類別內部點的離散程度是最小的（聚集程度是最大的）。

PCA v.s. LDA



- PCA (Eigenfaces)

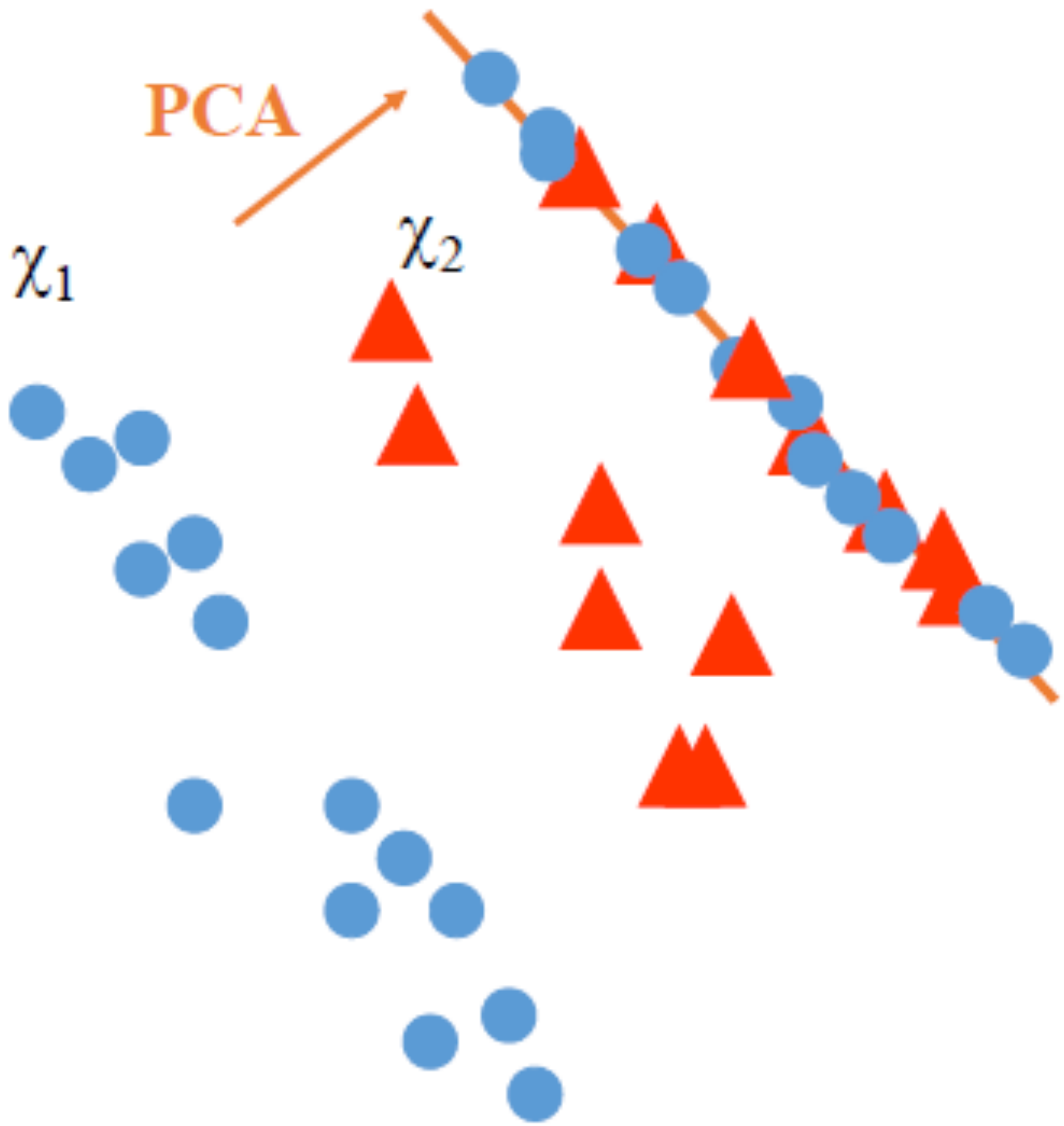
$$\boxed{W_{PCA} = \arg\max_{W} \left| W^T S_T W \right|}$$

Maximizes projected total scatter

$$S_T = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T$$

PCA v.s. LDA

找出最大化變異數的
“正交主成份軸”



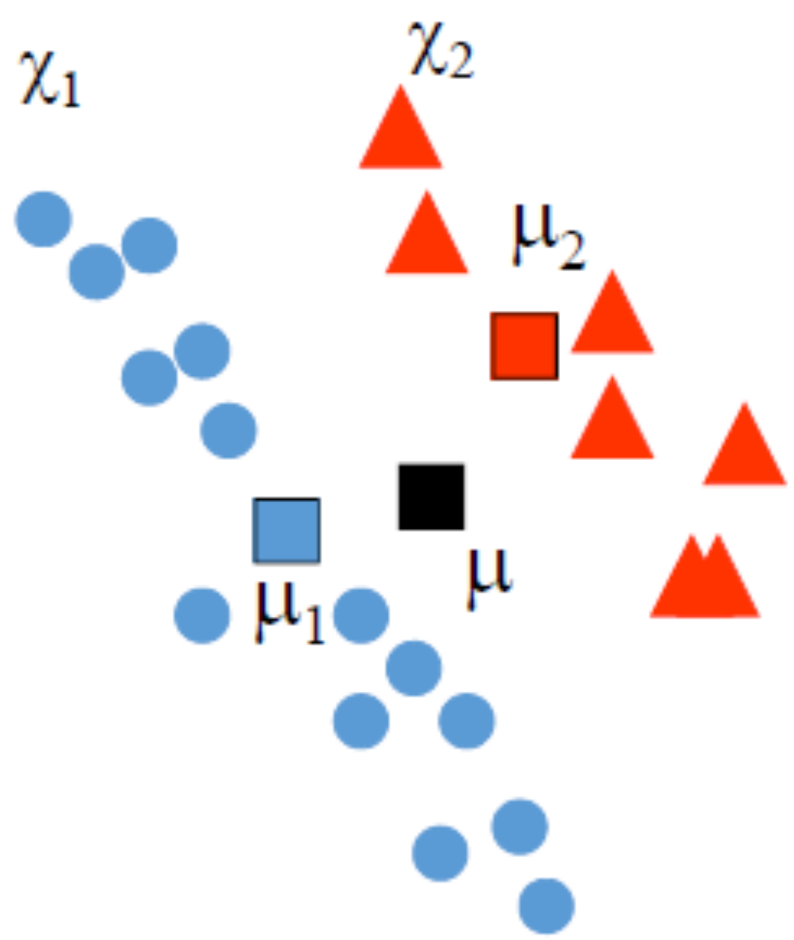
- PCA (Eigenfaces)

$$\boxed{W_{PCA} = \arg\max_{W} \left| W^T S_T W \right|}$$

Maximizes projected total scatter

$$S_T = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T$$

PCA v.s. LDA



- PCA (Eigenfaces)

$$\boxed{W_{PCA} = \arg\max_{W} \left| W^T S_T W \right|}$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

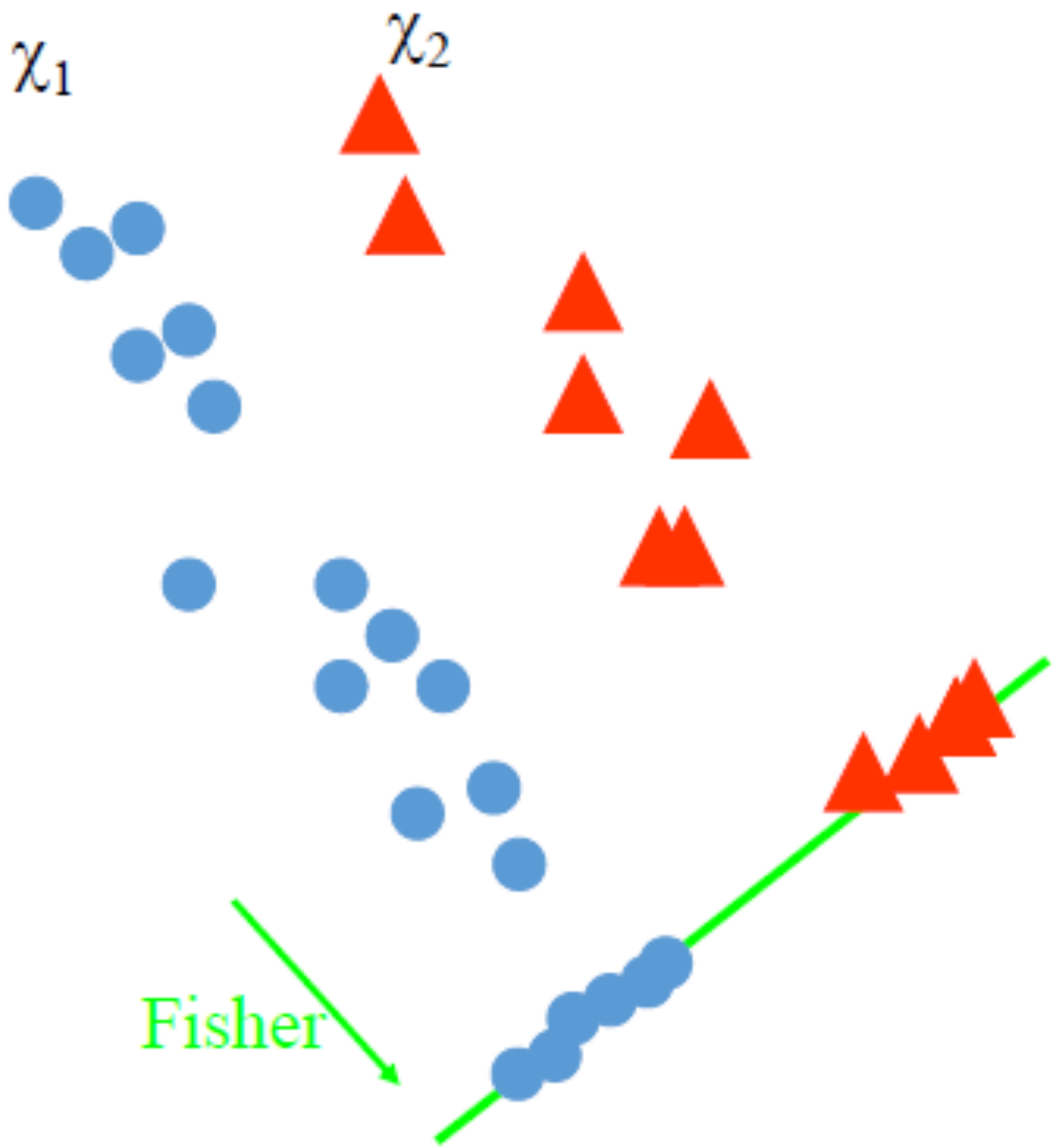
$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

where c is the number of classes

μ_i **is the mean of class** X_i

N_i **is the number of** X_i

PCA v.s. LDA



- PCA (Eigenfaces)

$$\boxed{W_{PCA} = \arg\max_{W} \left| W^T S_T W \right|}$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

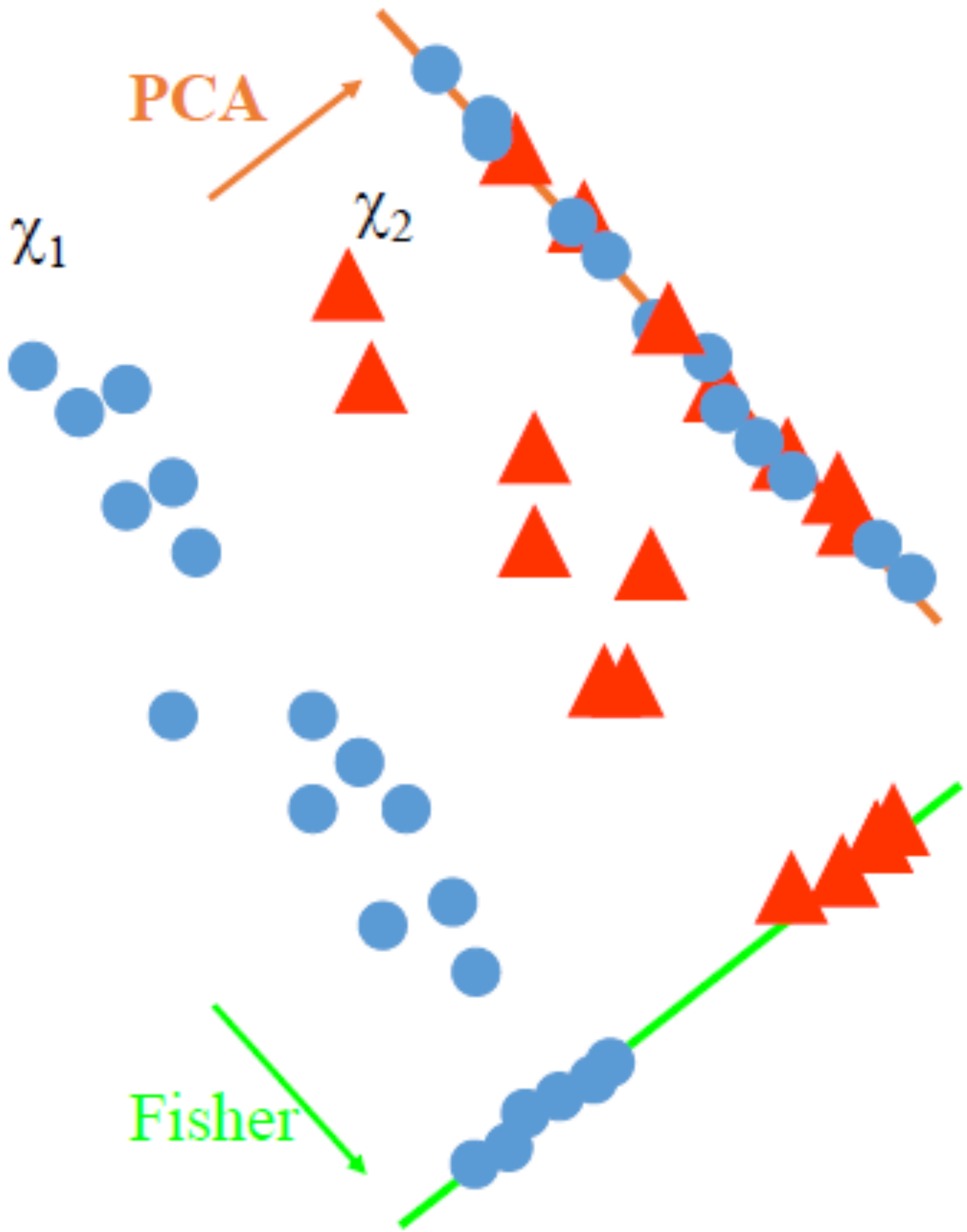
$$\boxed{W_{fld} = \arg\max_{W} \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|}}$$

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

PCA v.s. LDA

找出最大化變異數的
“正交主成份軸”



- PCA (Eigenfaces)

$$\boxed{W_{PCA} = \arg\max_{W} \left| W^T S_T W \right|}$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$\boxed{W_{fld} = \arg\max_{W} \frac{\left| W^T S_B W \right|}{\left| W^T S_W W \right|}}$$

Maximizes ratio of projected between-class to projected within-class scatter

找出可以最佳化類別
分離的“特徵子空間”

LDA

1. 標準化數據集
2. 建立「類別間」（between-class）的「散佈矩陣」(scatter matrix) S_B 與「類別內」（within-class）的「散佈矩陣」S_W
3. 計算 $S_W^{-1}S_B$ 的特徵值和特徵向量
4. 選取 k 個最大的特徵值
5. 用此 k 個特徵值對應的特徵向量建立「投影矩陣」(project matrix) W
6. 利用 W 轉換數據集

PCA v.s. LDA

- Principal Component Analysis (PCA，主成分分析): 對「非監督式數據」壓縮
- Linear Discriminant Analysis (LDA，線性判別分析)：對「監督式數據」降維來最大化類別分離性
- 直觀上，對分類問題，LDA是比PCA更好的一種特徵選取的技術
- 但有研究發現，對某些“圖像識別”的情況，使用PCA往往會得到較好的結果 (A. M. Martinez and A. C. Kak, “PCA Versus LDA.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2):228-233, 2001)

Python code

- 輸出各「主成分」的「解釋變異數比率」

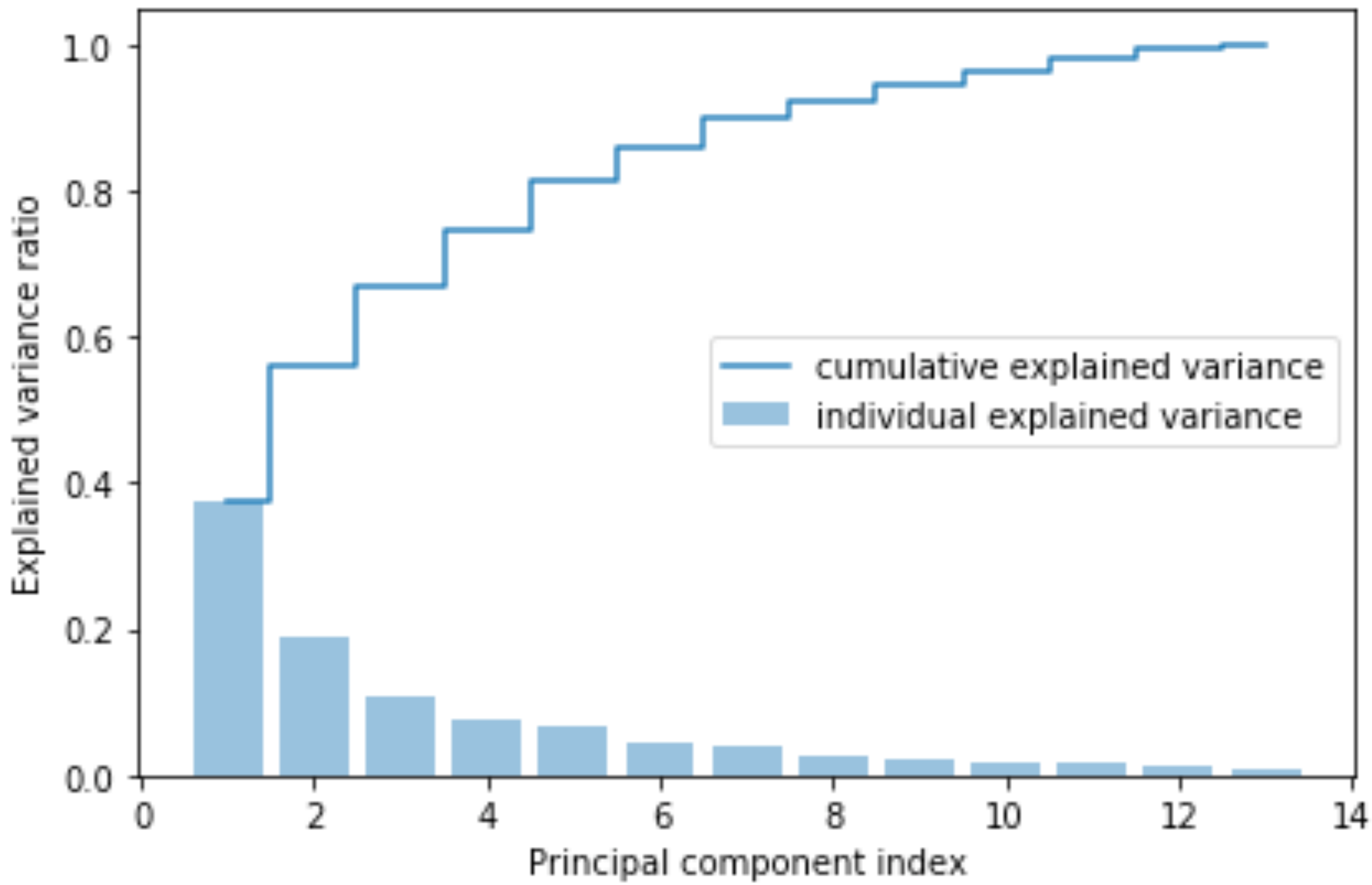
保留所有「主成分」，並以排序好的方式回傳

```
pca = PCA(n_components = None)
X_train_pca = pca.fit_transform(X_train_std)
var_ratio = pca.explained_variance_ratio_
cum_var_ratio = np.cumsum(var_ratio)

plt.bar(range(1, 14), var_ratio, alpha=0.5, align='center',
        label='individual explained variance')
plt.step(range(1, 14), cum_var_ratio, where='mid',
         label='cumulative explained variance')
plt.ylabel('Explained variance ratio')
plt.xlabel('Principal component index')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```

透明度

Python



1.0
0.8
0.6
0.4
0.2
0.0
Explained variance ratio
0
2
4
6
8
10
12
14
Principal component index
cumulative explained variance
individual explained variance