6. k-means and PCA

黃志勝（Tommy Huang）
義隆電子 人工智慧研發部
國立陽明交通大學 AI學院 合聘助理教授
國立台北科技大學 電資學院合聘助理教授

基礎機器學習

針對前述的介紹，每個topic都介紹一個演算算法

1. Regression: Linear regression & Regularization

2. Classification: Linear and Quadratic Discriminant Analysis

3. Clustering: K-means (Unsupervised learning)

4. Dimension Reduction: PCA (Unsupervised learning)

5. Ensemble learning: 不介紹。

k-means Clustering(Unsupervised learning)

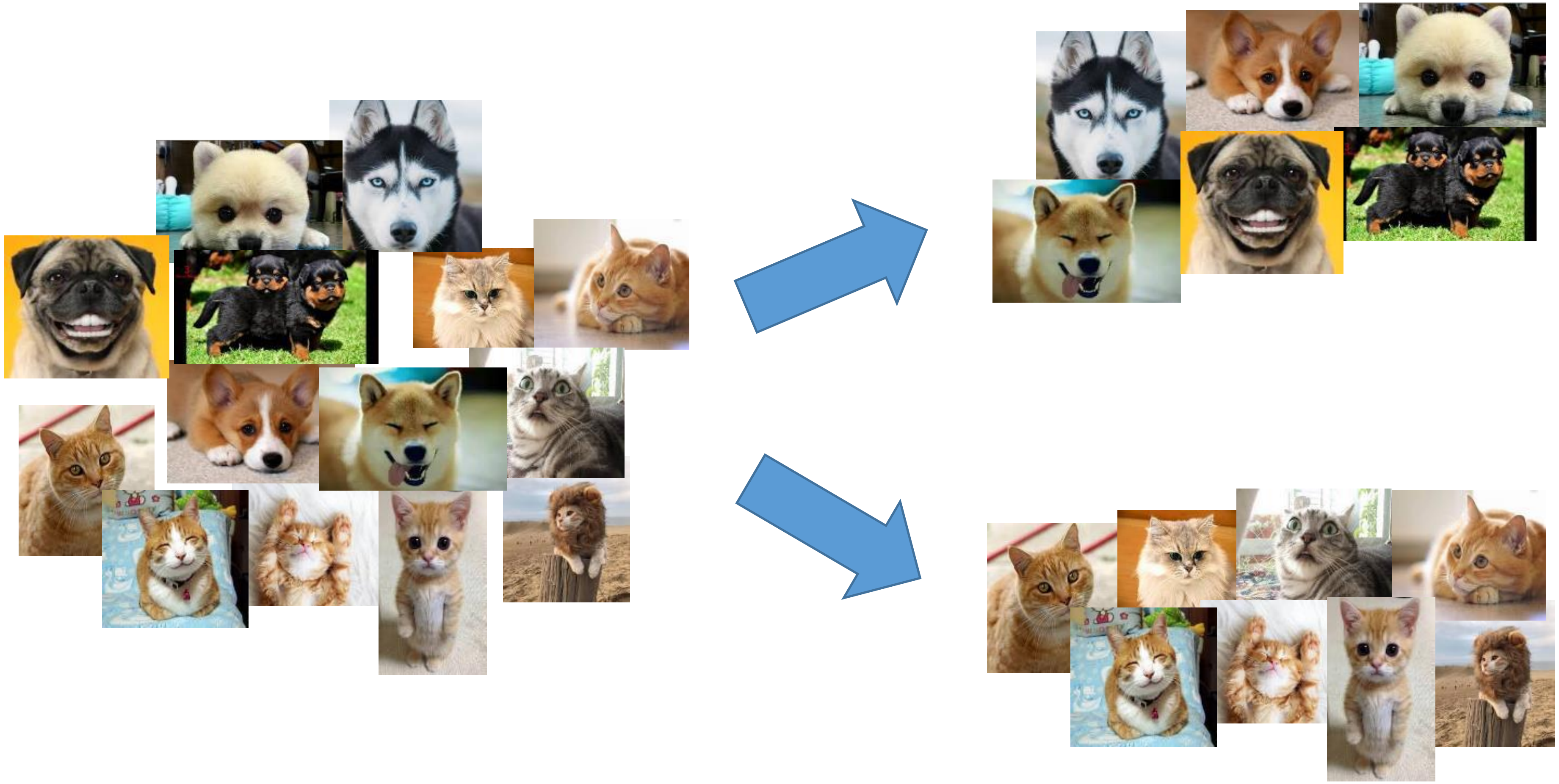
- *k*-means Clustering: 物以類聚 (類似歸納法)

不斷學習(iteration)，直到收斂為止。

為什麼叫*k*-means顧名思義就是有*k*個群心，我們將資料學習後判斷這些資料屬於哪個群心。

EX: 給你一組身高和體重資料，但我沒有跟你說這組資料哪些是男生哪些是女生。我希望你用這組資料分出兩群，這種時候就是用非監督式學習。→ 2-means clustering

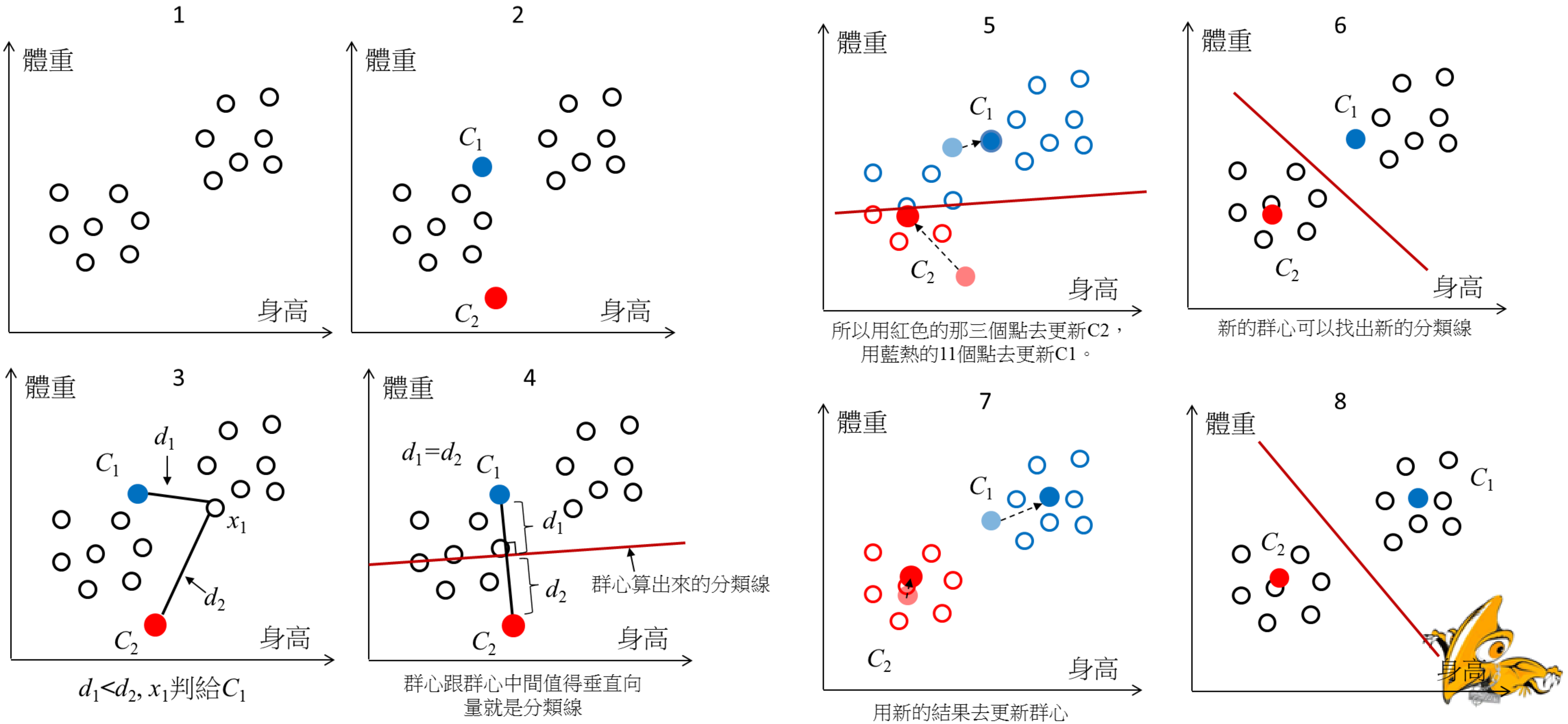
k-means Clustering(Unsupervised learning)



k-means Clustering(Unsupervised learning)

- *k*-means: with a name “means”.
- Most important information is “mean vector”.
- *k*-means is based on learning the mean vector for each cluster.
- Number of cluster is setting by user.

k-means Clustering(Unsupervised learning)



$d_1<d_2$, x_1判給C_1

群心算出來的分類線

群心跟群心中間值得垂直向量就是分類線

所以用紅色的那三個點去更新C2，用藍熱的11個點去更新C1。

新的群心可以找出新的分類線

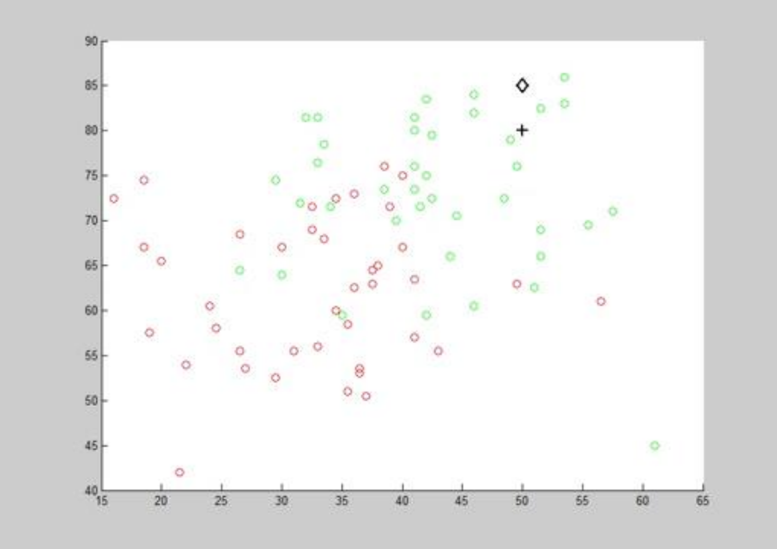
用新的結果去更新群心

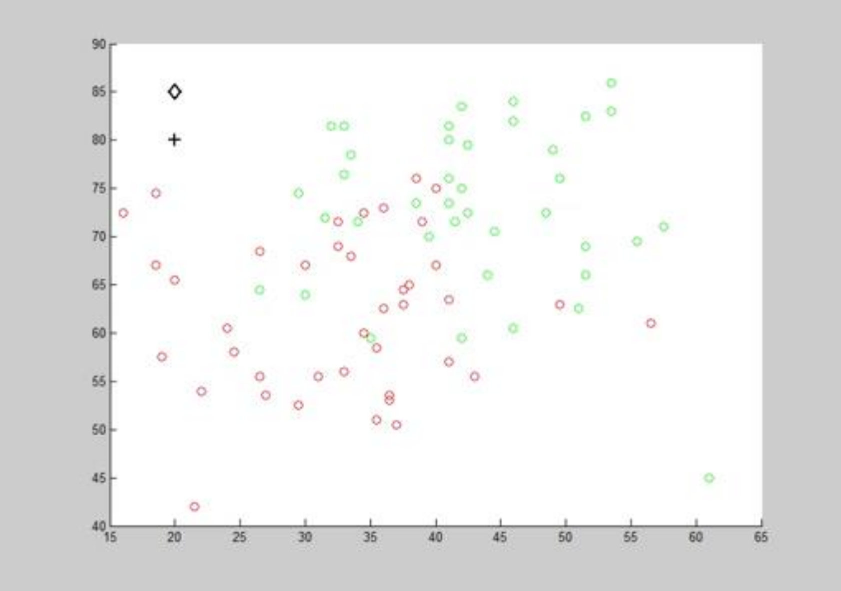
k-means Clustering(Unsupervised learning)

- 設定有*k*(必須≤*n*)個Clusters {**S**1,**S**2,...,**S***k*}，*k*-means clustering就是希望可以最小化群內的資料和群心的誤差平方和越小越好，數學公式如下:

$$\arg\min_{\mu} \sum_{c=1}^{K} \sum_{i=1}^{n_c} \|x_i - \mu_c\|^2 \Big|_{x_i \in S_c}$$

k-means Clustering





1. 初始隨機設定k個群心.

$$\mu_c^{(0)} \in R^d, c = 1,2, \dots, K$$

2. 計算分類到每一群體的樣本，(t)為第t次運算

$$S_c^{(t)} = \left\{x_i : \left\|x_i - \mu_c^{(t)}\right\| \le \left\|x_i - \mu_{c^*}^{(t)}\right\|, \forall i = 1, \dots, n\right\}$$

3. 更新群心(nc個資料在第c群內。)

$$\mu_c^{(t+1)} = \frac{sum(S_c^{(t)})}{n_c} = \sum_{i=1}^{n_c} x_i \Bigg|_{x_i \in S_c^{(t)}}$$

4. 重複2–3，直到群心不變動，也就是

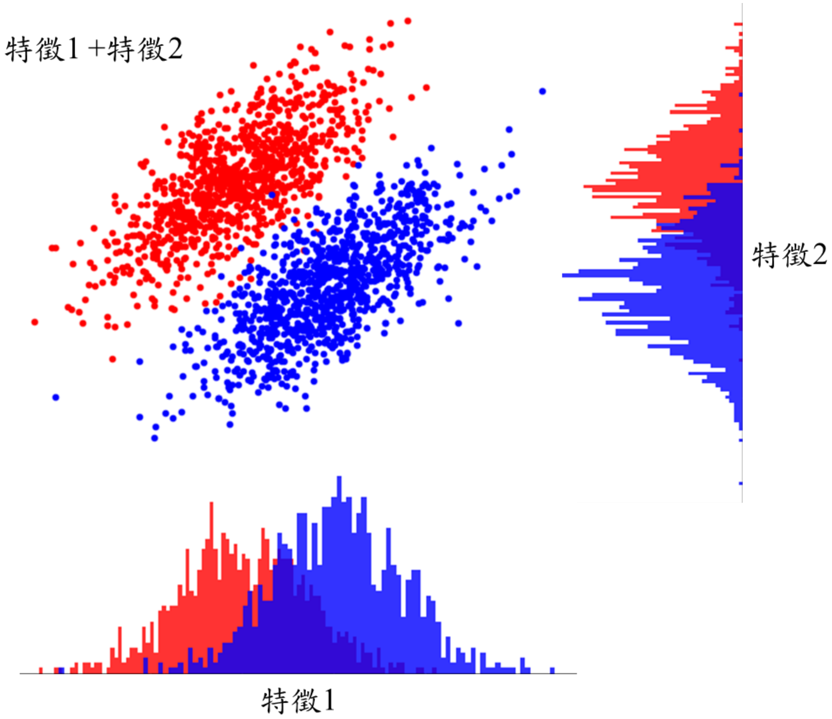
$$S_c^{(t+1)} = S_c^{(t)}, \forall c = 1, \dots, K$$

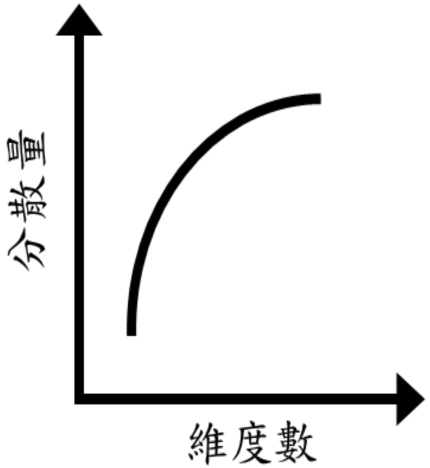
Example

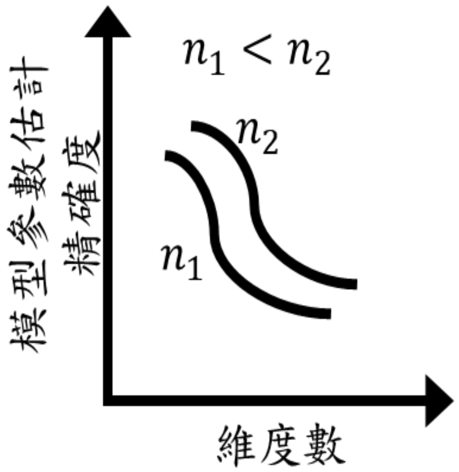
- 利用課程大家填寫的資料，用sklearn做cluster給大家看。

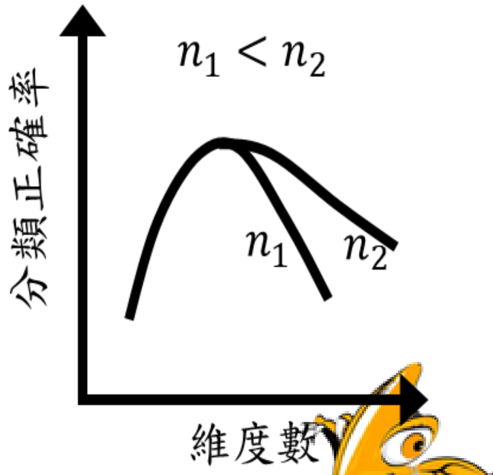
Dimension Reduction

在建立預測模型時，容易因為**特徵數大於資料樣本數**造成模型參數估計錯誤，導致模型無法有效進行任務預測，在機器學習稱此現象為「休斯現象(Hughes phenomenon)」，也稱為「維度詛咒(Curse of dimensionality)」，









Dimension Reduction

Example:

Model 1: “body fat (bf)”

Model 2: “body fat (bf)”, “weight (w)”, “hair length (hl)”

$$\operatorname{cov}(\text{model}1) = \left[\operatorname{cov}(bf, bf)\right]$$

$$\operatorname{cov}(\text{model}2) = \begin{bmatrix} \operatorname{cov}(bf, bf) & \operatorname{cov}(w, bf) & \operatorname{cov}(hl, bf) \\ \operatorname{cov}(bf, w) & \operatorname{cov}(w, w) & \operatorname{cov}(hl, w) \\ \operatorname{cov}(bf, hl) & \operatorname{cov}(w, hl) & \operatorname{cov}(hl, hl) \end{bmatrix}$$

Example for single variable

If we only get one sample, and try to calculate covariance.

$$\mu = x_i$$

$$cov(model\ 1) = \sigma = \frac{1}{1}\sum_{i=1}^{1}(x_i - \mu_x)^2 = 0$$

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Example for multi-variables

If we only get two sample, and try to calculate covariance matrix.

$$\Sigma = \begin{bmatrix} \operatorname{cov}(bf, bf) & \operatorname{cov}(w, bf) & \operatorname{cov}(hl, bf) \\ \operatorname{cov}(bf, w) & \operatorname{cov}(w, w) & \operatorname{cov}(hl, w) \\ \operatorname{cov}(bf, hl) & \operatorname{cov}(w, hl) & \operatorname{cov}(hl, hl) \end{bmatrix}$$

The elements in covariance matrix are larger than 0, but the covariance matrix would be singular.

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = (2\pi)^{-d/2}|\Sigma|^{-0.5} exp\{-0.5(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\mu)\}$$

Dimension Reduction

- Dimension Reduction is proposed to overcome this issue.

1. Feature selection

   Using only “import” features.

2. Feature extraction

   Feature Fusion.

Feature Selection

In statistics,

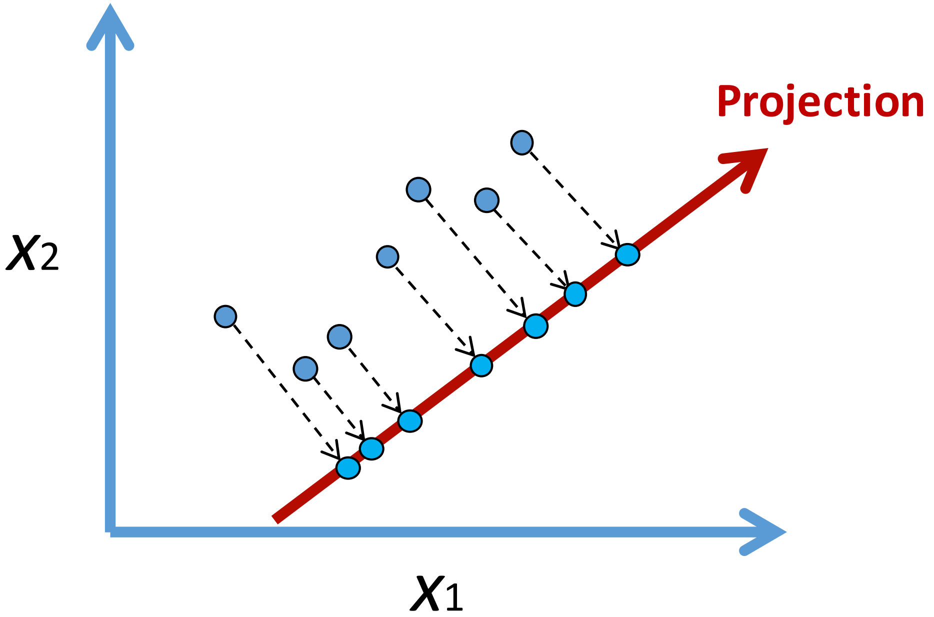
- Forward sequential feature selection
- Backward sequential feature selection
- Stepwise feature selection
- LASSO

In machine learning,

- Random subspace.

Feature Extraction

- Feature Fusion (Projection)



Feature extraction:
Just finding the projection vectors for input features.

Feature Extraction

- **Principal component analysis (PCA)**
- Independent component analysis (ICA)
- Canonical component analysis (CCA)
- Non-negative matrix factorization
- Discriminant Analysis Feature Extraction(DAFE)
- Neural Network

Principal component analysis (PCA)

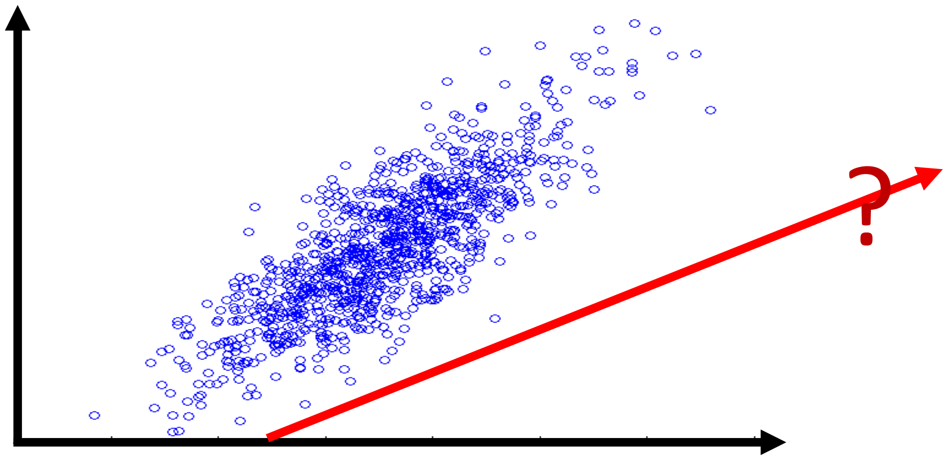
Why do I introduce PCA?

1. Stronger knowledge.
2. Unsupervised.
3. Most popular
4. Basic

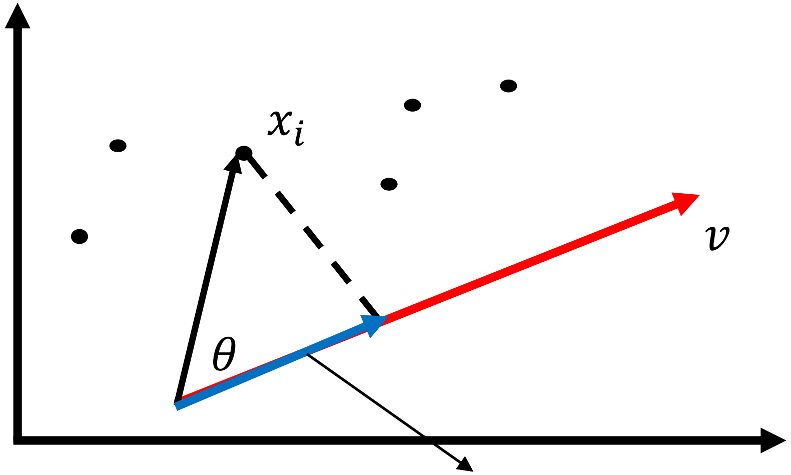
Principle Component Analysis

DL/ML/Statistics are developed by a given goal.

- PCA aims to find a set of vector containing the maximum amount of variance in the data.



Principle Component Analysis



$$\|x_i\|\cos\theta$$

$$\cos\theta = \frac{\langle x_i, v\rangle}{\|x_i\|\|v\|}$$

$$\|x_i\|\cos\theta\frac{v}{\|v\|}$$

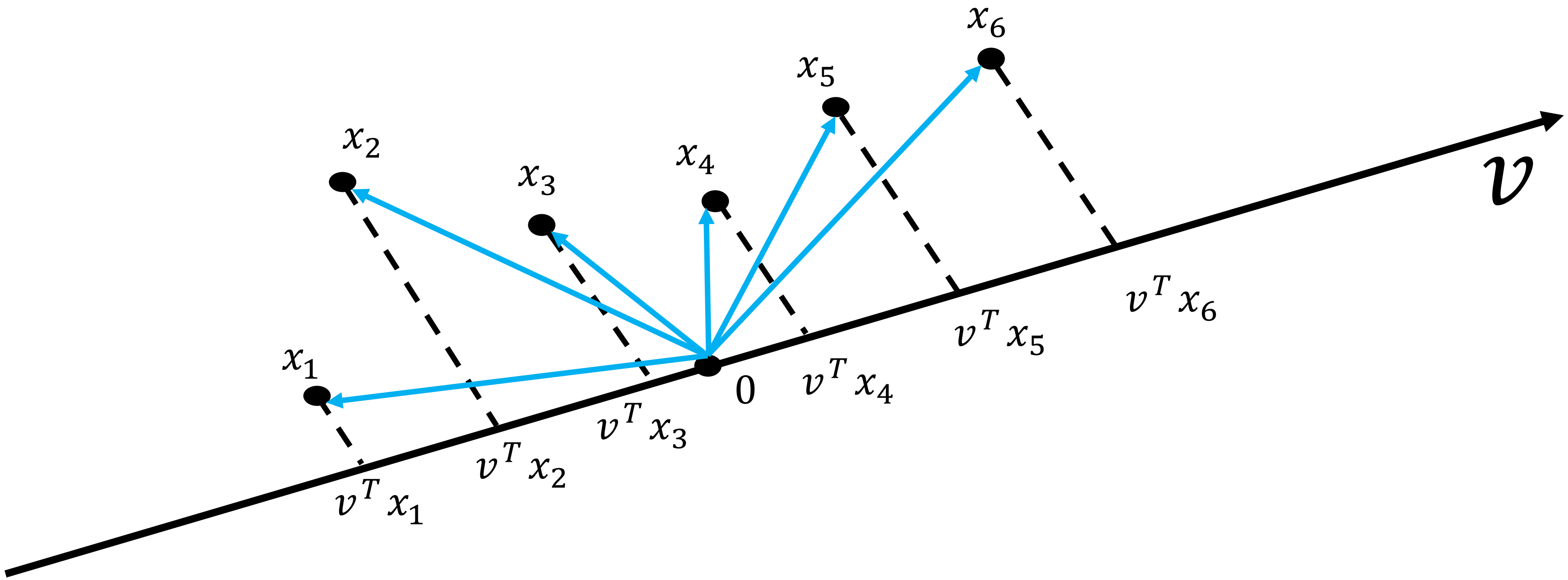
$$= \|x_i\|\frac{\langle x_i, v\rangle}{\|x_i\|\|v\|}\frac{v}{\|v\|} = \frac{\langle x_i, v\rangle}{\|v\|^2}v$$

- If $\|v\|$ is unit, then $\langle x_i, v\rangle v$

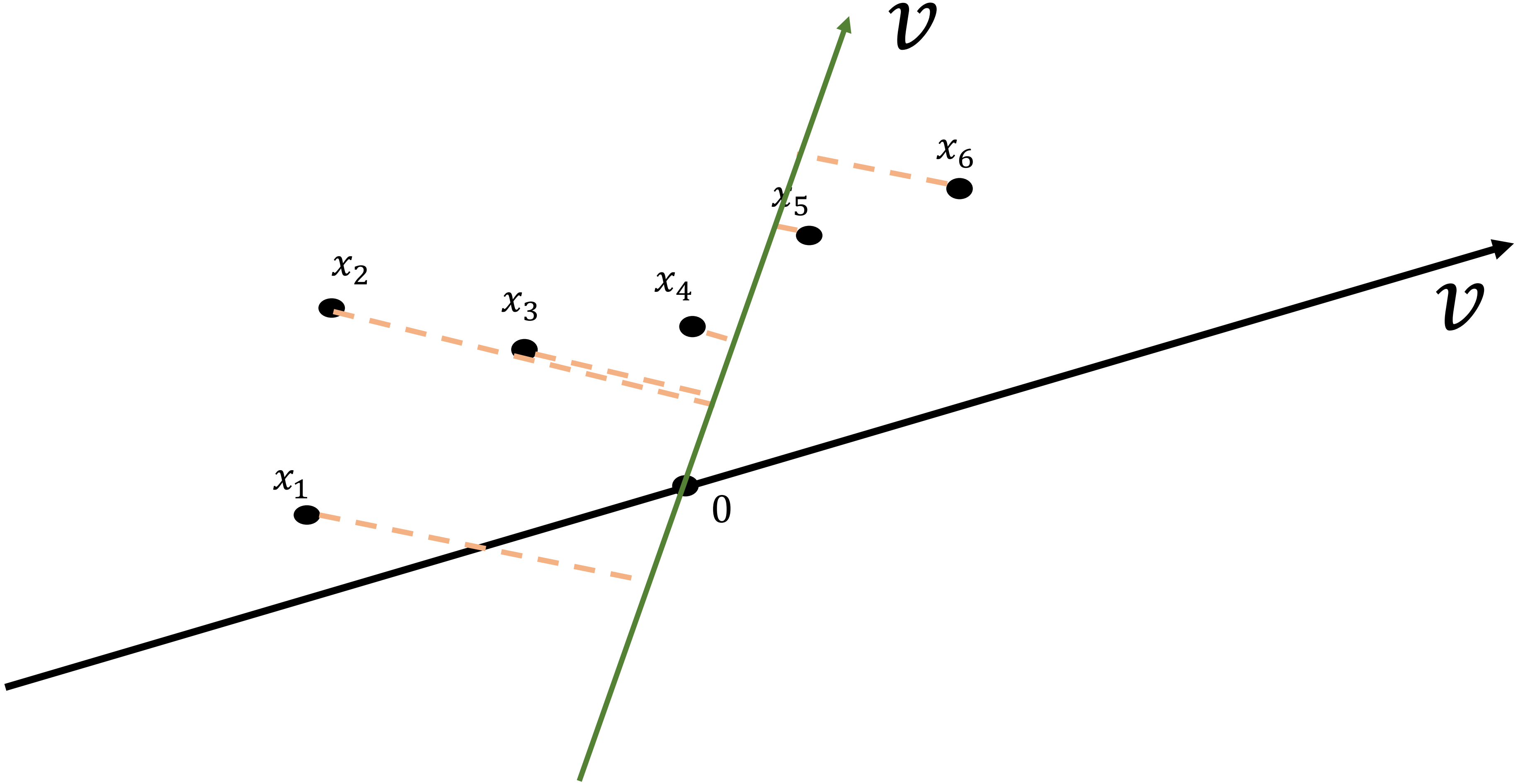
$$\langle x_i, v\rangle = x_i^T v = v^T x_i$$

$$y_i = v^T x_i$$

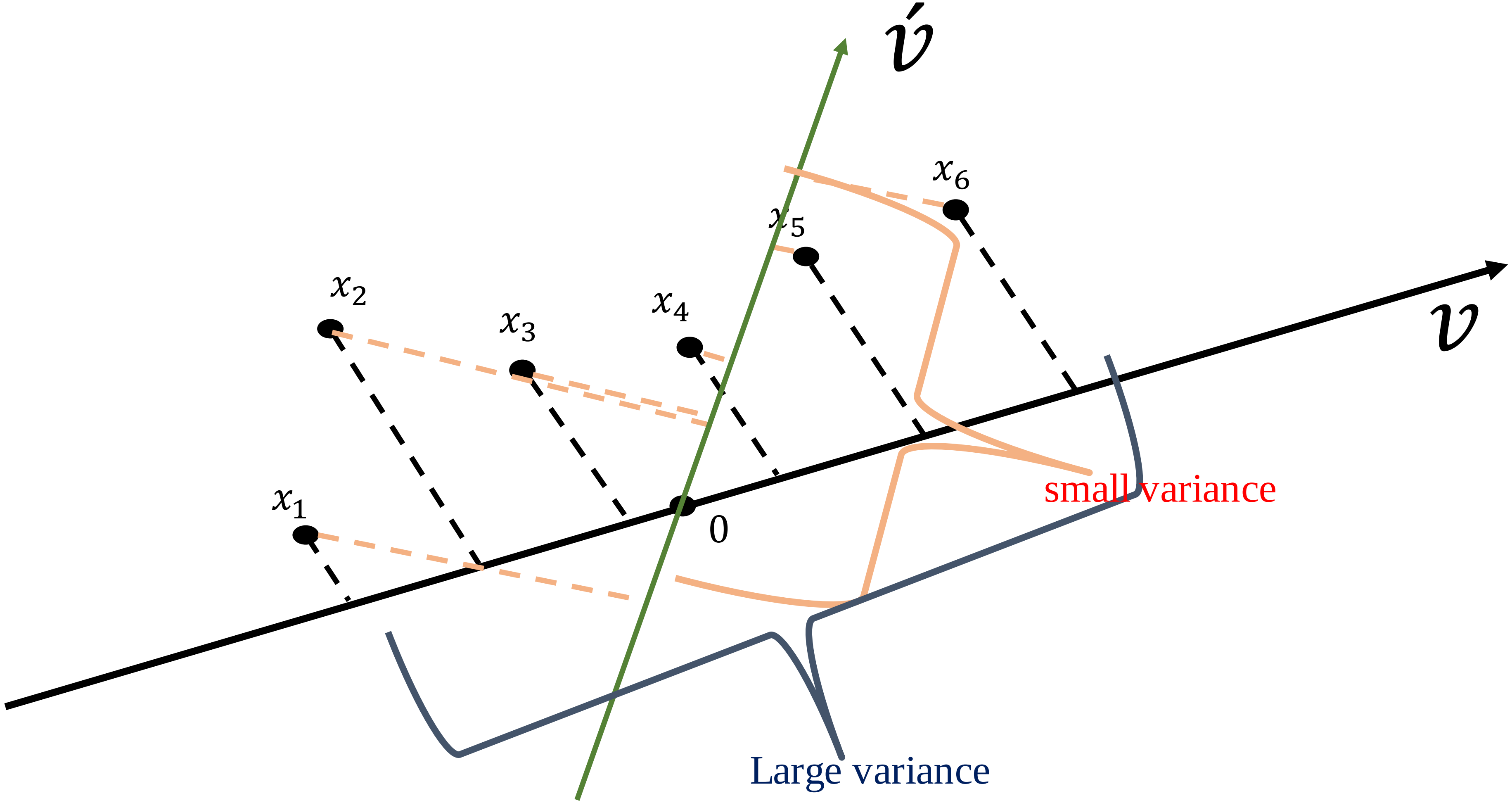
Principle Component Analysis



Principle Component Analysis



Principle Component Analysis



Principle Component Analysis

- The projections of the all points x_i into the direction v are

$$v^T x_1, v^T x_2, \dots, v^T x_N$$

The variance of the projections is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i - 0)^2 = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i)^2$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i)(v^T x_i)^T = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i x_i^T v) = v^T\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T\right)v$$

$$= v^T C v$$

C covariance matrix

Principle Component Analysis

- The first principal vector can be found by the following equation:

$$v = \arg\max_{v \in R^d, \|v\|=1} v^T C v$$

Principle Component Analysis

$$v = \arg\max_{v \in R^d, \|v\|=1} v^T C v$$

Lagrange function:

$$f(v, \lambda) = v^T C v - \lambda (v^T v - 1)$$

$$\frac{\partial f(v,\lambda)}{\partial v} = 0 \Rightarrow 2Cv - \lambda v = 0 \Rightarrow Cv = \lambda v$$

$$\frac{\partial f(v,\lambda)}{\partial \lambda} = 0 \Rightarrow v^T v - 1 = 0 \Rightarrow v^T v = 1$$

Principle Component Analysis

- The first principal vector can be found by the following equation:

$$v = \underset{v \in R^d,\ \|v\|=1}{\operatorname{argmax}} (v^T C v)$$

- This is equivalent to find the largest eigenvalue of the following eigenvalue problem:

$$Cv = \lambda v$$
$$\|v\| = 1$$

$$C = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T = \frac{1}{N}\left[x_1 \ \dots \ x_N\right]\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \frac{1}{N} X^T X$$

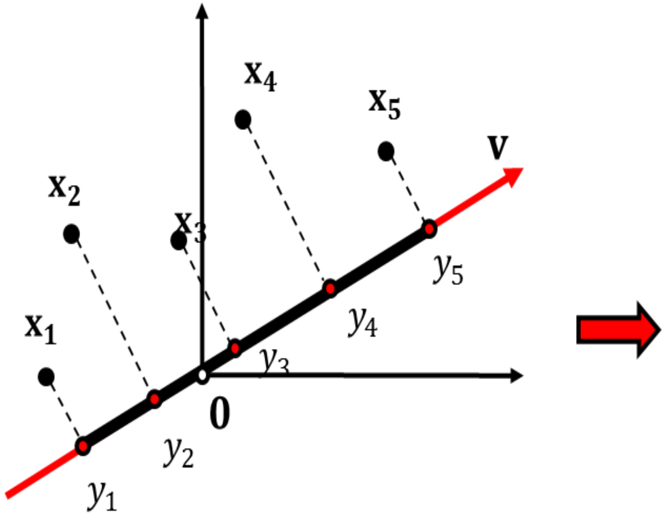
Principle Component Analysis

Eigenvalue vector is the corresponding variance vector.

$$v^T C v = \sigma^2$$

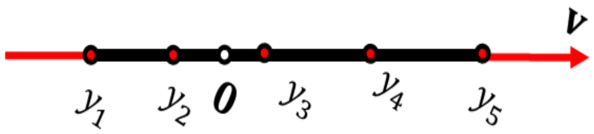
$$\Rightarrow v^T \lambda v = \lambda v^T v = \lambda = \sigma^2$$

Projection



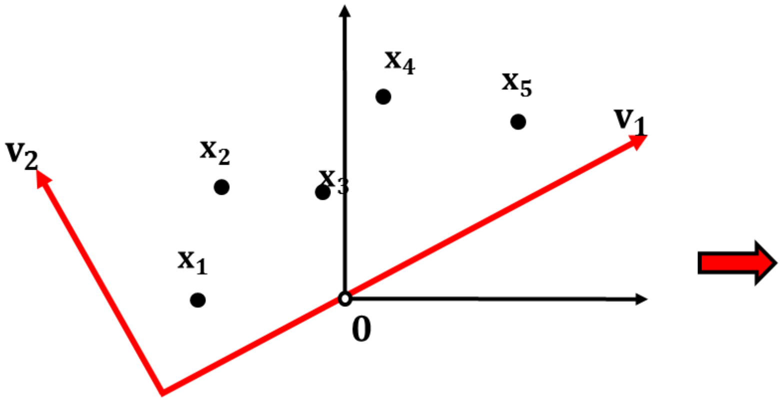
X空間轉換到特徵空間

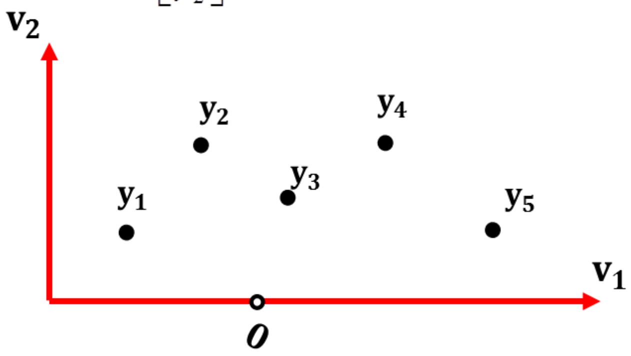
$$y = \mathbf{v}^T\mathbf{x}$$



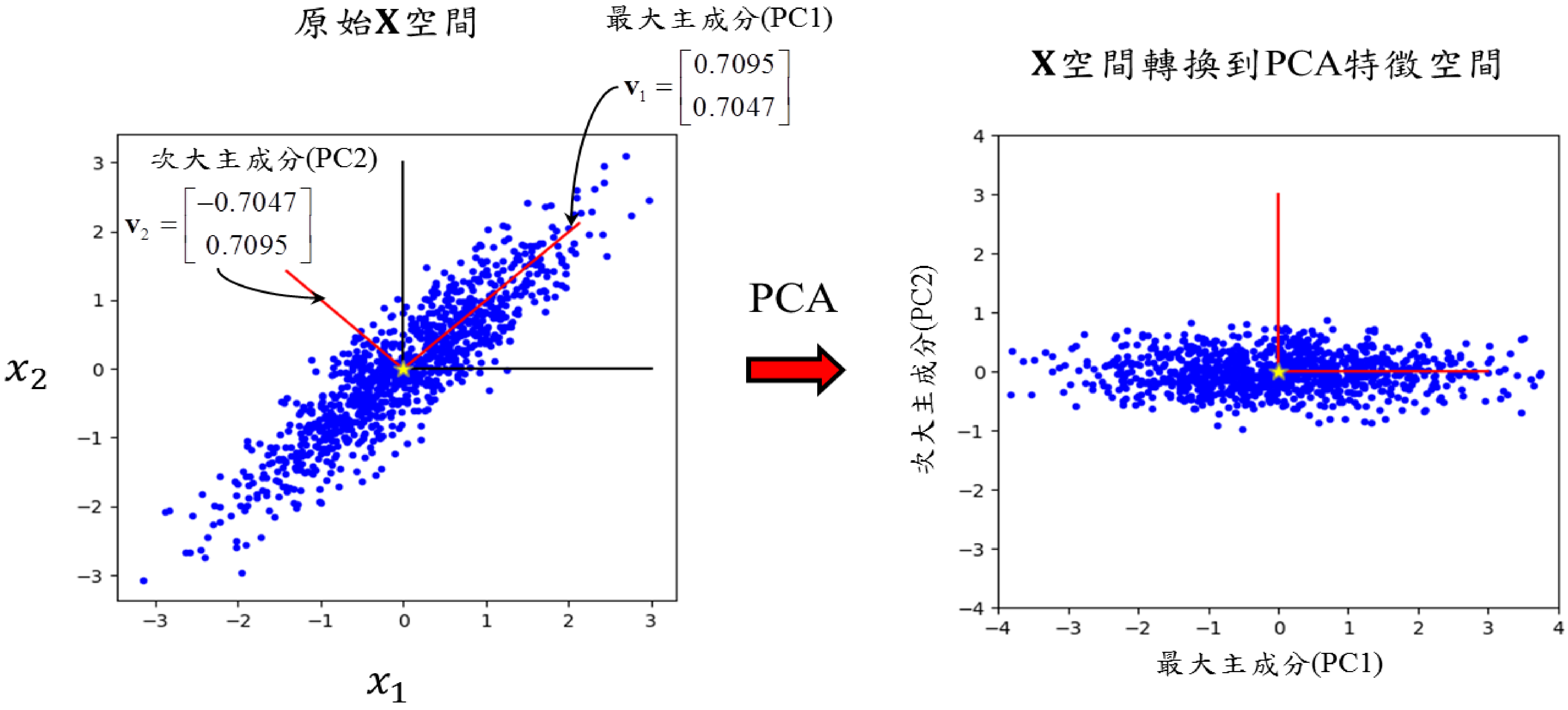
X空間轉換到**Y**空間

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{V}^T\mathbf{x} = [\mathbf{v}_1 \quad \mathbf{v}_2]\mathbf{x}$$





Exercise

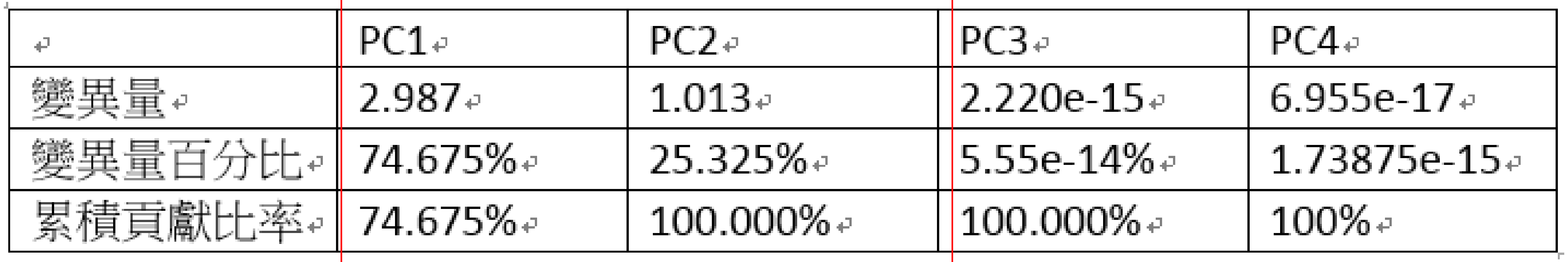

原始X空間
最大主成分(PC1)
$\mathbf{v}_1 = \begin{bmatrix} 0.7095 \\ 0.7047 \end{bmatrix}$
次大主成分(PC2)
$\mathbf{v}_2 = \begin{bmatrix} -0.7047 \\ 0.7095 \end{bmatrix}$
x_2
x_1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
PCA
X空間轉換到PCA特徵空間
次大主成分(PC2)
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4
最大主成分(PC1)

統計學資料科學的觀點

- **統計學要怎麼決定多少主要成分出來?**
- 答案是從由累積貢獻比率 (Cumulative Proportion)去決定需要取多少主要成分出來。
- **累積貢獻比率 (Cumulative Proportion)是什麼?**

累積貢獻比率 (Cumulative Proportion)

- 假設有4個變數(X1~X4)，所以萃取出的主成份會有(PC1~PC4)，累積貢獻比率則是看前幾個主成份可以表是原始資料多少百分比的變異量。



	PC1	PC2	PC3	PC4
變異量	2.987	1.013	2.220e-15	6.955e-17
變異量百分比	74.675%	25.325%	5.55e-14%	1.73875e-15
累積貢獻比率	74.675%	100.000%	100.000%	100%

兩個ＰＣ就夠代表整筆資料

Example

- 1. IRIS資料做PCA降維和LDA分類
- 2. MNIST資料做PCA降維和LDA分類