



113-2 資料探勘概論

Data-preprocessing 2: Data Reduction

授課教師：蔡孟勳 (Mason Tsai) 教授

助教聯絡方式 & 實驗室位置：

段浩恩 (社管680) howardtuan@smail.nchu.edu.tw

羅子芃 (社管680) 7113029021@smail.nchu.edu.tw





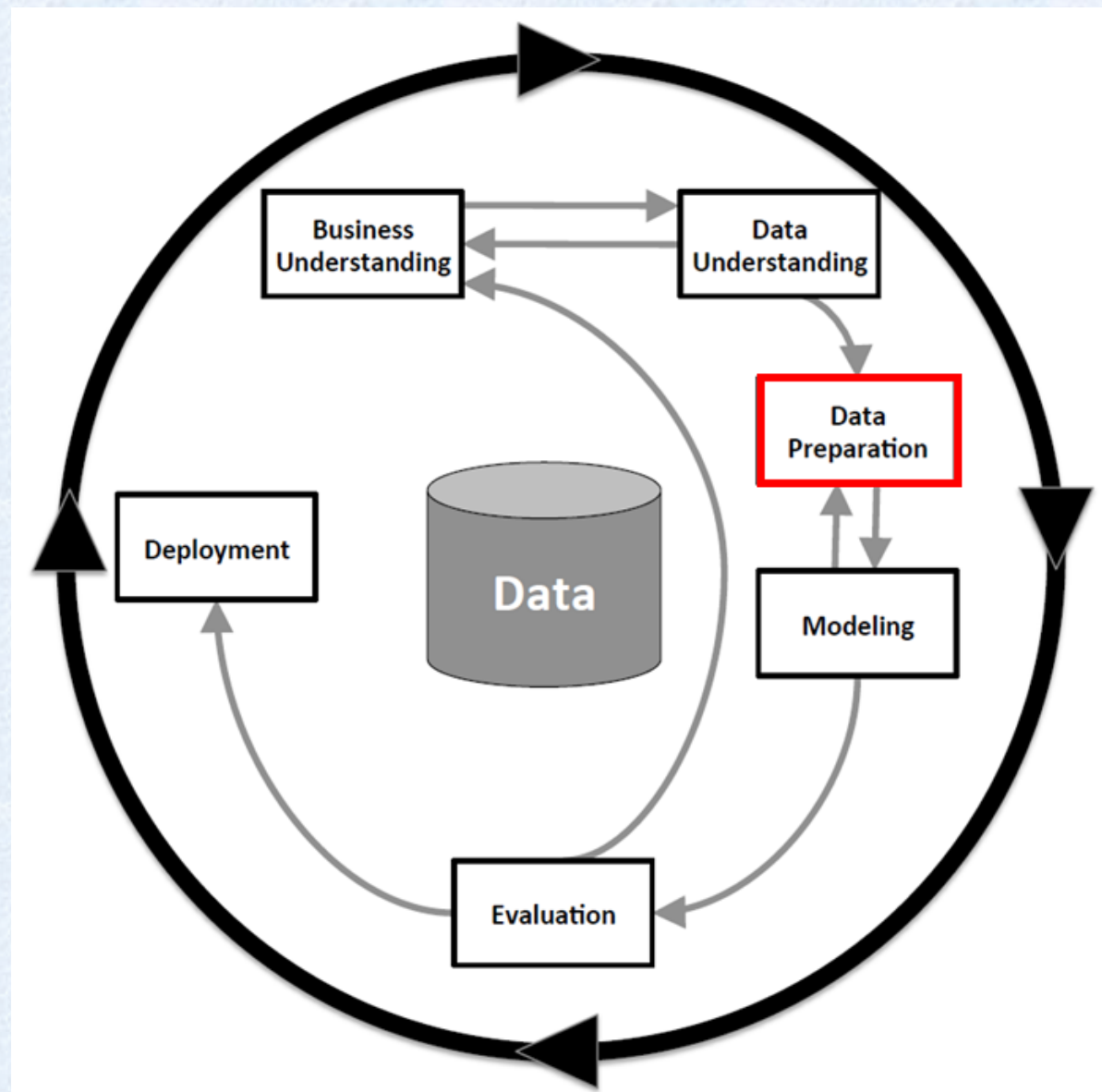
本章大綱

- Data Reduction
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression





Data Preparation





Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression



為甚麼要做資料精簡Data Reduction ?

- 資料庫/資料倉儲可以存儲 **TB 級的資料**。
- 但是**太過「複雜」的資料分析**可能需要**很長時間**才能在完整的資料集上**運行**。
- 資料精簡(Data reduction)
 - **維度縮減** Dimensionality reduction
 - **數量縮減** Numerosity reduction



Dimensionality reduction



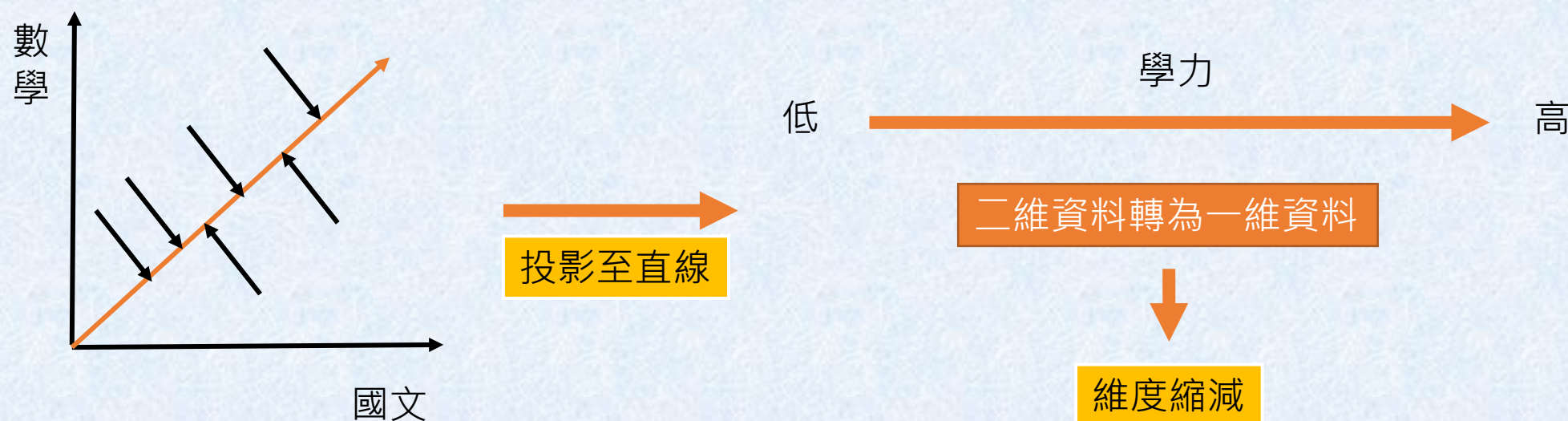
特徵選擇 Feature Selection

- 特徵(Feature)為資料減量的**主要減量目標**
- 特徵減量之後,資料的優點
 - **資料量減少**
 - 提高資料探勘處理正確率
 - 資料探勘後的結果較為簡單，並**減少探勘的時間**
 - **不用浪費太多時間在蒐集不相關或不需要的資料及屬性(Attribute)**
- 特徵減量的相關技術，可分為處理：
 - **非數值型資料 (Non-numerical)**
 - **數值型資料 (Numerical)**



維度縮減 Dimensionality Reduction

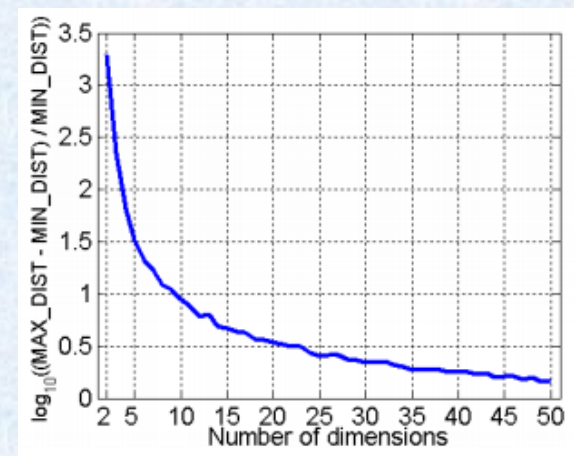
- 維度縮減(dimensionality reduction)
 - **減少維度資料的數量**處理
 - 資料維度:學生成績資料中的國文分數、數學分數、英文分數等
 - 透過下圖將二維資料轉換為一維資料





維度縮減 Dimensionality Reduction

- 維度災難 (Curse of Dimensionality)
 - 最早由**理察·貝爾曼** (Richard E. Bellman) 在考慮優化問題時首次提出來的術語
 - 用來描述當 (數學) 空間維度增加時，分析和組織高維空間 (通常有成百上千維)，因**體積指數增加**而遇到各種問題場景。
 - 當**維度增加**時，**資料會變得越來越稀疏**。
 - 對**聚類、異常值分析**至關重要的**點之間的密度和距離變得不那麼有意義**了。
 - 子空間的**可能組合將呈指數增長**。





維度縮減 Dimensionality Reduction

- 維度災難 (Curse of Dimensionality)
 - 使用**維度縮減**，能夠**迴避**維數災難
 - 一般會認為資料的維度數越高，越能夠充分表達資料的特徵，但在**機器學習**上，維度數**過大**時會碰到「**維數災難**」的現象
 - 簡單來說，維數災難就是「**比較的重點過多，反而分不清楚差異**」
 - 比如，在數學上碰到因**資料間的差異（距離）大小**，而嚴重**影響演算法的性能**

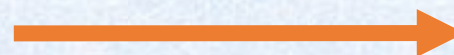


維度縮減 Dimensionality Reduction

- 以維度縮減壓縮資料
 - 將高維度資料轉換為低維度資料可以進行**資料壓縮**
 - 在機器學習處理的資料中，存在多達數十萬、數百萬維度的資料，維度縮減能夠**大幅減少運算量，進行快速的計算。**

	國文	數學
A同學	60	50
B同學	80	40
.	.	.
.	.	.
.	.	.

壓縮



	學力
A同學	4
B同學	5
.	.
.	.
.	.

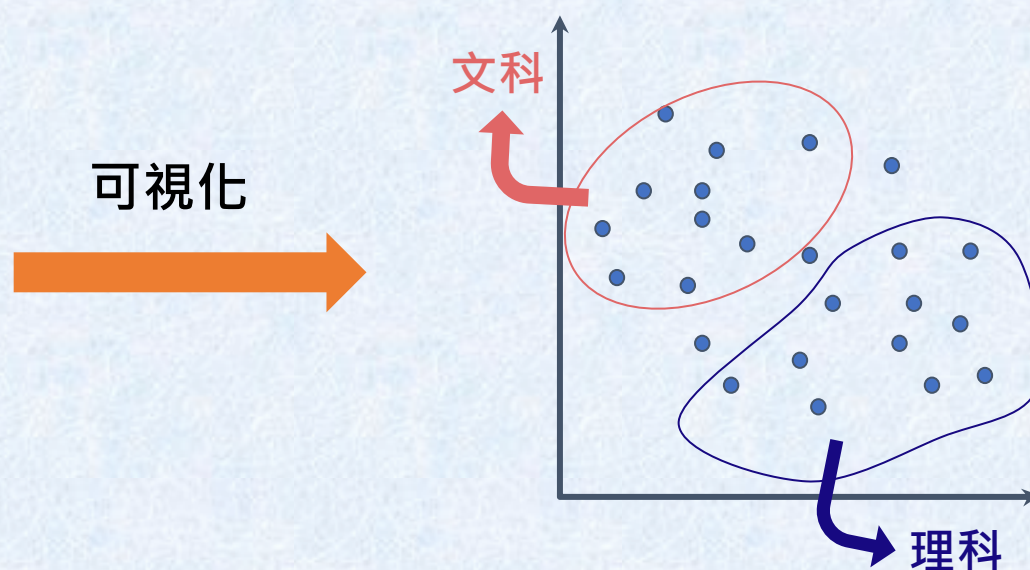


維度縮減 Dimensionality Reduction

- 以維度縮減可視化資料
 - 使用維度減縮，將高維度資料可視化為容易理解的形式，直接表達說明
 - 資料的可視化：

國文	生物
數學	地理
英文	日本史
物理	世界史
化學	

可視化





Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Dimensionality Reduction Techniques
 - Non-numerical
 - Numerical
 - Numerosity reduction
 - Data compression



降維方法

Dimensionality Reduction Techniques

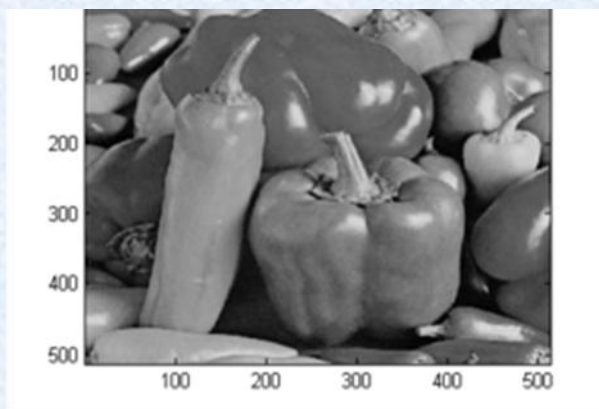
- 離散小波轉換 (Mapping Data to a New Space)
- 主成分分析 (PCA)
- t-隨機鄰近嵌入法 (t-SNE)



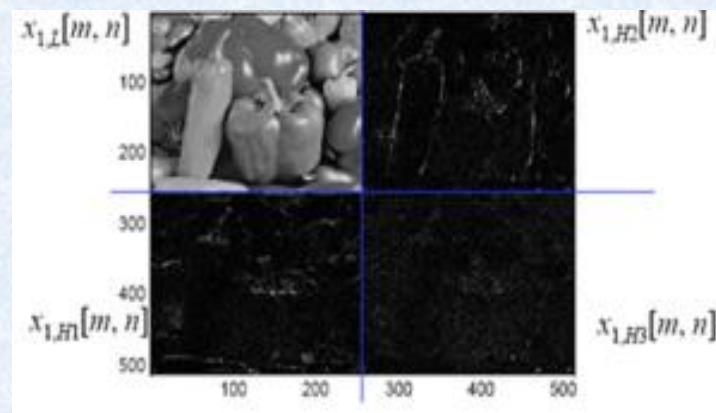
離散小波變換

Discrete Wavelet Transform

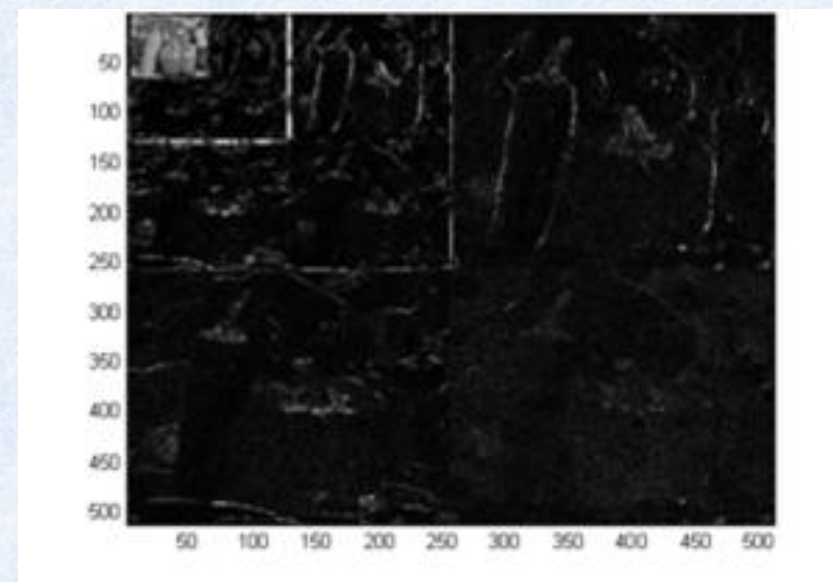
- **離散小波轉換(DWT)可以應用於多維度資料**，例如資料方塊，他的作法如下，首先將此轉換應用在第一維度，再套用於第二維度，以此類推，它們需要的**計算複雜度與資料方塊中的單元數目是線性關係**。
- 對於**稀疏、偏斜與使用順序屬性的資料**，小波轉換能得到很好的**結果**。
- 小波轉換的應用包括：**指紋圖像壓縮、電腦視覺、時間序列分析與資料清理**。



原始圖片



2D DWT的結果





主成分分析法 (Principal Component Analysis, PCA)

在介紹PCA前，先想幾個問題：

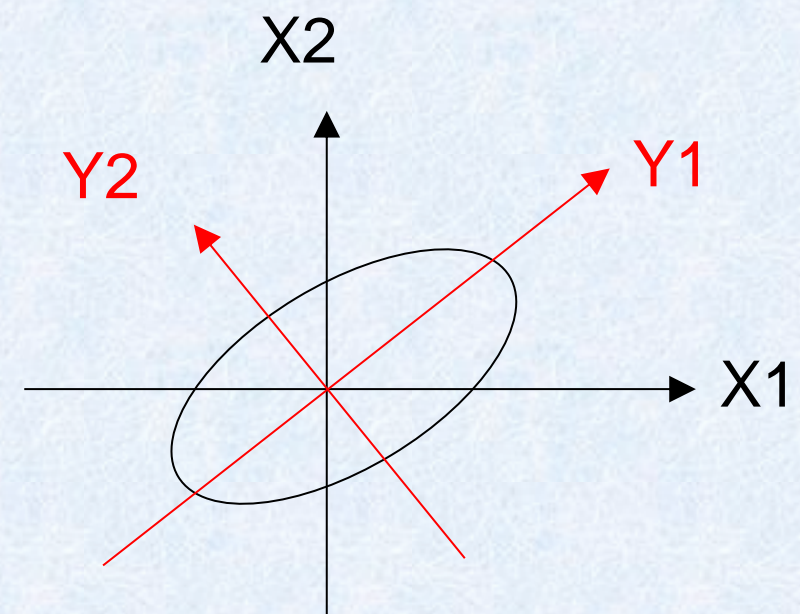
1. 比如拿到一個汽車的資料集，裡面既有以 “km/h(公里/小時)” 度量的最大速度特徵，也有 “mph(英里/小時)” 的最大速度特徵，顯然這兩個特徵有一個多餘。
2. 拿到一個樣本，**特徵非常多，而樣例特別少**，這樣用回歸去直接擬合非常困難，容易**過度擬合**。



主成分分析法 - 簡介

- 假設資料包括了**n個屬性**的數值或是資料向量，挑選最能表示資料變異的**k個維度**的正交向量，因而產生維度的縮減
- 將**原始資料**轉換至另外幾個**主成分變數**，即仍須輸入其原始資料以產生新的主成分，因此僅是計算維度的減少，資料輸入的維度則未改變

若原本資料集有**K個變數**（及K個維度），
則可透過**線性轉換**的方式找到**C個新的變數**（ $C \leq K$ ）來表示原有的變異量





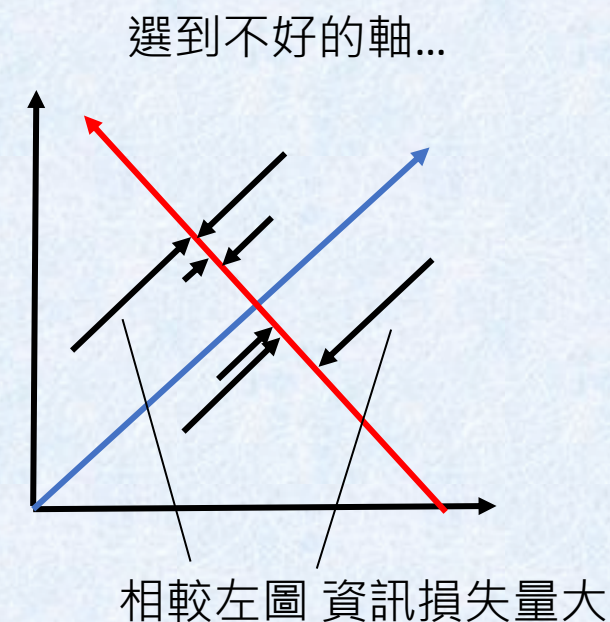
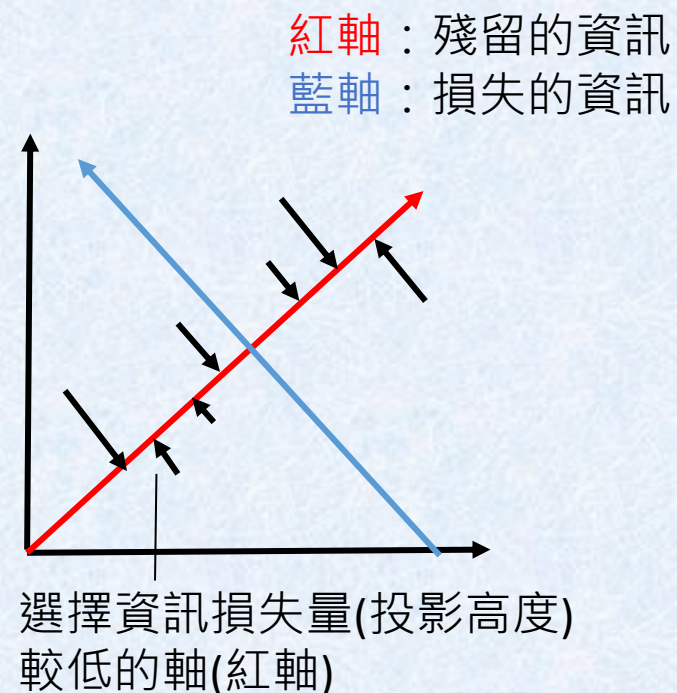
主成分分析法 - 流程

1. 將輸入資料正規化，使得每個屬性落在相同的值域，**此步驟能確保值域大的屬性，不會去支配值域小的屬性**。
2. PCA計算k個單範正交（orthonormal），作為正規化資料輸入的基底，這些是單位向量稱為主成分（principal component），使得每一向量都是垂直於其他向量。
3. 將這些主成分按照其重要性來排序，並作為資料的新座標，提供關於變異量（variance）的重要訊息；也就是說，對於排序過的座標軸，**第一個座標顯示資料集最大的變異量，第二個座標顯示第二高的變異量**
4. 由於這些主成分是根據「**重要度**」遞減排序，所以可以剔除不重要的主成分來精簡資料。



主成分分析法 - 視覺化解釋

- 下圖為沿著最為分散的方向取紅軸，沿著不太分散的方向取藍軸，由外觀可知，**紅軸的投影高度比較低**、**藍軸的投影高度較高**。

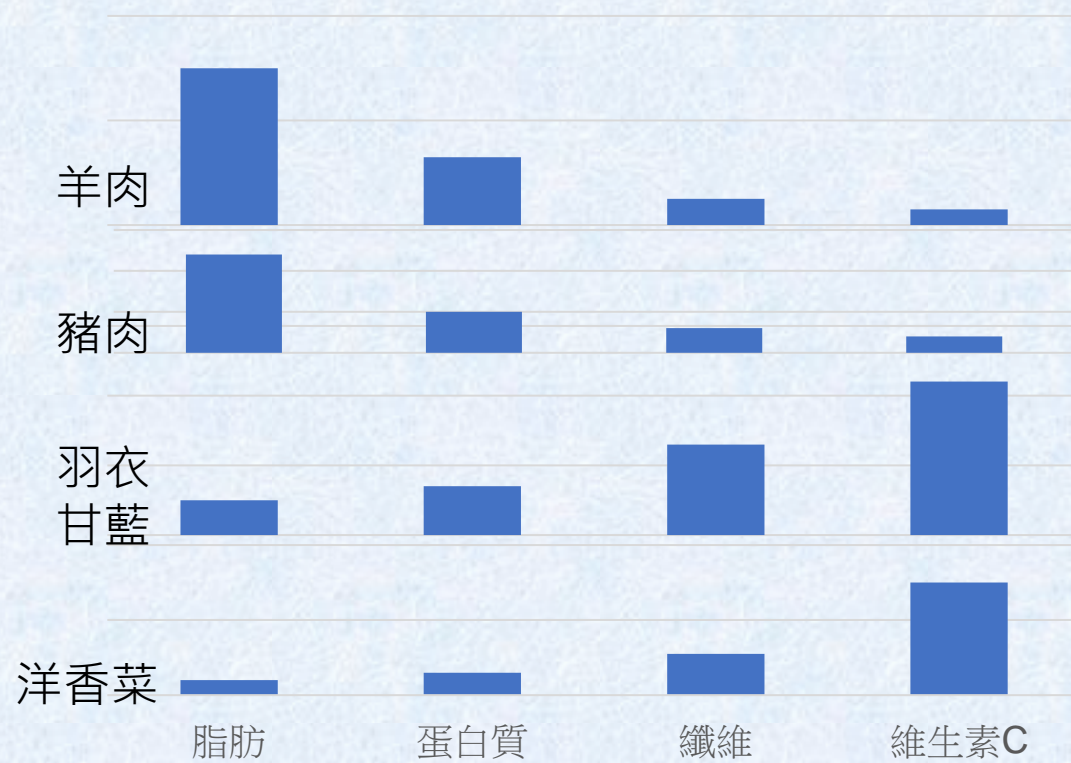




Case Study：食物營養含量分析

- 主成分分析利用**合併高度相關的變數**達到**降維**的效果，在最主要的主成分中可得出一個結論：脂肪、蛋白質和纖維、維生素C各成一對

$$0.55(\text{纖維}) + 0.44(\text{維生素}) - 0.45(\text{脂肪}) - 0.55(\text{蛋白質})$$



不同食物營養程度比較圖

營養變數的加權組合，同一主成分中粉色區塊代表同向加權的變數

	第一主成分(PC1)	第二主成分(PC2)	第三主成分(PC3)	第四主成分(PC4)
脂肪	-0.45	0.66	0.58	0.18
蛋白質	-0.55	0.21	-0.46	-0.67
纖維	0.55	0.19	0.43	-0.69
維生素C	0.44	0.70	-0.52	0.22

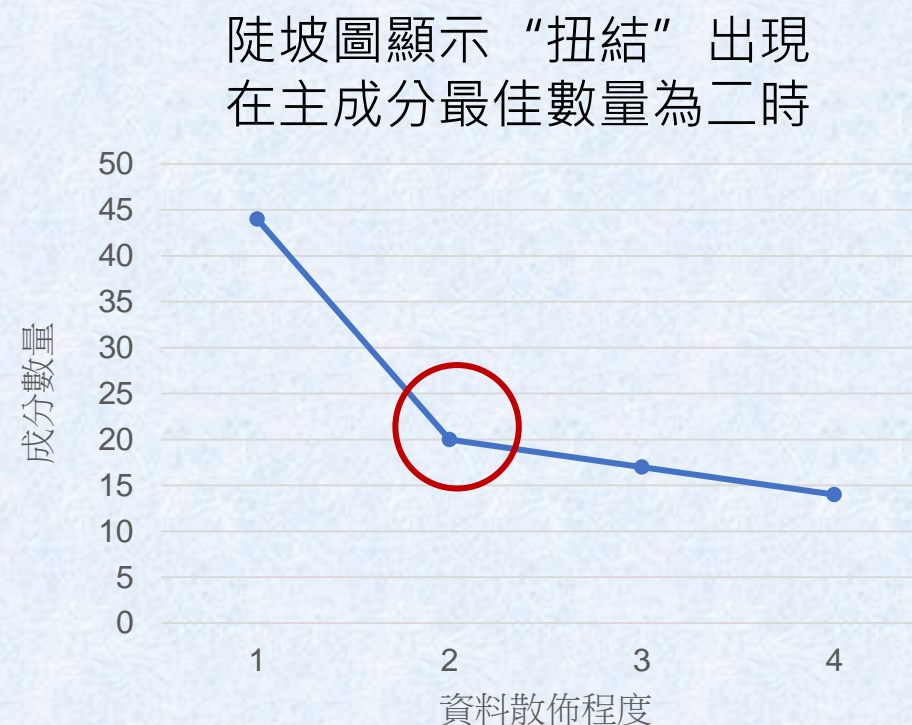


Case Study：食物營養含量分析

- 在第一個主成分(PC1)區分了肉類和蔬菜的資料，在第二主成分(PC2)中以脂肪含量區分肉類，以維生素C含量區分各蔬菜
- 陡坡圖可顯示主成分能區分資料點的有效性，而「扭結」正是陡坡圖中急遽彎曲處，使用扭結對應的主成分數量進行分析可得到最佳效果



以前兩個主要成分繪製的食物品項圖





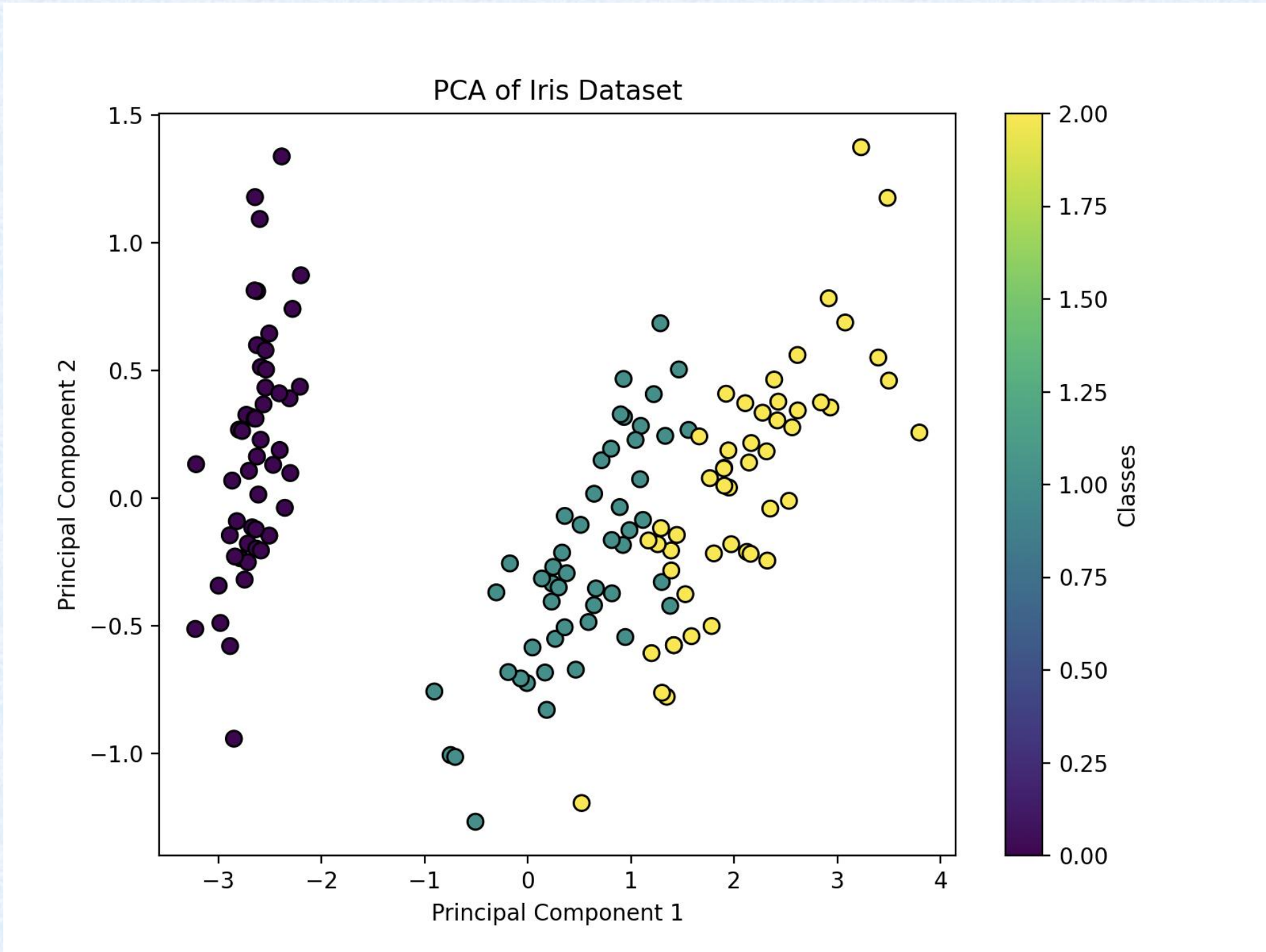
Python實作-PCA



```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from sklearn.decomposition import PCA
4  from sklearn.datasets import load_iris
5
6  # 載入資料集
7  data = load_iris()
8  X = data.data # 特徵
9  y = data.target # 標籤
10
11 # 建立PCA模型，設定要降低到的維度數量
12 pca = PCA(n_components=2)
13 X_pca = pca.fit_transform(X)
14
15 # 顯示各主成分所佔的變異量比例
16 print("Explained variance ratio:", pca.explained_variance_ratio_)
17
18 # 繪製降維後的資料點
19 plt.figure(figsize=(8, 6))
20 scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis', edgecolor='k', s=50)
21 plt.xlabel('Principal Component 1')
22 plt.ylabel('Principal Component 2')
23 plt.title('PCA of Iris Dataset')
24 plt.colorbar(scatter, label='Classes')
25 plt.show()
```



Python實作-PCA





t-隨機鄰近嵌入法

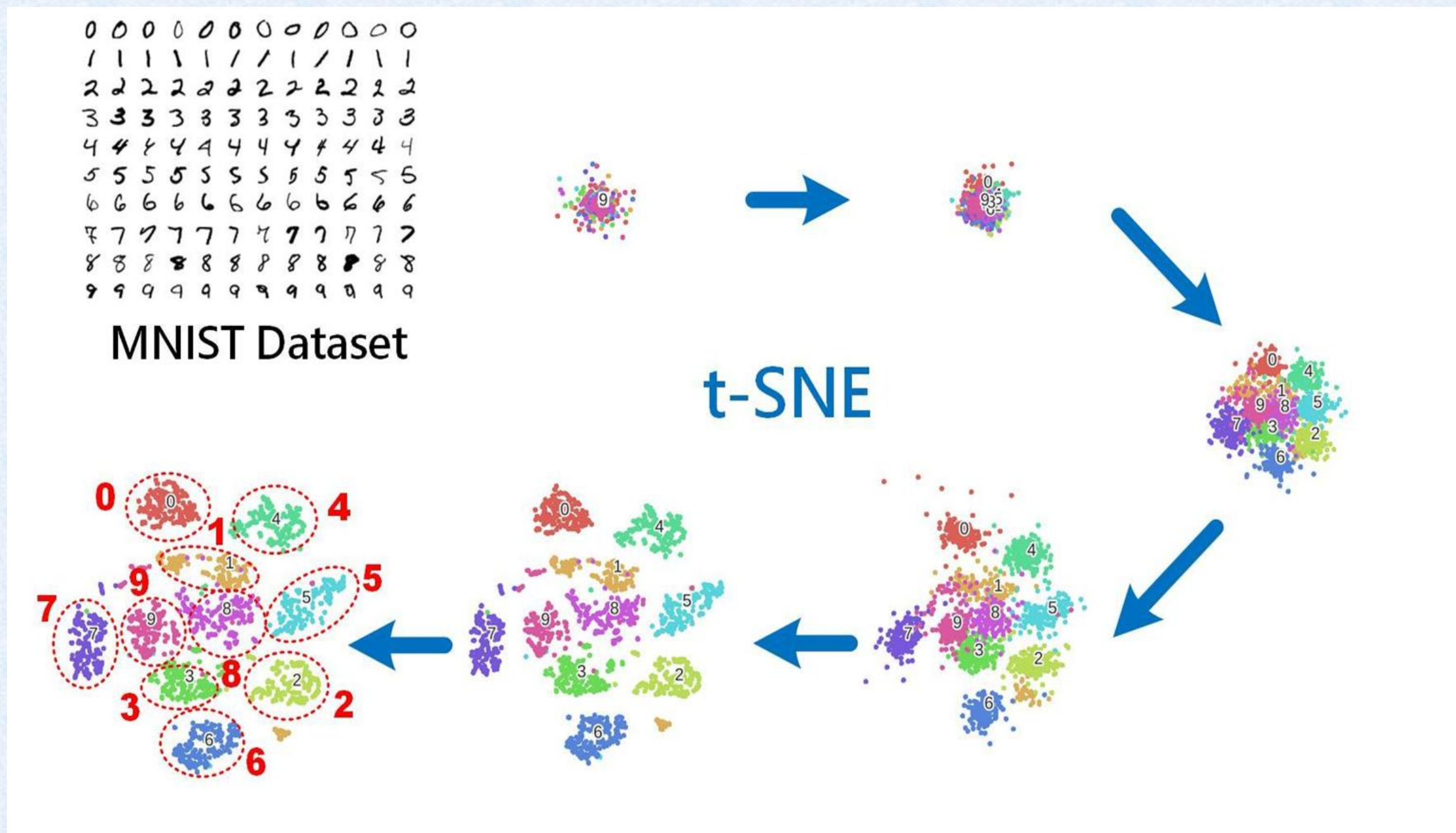
(t-distributed Stochastic Neighbor Embedding, t-SNE)

- 應用上，t-SNE 常用來將高維度的資料進行低維度的轉換達成視覺化，通過視覺化直觀的驗證某資料集或演算法的有效性。
- 求算實際資料 (P) 與理論資料 (Q) 間分布的相似度，經常用 **KL 散度 (Kullback-Leibler Divergence)** 來表示，也叫做**相對熵 (Relative Entropy)**，兩者差異越大則 **KL 散度越大**。而SNE 使用**條件機率和高斯分佈**來定義高維和低維中樣本點之間的相似度，並用 **KL 散度**來衡量兩條件機率分佈總和之間的相似度，並將其作為價值函數以**梯度下降法**求解。
- t-SNE 使用 **t 分佈**定義低維時的機率分佈來減緩**維數災難 (curse of dimensionality)**造成的**擁擠問題 (crowding problem)**。



t-SNE 流程

MNIST 資料集經 t-SNE 最佳化過程。





Python 實作 t-SNE

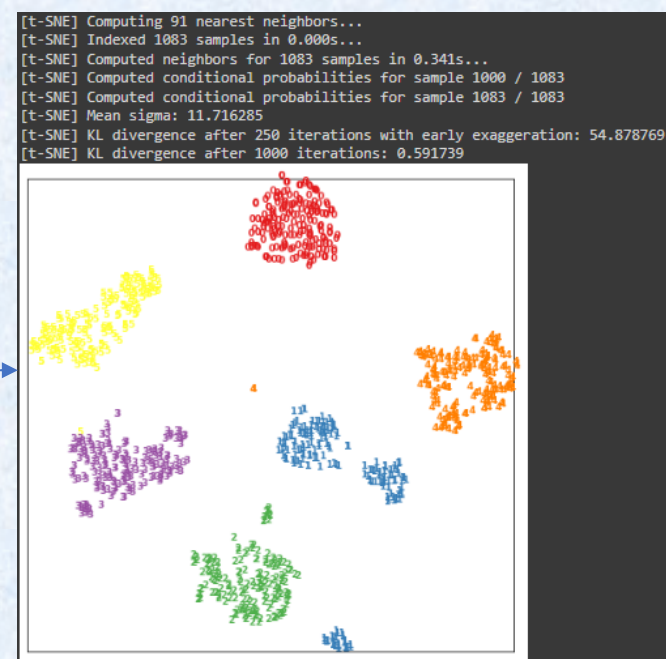
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn import manifold, datasets
4 #Prepare the data
5 digits = datasets.load_digits(n_class=6)
6 X, y = digits.data, digits.target
7 n_samples, n_features = X.shape
8 n = 20
9 img = np.zeros((10 * n, 10 * n))
10 for i in range(n):
11     ix = 10 * i + 1
12     for j in range(n):
13         iy = 10 * j + 1
14         img[ix:ix + 8, iy:iy + 8] = X[i * n + j].reshape((8, 8))
15 plt.figure(figsize=(8, 8))
16 plt.imshow(img, cmap=plt.cm.binary)
17 plt.xticks([])
18 plt.yticks([])
19 plt.show()
```

載入資料集
並視覺化呈現



```
21 #t-SNE
22 X_tsne = manifold.TSNE(n_components=2, init='random', random_state=5, verbose=1).fit_transform(X)
#Data Visualization
x_min, x_max = X_tsne.min(0), X_tsne.max(0)
X_norm = (X_tsne - x_min) / (x_max - x_min) #Normalize
plt.figure(figsize=(8, 8))
for i in range(X_norm.shape[0]):
    plt.text(X_norm[i, 0], X_norm[i, 1], str(y[i]), color=plt.cm.Set1(y[i]),
            fontdict={'weight': 'bold', 'size': 9})
plt.xticks([])
plt.yticks([])
plt.show()
```

執行 t-SNE 降維
以圖表方式呈現





Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Dimensionality Reduction Techniques
 - Non-numerical
 - Numerical
 - Numerosity reduction
 - Data compression



熵/亂度 Entropy

- 用來計算某一系統當中的失序情形，為一個描述性的函數且**經常使用參考值以及變化量來進行比較與分析**
- 例如：當每一種資訊結果發生的機率越平均的時候，我們所求得
的資訊量也就越大，因此資訊量就可以被視為是熵的指標，當**資訊
量越大的時候也表示亂度是越大**的。
- 可先將資料分為多個區間，若區間合併後亂度有下降，則考慮將該
區間合併，直到所有定義的區間均檢測完或合併區間後亂度不再下
降時為止。
- E值即亂度：
$$E = -\sum_{i=1}^{N-1} \sum_{j=i+1}^N (S_{ij} \times \log S_{ij}) + ((1 - S_{ij}) \times \log(1 - S_{ij}))$$



漢明距離 Hamming Distance

- 用來衡量非數值(Non-numerical)的資料的**相似度**
- 公式為：

$$s_{ij} = (\sum_{k=1}^n |x_{ik} - x_{jk}|) / n$$



相似度(Similarity)及不相似度(Dissimilarity)

- **鄰近值(Proximity)**：來表示相似度與不相似度
- 相似度
 - 相似度表示物件間相同的程度
 - 物件之間的**相似度愈高，其物件愈相像**
 - 其值大部分介於0 ~ 1之間
- 不相似度
 - 不相似度表示兩個物件間差異的程度
 - 不相似度和距離其實是同義字，**距離愈大，不相似度愈高**
 - 其值大部分介於0 ~ 1之間，但有時其範圍可到無限



相似度(Similarity)及不相似度(Dissimilarity)

- 下表是各種屬性型態的不相似度及相似度之計算方法，其中兩個物件 x 與 y ，各有一個屬性，而 $d(x,y)$ 與 $s(x,y)$ 分別表示不相似度及相似度

屬性型態	不相似度	相似度
名目	$d = \begin{cases} 0 & \text{若 } x = y \\ 1 & \text{若 } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{若 } x = y \\ 0 & \text{若 } x \neq y \end{cases}$
順序	$d = x - y / (n - 1)$ (將值對映至整數 $0 \sim n-1$ 之值，其中 n 為數值)	$s = 1 - d$
區間或比例	$d = x - y $	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$



漢明距離 Hamming Distance

特性減量 – 漢明距離範例

表 3.2 具有 3 個特性值的 5 個資料

樣本	特性 1	特性 2	特性 3
R1	A	X	1
R2	B	Y	2
R3	C	Y	2
R4	B	X	1
R5	C	Z	3



漢明距離 Hamming Distance

- 漢明距離：計算相似度

表 3.2 具有 3 個特性值的 5 個資料

樣本	特性 1	特性 2	特性 3
R1	A	X	1
R2	B	Y	2
R3	C	Y	2
R4	B	X	1
R5	C	Z	3



表 3.3 樣本之間的相似度

	R1	R2	R3	R4	R5
R1		0/3	0/3	2/3	0/3
R2			2/3	1/3	0/3
R3				0/3	1/3
R4					0/3



資訊理論 Information Theory

- Information Theory中的Entropy概念，最早由**Claude Shannon**在**1948年的論文**所提出
- 假設一個事件有n種結果，發生的機率分別為 $P(V_1), \dots, P(V_n)$ ，這些機率都是已知的，則定義這個事件發生後所得到的資訊量為：

$$I(P(\mathbf{v}_1), \dots, P(\mathbf{v}_n)) = \sum_{i=1}^n -P(\mathbf{v}_i) \log_2 P(\mathbf{v}_i)$$

- 各種結果**發生機率愈平均**，所求**資訊量也愈大**
- 資訊量可以當作亂度 (Entropy) 的指標，**資訊量愈大，表示亂度愈大**
- 解決**屬性選擇**的問題



漢明距離：計算原始資訊量(H0)

- 接著，利用「資訊理論」來計算資訊量（亂度）。
- 請注意，機率的總合為 1，因此，只有漢明距離是代入不了公式的。
- 漢明算出來的機率僅為「相似度」，因此，還需加上「不相似度」帶來的資訊量（亂度）。也就是說，資訊量的總和是這些正例負例加總而來的。
- 如果計算時出現 $\log(0)$ ，代表他一定不發生（沒有資訊價值），亂度 = 0

樣本	特性 1	特性 2	特性 3		R1	R2	R3	R4	R5
R1	A	X	1		R1	0/3	0/3	2/3	0/3
R2	B	Y	2		R2		2/3	1/3	0/3
R3	C	Y	2		R3			0/3	1/3
R4	B	X	1		R4				0/3
R5	C	Z	3						

→

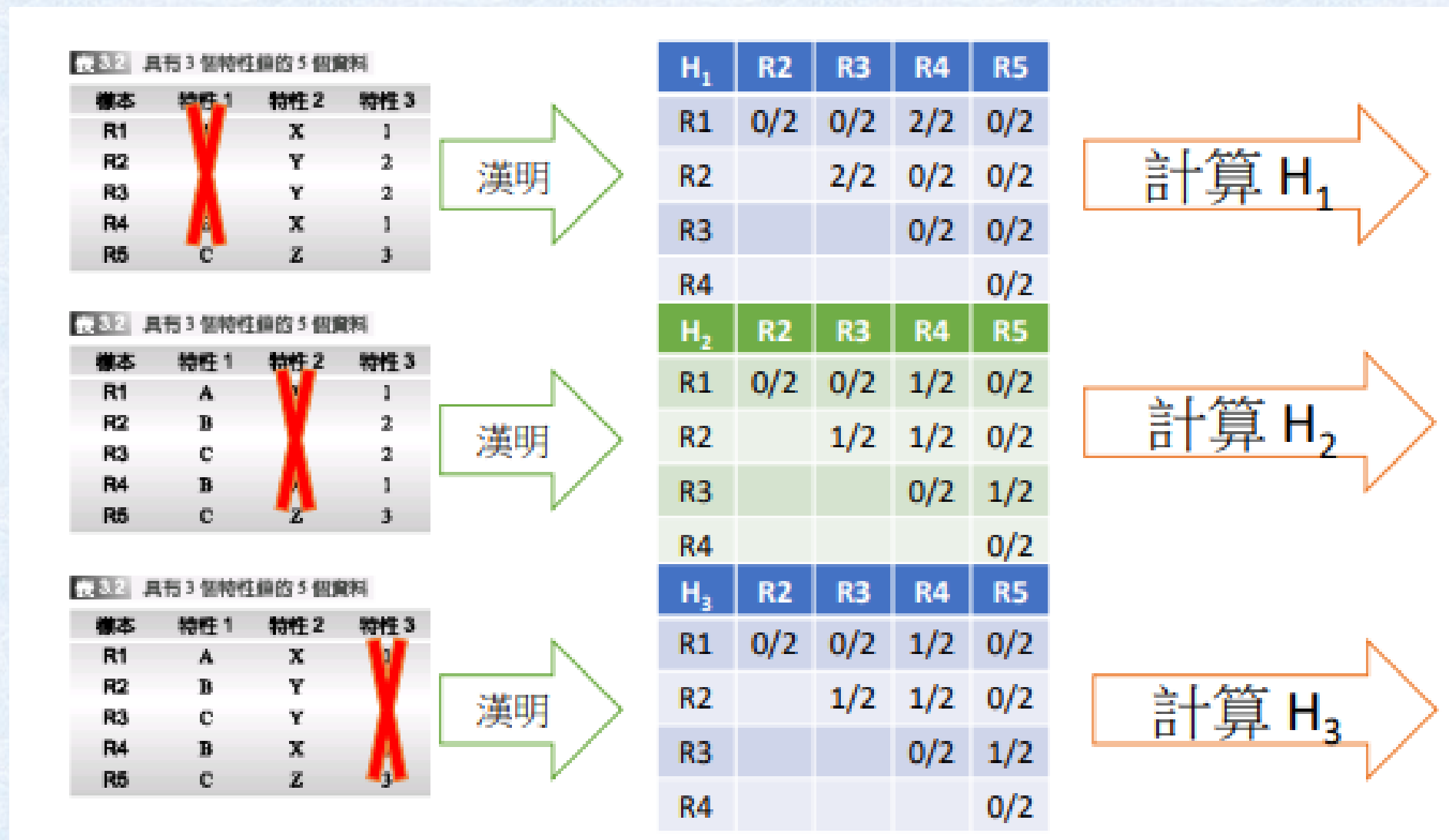
• $-\left[\frac{0}{3} * \log_2 \frac{0}{3} + \left(1 - \frac{0}{3}\right) * \log_2 \left(1 - \frac{0}{3}\right)\right] = 0$

$$E = -\sum_{i=1}^{N-1} \sum_{j=i+1}^N (S_{ij} \times \log S_{ij}) + ((1 - S_{ij}) \times \log(1 - S_{ij}))$$



漢明距離計算方式

- 漢明距離：**計算個別移除特徵的資訊量**（也就是前述 Step-wise backward elimination 的方法）





Python實作-漢明距離

- 漢明距離：計算資訊量

```
from math import log
P0 = [0/3,0/3,2/3,0/3,2/3,1/3,0/3,0/3,1/3,0/3 ]
P1 = [0/2,0/2,2/2,0/2,2/2,0/2,0/2,0/2,0/2,0/2 ]
P2 = [0/2,0/2,1/2,0/2,1/2,1/2,0/2,0/2,1/2,0/2 ]
P3 = [0/2,0/2,1/2,0/2,1/2,1/2,0/2,0/2,1/2,0/2]

def H(arr):
    info = 0
    for p in arr:
        if(p!=0 and p!=1): # 機率為1或0都不具資訊價值
            info += -p*log(p,2)-(1-p)*log((1-p),2) # 資訊理論公式
    return round(info,2)
print("H0:{}, H1:{}, H2:{}, H3:{}".format(H(P0),H(P1),H(P2),H(P3) ))
```

H0:3.67, H1:0, H2:4.0, H3:4.0

在輸入程式的時候，請務必注意def, for, if 程式當中的階層關係



Python實作-漢明距離

- 利用結果做 Dimension Reduction :
 - 從結果：H0:3.67, H1:0, H2:4.0, H3:4.0 可發現「特徵1」移除之後，**資訊量（亂度）變化量最大**，也就是說，**此特徵為最重要的特徵！**
 - 此外，亂度掉到0，也就是說，所有的資訊都是由「特徵1」所提供，其他特徵可以拿掉。
 - 這邊的資訊，指的是 sample 之間相似與否的資訊。可觀察當我們拿掉「特徵1」之後，其他特徵資料「不是完全相同，就是完全不同」，**這在判斷相似與否這件事情上，沒有提供任何資訊。**



Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Dimensionality Reduction Techniques
 - Non-numerical
 - Numerical
 - Numerosity reduction
 - Data compression



常用的距離公式

- Euclidean

$$\text{dist}(p, q) = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

- Manhattan

$$\text{dist}(p, q) = \sum_{k=1}^n |p_k - q_k|$$

- Chebyshev

$$\max(|x_2 - x_1|, |y_2 - y_1|)$$

- Minkowski

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{1/r}$$

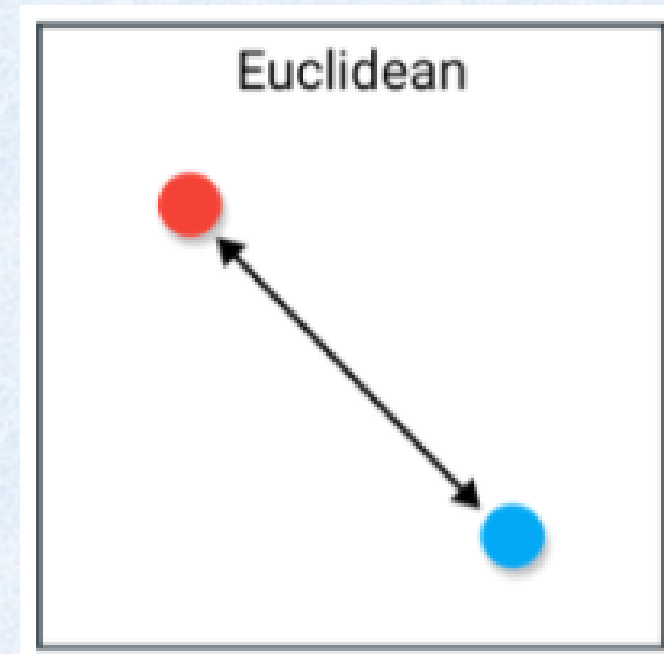
「距離公式」
常在數值型資料當中，用來
求出兩個數值
之間的關係。

亦常來計算非
數值型的距離
公式



歐幾里德距離 Euclidean

- 優點：
 - 在**低維數據中效果很好**。
 - 當今**最常用的距離度量**之一。



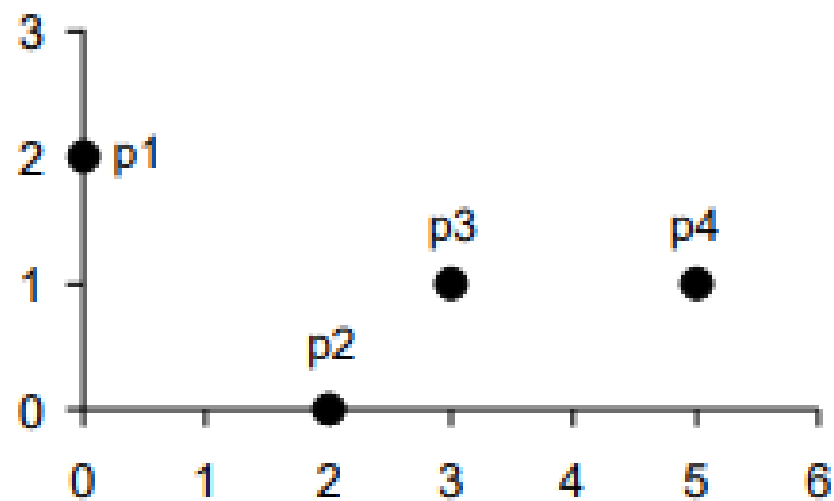
- 缺點：
 - 在使用此度量之前，需要對傾斜的資料進行正規化。
 - 高維空間表現不佳。

- 公式：

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



歐幾里德距離 Euclidean



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

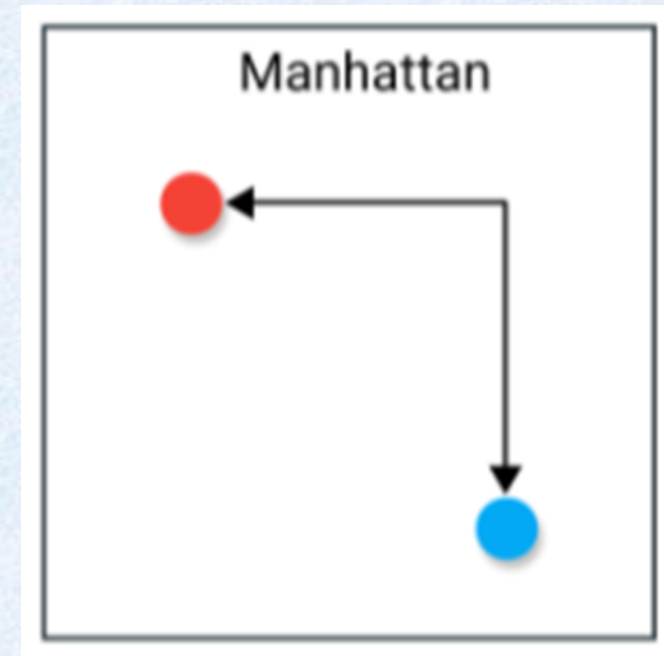
	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

距離矩陣 (Distance Matrix)



曼哈頓距離 Manhattan Distance

- 用例：
 - 通常稱為計程車距離或城市街區距離，**計算實值向量之間的距離**。



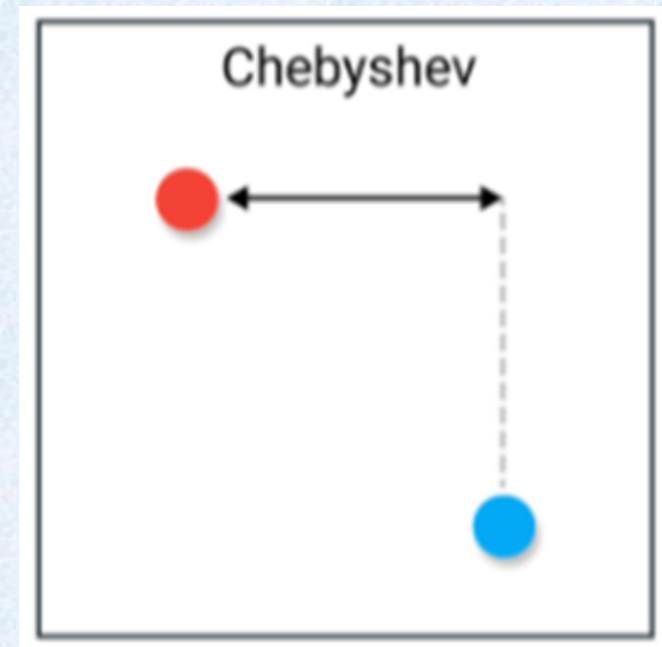
- 缺點：
 - 它更有可能給出比歐幾里德距離更高的距離值，因為它**可能不是的最短路徑**。

- 公式：
$$D(x, y) = \sum_{i=1}^k |x_i - y_i|$$



切比雪夫距離 Chebyshev Distance

- 用例：
 - 它用於提取要求從一個方格移動到另一個方格所需的**最少移動次數**。
- 缺點：
 - 它通常**用於非常特定的用例**，這使得它難以用作通用距離度量。

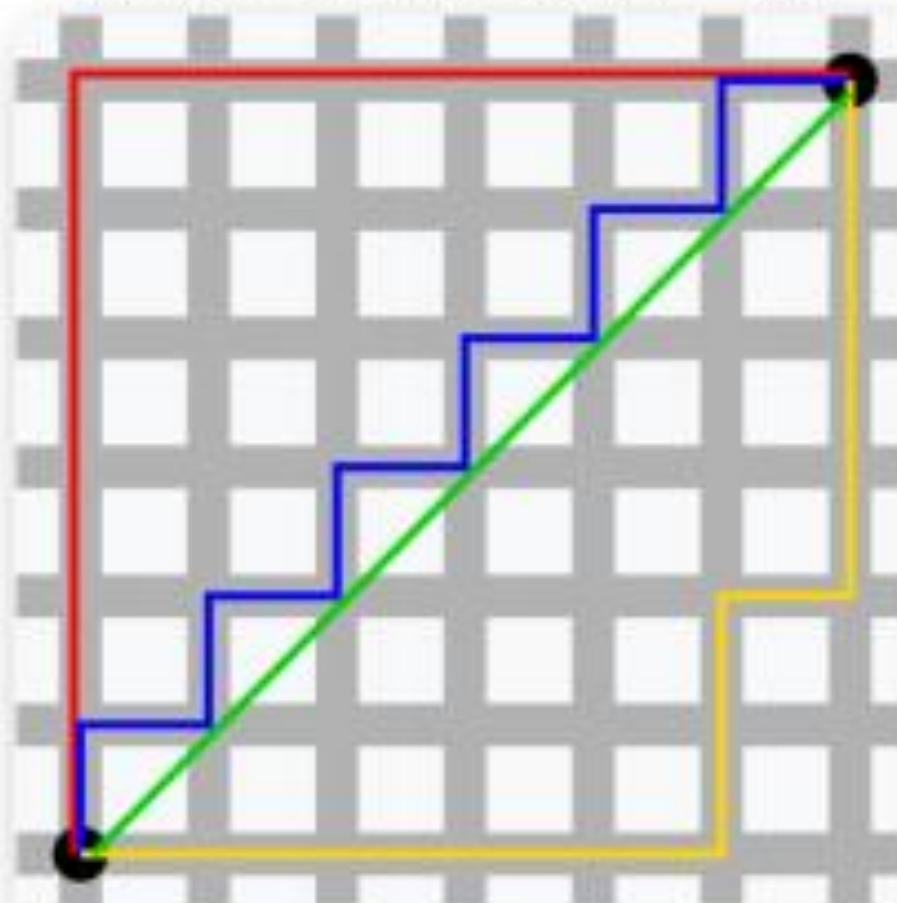


- 公式：
$$D(x, y) = \max_i (|x_i - y_i|)$$




曼哈頓距離與切比雪夫距離

所有曼哈頓距離都一樣



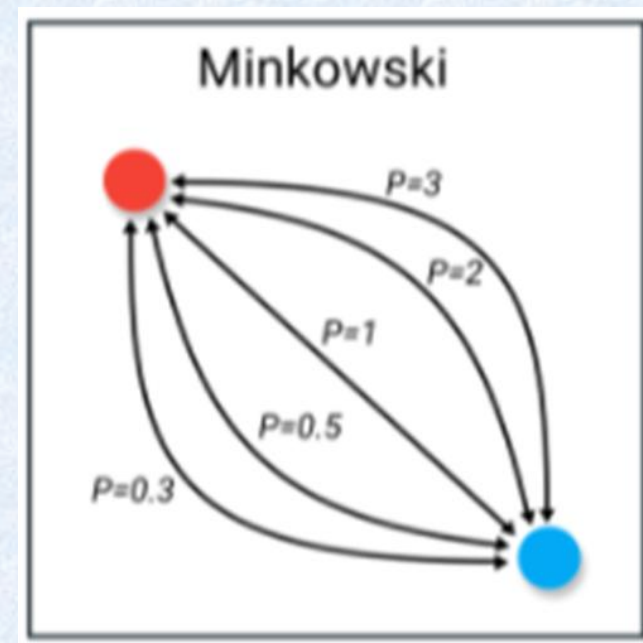
切比雪夫距離又稱為棋盤距離

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	



敏高斯基距離 Minkowski Distance

- 用例：
 - 它可以在距離能表示為**具有長度的向量的空間中**使用。
 - 該度量具有三個要求：**零向量 (Zero Vector)**、**比例係數 (Scalar Factor)**、**三角不等式 (Triangle Inequality)**



- 壞處：
 - 參數 p 實際上可能很難使用，因為找到正確的值在計算上可能非常低效。

- 公式：

$$D(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

$p=1$ — Manhattan distance

$p=2$ — Euclidean distance

$p=\infty$ — Chebyshev distance



敏高斯基距離 Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$D(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Where p is a parameter, n is the number of dimensions (attributes) and x_i and y_i are, respectively, the i^{th} attributes (components) or data objects x and y .



敏高斯基距離 Minkowski Distance

- $p = 1$. City block (Manhattan, taxicab, L1 norm) distance.
 - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $p = 2$. Euclidean distance(L2 norm)
- $p \rightarrow \infty$. “supremum” (Lmax norm, $L \infty$ norm) distance.
 - This is the maximum difference between any component of the vectors
- Do not confuse p with n , i.e., all these distances are defined for all numbers of dimensions.
- The proof process of formula



敏高斯基距離 Minkowski Distance

- p 不同即代表不同距離

Point	x	y
r1	0	2
r2	2	0
r3	3	1
r4	5	1

曼哈頓距離

歐氏距離

切比雪夫距離

$p=1$	r1	r2	r3	r4
r1	0	4	4	6
r2	4	0	2	4
r3	4	2	0	2
r4	6	4	2	0
$p=2$	r1	r2	r3	r4
r1	0	2.828	3.162	5.099
r2	2.828	0	3.162	5.099
r3	3.162	1.414	0	2
r4	5.099	3.162	2	0
$p=\infty$	r1	r2	r3	r4
r1	0	2	3	5
r2	2	0	1	3
r3	3	1	0	2
r4	5	3	2	0



Numerosity reduction



Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Numerosity reduction
 - Parametric
 - Non-parametric
 - Data compression



數量縮減 Numerosity Reduction

- Reduce data volume by **choosing alternative, smaller forms of data representation**
- 有母數 **Parametric** methods (e.g., regression)
 - Assuming the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
- 無母數 **Non-parametric** methods
 - Do not assume models
 - Major families: **histograms, clustering, sampling, ...**



Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Numerosity reduction
 - Parametric
 - Non-parametric
 - Data compression



聚合 Agglomerative Clustering

- 假設有一個記錄地區雨量標準差的資料，我們可以用聚合的觀念將**每個月降雨標準差、年降雨量標準差利用圖表呈現**，如此一來資料量就可以大幅降低，需選取的資料量

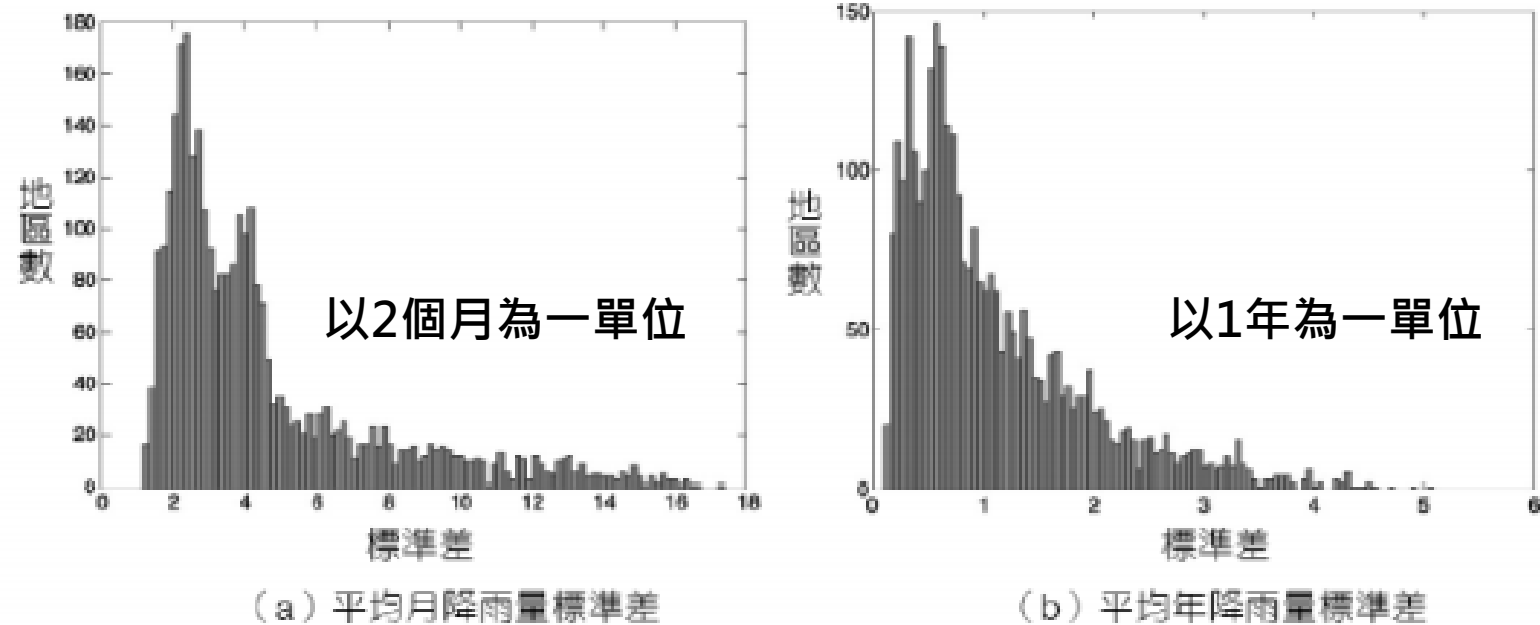
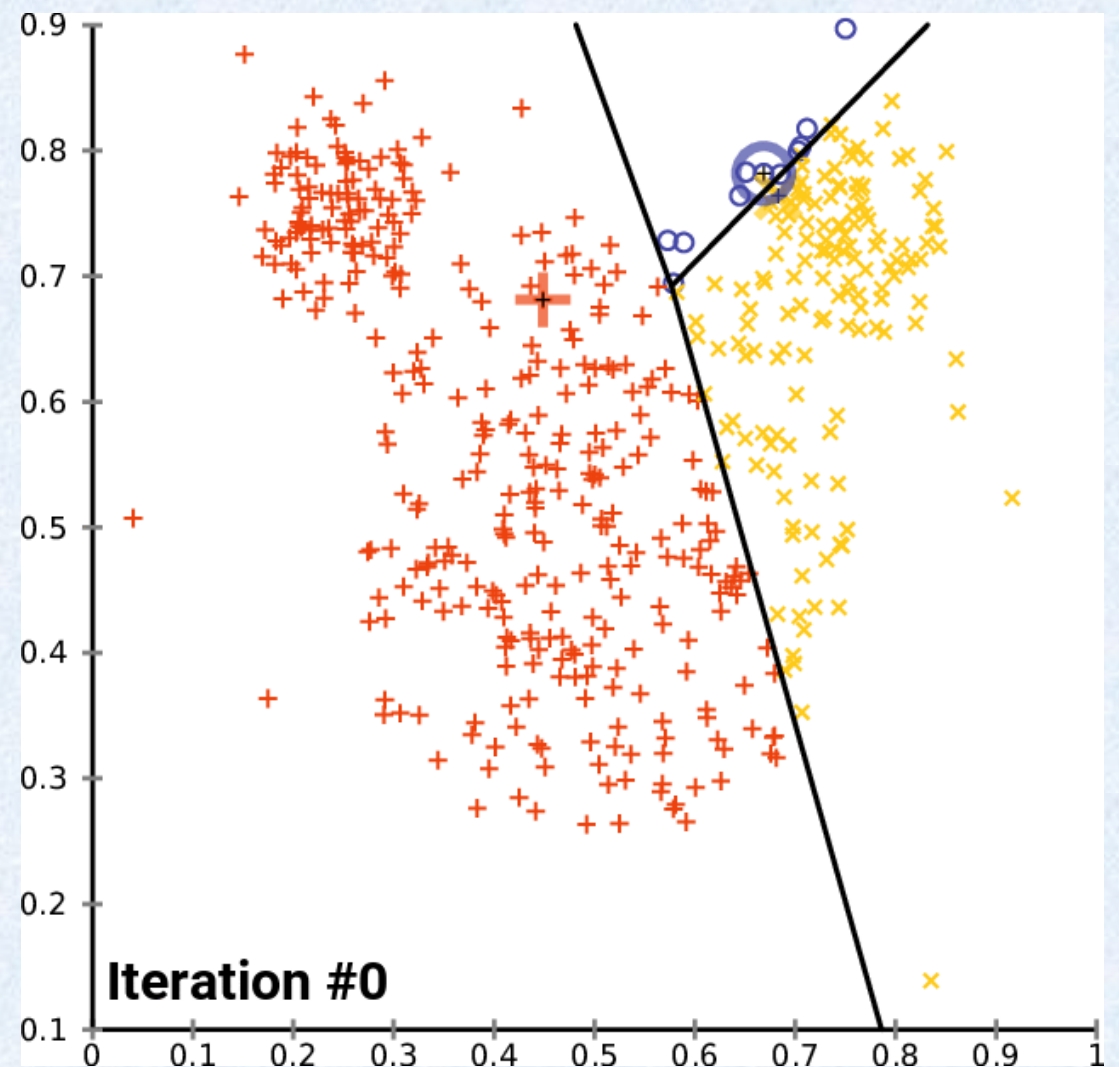


圖 2.8 ▶ 1982－1993 年澳洲降雨量的資料



聚類 Clustering

- **Partition data set into clusters** based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is **"smeared"**
- There are many choices of clustering definitions and clustering algorithms





抽樣 Sampling

- 抽樣是用來選取欲分析資料的主要技術
 - 通常用在資料調查及資料分析上
- **統計學**上的抽樣主要在於**要得到所有資料太過耗時**
- **資料探勘**的抽樣主要在於**計算的時間太過耗時**
- 有效的抽樣原則在於樣本必須是具有代表性
 - 抽樣的樣本所得到的結果會和整個原始資料的結果很接近
 - 如果某一個資料的平均數很接近整體資料的平均數，那麼就具有代表性



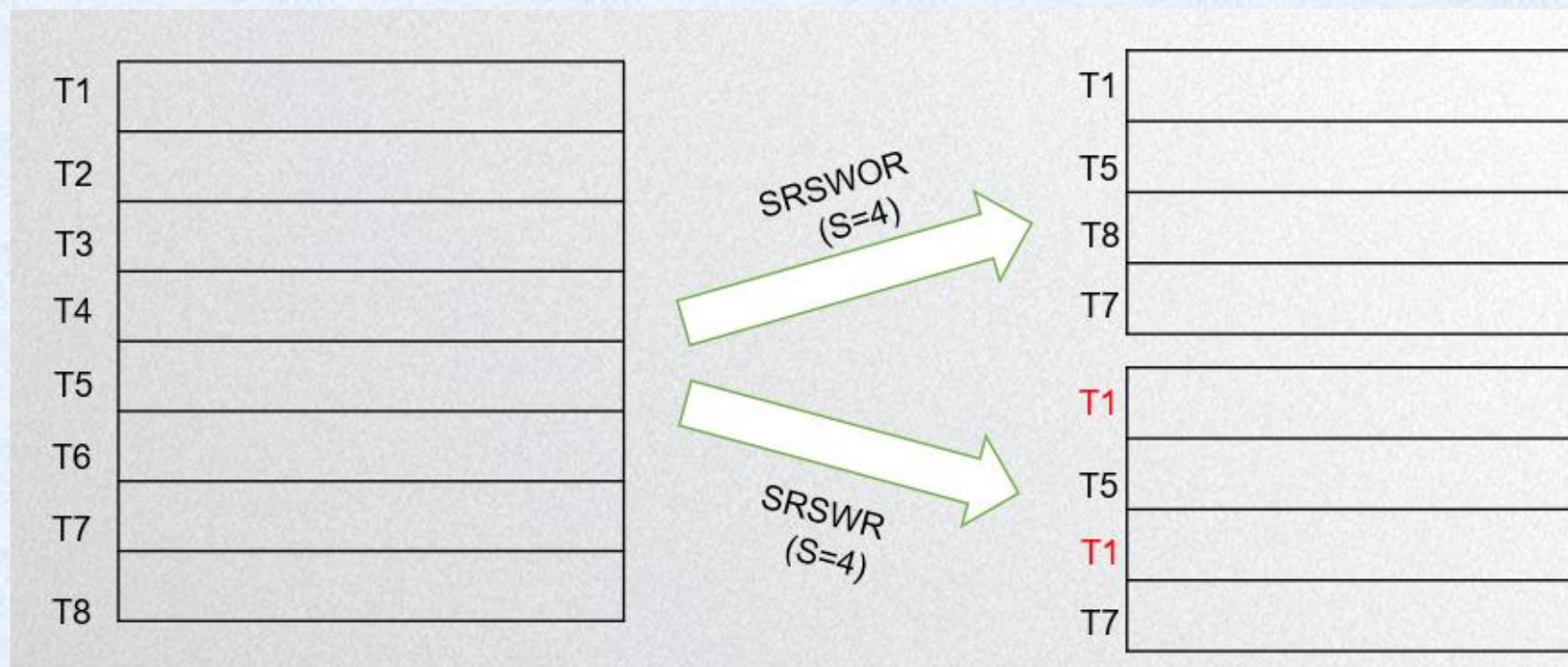
抽樣 Sampling

- **簡單隨機抽樣** (Simple Random Sampling)：每筆資料機率 **$1/N$**
- **分層抽樣** (Stratified Sampling)：不重疊分層，層間差異大，層內差異小。是先將**母群分為相關的層**，才在每層中隨機抽取樣本。
◦ Ex：性別、年齡...等
- **群集抽樣** (Cluster Sampling)：群間差異小，群內差異大。是將總體中各單位歸併成**若干個互不交叉、互不重複的集合**，稱之為**群(圈選的群體沒有特別性質)**。



簡單隨機抽樣 Simple Random Sampling

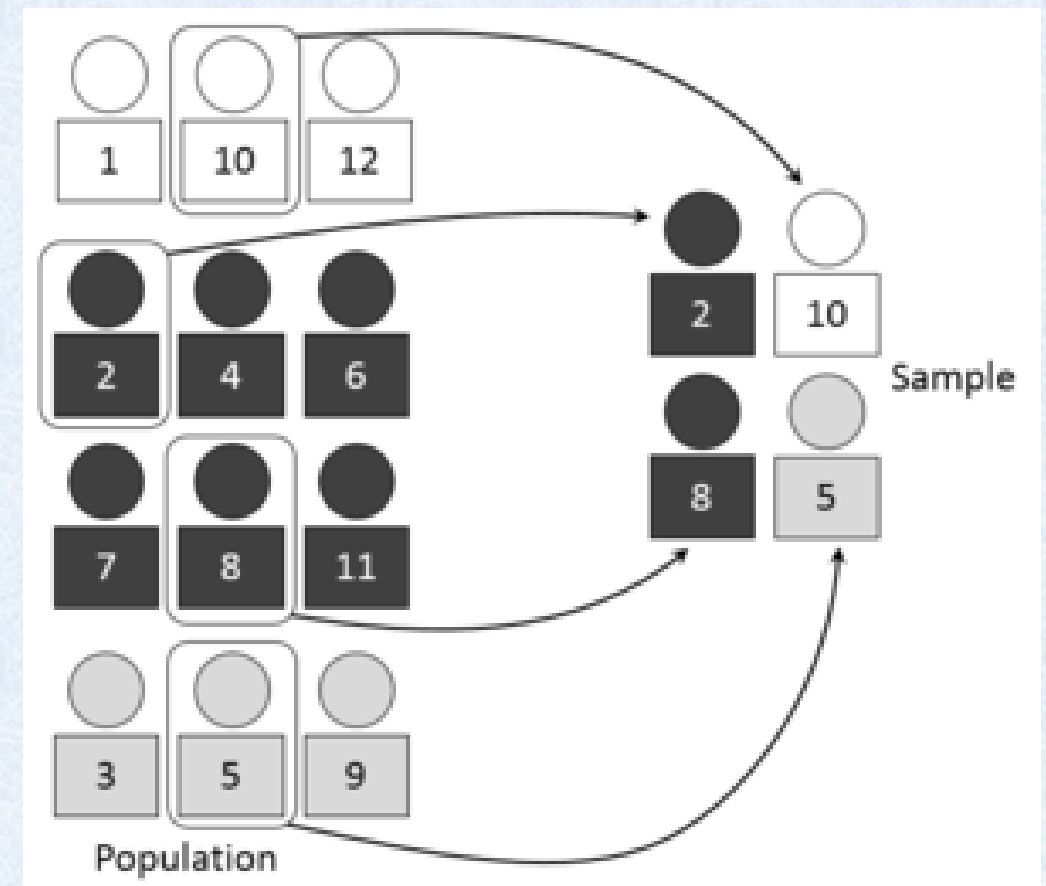
- 假設每一筆資料記錄具有相同機率會被抽出，若資料集合中有**N**筆資料，隨機抽取**S**個樣本，則每一筆被抽到的機率為 $1/N$ 。
- 每筆資料被抽到的機率相同
 - 放回式簡單隨機抽樣 (Simple random sample With replacement, SRSWR)
 - 不放回式簡單隨機抽樣 (Simple random sample Without replacement, SRSWOR)





分層隨機抽樣 Stratified Sampling

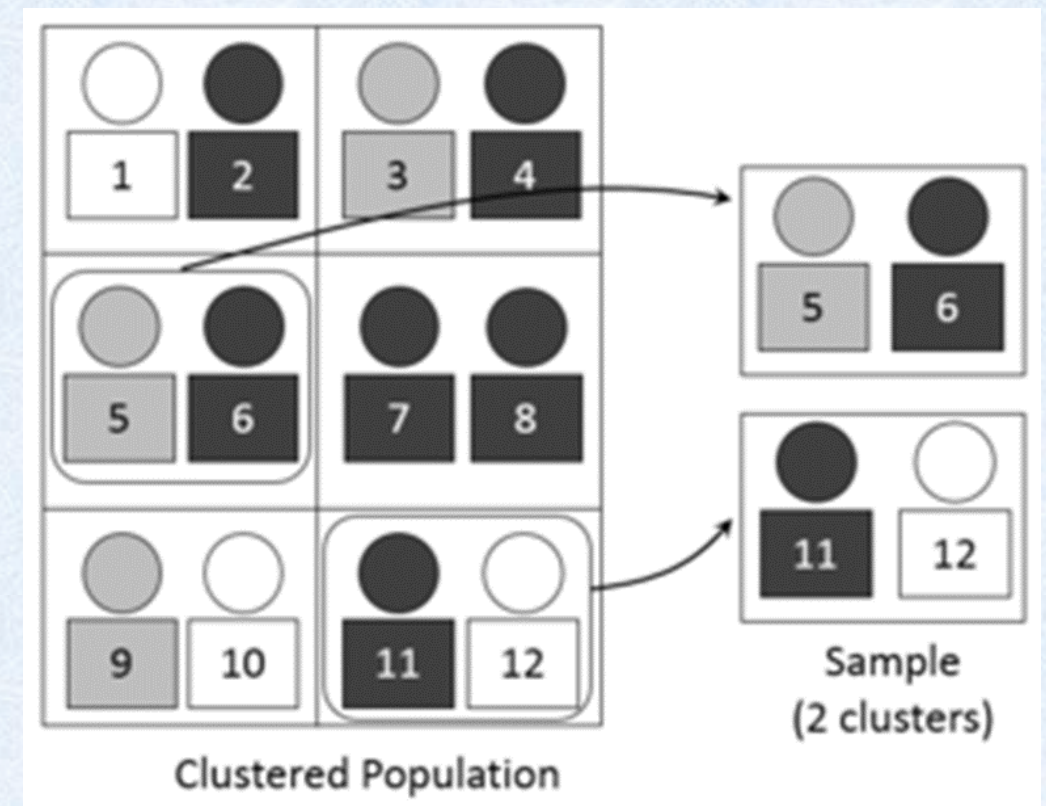
- 已知的資料特徵，將資料區分為數個不重疊的分層，**使層與層間的差異大，而層內的資料差異小**
- 樣本大小與各層資料比例，對各分層隨機抽取資料記錄
- **樣本較有代表性**，較不易失去過多的資料訊息





群集抽樣 Cluster Sampling

- 資料母體集合中依照已知標準或特徵所排列的集群作為抽樣單位，然後再依據要抽取的集群數量，選取抽樣集群中所有資料作為樣本
- 群間差異越小，則抽出的樣本越準確**
- 從所有群中**抽取部分群集而群內差異越大時減少抽樣樣本**





下採樣及上採樣

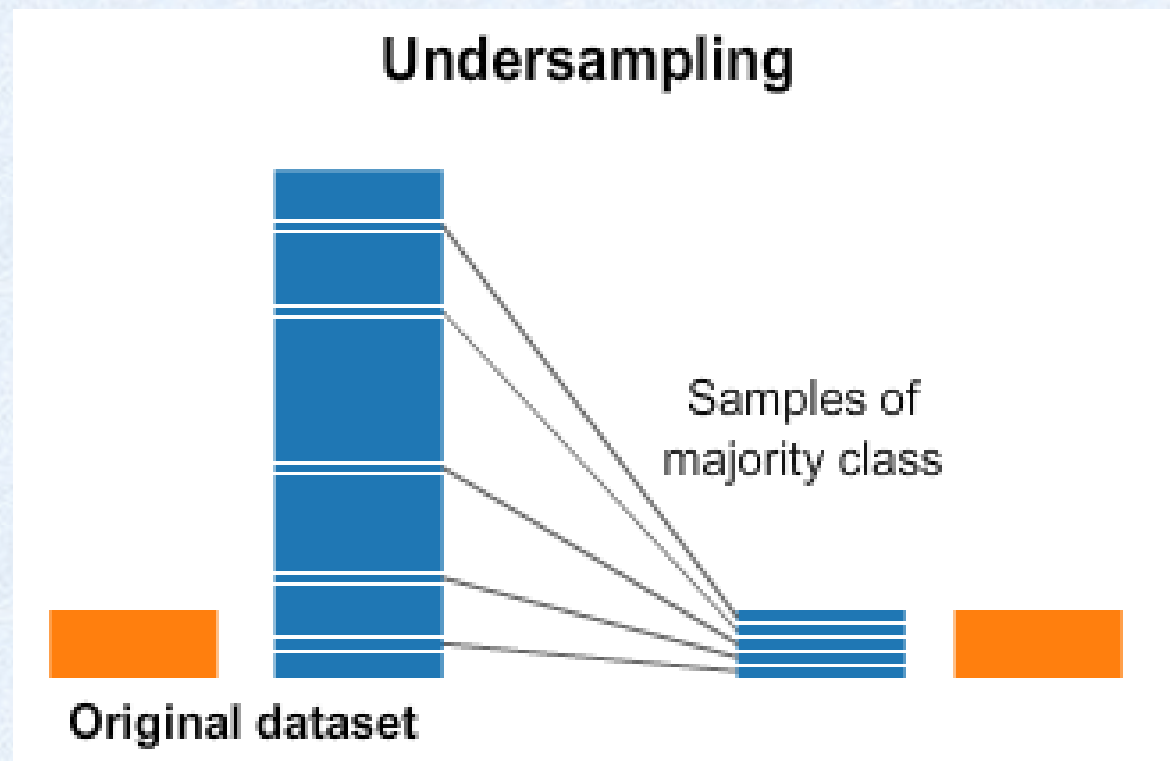
Under Sampling & Over sampling

- In contrast to stratified sampling, sometimes we would like a sample to contain different relative frequencies of the levels of a particular feature to the distribution in the original dataset.
- To do this, we can use **under-sampling** or **over-sampling**.



下採樣 Under Sampling

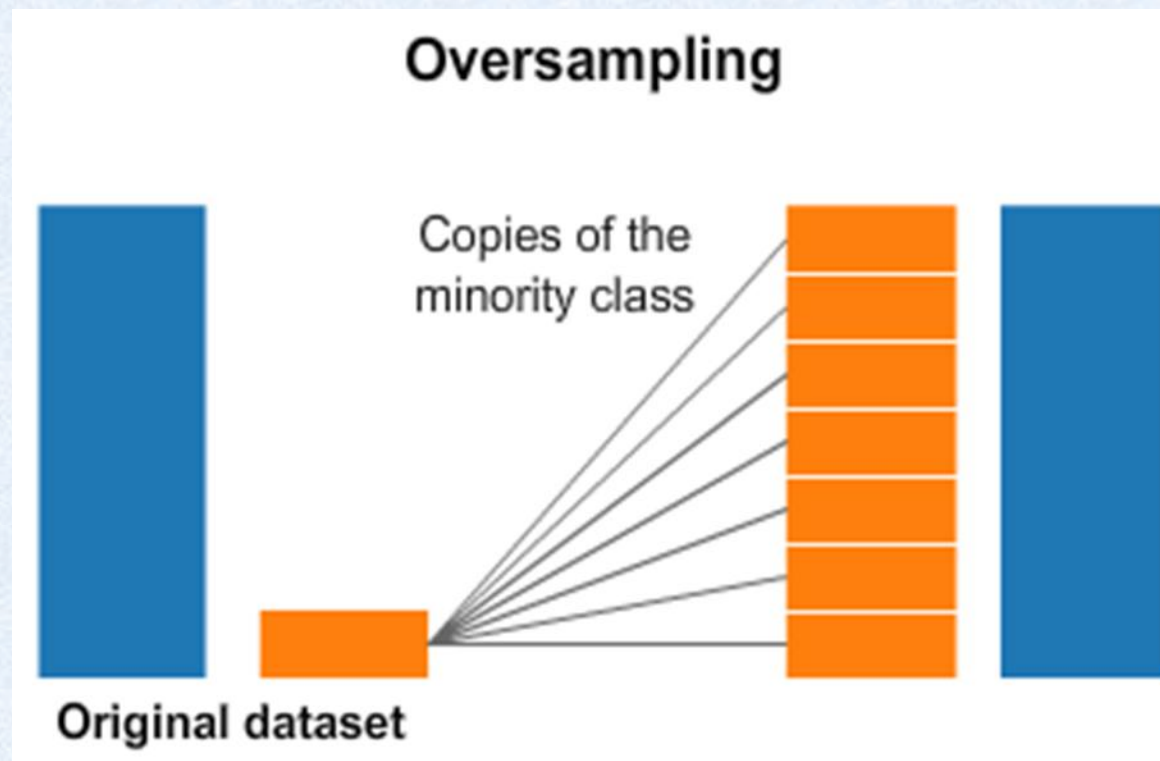
- 下採樣（Under Sampling）是通過減少樣本數，讓兩個類別的資料達成平衡。而在採樣的過程使用**隨機採樣法**來採樣
- 下採樣（Under Sampling）的**缺點**在於會刪除大量資料，可能會導致有重要特徵遭到刪除





上採樣 Over Sampling

- 上採樣（ Over-Sampling ）的**作法與下採樣相反**
- 上採樣通過**複製樣本數較少的資料**，進而讓「資料增量」。例如：SMOTE 方法
- 上採樣的缺點在於過度重複的資料可能會導致「**過度擬合**」(Overfitting)





Data compression



Tasks of Data Preprocessing

- 資料清理(Data cleaning)
 - Missing, Noisy, Inconsistent, Intentional, outliers, repetition.
- 資料整合(Data integration)
 - Schema integration, Entity identification problem, Different representations, different scales, Remove redundancies, Detect inconsistencies (chi squared, covariance, correlation)
- 資料轉換(Data transformation)
 - Normalization
 - discretization
 - Concept hierarchy generation
- 資料精簡(Data reduction)
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression

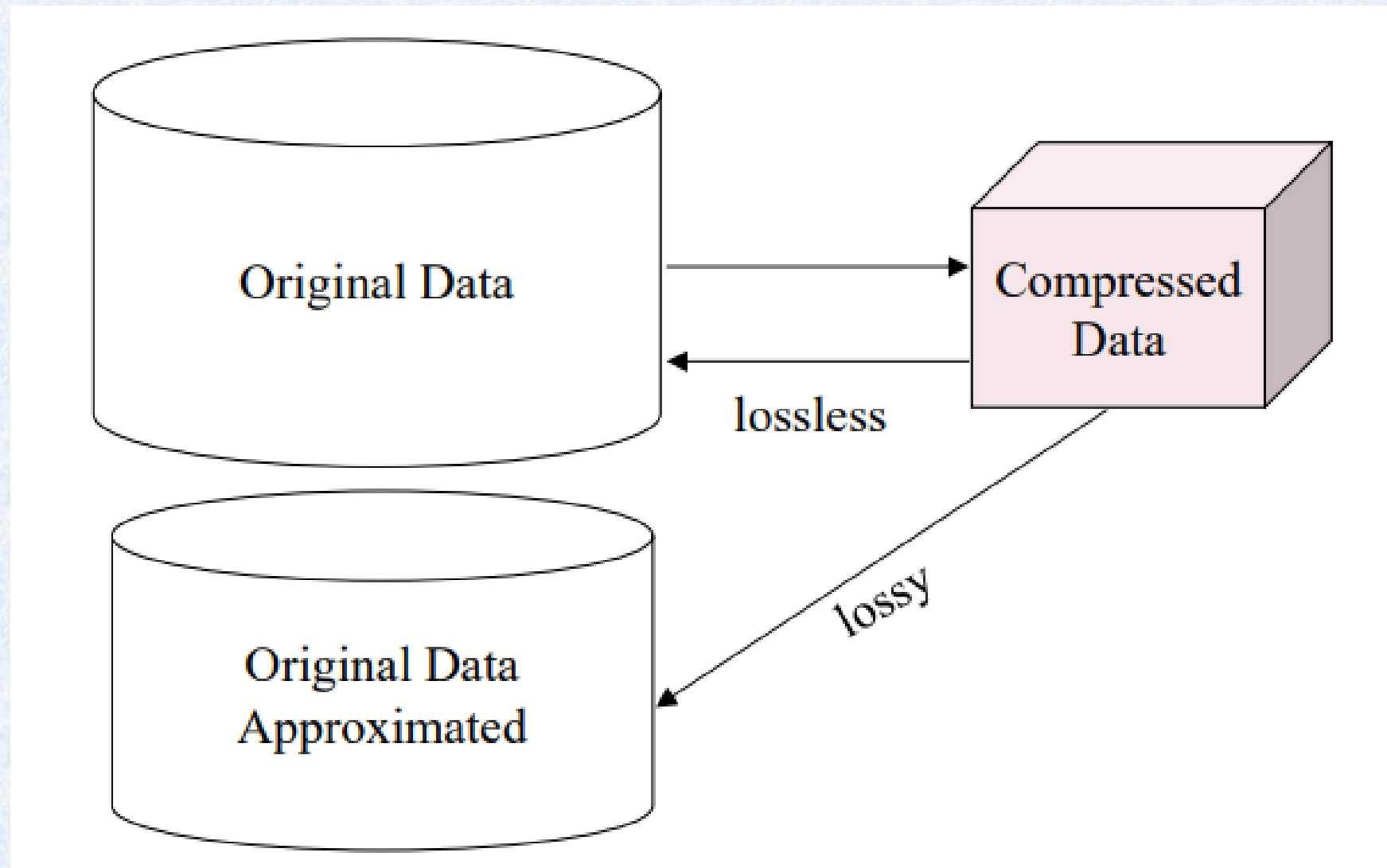


資料壓縮 Data compression

- 串流壓縮（字符串壓縮）
 - 有廣泛的理論和經過良好調整的算法
 - **通常是無損的**，但在不擴展的情況下只能進行有限的操作
- 影音/視頻壓縮（音視頻壓縮）
 - 典型的**有損壓縮**
 - 有時可以重建信號的小片段而不重建整個信號
- **「維度縮減」和「數量縮減」**也可以被視為資料壓縮的其中一種形式



資料壓縮 Data compression





Take a break...

