Kmeans的補充教材

- 經典四步驟
  - 載入模型
  - 建立模型
  - 訓練模型
  - 使用模型預測

```
# 1. 載入想要用的模型
from sklearn.cluster import KMeans
```

```
#2. 建立模型
clf = KMeans(n_clusters=3)
```

```
#3. 訓練模型
clf.fit(x)
```

```
#4. 使用模型來預測
clf.predict(y)
```

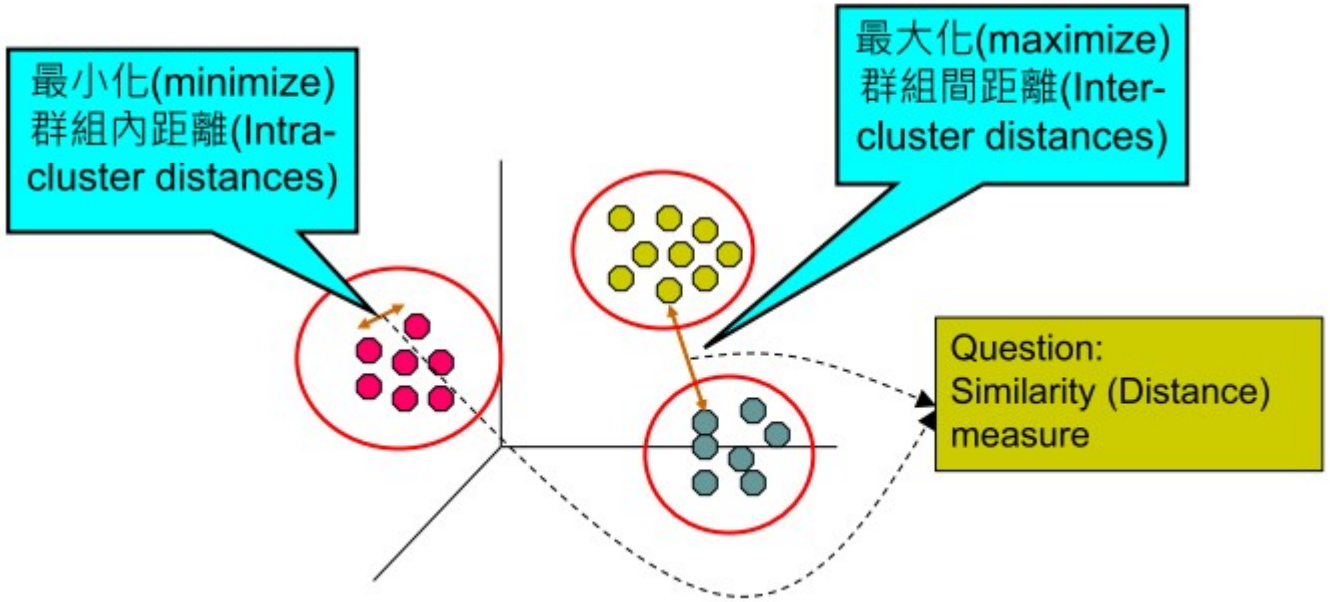
資料分群

- 基本概念

  給定一組資料(具有多個屬性)，將資料分成數個群組，使得
  - 在**同一群組**的資料**愈像愈好**
  - 在**不同群組**的資料**愈不像愈好**
- 關鍵：如何計算資料間的**相似度**(Similarity)
  - 最常用的相似度計算方式：歐幾里得距離(Euclidean Distance)；只適合數值欄位

資料分群

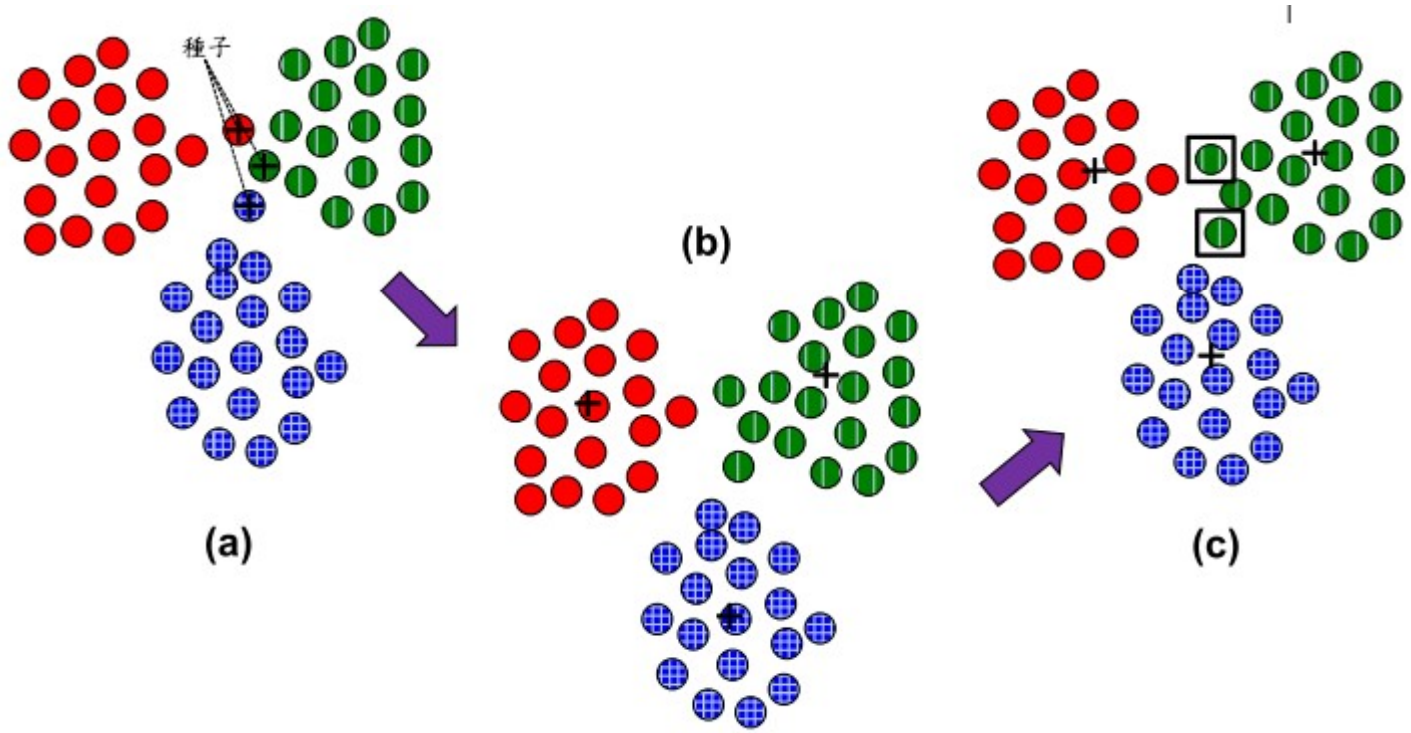
- 示意圖


最小化(minimize)
群組內距離(Intra-
cluster distances)
最大化(maximize)
群組間距離(Inter-
cluster distances)
Question:
Similarity (Distance)
measure

資料分群-Kmean為例

- 基本概念
  - 屬於切割型分群法
  - 每一群都以中心點(centroid)表示
  - 每一點都需分配給最靠近的中心點
  - 需要使用者設定分群的數目 *K*

資料分群-Kmean為例

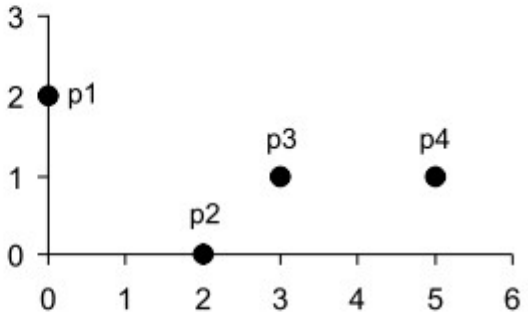


資料來源: 簡禎富、許嘉裕，資料挖礦與大數據分析，前程，2014。教材投影片

資料分群-Kmean為例

- 歐幾里得距離(最常用)

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



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

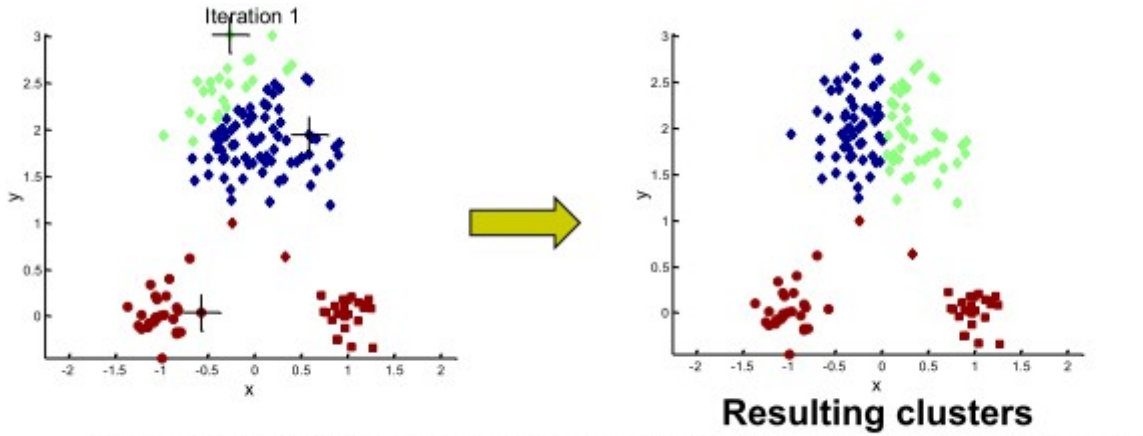
	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

資料分群-Kmean為例

- 群組中心初始值(seed)選取問題
  - 一般是隨機亂選，可能導致不佳的結果
  - 解決方法相當多，例如選距離最選的初始點(Python sk-learn中的K-means++)

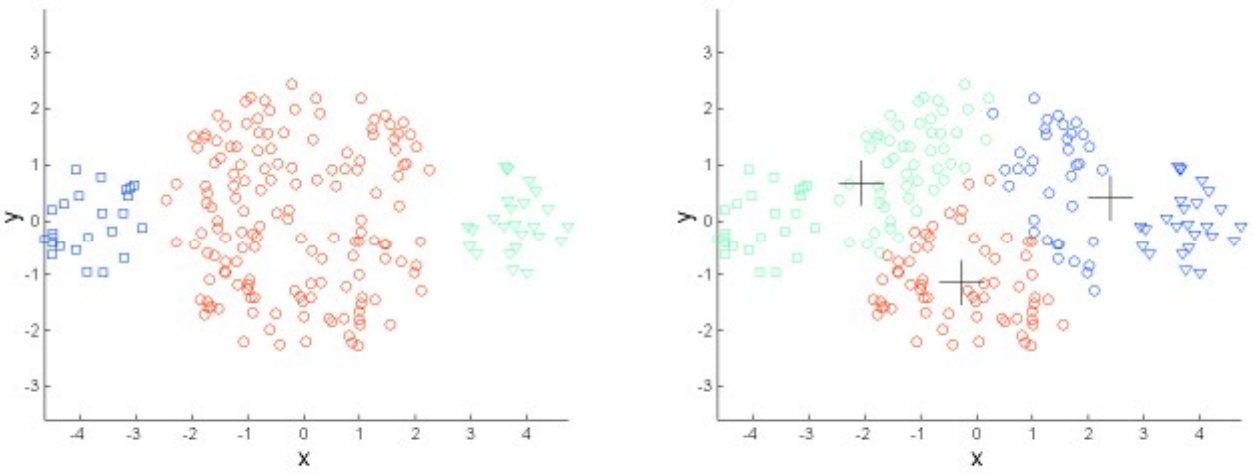


Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

資料分群-Kmean為例

- K-means以中心點距離最小化形成群組，造成此方法不利於下列特性的資料群組
  - Sizes: 大小差異較大
  - Densities: 群組密度不同
  - Non-globular shapes: 群組形狀較不規則
  - 此外，容易受離群值(outlier)影響
  - 類別型欄位需要進行轉換，例如One-Hot Encoding

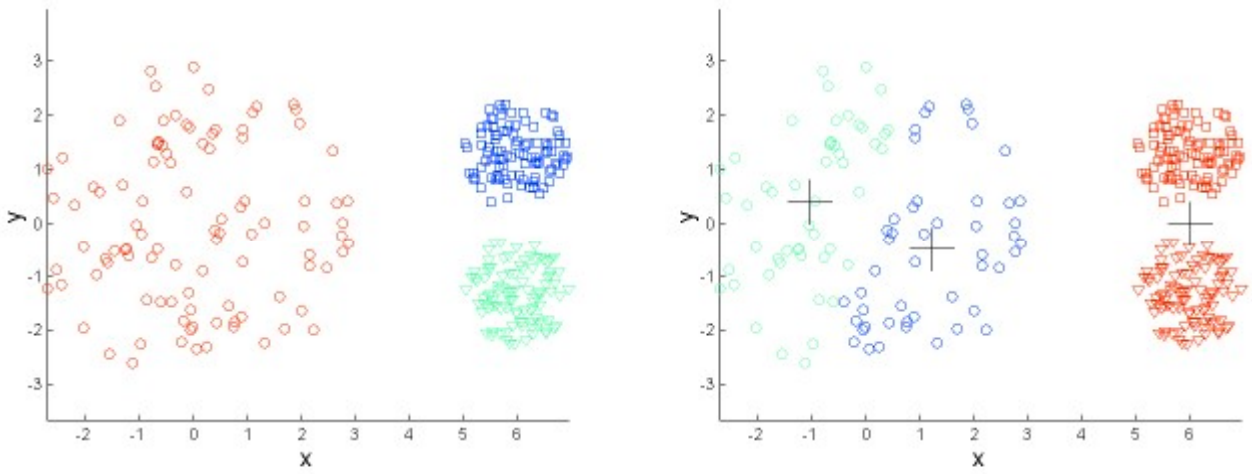
Size difference



Original Points **K-means (3 Clusters)**

Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

Density Difference

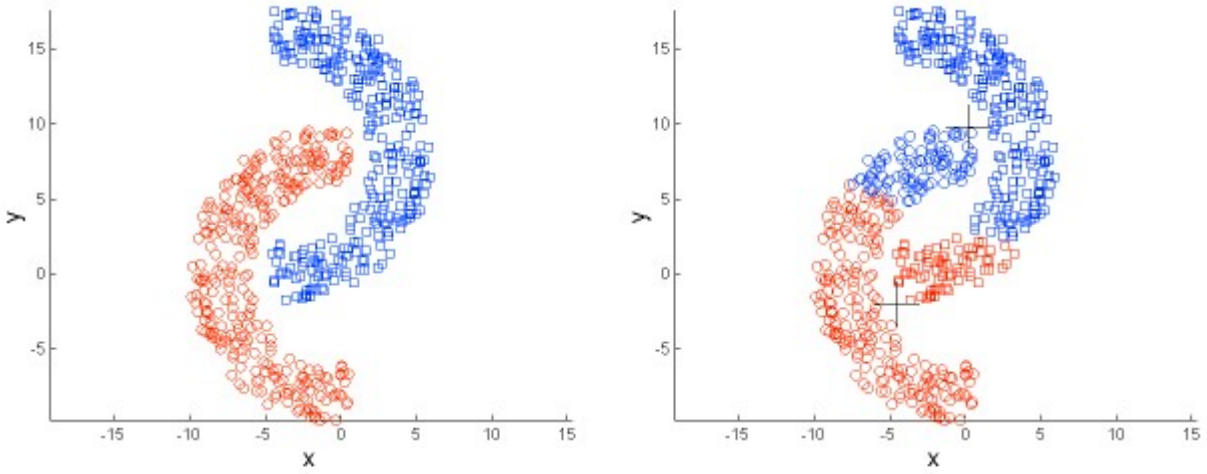


Original Points

***K*-means (3 Clusters)**

Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

Non-global shapes



Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

合理的分k群

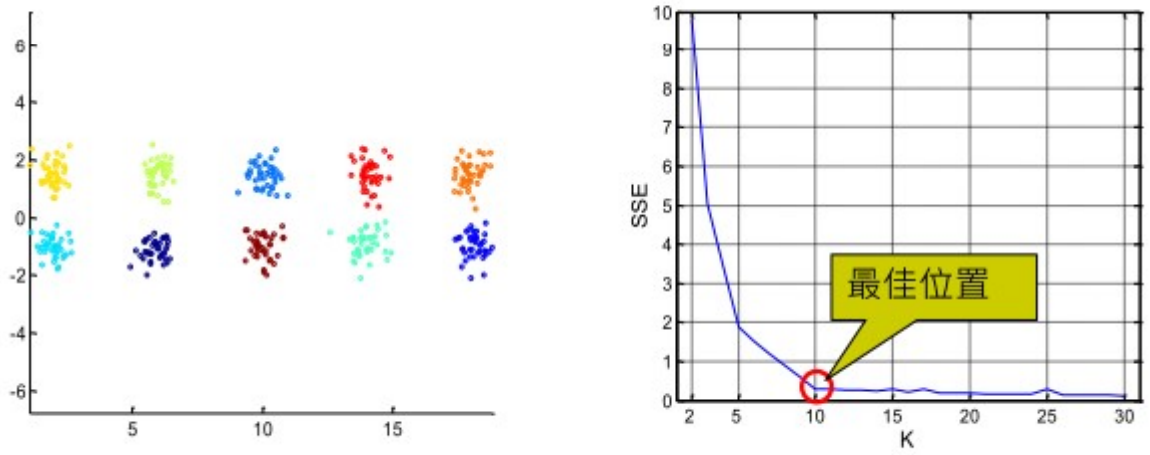
- 首先要了解如何評估合理的分群結果
- 以K-means而言，其目標是最小化下列公式，稱為Sum of Square Error (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - C_i: 表第i群，m_i: 第i群的中心點
- 簡而言之，即最小化所有群組內(intra-distance)距離的總和

合理的分k群

- 繪製不同*k*群 vs SSE的圖形(稱為Elbow plot)
- 從圖形中找到最明顯的轉折點(即梯度下降最多)，對應的*k*值
  即為首先要了解如何評估合理的分群結果



Source: P.N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

結論

- 分群方法有相當多，但沒有任一方法是最好的，只有最適合的方法
  - 可參考 Python scikit-learn clustering 官網的示範
  - 2.3. Clustering — scikit-learn 1.1.2 documentation
- 分群是屬於非監督學習的一種，即沒有標準答案，所以如何評估分群結果的合理性是最困難的