

K-Means Algorithm and Application in Data Compression using Python, Pygame and Scikit-Learn

阮文詠-Roan, Wen-Yong

Faculty of Information Management - Ming Chuan University

Taoyuan City, Taiwan

Email: vinhintw2014@gmail.com

Abstract—聚類問題是一項任務，其任務是根據對象的特徵將一組對象（也稱為成員）劃分為不同的組（稱為簇）。一組成員與其他組中的成員相比具有更多的相似性。本文從數學角度討論了一種傳統的聚類方法—K-Means 演算法。此外，提供了一個實驗，以檢驗該算法在二維空間中的應用，並在圖像壓縮中應用該算法。

I. 緒論

聚類問題在許多不同的應用程式中出現：機器學習數據挖掘和知識發現、數據壓縮和向量量化、模式識別和模式分類。^[1]

K-Means 演算法的目標是根據對象的屬性將數據集中的對象正確分成不同的組。例如，對象可以是房子，它們的屬性是大小、樓層數、位置、每年的耗電量等。目標是將房屋數據集分類成豪華、普通和貧窮等組。在這種情況下，必須處理房屋的所有屬性，將其轉化為數字以創建向量，這個過程被稱為向量化。

另一個例子是，將面板中的每個點作為一個對象，每個對像有兩個屬性，即 x 軸和 y 軸的位置。設置 K = 3。該算法正確找到了簇。(Fig. 1.a)

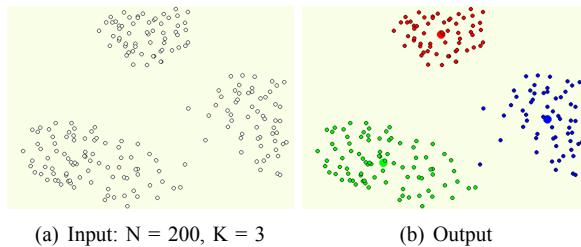


Fig. 1. 二維點上的 K-Means 演算法

II. 數學分析

A. 輸入與輸出

K-Means 演算法以一組觀察值 $X = [x_1, x_2, \dots, x_N] \in R^{d \times N}$ 為輸入，其中每個觀察值都是 d 維向量，N 是觀察值（成員）的數量，K 是組的數量 ($K, K < N$)。算法輸出 K 組的中心 $[m_1, m_2, \dots, m_K] \in R^{d \times N}$ 和每個成員所屬的組的索引或名稱（標籤）。

B. 誤差函數和優化問題

假設 $x_i (i \in [1, N])$ 屬於簇 $k (k \in [1, K])$ 則觀測值 x_i 的誤差值是觀測值 x_i 到歐幾里得空間 (euclidean space) 中心 m_k 的距離，由 $(x_i - m_k)$ 定義。設 $y_i = [y_{i1}, y_{i2}, \dots, y_{iK}]$ 是每個觀測值 x_i 的標籤向量，如果 x_i 屬於第 k 組，則 $y_{iK} = 1$ ，如果 x_i 不屬於第 k 組，則 $y_{iK} = 0 (\forall j \neq k)$ 。每個觀測值的標籤向量只包含一個數字 1，因為每個觀測值只屬於一個組，從而得到以下方程式

$$\sum_{k=1}^K y_{iK} = 1 \quad (1)$$

目標是最小化簇內平方和（方差），也稱為方差平方誤差，其中每個觀測值 x_i 與組 m_k 的平方誤差由以下式子定義

$$\|x_i - m_k\|^2 = y_{ik} \|x_i - m_k\|^2 \quad (2)$$

從方程式 1 中，標籤向量中的所有元素之和等於 1。每個觀測值的平方誤差是

$$y_{ik} \|x_i - m_k\|^2 = \sum_{j=1}^K y_{ij} \|x_i - m_j\|^2 \quad (3)$$

所有觀測值的平方誤差是給定觀測值集合中每個平方誤差的總和。目標是最小化誤差函數，即方程式 4，其中

$y_i = [y_{i1}, y_{i2}, \dots, y_{iN}]$ 是包含 N 個觀測值的所有標籤向量的矩陣， $M = [m_1, m_2, \dots, m_{mK}]$ 是 K 個組（簇）的中心。

$$f(Y, M) = \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|x_i - m_j\|^2 \quad (4)$$

目標也是要找到每個觀測值的中心和標籤向量，即 Y 與 M，這兩個輸出在 II-A 中提到。

$$Y, M = \operatorname{argmin}_{Y, M} \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|x_i - m_j\|^2 \quad (5)$$

C. 解決優化問題

方程 5 中有兩個變量，分別是每個觀測組的中心和每個觀測的標籤向量。可以通過固定其中一個變量，然後最小化另一個變量來解決問題。

1) 固定 M，觀測組的中心：因為所有中心 (M) 都是常數，所以目標是正確識別標籤向量，即確定每個觀測所屬的組，以便最小化方程 4 中的平方誤差。

$$y_i = \operatorname{argmin}_{y_i} \sum_{j=1}^K y_{ij} \|x_i - m_j\|^2 \quad (6)$$

從方程式 1 中得出，因為向量 y_i $i \in [1, K] = 1$ 中只有一個元素為 1，因此可以將方程 6 重寫為。

$$j = \operatorname{argmin}_j \|x_i - m_j\|^2 \quad (7)$$

$\|x_i - m_j\|^2$ 的值是觀測到組中心在歐幾里得空間中 (euclidean space) 的距離的平方。具體來說，當 M 是常數時，方程式 7 表明通過選擇標籤向量，使中心距離觀測最近，可以實現最小化平方誤差的目標。

2) 固定 Y，每個觀測的標籤向量：當標籤向量 (Y) 是常數時，目標是正確識別中心，以便最小化方程 4 中的平方誤差。在這種情況下，方程 5 中的優化問題可以通過以下方程式重寫。

$$m_j = \operatorname{argmin}_{m_j} \sum_{i=1}^N y_{ij} \|x_i - m_j\|^2 \quad (8)$$

方程式 8 是一個凸函數，對於每個 $i \in [1, N]$ 是可微的。因此，可以通過找到偏導函數的根來解決方程式 8。這種方法可以確保根是使函數達到最優值的值。假設 $g(m_j) = \sum_{i=1}^N y_{ij} \|x_i - m_j\|^2$ 從方程式 8 獲取並對 $g(m_j)$ 求偏導：

$$\frac{\partial g(m_j)}{\partial m_j} = 2 \sum_{i=1}^N y_{ij} (m_j - x_i) \quad (9)$$

方程式 9 等於 0 等價於：

$$m_j \sum_{i=1}^N y_{ij} = \sum_{i=1}^N y_{ij} x_i \quad (10)$$

$$\Leftrightarrow m_j = \frac{\sum_{i=1}^N y_{ij} x_i}{\sum_{i=1}^N y_{ij}} \quad (11)$$

當觀測值 x_i 屬於 m_j 組時， $y_{ij} = 1$ 。因此，方程式 11 的分母 $\sum_{i=1}^N y_{ij}$ 是屬於組 m_j 的觀測值的數量，而分子 $\sum_{i=1}^N y_{ij} x_i$ 是屬於組 m_j 的所有觀測值的總和。

換句話說，當 Y 是常數時，通過將中心分配給所屬組中觀測值的平均值，可以將平方誤差最小化。

D. 算法總結和流程圖

1) 總結：該算法可以通過連續完成將 Y 和 M 常數化，每次一個，如 II-C1 中所述和 II-C2

步驟 1. 將數據聚類為 k 組，其中 k 是預定義的。

步驟 2. 隨機選擇 k 個點作為聚類中心。

步驟 3. 根據歐幾里得距離函數將對象分配給最近的聚類中心。

步驟 4. 計算每個聚類中所有對象的質心或平均值。

步驟 5. 重複步驟 2。

2) 流程圖：下圖描述了 K-Means 演算法

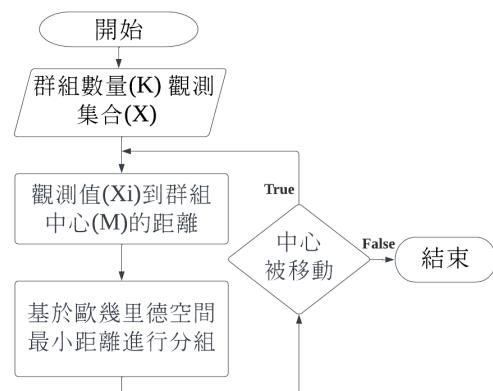


Fig. 2. K-Means 演算法流程圖

E. 討論

1) 收斂性：該算法將在一定的迭代次數後停止，因為平方誤差函數是一個嚴格遞減的序列，而平方誤差總是大於 0 的。但是，該算法不能保證找到全局最優解，因為通過找到偏導數為 0 時的根來解決方程式 8 只會返回局部最優解，但不能保證局部最優解是全局最小值。

以下圖表描述種子選擇不當導致出現局部最優情況的情況。

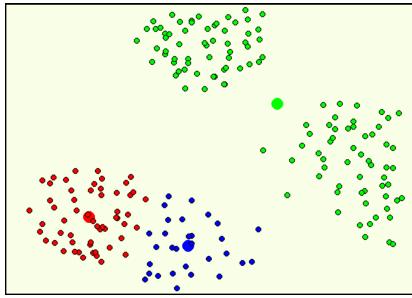


Fig. 3. 播種不良的 K-Means

在這種情況下，平方誤差為 25277，大約是圖 1b 產生的平方誤差 (16370) 的 1.54 倍。

2) 對初始聚類的敏感性：K-Means 演算法需要謹慎的種子設置，這意味著最終結果非常敏感於聚類的初始值。由於其缺點，已經進行了大量的努力來改進 K-Means 聚類演算法。^[2]

III. 應用資料壓縮

將重新進行一項實驗，即應用 K-Means 演算法來減小圖像的大小並輸出一張新的圖像，與原始圖像相比，新的圖像中顏色較少。該實驗使用 Python 程語言和 Pygame 以及 Scikit-Learn 套件進行。圖像的每個像素包含三個元素，即紅色、綠色、藍色 (RGB) 值。讓每個像素作為一個觀測值 (X)，然後圖像中的像素數量為觀測數量。每個觀測值都有三個屬性，即 RGB 值。在這種情況下，K-Means 演算法被應用於識別該圖像中的 K 個主要顏色。

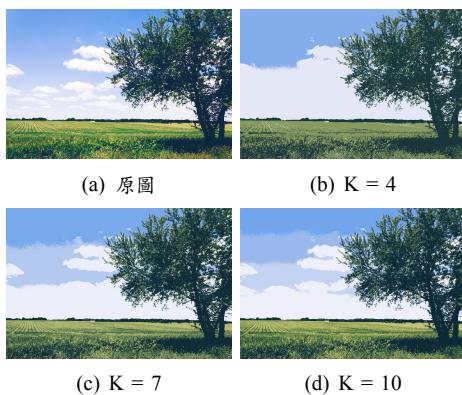


Fig. 4. 1920x1280 圖像上的圖像分割

下表顯示原始圖像的文件大小降低。

K(顏色)	文件大小 (KB)
原圖	4483
10	792
7	634
4	347

IV. 結論

K-Means 演算法非常簡單且快速易用，可以解決所有集群都是中心且分離的聚類問題。然而，當數據集和聚類更加複雜時，它將不再有效。

本報告沒有提出改進算法有效性的新思路，其目的是向讀者介紹基本的入門級聚類方法，並提供了二維和三維數據集的一些視覺示例。

V. 進一步的研究

該演算法的實現簡單，但是它對初始中心點的敏感性和數據集的嚴格結構等缺點是無法避免的。進一步的研究可以著眼於提高初始中心點的價值。傳統算法依賴於隨機性，可以嘗試尋找一種確定的初始中心點的方法。此外，對於大型數據集，該算法可能收斂速度較慢。可以研究如何加快迭代收斂的方法。

致謝

作者感謝 Tiep V. Huu 的機器學習博客提供的資訊，以及 Andrew Ng 教授在 Coursera 上啟發性的機器學習課程。

參考文献

- [1] Joaquin Prez Ortega, Ma. Del, Roco Boone Rojas, and Mara J. Somod-evilla "Research issues on k-means algorithm: An experimental trial using matlab".
- [2] Arthur, D., Vassilvitskii, S., 2016. *k-Means++: The Advantages of Careful Seeding*. Technical Report, Stanford.