

Homework 4: Clustering Techniques

Student ID: 18329015

Student Name: 郝裕玮

Lectured by: Shangsong Liang
Machine Learning and Data Mining
Sun Yat-sen University
Deadline for Submission: April 5, 2022

Exercise 1: Implement K-Means Manually

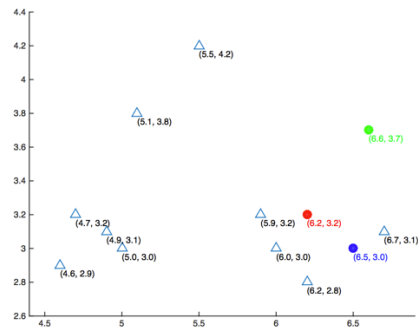


Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

Given the matrix \mathbf{X} (see the matrix below) whose rows represent different data points, you are asked to perform a k-means clustering on this dataset using the Euclidean distance as the distance function. Here k is chosen as 3. The Euclidean distance d between a vector x and a vector y both in \mathcal{R}^p is defined as $d = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$, where p is the number of dimensions of each data point. All data in \mathbf{X} were plotted in Figure 1. The centers of 3 clusters were initialized as $\mu_1 = (6.2, 3.2)$ (red), $\mu_2 = (6.6, 3.7)$ (green), $\mu_3 = (6.5, 3.0)$ (blue). Answer the following questions (a) to (d).

$$\mathbf{X} = \begin{bmatrix} 5.9 & 3.2 \\ 4.6 & 2.9 \\ 6.2 & 2.8 \\ 4.7 & 3.2 \\ 5.5 & 4.2 \\ 5.0 & 3.0 \\ 4.9 & 3.1 \\ 6.7 & 3.1 \\ 5.1 & 3.8 \\ 6.0 & 3.0 \end{bmatrix}$$

代码如下（算法思路和具体分析均已包含在代码注释中）：

```
import matplotlib.pyplot as plt
import numpy as np
import math

# 计算两个向量间的距离
def dis(a, b, ax):
    # 计算向量之间距离, axis = 1 代表按行向量进行计算处理
    return np.linalg.norm(a-b, axis = ax)

# K_means 聚类迭代算法
```

```

def K_means(X, center, total_times):
    """
        X: 需要进行分类的多个向量
        center: 聚类的初始中心向量（红绿蓝 3 个聚类中心）
        total_times: 迭代次数
    """
    # 储存更新后的聚类中心坐标
    center_new = np.zeros(center.shape)
    # 储存每个点所在的聚类(即红绿蓝 3 个聚类中心)
    X_cluster = np.zeros(len(X))

    #开始迭代
    times = 1
    while times < total_times :
        # 遍历 X 中的每个点
        for i in range(len(X)):
            # 计算当前点与 3 个中心点的距离
            distance_X = dis(X[i], center, 1)
            # 选出与该点最近的聚类中心点的下标
            cluster = np.argmin(distance_X)
            # 存储该点所在的聚类
            X_cluster[i] = cluster

        # 计算新的 3 个聚类中心点的坐标
        for i in range(3):
            # 用于存储每个聚类内部的点坐标
            cluster_point = []
            # 寻找 X 中相应聚类的点并将其统一存储到 cluster_point 中
            for j in range(len(X)):
                if X_cluster[j] == i:
                    cluster_point.append(X[j])
            cluster_point = np.array(cluster_point)
            # 计算新的聚类中心点坐标
            center_new[i] = np.mean(cluster_point, axis=0)

        # 输出每次迭代的结果
        print("当前迭代次数为%d, 各簇的中心点为: \n" %(times))
        center = center_new
        print("u1 = %s\nu2 = %s\nu3 = %s\n" %(center[0], center[1],
center[2]))
        times = times + 1

X = np.array([[5.9, 3.2], [4.6, 2.9], [6.2, 2.8], [4.7, 3.2], [5.5, 4.2],
[5.0, 3.0], [4.9, 3.1], [6.7, 3.1], [5.1, 3.8], [6.0, 3.0]])
center = np.array([[6.2, 3.2], [6.6, 3.7], [6.5, 3.0]])
K_means(X,center,5)


```

(a). What's the center of the first cluster (red) after one iteration? (Answer in the format of [x1, x2], round your results to three decimal places, same as problems 2 and 3)

当前迭代次数为1，各簇的中心点为：

```
u1 = [5.17142857 3.17142857]
u2 = [5.5 4.2]
u3 = [6.45 2.95]
```

答：由上图结果可知，红色簇的中心为：[5.171, 3.171]

(b). What's the center of the second cluster (green) after two iterations? 

当前迭代次数为2，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

答：由上图结果可知，绿色簇的中心为：[5.300, 4.000]

(c). What's the center of the third cluster (blue) when the clustering converges?

答案见下页。

当前迭代次数为1，各簇的中心点为：

```
u1 = [5.17142857 3.17142857]
u2 = [5.5 4.2]
u3 = [6.45 2.95]
```

当前迭代次数为2，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

当前迭代次数为3，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

当前迭代次数为4，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

由上图可知，聚类在第3次迭代后已经收敛。此时蓝色簇的中心点为：[6.200, 3.025]

(d). How many iterations are required for the clusters to converge?

当前迭代次数为1，各簇的中心点为：

```
u1 = [5.17142857 3.17142857]
u2 = [5.5 4.2]
u3 = [6.45 2.95]
```

当前迭代次数为2，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

当前迭代次数为3，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

当前迭代次数为4，各簇的中心点为：

```
u1 = [4.8 3.05]
u2 = [5.3 4. ]
u3 = [6.2 3.025]
```

由上图可知，聚类在第3次迭代后已经收敛。

Exercise 2: Application of K-Means

K-Means 的应用

There are 6 different datasets noted as A, B, C, D, E, and F. Each dataset is clustered using two different methods, and one of them is K-means. All results are shown in Figure 2. You are required to determine which result is more likely to be generated by K-means method. The distance measure used here is the Euclidean distance. Answer the following questions (a) to (h).

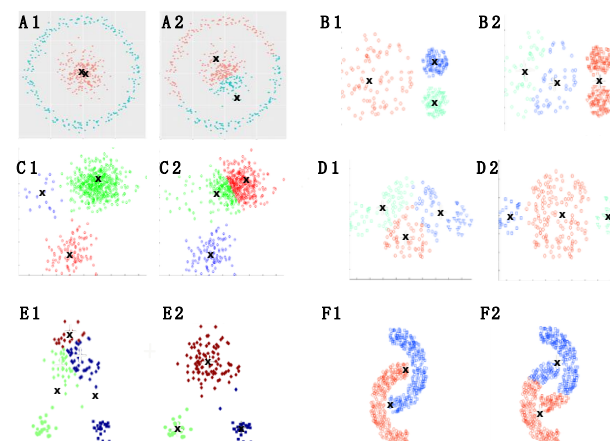


Figure 2: Clustered results for 6 datasets

(a). For dataset A, which result is more likely to be generated by K-means method? (write A1 or A2, same in the following questions (b) to (f))

答：在 A1 中，外圆上的某些点一定更接近于另一个的中心点（不可能外围一整圈都属于同一个聚类类）。所以很显然A2正确。

(b). Dataset B (B1 or B2?)

答：在 B1 中，红色类最右侧的点很显然更接近于蓝色和绿色聚类的中心点。所以很显然B2正确。

(c). Dataset C (C1 or C2?)

答：在 C1 中，绿色类最左侧的点很显然更接近于蓝色类的中心点。所以很显然C2正确。

(d). Dataset D (D1 or D2?)

答：在 D2 中，红色类最左侧的点很显然更接近于蓝色聚类的中心点。所以很显然D1正确。

(e). Dataset E (E1 or E2?)

答：在 E1 中，红色类下方的部分绿色点和蓝色点很显然更接近于红色类的中心点。所以很显然E2正确。

(f). Dataset F (F1 or F2?)

答：在 F1 中，蓝色簇的中心点左边的红色点很显然离蓝色聚类中心点更近。所以很显然F2正确。

(g). Provide the reasons/principles that draw your answers to the questions (a) to (f).

答：详见a-f题。

(h). For dataset F, do you think k-means perform well? Why? Are there other better clustering algorithms to be used to cluster data distributing like the data in the dataset F?

答：表现一般，因为没有与不同k值的分类结果进行对比，无法判定准确率。更好的聚类算法有：K-means++, Canopy等等。

Exercise 3: Applications of Clustering Techniques in IR and DM

聚类技术在 IR 和 DM 中的应用

In information retrieval and data mining, are there any applications where we can apply clustering algorithms to improve the performance? Explain how clustering algorithms can improve the performance of such applications.

答：数据挖掘中聚类算法的应用很广泛。

（1）在商务上，聚类能帮助市场分析人员从客户基本库中发现不同的客户群，并且用不同的购买模式来刻画不同的消费群体的特征；

（2）在生物学上，聚类能用于帮助推导植物和动物的种类，基因和蛋白质的分类，获得对种群中固定结构的认识；

（3）聚类在地球观测数据中相似地区的确定，根据房屋的类型、价值和位置对一个城市中房屋的分类发挥作用；

（4）聚类也能用来对web上的文档进行分类，以发现有用的信息；

（5）此外，聚类还可以作为其他方法的预处理步骤，即类似于降维处理，减少算法的初始参数输入数量。