Homework 4: Clustering Techniques

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

Exercise 1

(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)

解：[5.171, 3.171]

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

解：[5.300, 4.000]

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

解：[6.200, 3.025]

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

解：2次

具体迭代过程如下，可以看到，第二次和第三次的结果相同（代码见文末）：

```
第1次迭代：
color:red,x:5.171428571428572,y:3.1714285714285713
color:green,x:5.5,y:4.2
color:blue,x:6.45,y:2.95

第2次迭代：
color:red,x:4.800000000000001,y:3.05
color:green,x:5.3,y:4.0
color:blue,x:6.2,y:3.025

第3次迭代：
color:red,x:4.800000000000001,y:3.05
color:green,x:5.3,y:4.0
color:blue,x:6.2,y:3.025

Process finished with exit code 0
```

Exercise 2

(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))

解：A2。因为对于A2同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：B2。因为对于B2同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：C2。因为对于C2同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：D1。因为对于D1同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：E2。因为对于E2同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：F2。因为对于F2同一簇中的任意一点，该点距离簇心的距离比距离其他簇簇心的距离近。

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

解：根据K-means method可知，对于当前簇中的任意一点，该点距离簇心的距离比距离任何其他簇心的距离近。根据这一原则，可以得出 a-f 的答案

(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?

解：
对于数据集 F，我认为k-means的效果不好。
原因：显然，数据明显展示出左右两簇的特点，由此进行划分更加符合数据的特性，而不是按照k-means的结果进行划分。

其他算法：密度聚类，层次聚类等。

Exercise 3

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.

解：

1. In information retrieval：
    - 应用：文档自动分类，文献搜索结果聚类，图像信息检索聚类，XML文档聚类等。
    - 原因：
      在上述的应用项目中，其数据集均可以视为大量相似元素的集合。针对这类应用，很大部分的信息检索任务本质上就是分类问题。
      通过聚类算法将相似内容进行聚类，当出现检索任务时，直接将聚类好的对应类的结果反馈给用户，可以显著降低响应时间。另外，还可以根据用户的选择信息对数据进一步聚类，从而匹配到更符合用户需求的簇，有利于提升用户体验。

2. In data mining：
    - 应用：用户个性化推荐，商品布局等。
    - 原因：
      在上述的应用项目中，其数据集隐含有大量相似元素的信息。通过聚类算法，把归属于同一类的元素聚合在一起，从而更好的实现数据挖掘任务。
      以上面两个应用为例：前者可以根据用户之间兴趣爱好等信息的相似性，推荐其同类别下其他用户的选择，从而更大概率地匹配上该用户的兴趣点。后者则可以通过聚类发现不同商品类别之间的联系，从而将更可能同时购买的商品放在一起，典型的例子有“啤酒和尿布”。

code for exercise 1

```python
from numpy import *

dataSet = [[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]]
clusters = [
    {"color": "red", "x": 6.2, "y": 3.2, "kind": 0, "num": 0},
    {"color": "green", "x": 6.6, "y": 3.7, "kind": 1, "num": 0},
    {"color": "blue", "x": 6.5, "y": 3.0, "kind": 2, "num": 0},
]


# calculate Euclidean distance
def euclDistance(x1, y1, x2, y2):
    return sqrt(power(x1 - x2, 2) + power(y1 - y2, 2))


## step 1: init centroids
numSamples = 10
# first column stores which cluster this sample belongs to,
# second column stores the error between this sample and its centroid
clusterAssment = [[-1] * 2 for _ in range(numSamples)]
for i in range(numSamples):
    clusterAssment[i][0] = -1
clusterChanged = True
k = len(clusters)

count = 0
while clusterChanged and count < 10:
    count += 1
    print(f"\n第{count}次迭代：")
    clusterChanged = False
    ## for each sample
    for i in range(numSamples):
        minDist = 100000.0
        minIndex = -1
        ## for each centroid
        ## step 2: find the centroid who is closest
        for j in range(k):
            distance = euclDistance(dataSet[i][0], dataSet[i][1],
                                    clusters[j]["x"], clusters[j]["y"])
            if distance < minDist:
                minDist = distance
                minIndex = j

        ## step 3: update its cluster
        if clusterAssment[i][0] != minIndex:
            if clusterAssment[i][0] != -1:
                clusters[clusterAssment[i][0]]["num"] -= 1
            clusterChanged = True
            clusterAssment[i][0], clusterAssment[i][1] = minIndex, minDist ** 2
            clusters[minIndex]["num"] += 1

    ## step 4: update centroids
    newx = [0, 0, 0]
```

```
    newy = [0, 0, 0]
    for i in range(numSamples):
        belong_kind = clusterAssment[i][0]
        newx[belong_kind] += dataSet[i][0]
        newy[belong_kind] += dataSet[i][1]
    for j in range(k):
        clusters[j]["x"] = newx[j] / clusters[j]["num"]
        clusters[j]["y"] = newy[j] / clusters[j]["num"]
        print(f"color:{clusters[j]['color']},x:{clusters[j]['x']},"
              f"y:{clusters[j]['y']}")

```