作业二

k-均值的 Python 实现（100 分）

注意： 这一问题需要用 Python 实现。

* * *

这个问题将帮助你理解在 Python 上实现聚类算法的具体细节。此外，这个问题还将帮助你理解在实践中使用不同的距离度量和初始化策略所产生的影响。假设我们有一个包含 n 个数据点的集合 $\mathcal{X}$，这些数据点存在于 d 维空间 $\mathbb{R}^d$ 中。给定聚类数 k 和 k 个质心的集合 $\mathcal{C}$，我们现在将定义各种距离度量以及它们所最小化的相应成本函数。

欧几里得距离： 在 d 维空间中给定两个点 A 和 B，其中 $A = [a_1, a_2, \cdots, a_d]$，$B = [b_1, b_2, \cdots, b_d]$，$A$ 和 B 之间的欧几里德距离定义如下：

$$\|a - b\| = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2} \tag{1}$$

使用欧几里德距离度量来将点分配给各类别时，相应的成本函数 ϕ 如下：

$$\phi = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} \|x - c\| \tag{2}$$

请注意，在成本函数中，使用的是距离的平方。这是因为算法保证最小化的是平方欧几里德距离。

曼哈顿距离： 在 d 维空间中给定两个随机点 A 和 B，其中 $A = [a_1, a_2, \cdots, a_d]$，$B = [b_1, b_2, \cdots, b_d]$，$A$ 和 B 之间的曼哈顿距离定义如下：

$$|a - b| = \sum_{i=1}^{d} |a_i - b_i| \tag{3}$$

当我们使用曼哈顿距离度量来将点分配给各类别时，相应的成本函数 ψ 如下：

$$\psi = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} |x - c| \tag{4}$$

迭代 k-均值算法：我们在课堂上学习了基本的 k-均值算法，其步骤如下：首先初始化 k 个质心，然后将每个数据点分配到最近的质心，接着根据数据点的分配情况重新计算质心。在实践中，通常会对上述步骤进行多次迭代。我们在 **Algorithm 1** 中呈现了迭代版本的 k-均值算法。

Algorithm 1 Iterative k-Means Algorithm

```
procedure ITERATIVE k-MEANS
    Select k points as initial centroids of the k clusters.
    for iterations := 1 to MAX_ITER do
        for each point p in the dataset do
            Assign point p to the cluster with the closest centroid
        end for
        Calculate the cost for this iteration.
        for each cluster c do
            Recompute the centroid of c by minimizing the cost
        end for
    end for
end procedure
```

Python 中的迭代 k-均值聚类：使用 Python 实现迭代 k-均值聚类。请使用 *data* 中的数据集。

文件夹中包含 3 个文件：

1. *data.txt* 包含一个数据集，该数据集具有 4601 行和 58 列。每一行都是一个以 58 维特征向量表示的文档。向量中的每个分量表示文档中一个单词的重要性。

2. *c1.txt* 包含 k 个初始聚类质心。这些质心是从输入数据中随机选取的 $k = 10$ 个数据点。

3. *c2.txt* 包含初始的聚类质心，这些质心尽可能远离彼此，使用欧几里德距离作为距离度量。（实现上，可以随机选择第一个质心 c1，然后找到距离 c1 最远的点 c2，接着选择距离 c1 和 c2 都最远的点 c3，以此类推）。

在本问题中，对所有实验设置迭代次数（MAX_ITER）为 20，将聚类数 k 设置为 10。您在编写程序时也应参照这一要求。

在将数据点分配给质心时，如果有多个等距的质心，请选择按字典顺序排列的第一个质心。

(a) 使用欧几里德距离探索初始化策略 [50 分]

1. **[25 分]** 使用欧几里德距离（参考式1）作为距离度量，在每次迭代 i 中计算成本函数 $\phi(i)$（参考式2）。这意味着，在第一次迭代中，您将使用 *c1.txt* 或 *c2.txt* 中给出的初始质心计算成本函数。使用 *c1.txt* 和 *c2.txt* 在 *data.txt*

上运行 k-均值算法。生成图表，分别针对 *c1.txt* 和 *c2.txt*，绘制了成本函数 $\phi(i)$ 作为迭代次数 $i = 1, \ldots, 20$ 的函数。（统一要求画三张图，*c1.txt* 和 *c2.txt* 各一张，第三张是将两条成本曲线画在一张图上）

2. **[25 分]** 当使用欧几里德距离度量时，用 *c1.txt* 和 *c2.txt* 初始化聚类质心后，在 k-均值算法的前 10 次迭代中，成本的百分比变化是多少？从成本 $\phi(i)$ 的角度来看，使用 *c1.txt* 进行 k-均值的随机初始化是否比使用 *c2.txt* 更好？请解释您的理由。

   （提示：明确一点，百分比是指（成本 *[0]* - 成本 *[10])* / 成本 *[0]*。）

(b) 使用曼哈顿距离探索初始化策略 [50 分]

1. **[25 分]** 使用曼哈顿距离（参考式3）作为距离度量，在每次迭代 i 中计算成本函数 $\psi(i)$（参考式4）。这意味着，在第一次迭代中，您将使用 *c1.txt* 或 *c2.txt* 中给出的初始质心计算成本函数。使用 *c1.txt* 和 *c2.txt* 在 *data.txt* 上运行 k-均值算法。生成图表，分别针对 *c1.txt* 和 *c2.txt*，绘制了成本函数 $\phi(i)$ 作为迭代次数 $i = 1, \ldots, 20$ 的函数。（统一要求画三张图，*c1.txt* 和 *c2.txt* 各一张，第三张是将两条成本曲线画在一张图上）

   （提示：这个问题可以以与（a）部分类似的方式解决。另外请注意，对于曼哈顿距离，成本不一定总是递减的。k-均值只能保证在平方欧几里德距离的情况下成本单调递减。查阅 k-中位数以了解更多信息。）

2. **[25 分]** 当使用曼哈顿距离度量时，用 *c1.txt* 和 *c2.txt* 初始化聚类质心后，在 k-均值算法的前 10 次迭代中，成本的百分比变化是多少？从成本 $\phi(i)$ 的角度来看，使用 *c1.txt* 进行 k-均值的随机初始化是否比使用 *c2.txt* 更好？请解释您的理由。

需要提交的内容:

(1) （a）和（b）两部分的代码
(2) 两种初始化策略对应的成本 vs. 迭代次数的图表 [(a)]
(3) 成本的百分比变化以及您的解释 [(a)]
(4) 两种初始化策略对应的成本 vs. 迭代次数的图表 [(b)]
(5) 成本的百分比变化以及您的解释 [(b)]