# 山东大学 计算机科学与技术 学院

# 人工智能导论 课程实验报告

| 学号：202300130183 | 姓名： 宋浩宇 | 邮 箱 ： 202300130183 @mail.sdu.edu.cn |
|---|---|---|

实验题目：六、KMeans 异常检测（统计建模）

实验过程：

（记录实验过程、遇到的问题和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

实现 Kmeans（K 均值）算法的步骤如下（原文本为 markdown 格式，因此此处放截图）：

**实现方式**

| | |
|---|---|
| $D$ | 接受输入的数据集 |
| $k$ | 接受输入的k值 |
| $C_i$ | 聚类集合 |
| $\mathbb{M}$ | 均值向量集合 |
| $\mu_i$ | 均值向量 |
| $d(\boldsymbol{x},\mu)$ | 两个向量的距离 |

1. 在数据集$D$中随机挑选出$k$个样本$\{\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_k}\}$用于初始化均值向量集合$\mathbb{M}=\{\mu_1,\mu_2,\ldots,\mu_k\}$
2. 分别计算数据集中除这$k$个样本外每个样本与$\mu_i$的距离，并将该样本加入与之距离最短的均值向量所对应的聚类集合$C_i$中
3. 计算每一个聚类集合新的均值向量$\mu_i=\frac{1}{|C_i|}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x_i}$
4. 重复2-3步直到聚类集合不再改变

在 python 中的简单实现如下：

数据输入部分略，前期准备包括距离计算函数的定义，初始质心的选择

```
# 欧氏距离(L2范数)计算
1 个用法
def euclidean_distance(x,y):
    sum=0
    for i in range(len(x)):
        sum+=(x[i]-y[i])**2
    return np.sqrt(sum)
```

```
# 初始化
1 个用法
def initialize():
    for i in range(k):
        mu.append(data[i])
```

对于 kmeans 聚类过程的实现主要有一下两部分：

1. 质心向量的重计算

```python
# 质心向量重置
1个用法
def heart():
    global data
    global sets
    global mu
    if sets==[]:
        return True
    temmu=[]
    for i in range(k):
        temlist = [0 for l in range(n)]
        for j in sets[i]:
            for l in range(n):
                temlist[l]=temlist[l]+j[l]
        for l in range(n):
            temlist[l]=temlist[l]/len(sets[i])
        temmu.append(temlist)
    if temmu==mu:
        return False
    mu=temmu
    return True
```

2. 数据的重分组

```python
# 聚类
1 个用法
def kmeans():
    global sets
    global data
    global mu
    global wrong
    while heart():
        temset=[[] for _ in range(k)]
        for i in data:
            temdistance=[0 for _ in range(k)]
            for j in range(k):
                temdistance[j]=euclidean_distance(i,mu[j])
            if min(temdistance) >= thresold:
                wrong.append(i)
                data.remove(i)
            else:
                temset[temdistance.index(min(temdistance))].append(i)
        if temset != sets:
            sets=temset
```

而为了实现异常值检测的目的，在此处使用最简单的方法，即人为输入一个阈值，当某一项数据与每一个质心的距离都大于该阈值时，将该向量归为异常值。

结果分析与体会：

## 算法分析

可以证明，在使用$L_2$范数来作为距离计算依据时，k-means算法可以做到最小化平方误差$E=\sum_{i}^{k}\sum_{x\in C_i}\|x-\mu_i\|_2$，而因为k-means算法使用的是贪心策略，因此在使用其他范数作为距离计算的依据时，需要考虑计算出的结果是否是最小化了误差$E$的解

因为需要靠人输入$k$这个重要参数，因此程序的运行结果除了和数据集有关，也和人为输入的聚类个数和选择的范数有关，因此该机器学习算法也和其他传统机器学习算法一样对于专家知识的有一定需求

而且因为每一轮循环都要进行$k|D|$次浮算，且循环数难以预测，所以k-means算法会进行大量的浮算，在这个过程中会消耗大量时间，也有可能会出现精度丢失的情况

因为在计算均值向量时，聚类集合中的每一个向量都有相同的权值，因此k-means算法对于数据集中的噪声很敏感，对于数据集的质量要求很高