模型-机器学习-聚类-K-MEANS算法与原型聚类【hxy】

1. 模型名称

k均值聚类算法（K-Means Clustering Algorithm，K-MEANS）

迭代求解的**原型聚类**（Prototype-based Clustering）

2. 评价

2.1 优点

- 容易理解，聚类效果不错，虽然是局部最优， 但往往局部最优就够了
- 处理大数据集的时候，该算法可以保证较好的伸缩性
- 当簇近似高斯分布的时候，效果非常不错
- 算法复杂度低

2.2 缺点

- K值需预先给定
- 对初始选取的聚类中心点敏感，不同的随机种子点得到的聚类结果不同
- 使用贪心思想迭代，算法常常收敛至局部最优解，无法获得全局最优解
- 不适合所有数据类型：如不适合处理非球形簇、不同尺寸和不同密度的簇
- 找到最优划分是NP难问题（Aloise2009）

3. 基本算法

1. 初始化：为给定样本集合$X=(x_{ij})_{m\times n}$随机初始化k个随机聚类中心
2. 分配样本点：计算每个点到聚类中心的距离，并将之分配到距离它最近的聚类中心
3. 更新聚类中心：利用每一类点的均值作为新的聚类中心
4. 重复第2步和第3步，当算法在新的一轮迭代中聚类中心没有更新，或者到达迭代的最大代数，算法终止，输出聚类中心

4. 实例

4.1 利用已有数据和Sklearn自带K-Means方法

4.1.1 数据介绍

city.txt

现有1999年全国31个省份城镇居民家庭平均每人全年消费性支出的八个主要变量数据，这八个变量分别是：食品、衣着、家庭设备用品及服务、医疗保健、交通和通讯、娱乐教育文化服务、居住以及杂项商品和服务。利用已有数据，对31个省份进行聚类。

城市	食品	衣着	家庭	医疗保险	交通和通讯	娱乐教育文化服务	居住	杂项商品和服务

```
北京,2959.19,730.79,749.41,513.34,467.87,1141.82,478.42,457.64
天津,2459.77,495.47,697.33,302.87,284.19,735.97,570.84,305.08
河北,1495.63,515.90,362.37,285.32,272.95,540.58,364.91,188.63
山西,1406.33,477.77,290.15,208.57,201.50,414.72,281.84,212.10
内蒙古,1303.97,524.29,254.83,192.17,249.81,463.09,287.87,192.96
辽宁,1730.84,553.90,246.91,279.81,239.18,445.20,330.24,163.86
吉林,1561.86,492.42,200.49,218.36,220.69,459.62,360.48,147.76
黑龙江,1410.11,510.71,211.88,277.11,224.65,376.82,317.61,152.85
上海,3712.31,550.74,893.37,346.93,527.00,1034.98,720.33,462.03
江苏,2207.58,449.37,572.40,211.92,302.09,585.23,429.77,252.54
浙江,2629.16,557.32,689.73,435.69,514.66,795.87,575.76,323.36
安徽,1844.78,430.29,271.28,126.33,250.56,513.18,314.00,151.39
福建,2709.46,428.11,334.12,160.77,405.14,461.67,535.13,232.29
江西,1563.78,303.65,233.81,107.90,209.70,393.99,509.39,160.12
山东,1675.75,613.32,550.71,219.79,272.59,599.43,371.62,211.84
河南,1427.65,431.79,288.55,208.14,217.00,337.76,421.31,165.32
湖南,1942.23,512.27,401.39,206.06,321.29,697.22,492.60,226.45
湖北,1783.43,511.88,282.84,201.01,237.60,617.74,523.52,182.52
广东,3055.17,353.23,564.56,356.27,811.88,873.06,1082.82,420.81
广西,2033.87,300.82,338.65,157.78,329.06,621.74,587.02,218.27
海南,2057.86,186.44,202.72,171.79,329.65,477.17,312.93,279.19
重庆,2303.29,589.99,516.21,236.55,403.92,730.05,438.41,225.80
四川,1974.28,507.76,344.79,203.21,240.24,575.10,430.36,223.46
贵州,1673.82,437.75,461.61,153.32,254.66,445.59,346.11,191.48
云南,2194.25,537.01,369.07,249.54,290.84,561.91,407.70,330.95
```

```
西藏,2646.61,839.70,204.44,209.11,379.30,371.04,269.59,389.33
陕西,1472.95,390.89,447.95,259.51,230.61,490.90,469.10,191.34
甘肃,1525.57,472.98,328.90,219.86,206.65,449.69,249.66,228.19
青海,1654.69,437.77,258.78,303.00,244.93,479.53,288.56,236.51
宁夏,1375.46,480.89,273.84,317.32,251.08,424.75,228.73,195.93
新疆,1608.82,536.05,432.46,235.82,250.28,541.30,344.85,214.40
```

4.1.2 实验目的

通过聚类，了解1999年各个省份的消费水平在国内的情况

4.1.3 代码实现

city.py

```python
# 导入sklearn相关包
import numpy as np
from sklearn.cluster import KMeans

# 自定义加载数据函数
def loadData(filePath):
  # 以读的兼容模式打开文件
  fr = open(filePath, 'r+')
  # 读取每一行
  lines = fr.readlines()
  # retData用来存储城市各项消费信息
  retData = []
  # retCityName用来存储城市名称
  retCityName = []
  # 将每一行城市信息分别存到retData和retCityName中
  for line in lines:
    items = line.strip().split(",")
    retCityName.append(items[0])
    retData.append([float(items[i]) for i in range(1,len(items))])
  # 返回retData和retCityName
  return retData, retCityName

# 加载数据，创建K-means算法实例，并进行训练，获得标签
if __name__ == '__main__':
  # 用自定义loadData方法读取数据
  data, cityName = loadData('/Users/xinyuanhe/Desktop/city.txt')
  # 创建实例
  # n_clusters用于指定聚类中心的个数，init初始聚类中心的初始化方法，max_iter最大迭代次数
  # init默认k-means++，max_iter默认300
  km = KMeans(n_clusters=3)
  # 调用fit_predict进行计算：fit算簇中心，predict指定x中每个点所属于的簇的位置
  label = km.fit_predict(data)
  # 计算不同簇的平均花费
  # np.sum(axis=1)计算每一行向量的和
  # km.cluster_centers_聚类中心
```

```
  expenses = np.sum(km.cluster_centers_,axis=1)
  # 创建簇
  CityCluster = [[],[],[]]
  # 将每个样本分到不同簇中
  for i in range(len(cityName)):
    CityCluster[label[i]].append(cityName[i])
  # 打印每个簇的复杂度和簇中样本
  for i in range(len(CityCluster)):
    print("Expenses:%.2f" % expenses[i])
    print(CityCluster[i])
```

结果：

```
Expenses:5113.54
['天津', '江苏', '浙江', '福建', '湖南', '广西', '海南', '重庆', '四川', '云南', '西藏'
]
Expenses:7754.66
['北京', '上海', '广东']
Expenses:3827.87
['河北', '山西', '内蒙古', '辽宁', '吉林', '黑龙江', '安徽', '江西', '山东', '河南', '湖
北', '贵州', '陕西', '甘肃', '青海', '宁夏', '新疆']
```

4.2 利用blobs随机生成数据和自定义K-means方法

4.2.1 实验目的

生成两类聚类，分别用K-Means实现，看聚类结果

4.2.2 代码实现

K-means.py

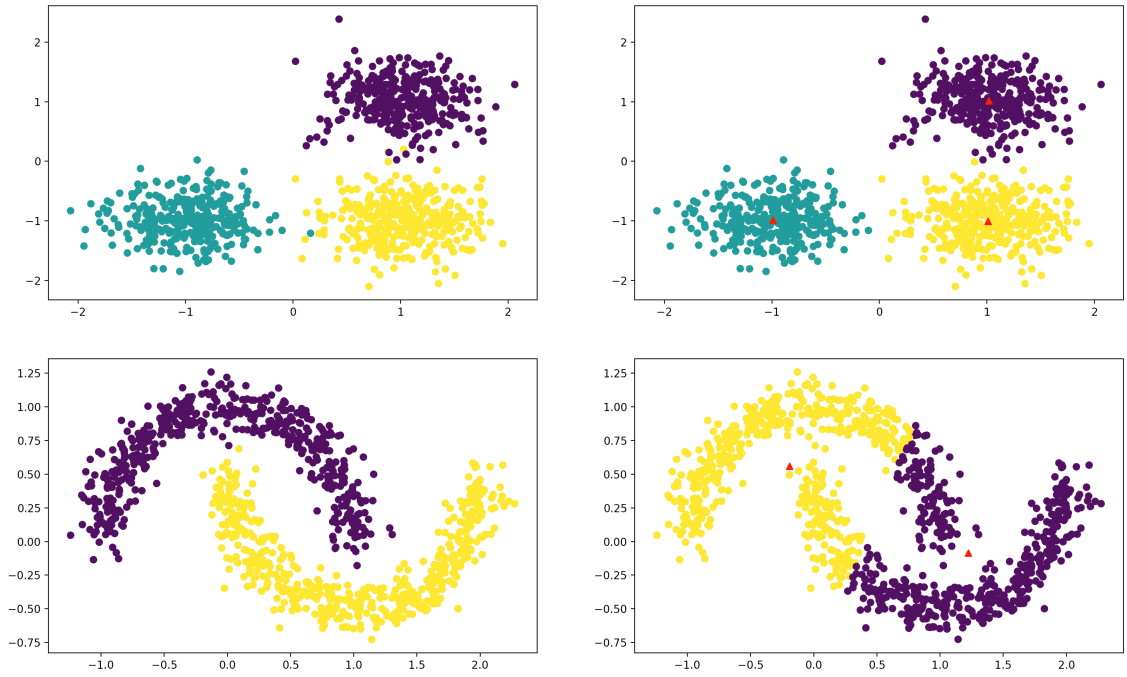
代码：

```
# 导入sklearn相关包
import numpy as np
from sklearn.cluster import KMeans

# 自定义加载数据函数
def loadData(filePath):
  # 以读的兼容模式打开文件
  fr = open(filePath, 'r+')
  # 读取每一行
  lines = fr.readlines()
  # retData用来存储城市各项消费信息
  retData = []
  # retCityName用来存储城市名称
  retCityName = []
  # 将每一行城市信息分别存到retData和retCityName中
  for line in lines:
    items = line.strip().split(",")
    retCityName.append(items[0])
    retData.append([float(items[i]) for i in range(1,len(items))])
```

```
    # 返回retData和retCityName
    return retData, retCityName

# 加载数据，创建K-means算法实例，并进行训练，获得标签
if __name__ == '__main__':
    # 用自定义loadData方法读取数据
    data, cityName = loadData('/Users/xinyuanhe/Desktop/city.txt')
    # 创建实例
    km = KMeans(n_clusters=3)
    # 调用fit_predict进行计算：fit算簇中心，predict指定x中每个点所属于的簇的位置
    label = km.fit_predict(data)
    # 计算不同簇的平均花费
    # np.sum(axis=1)计算每一行向量的和
    # km.cluster_centers_聚类中心
    expenses = np.sum(km.cluster_centers_,axis=1)
    # 创建簇
    CityCluster = [[],[],[]]
    # 将每个样本分到不同簇中
    for i in range(len(cityName)):
        CityCluster[label[i]].append(cityName[i])
    # 打印每个簇的复杂度和簇中样本
    for i in range(len(CityCluster)):
        print("Expenses:%.2f" % expenses[i])
        print(CityCluster[i])
```

结果：

```
= RESTART: /Users/xinyuanhe/Desktop/working/2021美赛/模型/模型-机器学习-聚类-K-MEANS
算法与原型聚类【hxy】/K-means.py
E = 439.72934380858874
E = 607.0190603988688
```



5. 参考资料

1. Mooc机器学习K-Means
2. 13-高晓泓-数据分析模型概览-请勿外传.pdf （P71-P74）
3. sklearn中的make_blobs的用法
4. K-means原理、优化及应用
5. 原型聚类（一）k均值算法和python实现