

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： 大数据2101班**

**学 号： U202115652**

**姓 名： 李嘉鹏**

**指导教师： 王蔚**

**报告日期： 2023年5月11日**

**计算机科学与技术学院**

**目录**

[实验四 kmeans算法及其实现 1](#_Toc135394765)

[**4.1实验目的** 1](#_Toc135394766)

[**4.2 实验内容** 1](#_Toc135394767)

[**4.3 实验过程** 2](#_Toc135394768)

[4.3.1 编程思路 2](#_Toc135394769)

[4.3.2 函数模块定义 4](#_Toc135394770)

[4.3.3 数据结构与存储形式 4](#_Toc135394771)

[4.3.4 对已归一化数据集的理解 5](#_Toc135394772)

[4.3.5 遇到的问题及解决方式 5](#_Toc135394773)

[4.3.6 实验测试与结果分析 6](#_Toc135394774)

[**4.4 实验总结** 7](#_Toc135394775)

[**4.5 参考文献** 7](#_Toc135394776)

# 实验四 kmeans算法及其实现

## **4.1实验目的**

1、加深对聚类算法的理解，进一步认识聚类算法的实现；

2、分析kmeans流程，探究聚类算法原理；

3、掌握kmeans算法核心要点；

4、将kmeans算法运用于实际，并掌握其度量好坏方式。

## **4.2 实验内容**

提供葡萄酒识别数据集，数据集已经被归一化。同学可以思考数据集为什么被归一化，如果没有被归一化，实验结果是怎么样的，以及为什么这样。

同时葡萄酒数据集中已经按照类别给出了1、2、3种葡萄酒数据，在csv文件中的第一列标注了出来，大家可以将聚类好的数据与标的数据做对比。

编写kmeans算法，算法的输入是葡萄酒数据集，葡萄酒数据集一共13维数据，代表着葡萄酒的13维特征，请在欧式距离下对葡萄酒的所有数据进行聚类，聚类的数量K值为3。

在本次实验中，最终评价kmean算法的精准度有两种，第一是葡萄酒数据集已经给出的三个聚类，和自己运行的三个聚类做准确度判断。第二个是计算所有数据点到各自质心距离的平方和。请各位同学在实验中计算出这两个值。

实验进阶部分：在聚类之后，任选两个维度，以三种不同的颜色对自己聚类的结果进行标注，最终以二维平面中点图的形式来展示三个质心和所有的样本点。效果展示图可如图4.1所示。



图4.1 葡萄酒数据集在黄酮和总酚维度下聚类图像（SSE为距离平方和，Acc为准确率）

## **4.3 实验过程**

### 4.3.1 编程思路

本次实验主要是通过kmeans算法对给定的葡萄酒数据进行分类（3种类别）。Kmeans算法会将给定数据集分成k个簇，且每个簇的中心采用簇中所有点在对应维度上数据的平均值。

具体的编程思路如下：

1. 使用csv.reader()函数从“归一化数据.csv”中读取各项数据，并以float的形式存储。这样做的原因是数据集中的归一化数据均为小数类型；
2. 迭代准备：使用rd.random()函数随机生成三个13维的初始中心点；
3. 迭代开始：每轮迭代中，首先按照距离最近原则，计算每一项输入数据到三个中心点的距离，并选取距离最小的中心点作为该点所在簇的中心点，将其划分至这个簇中。每轮遍历结束后，重新按照三个簇中所有点在13维上数据的平均值计算出新的中心点。重复这一过程，循环迭代。
4. 迭代结束：若某轮迭代最后，数据集某一项所在簇的中心点相对前一轮迭代发生变化，或者是它与中心点的距离发生变化，说明迭代没有收敛。反之，从这一轮开始之后的所有迭代结果均不会发生变化，因此迭代结束。
5. 计算三个聚类的SSE（误差平方和）。它的计算公式是：

其中是13维数据中的一项。SSE值越小，代表数据项越接近其各自的中心点，聚类效果越好。因此计算SSE可以评估最终聚类效果。

1. 使用get\_acc()函数计算最终分类的准确度。准确度的计算公式是：

其中“正确分类的项目数”是指三个类别中正确分类的数量之和，每个类别中，以出现次数最多的种类作为这个类的种类。

1. 使用matplotlib中的pyplot.scatter()函数绘制聚类示意图，图中三个簇分别对应红、蓝、绿三种不同的颜色。用户可以输入任意两个维度，得到的图形也具有多样性。

整体的程序流程图如图4-1所示。

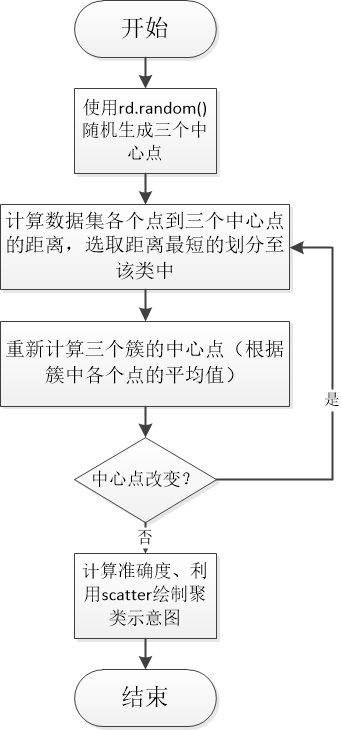


图4-1：程序流程图

上述迭代过程的代码实现如下。

|  |
| --- |
| **while** change:  count += 1  change = **False  for** i **in** range(m):  min\_s = 999999.0  min\_center = -1  **for** j **in** range(k):  distance = len\_s(centers[j], data[i])  **if** distance < min\_s:  min\_center = j + 1  min\_s = distance *# 计算每个点到中心点的距离并得到最小值* **if** data\_min[i, 0] != min\_center **or** data\_min[i, 1] != min\_s: *#中心点变化，或者是到中心点的距离发生变化* data\_min[i, :] = min\_center, min\_s  change = **True** *# 更新每个点的最近中心点和距离信息* **for** j **in** range(k):  point = get\_points(m, data, j, data\_min) *# 根据求得的每个点的信息分类* centers = np.array(centers)  centers[j, :] = np.mean(point, axis=0) *# 计算出新的中心点* |

计算聚类准确度的代码实现如下。

|  |
| --- |
| **def** get\_acc(m, data, k, data\_min):  hit = 0  **for** j **in** range(k):  cluster\_tmp = [0, 0, 0]  point = get\_points(m, data, j, data\_min)  **for** item **in** point:  cluster\_tmp[int(item[0])-1] += 1  hit += max(cluster\_tmp)  acc = hit / len(data)  print(**'准确度(acc)：'**, acc) *# 准确度为正确分类数与总数之比* |

### 4.3.2 函数模块定义

**（1）len\_s**

①函数参数：center, data

②函数功能：计算数据集中某个点到中心点的距离。

**（2）ssenum**

①函数参数：m, data\_min

②函数功能：分别计算三种聚类的误差平方和SSE。

**（3）get\_points**

①函数参数：m, data, j, data\_min

②函数功能：根据求得的数据集中每个点的信息（与其最近的中心点）将其分配到三个聚类中。

**（4）get\_acc**

①函数参数：m, data, k, data\_min

②函数功能：迭代结束后，计算最终的聚类准确度。

### 4.3.3 数据结构与存储形式

**（1）data**

①类型：二维数组

②作用：存储归一化数据集中每一项的信息。

**（2）data\_min**

①类型：二维数组

②作用：分别存储某个数据所在簇的中心点及其与中心点的距离。

**（3）centers**

①类型：列表

②作用：存储三个中心点的13维坐标。

**（4）clusters\_tmp**

①类型：列表

②作用：计算聚类准确度时，存储0、1、2三种类别的分类数并与真实值进行比较。

**（5）sse\_num**

①类型：列表

②作用：存储三个聚类各自的误差平方和SSE。

### 4.3.4 对已归一化数据集的理解

任务书前面提出“数据集已经被归一化”，并让我思考归一化的作用以及不进行归一化带来的影响。

我认为此处对原始数据集进行归一化可以使数据处理更加方便，它将13维的葡萄酒数据限制在0~1的区间范围内，使各个维度的数据在决定数据点的特征上具有同等的地位。

如果没有进行归一化，原始数据集会放大某些因素对聚类的影响，同时会在很大程度上忽略某些因素。如图4-2所示，可以看到第一维数据范围大约在13~15之间，但第二维数据范围大约在1~3之间，二者的尺度差别太大，需要在放缩后才能使用。

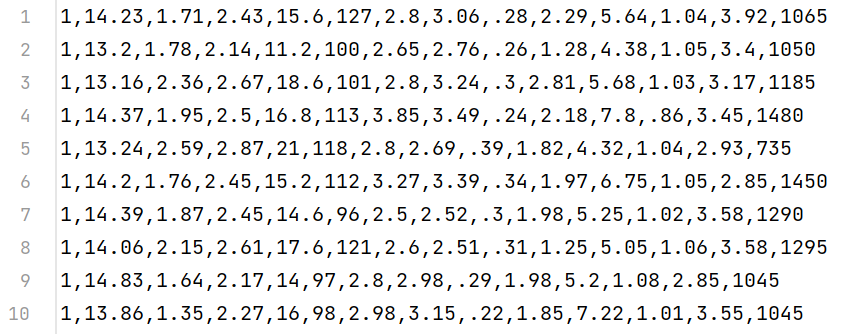


图4-2：原始数据集的13维数据示意图

### 4.3.5 遇到的问题及解决方式

实验中主要遇到了两个问题。第一是最开始用pyplot.scatter()函数绘制图像时，没有弹出聚类结果图，后来查阅资料发现，要再加一句plt.show语句才能把绘制好的图形展示出来。第二是在判断迭代终止条件时，一开始只认为数据集中每个点对应的中心点不变就能说明聚类完成，实际上还存在一种可能的情形就是到中心点的距离发生变化，这时该点需要再次计算到其它中心点的距离，因此可能会划分到另一个簇中。

### 4.3.6 实验测试与结果分析

运行kmans.py，程序输出了循环次数、三个聚类各自的SSE、聚类准确度，如图4-3所示。可以看出三个聚类的SSE分别为19.998、15.056、13.935，最终的准确度为0.9438，拟合效果较好，因此程序完成了应有的功能。

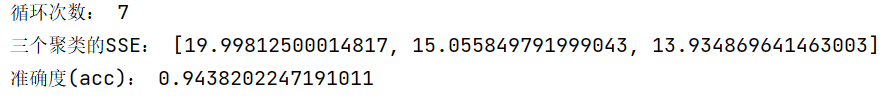


图4-3：聚类结束后输出的信息

此后程序提示输入两个需要绘制聚类关系图的属性，以1~13表示。输入“1 2”并回车，程序自动绘制出属性1和属性2的聚类关系图，如图4-4、图4-5所示。

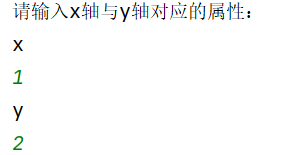


图4-4：输入要绘制的属性

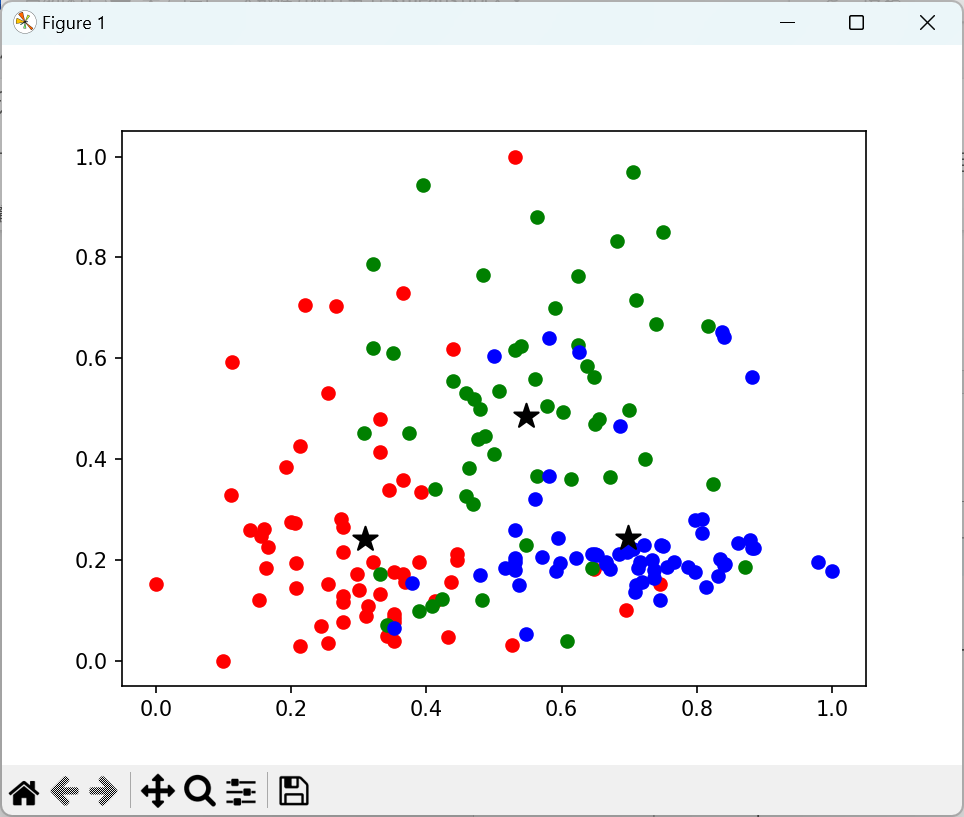


图4-5：属性1和属性2的聚类关系图

上图中，三个聚类的中心点用五角星“★”表示。

## **4.4 实验总结**

本次实验中，我通过对kmeans聚类算法的实现，掌握了一种重要的聚类处理算法。在面对大规模数据集时，聚类可以更快地掌握样本数据集的大体分布与特征。使用聚类算法时，要特别注意对所给数据进行归一化，以便于更好地保证迭代快速收敛。

## **4.5 参考文献**

[1] K-Means（K-均值）聚类算法<https://blog.csdn.net/sikh_0529/article/details/126806720>

[2] 聚类算法数据预处理——数据归一化 <https://it.cha138.com/jingpin/show-32640.html>