**聚类实验要求和内容**

1. 实验目的

1. 理解 k-means 算法的算法过程，并能够使用 Python 3 中 sklearn 包所提供的 KMeans 聚类包对数据集进行聚类

1. 理解层次聚类算法的算法过程，并能够使用 Python 3 中 scipy 包提供的hierarchy 方法对 sklearn 中一个特定大的数据集 iris 进行聚类

二、 算法过程

调用python里的库进行聚类实现

三、 实验步骤

(一)层次聚类

**实验先验知识：**

接下来介绍下层次聚类主要函数及参数含义：

(1)linkage(y, method=’single’, metric=’euclidean’)   
共包含3个参数:   
y是距离矩阵,由hcluster. distance.pdist方法得到;method是指计算类间距离的方法,比较常用的有3种:   
1)single:最近邻,把类与类间距离最近的作为类间距   
2)complete:最远邻,把类与类间距离最远的作为类间距   
3)average:平均距离,类与类间所有pairs距离的平均

(2)fcluster(Z, t, criterion, depth, R=None, monocrit=None)

第一个参数Z是linkage得到的矩阵,记录了层次聚类的层次信息;

criterion这个参数代表了判定条件，这里详细解释下各个参数的含义：

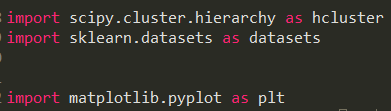
1). 当criterion为’inconsistent’时，t值应该在0-1之间波动，t越接近1代表两个数据之间的相关性越大，t越趋于0表明两个数据的相关性越小。这种相关性可以用来比较两个向量之间的相关性，可用于高维空间的聚类

2). 当criterion为’distance’时，t值代表了绝对的差值，如果小于这个差值，两个数据将会被合并，当大于这个差值，两个数据将会被分开。

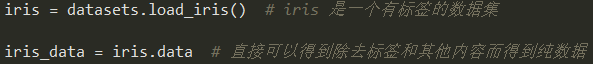
3). 当criterion为’maxclust’时,t代表了最大的聚类的个数，设置4则最大聚类数量为4类，当聚类满足4类的时候，迭代停止

**实验具体实施：**

1.在进行实验前，需要调用两个库，第一个是实现层次聚类的scipy库，第二个是iris数据集库



2.首先调用iris数据集，并得到原始数据：



3.接着使用hcluster包的函数实现层次聚类：

#生成点与点之间的距离矩阵,这里用的欧氏距离

disMat = hcluster.distance.pdist(iris\_data, 'euclidean')

#进行层次聚类:

Z = hcluster.linkage(disMat,method='average')

#将层级聚类结果以树状图表示出来

P = hcluster.dendrogram(Z)

#示例效果如图：



4.#根据linkage matrix 得到聚类结果:

cluster= hcluster.fcluster(Z, t=6, 'maxclust')

注意到这里使用的参数criterion为’maxclust’，t代表一共聚成6类，

也可以用先验知识的其他参数值来进行不同的层次聚类。

5.完成可视化工作

利用 matplotlib 中的 pyplot 包进行数据可视化

显示出结果。（要求：将iris\_data前二维作为平面点的坐标在坐标系中表示出来，相同聚类节点颜色相同。）

最后的大致效果：



(二)K-means 聚类

**实验先验知识：**

简介下KMeans包的主要函数及方法

#调用kmeans类

clf = KMeans(n\_clusters=9)

s = clf.fit(dataset)

#9个中心

print clf.cluster\_centers\_

#每个样本所属的簇

print clf.labels\_

#用来评估簇的个数是否合适，距离越小说明簇分的越好，选取临界点的簇个数

print clf.inertia\_

#进行预测

print clf.predict(dataset)

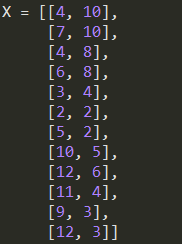
**实验具体实施：**

1．由于需要使用 sklearn 包，所以需要先进行安装，可直接在 Python 3 运行终端输入语句：pip install sklearn

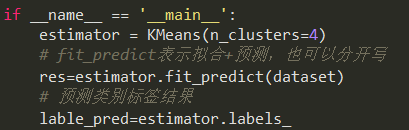
2.需要调用的包有两个，第一个是调用其聚类包 cluster，第二个是一个可视化工具

1530689662(1)

3.该部分使用的数据集如下图所示：

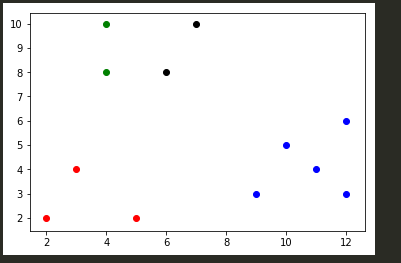


4.调用 KMeans 包中的函数，并且可以设置类数以及对输入的数据集进行预测。这个调用方法可以自行上网搜索查询或根据先验知识

具体实现时，可以设置为4个中心，将数据集聚类成4个簇

由之前函数讲解以知道，聚类的个数是4，且lable\_pred区分了不同聚类（0,1,2,3），res存储了聚类结果。

5.完成可视化工作，利用 matplotlib 中的 pyplot 包进行数据可视化显示出结果。 分别将不同聚类用不同颜色的点标记出来，可视化结果如下图：



四、思考题

UCI数据库是加州大学欧文分校(University of CaliforniaIrvine)提出的用于机器学习的数据库，这个数据库目前共有488个数据集，其数目还在不断增加，UCI数据集是一个常用的标准测试数据集。

其中上面实验用到的Iris数据集就是其中的一小部分，Iris数据集分为3类，每类样本有4个属性，样本数量为50；

在这类我们使用另一个wine葡萄酒样本，来尝试不同样本的层次聚类和K-means聚类算法不同效果，并和数据集标签对比，判断其准确率。

Wine数据集简介：

该实验的数据源是Wine data，这是对在意大利同一地区生产的三种不同品种的酒，做大量分析所得出的数据。如下图所示：

图片包含 屏幕截图

描述已自动生成

这些数据包括了三种酒中13种不同成分的数量。13种成分分别为：Alcohol，Malicacid，Ash，Alcalinity of ash，Magnesium，Total phenols，Flavanoids，Nonflavanoid phenols，Proanthocyanins，Color intensity，Hue，OD280/OD315 of diluted wines，Proline。在 “wine.data”文件中，每行代表一种酒的样本，共有178个样本；一共有14列，其中，第一列为类标志属性，共有三类，分别记为“1”，“2”，“3”；后面的13列为每个样本的对应属性的样本值。其中第1类有59个样本，第2类有71个样本，第3类有48个样本。

五、实验总结

总结自己对聚类算法的认识以及在实现过程中遇到的困难和解决方法