浙江工业大学

数据挖掘实验

计算机科学与技术学院

聚类分析

一、实验目的

掌握 K-均值、层次聚类算法，理解重要参数对结果的影响。

二、实验内容

1、k 均值聚类

从 scikit learn 导入 wine 数据集（真实类别为 3），实现 k 均值聚类，输出 cluster_label 向量表示聚类结果。计算并输出 NMI 值。可以直接调用 sklearn metrics 中的函数 metrics.normalized_mutual_info_score，具体为：

```
from sklearn import metrics
NMI_score= metrics.normalized_mutual_info_score(labels_true, labels)
print("normalized Mutual Information: %0.3f"
      % NMI_score)
========================================
```

可参考以下框架实现 k 均值算法：

输入： 数据 X， 簇的个数 K, 最大迭代次数 Max_iter, 最小改变量 eps

输出：label(代表样本所在簇的 id)

step 1. 初始化：随机选择 K 个样本作为中心点， 设置迭代次数 t=0;

step 2. 重复以下交叉迭代直到 t> Max_iter 或 改变量 Eps < eps

a. 更新 cluster assignment: 对每个样本，计算其到 K 个中心点的距离，把该样本分配到离他最近的中心对应的簇；

b.更新中心点：保存旧的 K 个中心点；对每个簇 c，计算属于该簇的样本的均值作为其新的中心点。计算旧的中心点δ_c^t和新的中心点δ_c^{t+1}的改变量$Eps_c = \|\delta_c^t - \delta_c^{t+1}\|$。

t=t+1; Eps=$max_c(Eps_c)$ (K 个中心点改变量的最大值)

==========

2、层次聚类

2.1 基于欧式距离，分别以最短距离（single linkage）、平均距离（average linkage）为簇间距离度量对 wine 数据集进行凝聚层次聚类，在簇数目为 3 的情况下比较层次聚类与 k 均值的 NMI。

```
#以下为层次聚类调用方法

from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram


model = AgglomerativeClustering(affinity=’ euclidean’ ,
distance_threshold=None, n_clusters=3, linkage=’ single’ )
model = model.fit(X)
labels=model.labels_
```

2.2 计算 manhattan 距离矩阵 D，以预先计算好的距离选项调用凝聚层次聚类，重复 1）中实验，比较与欧式距离的结果。

```
model = AgglomerativeClustering(affinity=’ precomputed’ ,
distance_threshold=None, n_clusters=3, linkage=’ single’ )
model = model.fit(D)
```

3、密度聚类（选做）

3.1 对 2D 数据集 noisy_moons(从外部文件夹导入)，打印散点图（用不同颜色表示不同的类）

3.2 调用 DBSCAN 算法，设置不同的 eps 和 MinPts，比较散点图结果并讨论。具体包括以下三种情况：eps=0.3,MinPts=10；eps=0.3,MinPts=5；eps=0.1，MinPts=10。

```
# DBSCAN 调用，返回每个样本对应的簇，若簇 id 为-1 则表示该样本被判断为噪声

from sklearn.cluster import DBSCAN

db = DBSCAN(eps=0.3, min_samples=10).fit(X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
```