链滴

AGNES 算法

作者：Lonery

原文链接：https://ld246.com/article/1611192828207

来源网站：链滴

层次凝聚算法——AGNES

AGNES算法是一种基于层次凝聚的聚类算法，它的思想十分朴素。假设现在有一个待聚类的数据集那么根据分而治之的思想我们可以首先将每一个样本点看成是一个类，然后根据一定的规则将这些比“小”的类进行合并，进而达到最终想要的结果。

那么这个合并的规则是什么？通常我们将样本点之间的距离看成相似度。在两个小类中，第一个类和二个类中的点它们之间的距离有很多，如果第一个类有n个样本点，第二个类有m个样本点，那么不的类点和点之间的距离就会有m*n个，到底如何定义这个规则呢？ 一般而言我们有三种方式可采用，同方式聚类出来的效果可能也不尽相同，即，最小距离，最大距离和平均距离，定义方式皆为字面意。最小距离就是分处在两个小类中的距离最近的两个点，两点分别处于第一个小类和第二个小类。最距离也是如此，平均距离就是将两个类之间的点所有距离加权求和，即所有距离除以距离的个数得出距离就是平均距离。

AGNES算法的伪代码:
</br>

输入：样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$；
聚类簇距离度量函数 d；
聚类簇数 k.

过程：

```
for j = 1,2,...,m do
    C_j = {x_j}
end for
for i = 1,2,...,m do
    for j = 1,2,...,m do
        M(i,j) = d(C_i, C_j);
        M(j,i) = M(i,j)
    end for
end for
设置当前聚类簇个数: q = m
while q > k do
    找出距离最近的两个聚类簇 C_i* 和 C_j*;
    合并 C_i* 和 C_j*: C_i* = C_i* ⋃ C_j*;
    for j = j* + 1, j* + 2, ..., q do
        将聚类簇 C_j 重编号为 C_{j-1}
    end for
    删除距离矩阵 M 的第 j* 行与第 j* 列;
    for j = 1,2,...,q-1 do
        M(i*, j) = d(C_i*, C_j);
        M(j, i*) = M(i*, j)
    end for
    q = q - 1
end while
```

输出：簇划分 $C=\{C_1,C_2,\ldots,C_k\}$

代码实现：

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets

def findmin(mat):   #找出距离最小的两个类
    temp=mat[0][1]
    x=0
    y=0
    for i in range(mat.shape[0]):
        for j in range(mat.shape[1]):
            if mat[i][j]<temp and i!=j:
                x=i
                y=j
                temp=mat[i][j]
    return [x,y,temp]
def dis(X,a,b):  #采用最大距离
    temp=0
    for i in a:
        for j in b:
            T=np.linalg.norm(X[i]-X[j],ord=2)
            if temp<T:
                temp=T
    return temp
def AGNES(X,k):
    C=[]
    m=len(X)
    for j in range(m):
        C.append([j])
    M=np.zeros((m,m))
    for i in range(m):
        for j in range(i):
            M[i][j]=np.linalg.norm(X[i]-X[j],ord=2)
            M[j][i]=M[i][j]
    q=m
    while q>k:
        [x,y,m]=findmin(M)
        C[x].extend(C[y])
        C.pop(y)

        M=np.delete(M,y,axis=0)  #删除第J行和第J列
        M=np.delete(M,y,axis=1)
        for j in range(q-1):
            M[x][j]=dis(X,C[x],C[j])  #更新距离(这一步可以优化，因为在代码开始阶段就已将所有点之
的距离计算完成。)
            M[j][x]=M[x][j]
        q=q-1
    return C

def plot(X,c):
    for i in c:
        x=[]
        y=[]
        for j in i:
            x.append(X[j][0])
```

```
            y.append(X[j][1])
        plt.scatter(x,y)
    plt.show()

if __name__=="__main__":
    X=datasets.make_blobs(n_samples=300,centers=3,cluster_std=1.0,shuffle=True,random_sta
e=None)[0]
    c=AGNES(X,3)
    plot(X,c)
```

测试结果：