模型-机器学习-聚类-AGNES算法与层次聚类

1. 模型名称

AGglomerative NESting，AGNES

2. 模型评价

2.1 优点

- 其参数很少，只需要输入要分类的总数以及数据样本即可

2.2 缺点

- 算法慢得多

3. 基本算法

输入：包含个对象的数据库
输出：满足终止条件的若干个簇

1. 将每个对象当成一个初始簇
2. 计算任意两个簇的距离，并找到最近的两个簇

   距离计算

   方法一：**最小距离**（对应**单链接算法**）

$$d_{min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} |p - q|$$

   方法二：**最大距离**（对应**全链接算法**）

$$d_{max}(C_i, C_j) = \max_{p \in C_i, q \in C_j} |p - q|$$

方法三：**均值距离**

$$d_{mean}(C_i, C_j) = |\overline{p} - \overline{q}| \quad \overline{p} = \frac{1}{|C_i|} \sum_{p \in C_i} p, \ \overline{q} = \frac{1}{|C_j|} \sum_{q \in C_j} q$$

方法四：**平均距离**（对应**均链接算法**）

$$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{p \in C_i} \sum_{q \in C_j} |p - q|$$

3. 合并两个簇，生成新的簇的集合
4. 重复2、3步直到终止条件得到满足

4. 实例

4.1 用Sklearn自带函数进行iris数据聚类

4.1.1 数据介绍

数据概览网址

鸢尾属约300种。Iris数据集中包含了其中的三种：山鸢尾(Setosa)，杂色鸢尾(Versicolour)，维吉尼亚鸢尾(Virginica)，每种50个数据, 共含150个数据。在每个数据包含四个属性：花萼长度，花萼宽度，花瓣长度，花瓣宽度，可通过这四个属性预测鸢尾花卉属于（山鸢尾，杂色鸢尾，维吉尼亚鸢尾）哪一类

4.1.2 实验目的

通过4个属性预测鸢尾花卉属于（山鸢尾，杂色鸢尾，维吉尼亚鸢尾）哪一类

4.1.3 代码实现

AGNES.py

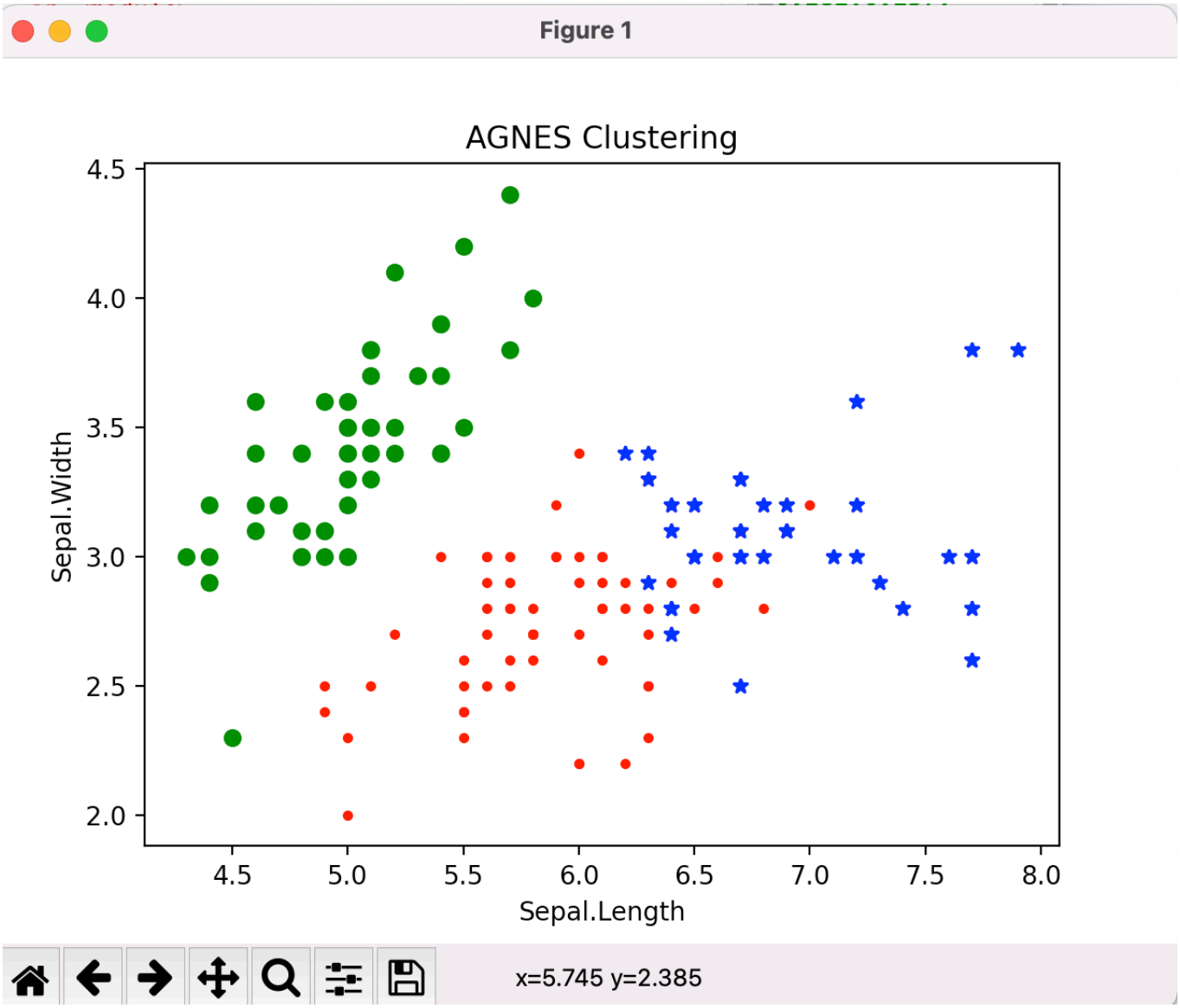
代码：

```
# 导入sklearn相关包
from sklearn import datasets
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import confusion_matrix
# 导入matplotlib
import matplotlib.pyplot as plt
# 导入pandas
import pandas as pd

# 加载iris数据
iris = datasets.load_iris()
irisdata = iris.data

# 建立AGNES模型
# linkage是一个字符串，用于指定链接算法，ward表示采用单链接，complete表示全链接，average表示均链接
```

```
# n_clusters指定分类簇的数量
clustering = AgglomerativeClustering(linkage='ward', n_clusters=3)

# 输入iris数据进行训练
res = clustering.fit(irisdata)

# 打印各个簇的样本数目
print('各个簇的样本数目：')
print(pd.Series(clustering.labels_).value_counts())
# 打印聚类结果
print('聚类结果：')
print(confusion_matrix(iris.target,clustering.labels_))

# 可视化
plt.figure()
# labels_为0的点
d0 = irisdata[clustering.labels_==0]
plt.plot(d0[:,0],d0[:,1],'r.')
# labels_为1的点
d1 = irisdata[clustering.labels_==1]
plt.plot(d1[:,0],d1[:,1],'go')
# labels_为2的点
d2 = irisdata[clustering.labels_==2]
plt.plot(d2[:,0],d2[:,1],'b*')
# 设置xlabel和ylabel
plt.xlabel('Sepal.Length')
plt.ylabel('Sepal.Width')
# 设置title
plt.title('AGNES Clustering')
# 显示
plt.show()
```

结果：

```
= RESTART: /Users/xinyuanhe/Desktop/working/2021美赛/模型/模型-机器学习-聚类-AGNES算法与层次聚类【hxy】/AGNES.py
各个簇的样本数目：
0    64
1    50
2    36
dtype: int64
聚类结果：
[[ 0 50  0]
 [49  0  1]
 [15  0 35]]
```



4.2 自定义函数实现西瓜聚类

4.2.1 数据介绍

每三个是一组分别是西瓜的编号，密度，含糖量

```
1,0.697,0.46,2,0.774,0.376,3,0.634,0.264,4,0.608,0.318,5,0.556,0.215,
6,0.403,0.237,7,0.481,0.149,8,0.437,0.211,9,0.666,0.091,10,0.243,0.267,
11,0.245,0.057,12,0.343,0.099,13,0.639,0.161,14,0.657,0.198,15,0.36,0.37,
16,0.593,0.042,17,0.719,0.103,18,0.359,0.188,19,0.339,0.241,20,0.282,0.257,
21,0.748,0.232,22,0.714,0.346,23,0.483,0.312,24,0.478,0.437,25,0.525,0.369,
26,0.751,0.489,27,0.532,0.472,28,0.473,0.376,29,0.725,0.445,30,0.446,0.459
```

4.2.2 实验目的

根据西瓜的密度和含糖量对西瓜进行聚类

4.2.3 代码实现

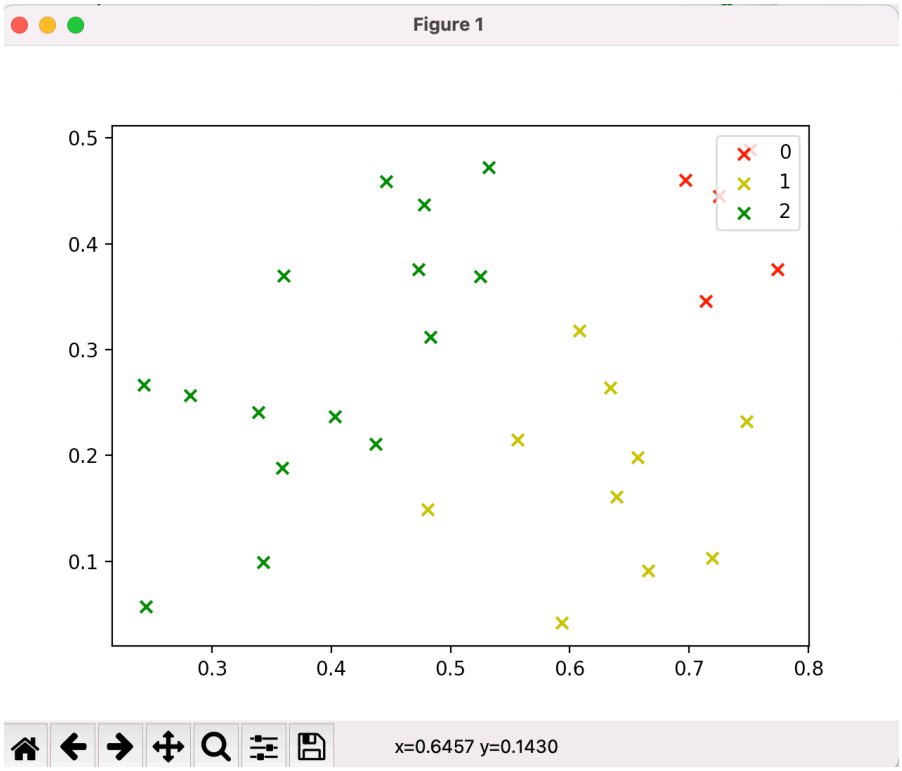
AGNES2.py

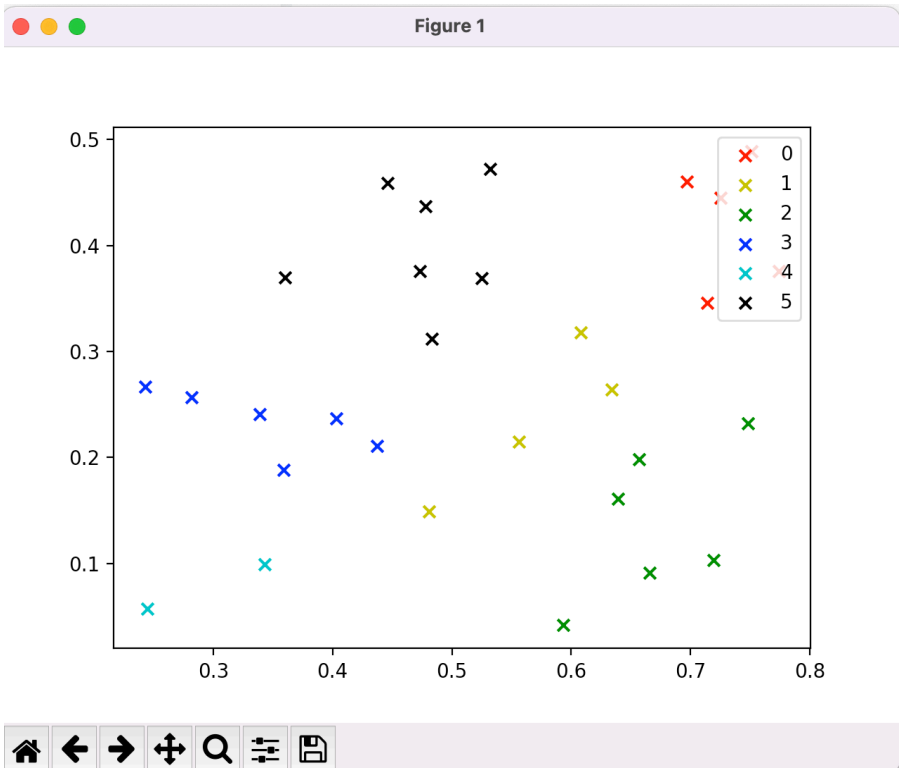
代码：

```
import math
import pylab as pl

#数据集：每三个是一组分别是西瓜的编号，密度，含糖量
data = """
1,0.697,0.46,2,0.774,0.376,3,0.634,0.264,4,0.608,0.318,5,0.556,0.215,
6,0.403,0.237,7,0.481,0.149,8,0.437,0.211,9,0.666,0.091,10,0.243,0.267,
11,0.245,0.057,12,0.343,0.099,13,0.639,0.161,14,0.657,0.198,15,0.36,0.37,
16,0.593,0.042,17,0.719,0.103,18,0.359,0.188,19,0.339,0.241,20,0.282,0.257,
21,0.748,0.232,22,0.714,0.346,23,0.483,0.312,24,0.478,0.437,25,0.525,0.369,
26,0.751,0.489,27,0.532,0.472,28,0.473,0.376,29,0.725,0.445,30,0.446,0.459"""

#数据处理 dataset是30个样本（密度，含糖量）的列表
a = data.split(',')
dataset = [(float(a[i]), float(a[i+1])) for i in range(1, len(a)-1, 3)]



#计算欧几里得距离,a,b分别为两个元组
def dist(a, b):
    return math.sqrt(math.pow(a[0]-b[0], 2)+math.pow(a[1]-b[1], 2))

#dist_min
def dist_min(Ci, Cj):
    return min(dist(i, j) for i in Ci for j in Cj)
#dist_max
def dist_max(Ci, Cj):
    return max(dist(i, j) for i in Ci for j in Cj)
#dist_avg
def dist_avg(Ci, Cj):
    return sum(dist(i, j) for i in Ci for j in Cj)/(len(Ci)*len(Cj))

#找到距离最小的下标
def find_Min(M):
    min = 1000
    x = 0; y = 0
    for i in range(len(M)):
        for j in range(len(M[i])):
            if i != j and M[i][j] < min:
                min = M[i][j];x = i; y = j
    return (x, y, min)

#算法模型：
```

```
def AGNES(dataset, dist, k):
    #初始化C和M
    C = [];M = []
    for i in dataset:
        Ci = []
        Ci.append(i)
        C.append(Ci)
    for i in C:
        Mi = []
        for j in C:
            Mi.append(dist(i, j))
        M.append(Mi)
    q = len(dataset)
    #合并更新
    while q > k:
        x, y, min = find_Min(M)
        C[x].extend(C[y])
        C.remove(C[y])
        M = []
        for i in C:
            Mi = []
            for j in C:
                Mi.append(dist(i, j))
            M.append(Mi)
        q -= 1
    return C
#画图
def draw(C):
    colValue = ['r', 'y', 'g', 'b', 'c', 'k', 'm']
    for i in range(len(C)):
        coo_X = []    #x坐标列表
        coo_Y = []    #y坐标列表
        for j in range(len(C[i])):
            coo_X.append(C[i][j][0])
            coo_Y.append(C[i][j][1])
        pl.scatter(coo_X, coo_Y, marker='x', color=colValue[i%len(colValue)], label=i)

    pl.legend(loc='upper right')
    pl.show()

C = AGNES(dataset, dist_avg, 3)
# C = AGNES(dataset, dist_avg, 5)
draw(C)
```

结果：





5. 参考资料

1. 机器学习算法-层次聚类AGNES-自带库函数实现
2. 聚类算法——python实现层次聚类（AGNES）
3. AGNES聚类算法的理解与应用C++