**scikit-learn——层次聚类**

**任务目标**

1.了解层次聚类的原理

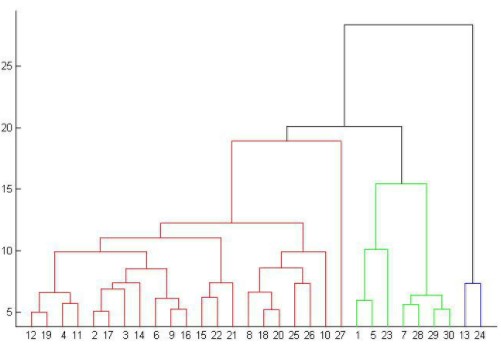
2.掌握sklearn中AgglomerativeClustering的使用

**相关知识**

层次聚类

层次聚类，是一种很直观的算法。顾名思义就是要一层一层地进行聚类，可以从下而上地把小的cluster合并聚集，也可以从上而下地将大的cluster进行分割。似乎一般用得比较多的是从下而上地聚集，代表算法有：CURE、CHAMELEON、ROCK、Hierarchical等因此这里我就只介绍这一种。

所谓从下而上地合并cluster，具体而言，就是每次找到距离最短的两个cluster，然后进行合并成一个大的cluster，直到全部合并为一个cluster。整个过程就是建立一个树结构，类似于下图。

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/e0a72992-16ff-4a4b-b5d6-c9a986e613a5.jpg)

AggregativeClustering是一种常用的层次聚类算法。

  其原理是：最初将每个对象看成一个簇，然后将这些簇根据某种规则被一步步合并，就这样不断合并直到达到预设的簇类个数。这里的关键在于：如何计算聚类簇之间的距离？

  由于每个簇就是一个集合，因此需要给出集合之间的距离。给定聚类簇Ci，Cj有如下三种距离：

最小距离：

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/f7d91c9f-cff8-4553-8f9b-5453cee39629.png)

它是两个簇的样本对之间距离的最小值。

最大距离：

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/599e80c7-90d0-471c-b3a8-feda99fa8bf9.png)

它是两个簇的样本对之间距离的最大值。

平均距离：

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/e2763e06-0962-4b8e-b7ad-f904d8886733.png)

它是两个簇的样本对之间距离的平均值。

当该算法的聚类簇采用dmin时，称为单链接single-linkage算法，当该算法的聚类簇采用dmax时，称为单链接complete-linkage算法，当该算法的聚类簇采用davg时，称为单链接average-linkage算法。下面给出算法：

输入：

* 数据集D=

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/59e16e88-493c-41db-9075-8f936ed1e6e2.png)

* 聚类簇距离度量函数
* 聚类簇数量K

输出：簇划分C=C={C1,C2,...,CK}

算法步骤如下：

* setp1:每个样本作为单独一类（c1,c2,...cn）
* setp2:计算每2类的距离d
* setp3:找出最小的d，合并这2类
* setp4:判断剩余类数是否是达到预类数（可以自己设置），没有达到，返回setp2，达到，跳出循环，输出每层类别

优点：

1、距离和规则的相似度容易定义，限制少；

2、可以发现类的层次关系；

3、可以聚类成其它形状

4、适用于任意形状和任意属性的数据集；灵活控制不同层次的聚类粒度，强聚类能力

缺点：1、计算复杂度太高；

2、奇异值也能产生很大影响；

3、算法很可能聚类成链状

4、大大延长了算法的执行时间，不能回溯处理；

5、需要预设类别数

AgglomerativeClustering是scikit-learn提供的层级聚类算法模型，其原型为：

[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

1. **class** sklearn.cluster.AgglomerativeClustering(n\_clusters=2, affinity=’euclidean’, memory=None, connectivity=None, compute\_full\_tree=’auto’, linkage=’ward’, pooling\_func=<**function** mean>)

**参数说明：**

n\_clusters：一个整数，指定分类簇的数量

connectivity：一个数组或者可调用对象或者None，用于指定连接矩阵

affinity：一个字符串或者可调用对象，用于计算距离。可以为：’euclidean’，’l1’，’l2’，’mantattan’，’cosine’，’precomputed’，如果linkage=’ward’，则affinity必须为’euclidean’

memory：用于缓存输出的结果，默认为不缓存

n\_components：在 v-0.18中移除

compute\_full\_tree：通常当训练了n\_clusters后，训练过程就会停止，但是如果compute\_full\_tree=True，则会继续训练从而生成一棵完整的树

linkage：一个字符串，用于指定链接算法

‘ward’：单链接single-linkage，采用dmindmin

‘complete’：全链接complete-linkage算法，采用dmaxdmax

‘average’：均连接average-linkage算法，采用davgdavg

pooling\_func：一个可调用对象，它的输入是一组特征的值，输出是一个数

**属性**

labels：每个样本的簇标记

n\_leaves\_：分层树的叶节点数量

n\_components：连接图中连通分量的估计值

children：一个数组，给出了每个非节点数量

**方法**

fit(X[,y])：训练样本

fit\_predict(X[,y])：训练模型并预测每个样本的簇标记

**系统环境**

Linux Ubuntu 16.04

Python 3.6.1

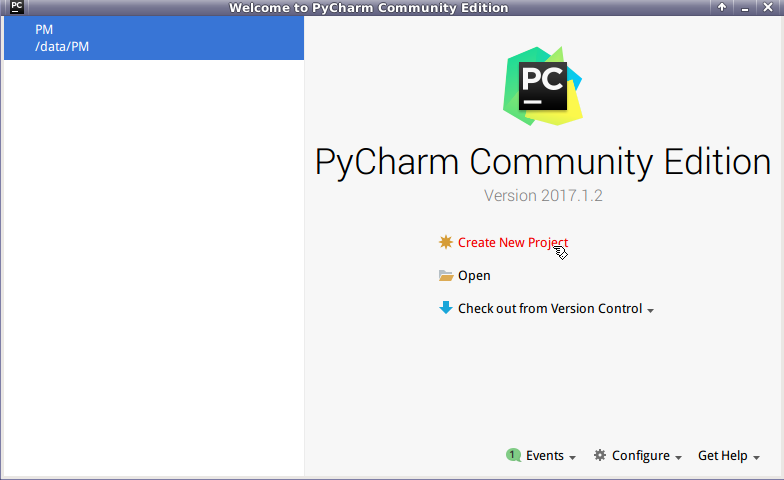
PyCharm

**任务内容**

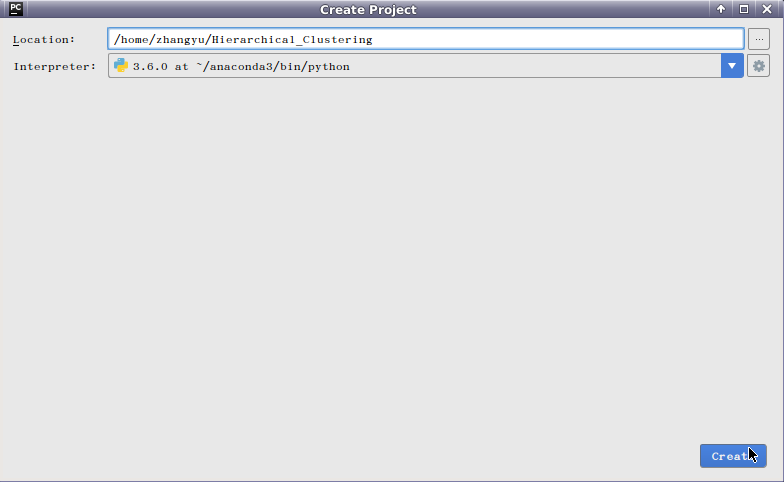
练习sklearn中层次聚类AgglomerativeClustering的操作过程。

**任务步骤**

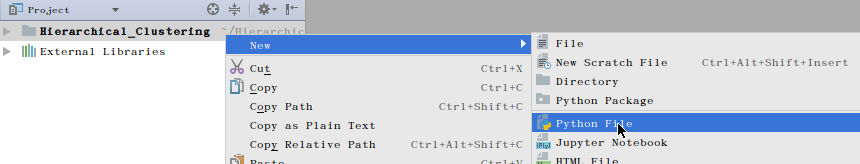
1.打开Pycharm，选择Create New Project，

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/804e776a-36d4-4c45-bbdc-c811ace41b67.png)

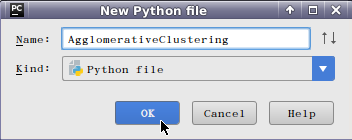
创建名为Hierarchical\_Clustering的项目。

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/e11397df-5296-406f-9386-3c4ac4517705.png)

2.打开Hierarchical\_Clustering项目，右键选择New=>Python File，

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/56520792-0830-49af-8480-574b56d063ae.png)

创建名为AgglomerativeClustering的Python文件。

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/f1d02d55-eec2-45ab-87f7-fa8585fe5124.png)

3.打开AgglomerativeClustering.py文件，编写代码，用于密度聚类。

导入实验所需要的模块

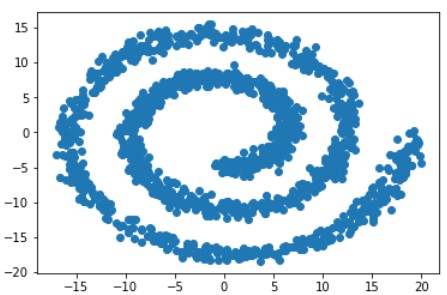
[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

1. **import** time
2. **import** matplotlib.pyplot as plt
3. from sklearn **import** metrics
4. **import** numpy as np
5. from sklearn.cluster **import** AgglomerativeClustering
6. from sklearn.neighbors **import** kneighbors\_graph

4.准备实验数据，X为样本特征，Y为样本簇类别的1500样本，并作图，查看数据的分布。

[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

1. n\_samples = 1500
2. np.random.seed(0)
3. t = 1.5 \* np.pi \* (1 + 3 \* np.random.rand(1, n\_samples))
4. x = t \* np.cos(t)
5. y = t \* np.sin(t)
6. X = np.concatenate((x, y))
7. X += .7 \* np.random.randn(2, n\_samples)
8. X = X.T
9. plt.scatter(X[:, 0], X[:, 1], marker='o')
10. plt.show()

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/3ba55ebf-b295-4988-8c90-e31bdef47133.jpg)

5.使用kneighbors\_graph函数计算X中k个临近点（列表）对应的权重，设置临近点个数为30，不包含自身点。

[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

1. knn\_graph = kneighbors\_graph(X, 30, include\_self=False)

6.用AgglomerativeClustering算法来聚类，使用for循环遍历设置参数的值，connectivity=None, knn\_graph，聚类数参数n\_clusters=30, 3，参数linkage='average', 'complete', 'ward'

作出各种参数下聚类的结果图，并使用Calinski-Harabasz分数评估各个聚类模型好坏。

[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

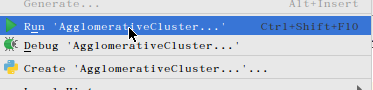
1. **for** connectivity **in** (None, knn\_graph):
2. **for** n\_clusters **in** (30, 3):
3. plt.figure(figsize=(10, 4))
4. **for** index, linkage **in** enumerate(('average', 'complete', 'ward')):
5. plt.subplot(1, 3, index + 1)
6. model = AgglomerativeClustering(linkage=linkage,
7. connectivity=connectivity,
8. n\_clusters=n\_clusters)
9. t0 = time.time()
10. model.fit(X)
11. elapsed\_time = time.time() - t0
12. plt.scatter(X[:, 0], X[:, 1], c=model.labels\_,
13. cmap=plt.cm.spectral)
14. plt.title('linkage=%s (time %.2fs)' % (linkage, elapsed\_time),
15. fontdict=dict(verticalalignment='top'))
16. plt.axis('equal')
17. plt.axis('off')
19. plt.subplots\_adjust(bottom=0, top=.89, wspace=0,
20. left=0, right=1)
21. plt.suptitle('n\_cluster=%i, connectivity=%r' %
22. (n\_clusters, connectivity is not None), size=17)
23. print("n\_cluster=%i connectivity=%r linkage=%s of Calinski-Harabasz Score:" % (n\_clusters, connectivity is not None,linkage),metrics.calinski\_harabaz\_score(X,model.labels\_))
24. plt.show()

7.完整代码：

[view plain](https://www.ipieuvre.com/e/408/311/9539) [copy](https://www.ipieuvre.com/e/408/311/9539)

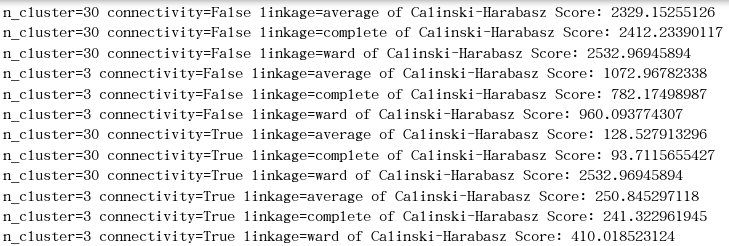
import time  
import matplotlib.pyplot as plt  
import numpy as np  
from sklearn import metrics  
from sklearn.cluster import AgglomerativeClustering  
from sklearn.neighbors import kneighbors\_graph  
n\_samples = 1500  
np.random.seed(0)  
t = 1.5 \* np.pi \* (1 + 3 \* np.random.rand(1, n\_samples))  
x = t \* np.cos(t)  
y = t \* np.sin(t)  
X = np.concatenate((x, y))  
X += .7 \* np.random.randn(2, n\_samples)  
X = X.T  
plt.scatter(X[:, 0], X[:, 1], marker='o')  
knn\_graph = kneighbors\_graph(X, 30, include\_self=False)  
for connectivity in (None, knn\_graph):  
 for n\_clusters in (30, 3):  
 plt.figure(figsize=(10, 4))  
 for index, linkage in enumerate(('average', 'complete', 'ward')):  
 plt.subplot(1, 3, index + 1)  
 model = AgglomerativeClustering(linkage=linkage,  
 connectivity=connectivity,  
 n\_clusters=n\_clusters)  
 t0 = time.time()  
 model.fit(X)  
 elapsed\_time = time.time() - t0  
 plt.scatter(X[:, 0], X[:, 1], c=model.labels\_,  
 cmap = plt.cm.jet  
)  
  
 plt.title('linkage=%s (time %.2fs)' % (linkage, elapsed\_time),  
 fontdict=dict(verticalalignment='top'))  
 plt.axis('equal')  
 plt.axis('off')  
 plt.subplots\_adjust(bottom=0, top=.89, wspace=0,  
 left=0, right=1)  
 plt.suptitle('n\_cluster=%i, connectivity=%r' %  
 (n\_clusters, connectivity is not None), size=17)  
 print("n\_cluster=%i connectivity=%r linkage=%s of Calinski-Harabasz Score:" % (  
 n\_clusters, connectivity is not None, linkage), metrics.calinski\_harabasz\_score(X, model.labels\_))  
  
plt.show()

8.代码编写完毕，在AgglomerativeClustering.py文件内，点击右键=》Run ‘AgglomerativeClustering’，执行AgglomerativeClustering.py文件。

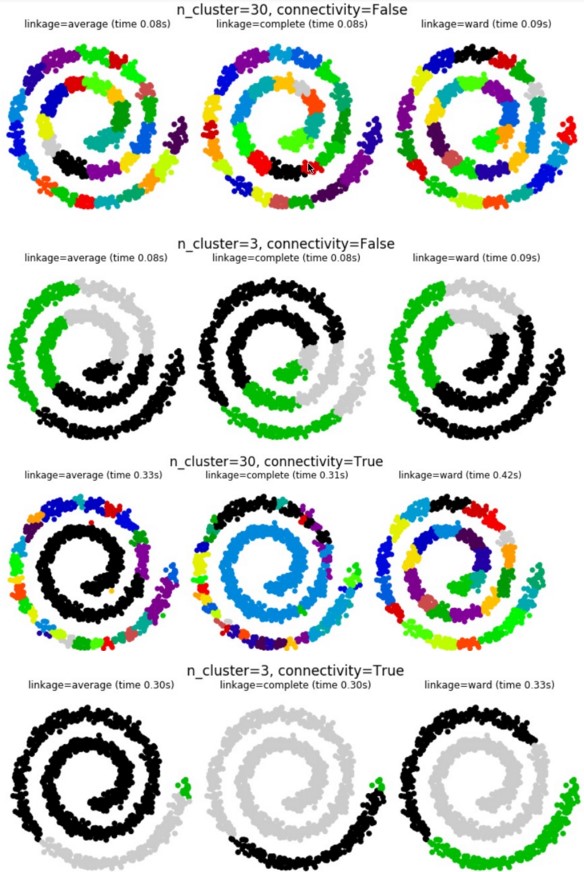
[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/8136e7be-be36-4adb-8b17-0c659295ee8f.png)

9.运行结果：

各个聚类模型对应的Calinski-Harabasz 分数输出为：

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/726c56c9-f3fd-4197-a4f0-ce5333c46488.png)

各个聚类模型的图像：

[](https://www.ipieuvre.com/doc/exper/367e266b-91ad-11e9-beeb-00215ec892f4/img/50fcd2ba-10ee-4519-bf4a-61abac723c7c.jpg)