Clustering 实验报告

一．项目描述

使用 sklearn 实现 K-Means, Affinity propagation, Mean-shift, Spectral clustering, Ward hierarchical clustering, Agglomerative clustering, DSCAN, Gaussian mixtures 聚类算法，使用 NMI 进行评估，并考虑运行时间。

二．过程实现

分为三部分实现：

Utils：对原始的 Tweets 数据进行处理，读取 text 和 cluster 并分开存储，根据 TFIDF 计算权重矩阵，将权重矩阵和标签都存为 pkl 文件。

Initial：第一次尝试，使用 sklearn 调用聚类函数，如果有参数 n_cluster，就使用 for 循环测试一些值，将 NMI 写入文件中。

Modify：对 Initial_Main.py 文件进行简化，去掉 for 循环，考虑 n_cluster 值相同的情况下进行比较。同样使用 sklearn 调用聚类函数，并考虑运行时间，将 NMI 和 run_time 写入文件中。

三．运行结果

通过分析 Modify_NMI_Result.txt 发现，Mean-shift 和 DBSCAN 的 NMI 值最小，Agglomerate clustering 的 NMI 值最大，Spectral clustering 运行最快，Ward hierarchical clustering、Agglomerate clustering、DBSCAN 和 Gaussian mixtures 相对来说

运行也比较快，但是 Mean-shift 最慢。所以总体来看，Agglomerate clustering 方法的结果相对最好。

四．问题分析及解决

1.权重矩阵存储问题。之前在每次调用聚类函数之前都要计算一次权重矩阵，后来使用 joblib.dump()将权重矩阵和标签存储为 pkl 文件，方便直接 joblib.load()使用

2.fit 与 fit_predict 的使用。调用聚类函数一开始使用 fit，然后使用 model.labels_得到其标签，虽然这样可以查看其聚类中心等属性，但是太麻烦。后来直接使用 fit_predict 得到标签。

3.列表和数组的相互转化问题。

一开始将实际标签存成列表，将预测标签逐个读入列表，然后计算 NMI；后来使用 tolist()将预测标签转化为列表；最后直接使用 np.array()将列表转化为数组，并存成 pkl 文件，得到预测标签后可以不做其他处理直接计算 NMI。

4.AP：preference 设置没效果，标签是 0-2472,相当于没有聚类，将该参数设置成默认之后，结果生成了 320 个类，准确率为 0.785614。

5.meanshift:一开始结果是 -0.726562 ，调参没有用还是一样的结果，后来再次运行结果变为-0.000002，比原来较好。

6.Ward hierarchical clustering 可以在 Agglomerate clustering 中通过调节 linkage 的值来实现；一开始 Agglomerate clustering 结果为 0.09，去掉 affinity="precomputed" ，结果就变为 0.887871。

8.DBSCAN:Too many open files: 'ProTweets'。在 LINUX 服务器上运行时出现此问题，是打开的文件或是 socket 没有正常关闭。解决：https://langyu.iteye.com/blog/763247。

9.GMM:无法导入 GMM。在从 0.18 开始的新版本中，GMM 已被弃用，GaussianMixture 用于代替它。同时我电脑上有一个旧版本的 scikit-learn，它还没有 GaussianMixture 类。所以升级 sklearn 出“错这是一个 distutils 安装的项目，因此我们无法准确确定哪些文件属于它，这将导致仅部分卸载”，所以忽略旧版本进行升级。