Data Mining 实验报告

Homework 3：clustering

学号：201844906　　姓名：宋春娇

一、 实验要求

1. 测试 sklearn 中聚类算法在数据集上的聚类效果；

2. 使用 NMI(Normalized Mutual Information)作为评价指标。

二、 数据集

1、 数据集 Tweets 以 Json 格式存储数据，格式如下：{"text": "centrepoint winter white gala london", "cluster": 65}。需要将其进行预处理，将文本和 label 存储在列表中，然后对文本进行分词处理，后续算法会用到 tf_idf 矩阵，可以用过 sklearn 中的 TfidfVectorizer 方法得到。

三、 实验过程

1. 安装 sklearn

   使用 pip install 命令进行安装，安装 sklearn 模块之前要确保已经安装了 numpy 和 scipy。

2. 对于实验要求指定的 8 种算法，通过 scikit-learn 官方文档进行学习，通过文档以及示例熟悉 8 种算法的使用方法。

   (1) K-Means 算法

   算法的目的是将 n 个向量分别归属到 K 个中心点里面去。算法首先会随机选择 K 个中心向量，然后通过迭代计算以及重新选择 K 个中心向量，使得 n 个向量各自被分配到距离最近的 K 中心

点，并且所有向量距离各自中心点的和最小。

值得注意的是：有两个地方是需要算法使用者去自己选择的：第一个就是 K 的值，简而言之就是数据集应该被分成多少个类，在 K-Means 算法里面是要求先给出 K 值的；第二个就是距离函数，即如何计算向量和向量或者向量和中心点之间的距离，最常见的自然是欧式距离，也可以用余弦相似度来作为距离函数。

(2) Affinity Propagation（AP 近邻传播聚类算法）

思想：根据 n 个点之间的相似度进行聚类（相似度可以用两个点之间的欧式距离来衡量）。两个点之间的相似度可以相同也可以不同，构成一个 n*n 的相似度矩阵。该算法不需要事先指定聚类数目，它将所有的数据点都作为潜在的聚类中心。

以 S 矩阵的对角线上的数值 s (k, k)作为 k 点能否成为聚类中心的评判标准,这意味着该值越大,这个点成为聚类中心的可能性也就越大,这个值又称作参考度 p (preference) 。聚类的数量受到参考度 p 的影响,如果认为每个数据点都有可能作为聚类中心,那么 p 就应取相同的值。如果取输入的相似度的均值作为 p 的值,得到聚类数量是中等的。如果取最小值,得到类数较少的聚类。

(3) MeanShift 算法

算法原理：在 n 个数据点中随机选择一个点作为圆心，找出距离圆心小于等于 bandwidth 的所有点（即以该点为圆心，以 bandwidth 为半径的园内所有点），这些点属于和圆心相同类别的概率要+1，然后计算出一个漂移向量，使圆沿着漂移向量移动，

移动后的圆重复概率+1 操作。直至所有的点均被标记类别。漂移向量的计算方法是将园内所有点到圆心的向量之和。这使得圆会向着点密集区域移动。也可以理解成该算法是沿着密度上升的方向寻找属于一个簇的点。

编程过程中将 tfidf_matrix 作为参数传递时会报如下错误：TypeError: A sparse matrix was passed, but dense data is required. Use X.toarray() to convert to a dense numpy array.于是将参数改为 tfidf_matrix.toarray()。

(4) spectral_clustering 算法

它是一种基于图论的聚类方法（这点上跟 AP 类似，而 K-Means 是基于点与点的距离计算），它能够识别任意形状的样本空间且收敛于全局最有解，其基本思想是利用样本数据的相似矩阵进行特征分解后得到的特征向量进行聚类。

(5) agglomerative_clustering 算法

凝聚是一开始将每个样本当做一个聚类，接着通过计算将距离最近的两个聚类合并，成为新聚类，每次合并聚类总数减少一个，不断循环合并操作，直到所有聚类合并成一个聚类或当聚类数量到达某预定值或当聚类直接距离达到某阀值后停止合并。而分裂则与凝聚相反，一开始将所有样本当做一个聚类，每次分裂一个聚类，直到满足某条件。

(6) GaussianMixture 算法

混合高斯模型是用高斯概率密度函数（正态分布曲线）精确

地量化事物，将一个事物分解为若干的基于高斯概率密度函数（正态分布曲线）形成的模型。

(7) DBSCAN 算法

基于密度的方法的特点是不依赖于距离，而是依赖于密度，从而克服基于距离的算法只能发现“球形”聚簇的缺点。

DBSCAN 的核心思想是从某个核心点出发，不断向密度可达的区域扩张，从而得到一个包含核心点和边界点的最大化区域，区域中任意两点密度相连。就是我们先找到一个核心对象，从它出发，确定若干个直接密度可达的对象，再从这若干个对象出发，寻找它们直接密度可达的点，直至最后没有可添加的对象了，那么一个簇的更新就完成了。我们也可以说，簇其实就是所有密度可达的点的集合。

(8) birch 算法

BIRCH 算法利用了一个树结构来帮助我们快速的聚类，这个数结构类似于平衡 B+树，一般将它称之为聚类特征树(Clustering Feature Tree，简称 CF Tree)。这颗树的每一个节点是由若干个聚类特征(Clustering Feature，简称 CF)组成。聚类特征树中，每个节点包括叶子节点都有若干个 CF，而内部节点的 CF 有指向孩子节点的指针，所有的叶子节点用一个双向链表链接起来。

四、 实验结果

```
使用kmeans算法进行聚类，准确率为：0.7765083908130822
使用AP近邻传播聚类算法进行聚类，准确率为：0.7624742003503602
使用MeanShift算法进行聚类，准确率为：0.721245221852964
使用spectral_clustering算法进行聚类，准确率为：0.6738977154419252
使用agglomerative_clustering算法进行聚类，准确率为：0.7868056884757556
使用GaussianMixture算法进行聚类，准确率为：0.7868056884757556
使用DBSCAN算法进行聚类，准确率为：0.669611144599745
使用birch算法进行聚类，准确率为：0.7685330121788387
```

图 1 各种算法聚类结果