作业三 实验报告

一、 实验目的

测试 sklearn 中以下聚类算法在 tweets 数据集上的聚类效果。

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

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers

二、 实验环境

Windows 10 + Python3.6.5

三、 实验步骤

首先使用 json 库将文本和标签分别提取出来并保存, 将文本向量化用于下一步的聚类。

- Kmeans

  k-means 算法将一组 N 样本 X 划分成 K 不相交的簇 C，每个都用该簇中的样本的均值 μ_j 描述。它们一般不是从 X 中挑选出的点，虽然它们是处在同一个空间。 K-means 算法旨在选择最小化簇内和的平方和的标准的聚类中心:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Kmeans 算法具有实现简单，收敛速度快的优点，但缺点是必须预先指定聚类的数目，在聚类数目未知时效果较差。

- Affinity Propagation

AP 聚类是通过在样本对之间发送消息直到收敛来创建聚类。然后使用少量示例样本作为聚类中心来描述数据集，聚类中心是数据集中最能代表一类数据的样本。在样本对之间发送的消息表示一个样本作为另一个样本的示例样本的 适合程度，适合程度值在根据通信的反馈不断更新。更新迭代直到收敛，完成聚类中心的选取，因此也给出了最终聚类。

样本之间传递的信息有两种。 第一种是 responsibility(吸引信息) $r(i,k)$，样本 k 适合作为样本 i 的聚类中心的程度。第二种是 availability(归属信息) $a(i,k)$ 样本 i 选择样本 k 作为聚类中心的适合程度,并且考虑其他所有样本选取 k做为聚类中心的合适程度。 通过这个方法，选取示例样本作为聚类中心如果 (1) 该样本与其许多样本相似，并且 (2) 被许多样本选取 为它们自己的示例样本。

样本 k 对样本 i 吸引度计算公式:

$$r(i,k) \leftarrow s(i,k) - max[a(i,k') + s(i,k') \forall k' \neq k]$$

其中 $s(i,k)$ 是样本 i 和样本 k 之间的相似度。 样本 k 作为样本 i 的示例样本的合适程度:

$$a(i,k) \leftarrow min[0, r(k,k) + \sum_{i' \ s.t. \ i' \notin \{i,k\}} r(i',k)]$$

算法开始时 r 和 a 都被置 0,然后开始迭代计算直到收敛。 为了防

止更新数据时出现数据振荡，在迭代过程中引入阻尼因子 λ:

$$r_{t+1}(i,k) = \lambda \cdot r_t(i,k) + (1-\lambda) \cdot r_{t+1}(i,k)$$
$$a_{t+1}(i,k) = \lambda \cdot a_t(i,k) + (1-\lambda) \cdot a_{t+1}(i,k)$$

其中 t 是迭代的次数。

- Mean Shift

MeanShift 算法旨在于发现一个样本密度平滑的 *blobs* 。 均值漂移算法是基于质心的算法，通过更新质心的候选位置为所选定区域的偏移均值。 然后，这些候选者在后处理阶段被过滤以消除近似重复，从而形成最终质心集合。

给定第 t 次迭代中的候选质心 x_i，候选质心的位置将被安装如下公式更新:

$$x_i^{t+1} = m(x_i^t)$$

其中 $N(x_i)$ 是围绕 x_i 周围一个给定距离范围内的样本空间 and m 是 *mean shift* vector（均值偏移向量） 是所有质心中指向 点密度增加最多的区域的偏移向量。使用以下等式计算，有效地将质心更新为其邻域内样本的平均值:

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$

- Spectral Clustering

  SpectralClustering 是在样本之间进行亲和力矩阵的低维度嵌入，其实是低维空间中的 KMeans。 如果亲和度矩阵稀疏，则这是非常有效的并且 SpectralClustering 需要指定聚类数。这个算法适用于聚类数少时，在聚类数多时不建议使用。

- Hierarchical Clustering

  Hierarchical clustering 是一个常用的聚类算法，它通过不断的合并或者分割来构建聚类。 聚类的层次被表示成树（或者 dendrogram（树形图））。树根是拥有所有样本的唯一聚类，叶子是仅有一个样本的聚类。

  The AgglomerativeClustering 使用自下而上的方法进行层次聚类:开始是每一个对象是一个聚类，并且聚类别相继合并在一起。linkage criteria 确定用于合并的策略的度量:

  Ward 最小化所有聚类内的平方差总和。这是一种 variance-minimizing（方差最小化）的优化方向， 这是与 k-means 的目标函数相似的优化方法，但是用 agglomerative hierarchical（聚类分层）的方法处理。

  Maximum 或 complete linkage 最小化聚类对两个样本之间的最大距离。

  Average linkage 最小化聚类两个聚类中样本距离的平均值。

- DBSCAN

  DBSCAN 算法将聚类视为被低密度区域分隔的高密度区域。由于这个相当普遍的观点， DBSCAN 发现的聚类可以是任何形状的，与假设聚类是凸区域的 K-means 相反。 DBSCAN 的核心概念是 *core samples*，是指位于高密度区域的样本。 因此一个聚类是一组核心样本，每个核心

样本彼此靠近（通过一定距离度量测量） 和一组接近核心样本的非核心样本 (但本身不是核心样本)。算法中的两个参数, min_samples 和 eps,正式的定义了我们所说的 dense（稠密)。较高的 min_samples 或者较低的 eps 表示形成聚类所需的较高密度。

聚类结果评估

考虑到 ground truth class assignments （标定过的真实数据类分配） labels_true 的知识和相同样本 labels_pred 的聚类算法分配， Mutual Information 是测量两者 agreement 分配的函数，忽略 permutations（排列)。

假设两个标签分配 (相同的 N 个对象)，U 和 V。 它们的 entropy （熵）是一个 partition set （分区集合）的不确定性量，定义如下:

$$H(U) = -\sum_{i=1}^{|U|} P(i)\log(P(i))$$

其中 $P(i) = |U_i|/N$ 是从 U 中随机选取的对象到类 U_i 的概率。同样对于 V:

$$H(V) = -\sum_{j=1}^{|V|} P'(j)\log(P'(j))$$

使用 $P'(j) = |V_j|/N$. U 和 V 之间的 mutual information (MI) 由下式计算:

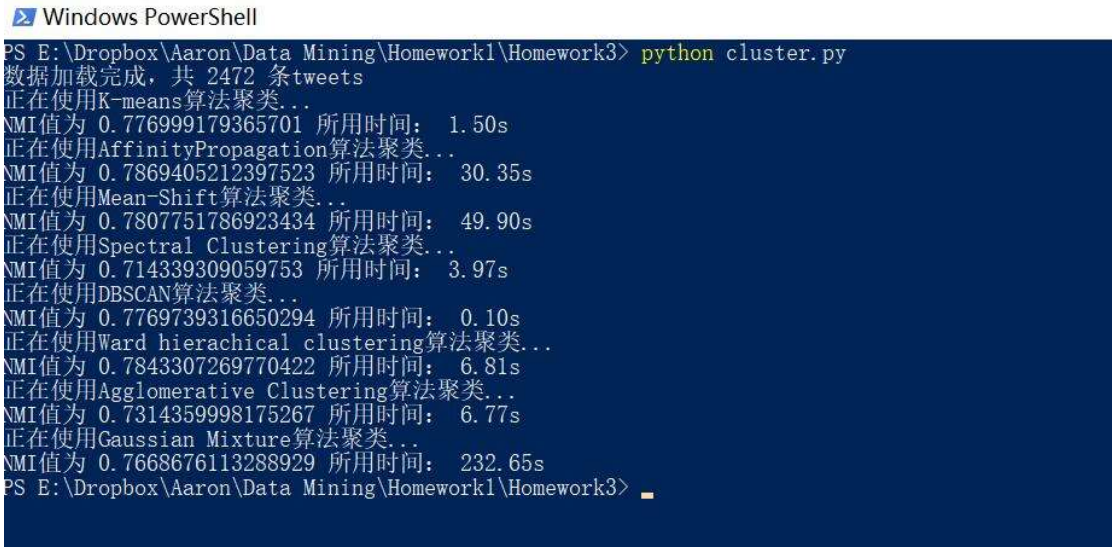
$$\mathrm{MI}(U,V)=\sum_{i=1}^{|U|}\sum_{j=1}^{|V|}P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$$

其中 $P(i,j)=|U_i\cap V_j|/N$ 是随机选择的对象落入两个类的概率 U_i 和 V_j 。

normalized (归一化) mutual information 被定义为

$$\mathrm{NMI}(U,V)=\frac{\mathrm{MI}(U,V)}{\mathrm{mean}(H(U),H(V))}$$

四、 实验结果



其中 K-means，Spectral Clustering 和 Aggolomerative Clustering 在默认参数下效果较差，给定类别个数后结果正常。Mean-Shift 算法使用默认的参数结果也不理想，通过实验，设定 bandwidth=0.9 可以达到最好的效果。由实验结果可以看出 DBSCAN 在聚类效果和效率的综合表现最好。