实验报告 3 Cluster

本次实验主要是运用 sklearn 中的方法，因此在具体的代码实现上就不再赘述，而是主要研究各方法的异同点及数据集上的表现。

一．评价标准

实验采用 NMI 评价聚类效果，即 Normalized Mutual information，该方法沿用了信息论中熵的思想，使用真实分布和求的分布的熵的和除以两者联合分布的熵。求得的结果属于[0,1]区间。

二．聚类算法

1.KMeans ：n_clusters 聚类个数 random_state 随机初始化

KMeans 算法在给定一个数 k 之后，通过最小化“最小二乘法”的函数控制变量，不断收敛，最终达到聚类的目的。总的目标函数如下

$$\text{argmin } l(\mu)=\Sigma\ \Sigma\ ||x-\mu i||^2$$

其中$\mu_i=\Sigma x/|C_i|\ (x\in Ci)$是簇 Ci 的均值向量，或者说是质心，$||x-\mu i||^2$代表每个样本点到均值点的距离。以上是对于数值属性来说的，对于一些离散属性也有相关的距离的定义。最后在现实中的数据如果要确定合适的距离计算式，可以通过“距离度量学习”来实现。也就是说上式的目的就是我们要找到 k 个簇，使得在每个簇内，所有的样本点都尽量接近。

Kmeans 方法对 k 的初始值较为敏感，不同的取值可能会获得不同的结果，在收敛速度和聚类表现上表现不错。

2.DBSCAN：eps 邻域的大小 min_samples 最小样本点

假设样本集是 D=(X1,X2,X3...Xm),则 DBSCAN 具体的描述定义如下：

1）ϵ-邻域：对于 xj∈D，其ϵ-邻域包含样本集 D 中与 xj 的距离不大于ϵ的子样本集，即 Nϵ(xj)={xi∈D|distance(xi,xj)≤ϵ}, 这个子样本集的个数记为|Nϵ(xj)|

2) 核心对象：对于任一样本 xj∈D，如果其ϵ-邻域对应的 Nϵ(xj)至少包含 MinPts 个样本，即如果|Nϵ(xj)|≥MinPts，则 xj 是核心对象。

3）密度直达：如果 xi 位于 xj 的ϵ-邻域中，且 xj 是核心对象，则称 xi 由 xj 密度直达。注意反之不一定成立，即此时不能说 xj 由 xi 密度直达, 除非且 xi 也是核心对象。

4)密度可达：对于 xi 和 xj,如果存在样本样本序列 p1,p2,...,pT,满足 p1=xi,pT=xj, 且 pt+1 由 pt 密度直达，则称 xj 由 xi 密度可达。也就是说，密度可达满足传递性。此时序列中的传递样 p1,p2,...,pT−1 均为核心对象，因为只有核心对象才能使其他样本密度直达。

5）密度相连：对于 xi 和 xj,如果存在核心对象样本 xk，使 xi 和 xj 均由 xk 密度可达，则称 xi 和 xj 密度相连。

和传统的 K-Means 算法相比，DBSCAN 最大的不同就是不需要输入类别数 k，当然它最大的优势是可以发现任意形状的聚类簇，而不是像 K-Means，一般仅仅使用于凸的样本集聚类。但同样，DBSCAN 方法也有自己的问题，即需要数据样本的分布较为稠密，否则可能会得出多余的类别。

3.谱聚类 n_clusters 聚类个数 random_state 随机初始化

它的主要思想是把所有的数据看做空间中的点，这些点之间可以用边连接起来。距离较远的两个点之间的边权重值较低，而距离较近的两个点之间的边权重值较高，通过对所有数据点组成的图进行切图，让切图后不同的子图间边权重和尽可能的低，而子图内的边权重和尽可能的高，从而达到聚类的目的。

为了避免最小切图导致的切图效果不佳，我们需要对每个子图的规模做出限定，一般来说，有两种切图方式，RatioCut 方法对每个切图，不光考虑最小化，它还同时考虑最大化每个子图点的个数，Ncut 切图和 RatioCut 切图很类似，将

子图中点的个数换为权重. 由于子图样本的个数多并不一定权重就大，我们切图时基于权重也更合我们的目标，因此一般来说 Ncut 切图优于 RatioCut 切图。

谱聚类只需要数据之间的相似度矩阵，因此对于处理稀疏数据的聚类很有效。这点传统聚类算法比如 K-Means 很难做到，由于使用了降维，因此在处理高维数据聚类时的复杂度比传统聚类算法好。但是如果最终聚类的维度非常高，则由于降维的幅度不够，谱聚类的运行速度和最后的聚类效果均不好。

4.层次聚类：n_clusters 聚类个数 linkage 确定计算方法(ward, average, complete)

层次聚类是通过从下往上不断合并簇，或者从上往下不断分离簇形成嵌套的簇。这种层次的类通过“树状图”来表示，最开始的时候将所有数据点本身作为簇，然后找出距离最近的两个簇将它们合为一个，不断重复以上步骤直到达到预设的簇的个数。

可以看到，一个很关键的地方就是判断簇之间的距离。判断的准则叫做链接准则。对于此类算法，scikit-learn 有三种准则

Ward:将所有集群内的差异平方和最小化。这是一种方差最小化的方法这种感觉类似于 k-means 目标函数，但采用了聚集层次方法。

Maximum or complete linkage:最小化对集群的观察之间的最大距离。

Average linkage:最小化了对集群的所有观测之间的距离的平均值。

时间复杂度比较高，运行速度较慢，并且可能得到的类大小不均匀，但是对分布怪异的样本有不错的聚类效果。

4. MeanShift：bandwidth 高斯核的参数 bin_seeding 优化加速

Mean Shift 算法是指一个迭代的步骤,即先算出当前点的偏移均值,移动该点到其偏移均值,然后以此为新的起始点,继续移动,直到满足一定的条件结束。

在数据集中选定一个点，然后以这个点为圆心，r 为半径，画一个圆，求出这个点到所有点的向量的平均值，而圆心与向量均值的和为新的圆心，然后迭代此过程，直到满足一点的条件结束。后来加了权重系数和核函数，目前在聚类，图像平滑，分割，跟踪等方面有着广泛的应用。

5. Affinity propagation Clustering Algorithm：damping 阻尼系数

AP 算法是根据数据点之间的相似度来进行聚类，可以是对称的，也可以是不对称的。 该算法不需要先确定聚类的数目，而是把所有的数据点都看成潜在意义上的聚类中心，这有别于 K-means 等聚类。

AP 算法进行交替两个消息传递的步骤，以更新两个矩阵：

吸引信息（responsibility）矩阵 R：r(i,k)

描述了数据对象 k 适合作为数据对象 i 的聚类中心的程度，表示的是从 i 到 k 的消息；

归属信息（availability）矩阵 A：a(i,k)

描述了数据对象 i 选择数据对象 k 作为其据聚类中心的适合程度，表示从 k 到 i 的消息。

通过对这两个信息进行迭代以获得聚类结果,同时为了避免振荡，AP 算法更新信息时引入了衰减系数 λ 。每条信息被设置为它前次迭代更新值的 λ 倍加上本次信息更新值的 1-λ 倍。

AP 聚类算法样本中的所有数据点都可能成为 AP 算法中的质心，叫做 Examplar，而不是由多个数据点求平均而得到的聚类中心（如 K-Means）。同时对初始值不敏感。多次执行 AP 聚类算法，得到的结果是完全一样的，即不需要进行随机选取初值步骤。

但是由于 AP 算法每次迭代都需要更新每个数据点的吸引度值和归属度值，算法复杂度较高，为 O(N*N*logN)，而 K-Means 只是 O(N*K)的复杂度。因此当 N 比较大时，AP 聚类算法往往需要算很久，在大数据量下运行时间较长。