Data Mining 实验报告

张浩 计算机技术 专硕班 201834886

实验名称

实验三 基于 sklearn 的聚类实验

实验要求

使用 sklearn python 工具包在给定数据集上进行聚类

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

实验环境

硬件环境: Intel(R) Core(TM) i5-6260U CPU @1.8GHz 4G RAM
软件环境: windows 10 1803
编程语言: python 3.7

实验数据

Tweets 数据集
2742 个 text
89 个类别

实验过程

规则化数据集文本

建立词典

计算 TF-IDF

使用各种聚类方法在数据集上进行实验

Kmeans Flow

Step1：对于给定的群集分配 C，在群集中查找最小化与该群集中其他点的总距离的观察：

$$i_k^* = \underset{\{i:C(i)=k\}}{\arg\min} \sum_{C(j)=k} d(x_i, x_j).$$

Step2：
$$m_k = x_{i_k^*},\ k = 1,2,\ldots,K$$

Step3：给定一组集群中心{m1，…，mK}，通过将每个观测值分配给当前集群中心来最小化总误差：

$$C(i) = \underset{1\le k\le K}{\arg\min}\, d(x_i, m_k),\ i = 1,\ldots,N$$

重复步骤 1-3

参考:

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006910.html

Mean-shift

Step1：在未被标记的数据点中随机选择一个点作为中心 center；

Step2：找出离 center 距离在 bandwidth 之内的所有点，记做集合 M，认为这些点属于簇 c。同时，把这些求内点属于这个类的概率加 1，这个参数将用于最后步骤的分类

Step3：以 center 为中心点，计算从 center 开始到集合 M 中每个元素的向量，将这些向量相加，得到向量 shift。

Step4：center = center+shift。即 center 沿着 shift 的方向移动，移动距离是||shift||。

Step5：重复步骤 2、3、4，直到 shift 的大小很小（就是迭代到收敛），记住此时的 center。注意，这个迭代过程中遇到的点都应该归类到簇 c。

Step6：如果收敛时当前簇 c 的 center 与其它已经存在的簇 c2 中心的距离小于阈值，那么把 c2 和 c 合并。否则，把 c 作为新的聚类，增加 1 类。

重复 1、2、3、4、5 直到所有的点都被标记访问。

Step7：分类：根据每个类，对每个点的访问频率，取访问频率最大的那个类，作为当前点集的所属类。

参考:

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html

https://blog.csdn.net/hjimce/article/details/45718593

Spectral clustering

谱聚类

是一种基于图论的聚类方法——将带权无向图划分为两个或两个以上的最优子图，使子图内部尽量相似，而子图间距离尽量距离较远，以达到常见的聚类的目的。其中的最优是指最优目标函数不同，可以是割边最小分割也可以是分割规模差不多且割边最小的分割。

参考:

https://www.cnblogs.com/sparkwen/p/3155850.html

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

Ward hierarchical clustering

分层聚类

自下而上的方法：每个观察都在它自己的集群中开始，并且当一个集群向上移动时，它们将被合并。

自上而下的方法：所有观察都在一个集群中开始，并且当一个集体向下移动时，递归地执行分割。

参考:

https://en.wikipedia.org/wiki/Hierarchical_clustering

http://scikit-learn.org/0.15/modules/generated/sklearn.cluster.Ward.html

DBSCAN Density-Based Spatial Clustering of Applications with Noise

(1) 首先将数据集 D 中的所有对象标记为未处理状态

(2) for（数据集 D 中每个对象 p） do

(3) if （p 已经归入某个簇或标记为噪声） then

(4) continue;

(5) else

(6) 检查对象 p 的 Eps 邻域 NEps(p) ；

(7) if (NEps(p)包含的对象数小于 MinPts) then

(8) 标记对象 p 为边界点或噪声点；

(9) else

(10) 标记对象 p 为核心点，并建立新簇 C, 并将 p 邻域内所有点加入 C

(11) for (NEps(p)中所有尚未被处理的对象 q) do

(12) 检查其 Eps 邻域 NEps(q)，若 NEps(q)包含至少 MinPts 个对象，则将 NEps(q)中未归入任何一个簇的对象加入 C；

(13) end for

(14) end if

(15) end if

(16) end for

参考:

https://blog.csdn.net/zhouxianen1987/article/details/68945844

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

实验结果

本次实验选用 NMI 作为评价指标，Normalized Mutual Information (NMI) 归一化互信息（NMI）是互信息（MI）分数的归一化，以将结果在 0（无互信息）和 1（完全相关）之间进行缩放。

	Kmeans	nityPropaga	Mean-shift	Spectral clustering	Ward hierarchical clustering	DBSCAN
NMI	0.6669257	0.6938525	0.2063121	0.1159622	0.8046852	0.6468122
Time	54.7192000	37.0428000	873.8014200	743.6899200	21.3175900	9.1729600

从实验结构上看效果最好的是 hierarchical clustering 方法，时间最长的是 Mean-shift 方法。如果样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差，这时用 DBSCAN 聚类一般不适合。经过查资料得知 Spectral clustering 计算复杂度比 K-means 要小但是我在实际计算时候发现用时也很长，而且还会报错，按照报错信息查找，得知这类方法时候聚类类别为 2-7 如果类别大于 7 通常情况下会报错。

总结

1. 之前写 python 程序很少调用现成的工具包，使用 sklearn 之后发现其功能非常强大，如果直接用 sklearn 处理之前实验的话，会提高编码效率。Python 相比与 c++或者 c 而言优点在于其有强大的库支持，所以以后在写代码时，先找找有没有相关的的库支持。这样可以减少好多重复的编码量。
2. 从算法上讲 k-means，实现起来非常简单，K 均值收敛，但它找到了成本函数的局部最小值。仅适用于数值观测，K 是用户输入这样其实对于好多场景并不适用; 扩展调查 BIC（贝叶斯信息准则）或 MDL（最小描述长度）可用于估计 K。