**《机器学习》课程实验报告**

|  |  |
| --- | --- |
| **实验名称:** | 体验K-means对文档聚类 |
| **姓 名:** | 李若昊 |
| **学 号：** | 5120180269 |
| **教 师:** | 赵旭剑 |
| **完成时间:** | 2019/5/30 |

|  |
| --- |
| **一、实验目的和内容**  **用sklearn.cluster实现K-means聚类，也就是选择出质心，使得各个聚类内部的inertia值最小化。** |
| **二、实验过程**  **查阅了函数原型**[**https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html**](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)  classsklearn.cluster.**KMeans**(n\_clusters=8, \*, init='k-means++', n\_init=10, max\_iter=300, tol=0.0001, precompute\_distances='deprecated', verbose=0, random\_state=None, copy\_x=True, n\_jobs='deprecated', algorithm='auto')  **大概理解了一下参数用法：**  **n\_clusters：类数**  **init：初始化方法，提供了‘kmeans++’选项可以使初始化的质心相互之间距离很远，这比随机初始质心能够取得更好的效果。考虑到聚类的结果高度依赖质心的初始化，所以就采用默认的这个值。**  **n\_init：这个参数的值代表使用不同的质心运行算法的次数。计算过程中会进行多次的聚类，最终选择一个较好的聚类结果。每次都采用init中选择的方法初始化不同的质心。最终输出结果为几次聚类中效果最好的，以组内距离衡量。**  **max\_iter：一次聚类算法所执行的最大迭代次数**  **tol：当两次迭代的组内距离之差小于该值时，停止迭代**  **precompute\_distances：是否进行距离的预先计算，默认是若样本总数乘以聚类数 n\_samples \* n\_clusters 大于 1200 万，就不进行预计算**  **random\_state：随机初始化聚类中心的随机规则。取值为整数时，随机规则固定，默认使用 numpy 随机数生成器。**  **n\_jobs：设置并行计算的处理器个数。取值为“-1”时，使用所有的处理器，这样可以通过设置它来采用并行计算的方式。**  **algorithm：选择聚类算法，auto表示稀疏数据选择经典的K-Means算法，否则选择elkan-K-means算法；full是经典的K-Means 算法。**  **然后利用上一次实验做出来的VSM进行聚类：** |
| **三、实验结果**  **20个类簇中心点和408个簇，对应408个景点，每个文档对应聚在相应的类0~19**  **可以看到14个质心们**  **143个日期后面对应的内容各属于14个类中的哪一个** |
| **四、实验总结**  **聚类结果不是很好，因为误差平方和还比较大。第一次做语言处理，通过初步体验了一下按文档内容自动分类，之后要在各种考试后自己尝试写一下K-means方法并且解决分类不够好的问题。** |