系统工程第 6 次作业

张博睿　自 75　2017011537

程序详见

$$./code/..$$

1.Kmeans 算法分析

（1）Kmeans 算法流程

在本次作业中，KMeans 算法的主要实现在"./code/mykmeans.py"文件中，具体算法如下

输入：原始数据$data$，聚类数量num
（1）初始化类中心和标签（默认采用随机初始化）； （2）下面开始迭代过程： （2.1）计算输入数据$data$与类中心之间的两两距离矩阵$distance\ matrix$。 （2.2）根据距离矩阵$matrix$利用样本最近类中心信息更新标签$label$。 （2.3）根据新的标签，更新类中心 $$centroids = \frac{1}{n_k}\sum_{i \in C_k} x_k$$ （2.4）判断类中心改变的幅度是否满足终止条件。
输出：聚类标签$label$。

（2）收敛性分析

已知 Kmeans 算法的目标是最小化

$$\min_{\Omega} L$$

$$s.t.\ L = \sum_{i=1}^{k}\sum_{t \in \omega_i}(x_t - c_i)^T(x_t - c_i)$$

对于一个更新的分类，需要满足

$$\widehat{\Omega} = \{\widehat{\omega}_i, 1 \le i \le k\}$$

其中

$$\widehat{\omega}_i = \left\{t \in J(N)\middle|(x_t - c_i)^T(x_t - c_i) \le \left(x_t - c_j\right)^T\left(x_t - c_j\right), \forall j\right\}$$

因此

$$\sum_{i=1}^{k}\sum_{t \in \widehat{\omega}_i}(x_t - c_i)^T(x_t - c_i) \le \sum_{i=1}^{k}\sum_{t \in \omega_i}(x_t - c_i)^T(x_t - c_i)$$

即$\hat{L} \le L$，说明L单调递减。

又因为$L \ge 0$，根据实数基本定理**单调有界必收敛**，k-means 算法必然会收敛。

2.代码函数功能

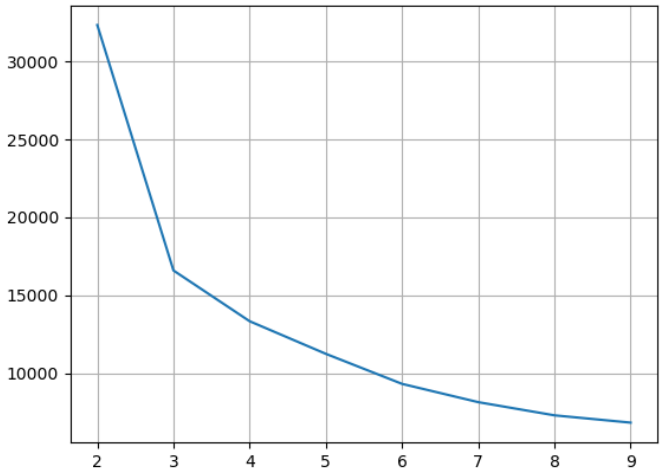
函数	功能
$MyKmeans.clustering(data, num)$	$KMeans$算法实现
$MyKmeans.visualize()$	结果可视化
$MyKmeans.sse()$	返回SSE值

3.聚类数目探究

这一步部分选择了聚类数目从 2 到 9 进行测试的结果，用SSE来代表聚类的效果

$$SSE = \sum_{i=1}^{k} \sum_{t \in \omega_i} (x_t - c_i)^T (x_t - c_i)$$

由于显然随着聚类数目的上升，SSE显然是呈现下降趋势的，因此采用肘部法则来判断最合适的聚类数目，结果如下：



结论：可以看到，随着聚类数目的增加，SSE逐渐下降，并且可以看到，当聚类数目增长到 3 时，继续增加聚类数目不会导致明显的SSE降低，可以认为这时的聚类数目比较合理（到达肘部）。

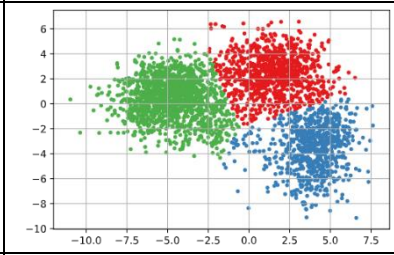
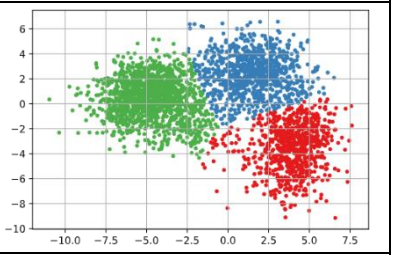
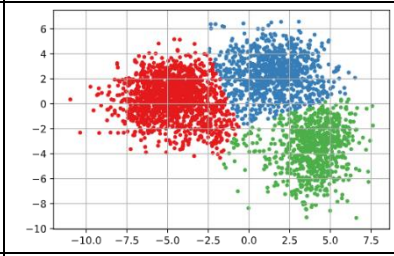
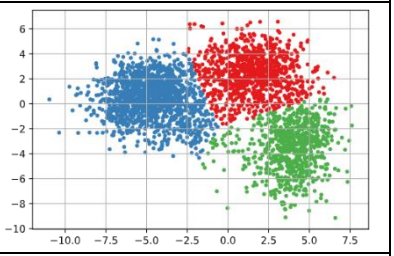
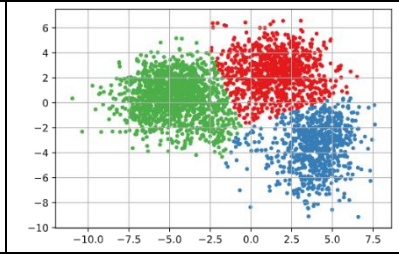
4.初始点选择

代码中主要实现了两种初始化方法

（1）随机初始化类中心；

（2）选择样本中前k个样本作为初始类中心。

聚类可视化结果如下：

实验组	时间	实验组	时间
$top\ k$		$random$	
$random$		$random$	
$random$			

结论：可以看到，采用不同的初始化方式，最终得到的划分结果比较稳定。这可能是因为本次实验的数据本身在分布上具有一定的聚团特点（否则，对于均匀随机分布的数据，不可能保证 KMeans 能够稳定收敛）。

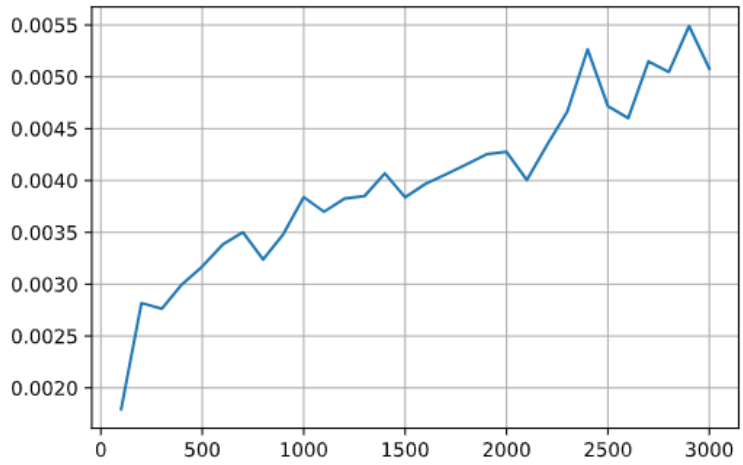
运行时间如下：

实验组	时间
$top\ k$	$0.017951s$
$random$	$0.022942s$
$random$	$0.005985s$
$random$	$0.009974s$
$random$	$0.009972s$

结论：可以看到，不同实验组的运行时间存在一定的波动。说明不同的初始化方式会影响算法的收敛速度。这一点可以用举例的方式说明：例如，如果有一种初始化的方法为将类中心初始设置为集中在某一点的分布，而另一种初始化方法为将类中心初始设置为空间的均匀分布，那么可想而知，前一种方法由于初始化中心选择和最终结果差别太大，一般会花费更多的计算时间。

5.耗时与数据规模

这一部分通过随机选择100、200、…、3000个数据进行$Kmeans$算法测试。由于为了保证运行时间的稳定性，每一组实验都运行了 100 次，取运行时间的平均值。最终结果如下：



可以看到，上述曲线基本呈现出正比的关系。下面可以从理论的层面进行分析。

分析：

由于本次作业$KMeans$的算法实现是类似于 Lloyd's algorithm的实现方式，因此复杂度为

$$O(nkdi)$$

其中，n是 d 维向量的个数；k是聚类的个数；i是迭代的次数。

对于大规模的数据，可以简单认为d, k, i是差别不大的，则算法复杂度近似为$O(n)$。这里的分析和实验结果是基本一致的。

参考资料

[1] https://en.wikipedia.org/wiki/K-means_clustering