《系统工程导论》

第七章作业

聚类分析 CA

姓名：卢志

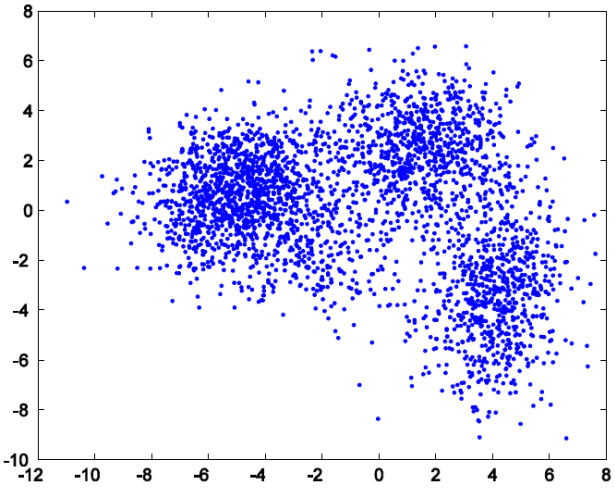
班级：自 43 班

学号：2014011497

2016 年 4 月 25 日

1. 题目要求

附件 data.mat 中包含 3000 个二维平面上的点，请根据课堂所学知识，编写 k-means 聚类方法对这些点进行聚类。



(1) Kmeans 聚类一定会收敛吗？为什么？

(2) 完成函数 function label = kmeans_clustering(data, num)，其中输入变量 data 为 N 行 m 列，每一行为一个数据点，num 表示聚类数目；输出变量 label 为 N 行 1 列，表示对应的数据点属于哪一类（比如属于第一类的点 label 就为 1）

(3) 聚类数目从 2 类开始逐渐增加，分别进行计算并分析聚类效果，决定最合适的聚类数目并说明理由。

(4) 选择不同的初始点多次实验，观察初始点的选择对最终结果的影响

(5) 选择不同的数据规模进行实验，计算你的程序耗时，观察耗时与数据规模之间的关系，从中你能得到什么结论？（此题选作） 提示：MATLAB 中可以使用 tic 和 toc 语句组合来计算某一段代码的耗时，具体可以查看帮助。

2. 作业内容

2.1 思考部分

Kmeans 聚类一定会收敛吗？为什么？

答：Kmeans 聚类一定会收敛，但不一定能保证收敛到全局最优解，只能收敛到局部最优解。所以说 Kmeans 方法的聚类结果与初始值选取有关。下面进行证明：

假设有一组数据，变量为 n 维，在 t=1, 2, …, N 时刻，记为：

$$x(t)=\begin{bmatrix}x_1(t) & x_2(t) & \dots & x_n(t)\end{bmatrix}^T, 1\le t\le N$$

而聚类的目标是：

$$\bigcup_{1\le i\le k}\varpi_i = J(M), \varpi_i\cap\varpi_j=\phi, \forall i\ne j$$

$$\Omega=\{\varpi_i\subseteq J(M), 1\le i\le k\}$$

并且使得下述目标函数最小：

$$\sum_{i=1}^{k}\rho(\varpi_i)$$

$$\rho(\varpi)=\sum_{t\in\varpi}(x(t)-e_\varpi(x))^T(x(t)-e_\varpi(x))$$

$$e_\varpi(x)=\frac{1}{|\varpi|}\sum_{t\in\varpi}x(t)$$

而 Kmeans 方法在一开始就固定了分类的个数 k，假设已经有了一个分类：

$$\Omega=\{\varpi_i\subseteq J(M), 1\le i\le k\}$$

此时 k 个分类中心点的值为：

$$c_i=e_{\varpi_i}(x)\in R^n, 1\le i\le k$$

再根据新的中心点可以更新一个新的分类：

$$\hat{\Omega}=\{\hat{\varpi}_i \subseteq J(M),1\le i\le k\}$$

而这一分类的标准为遍历每一个点，根据各个中心的距离，选择该点距离最近的中心所在的类。

$$\hat{\varpi}_i=\{t\in J(M)\,|\,(x(t)-c_i)^T(x(t)-c_i)\le(x(t)-c_j)^T(x(t)-c_j),\forall j\}$$

进一步可以得到：

$$\sum_{i=1}^{k}\sum_{t\in\hat{\varpi}_i}(x(t)-c_i)^T(x(t)-c_i)\le\sum_{i=1}^{k}\sum_{t\in\varpi_i}(x(t)-c_i)^T(x(t)-c_i)$$

$$\sum_{i=1}^{k}\sum_{t\in\hat{\varpi}_i}(x(t)-c_i)^T(x(t)-c_i)\le\sum_{i=1}^{k}\rho(\varpi_i)$$

可见：**每一步迭代之后，目标函数值都是在非严格下降的，最终一定会收敛到局部最优解。**

从另一角度分析，相当于求优化问题：

$$\min_{\alpha\in R^n}\sum_{t\in\hat{\varpi}}(x(t)-\alpha)^T(x(t)-\alpha)$$

求导得：

$$\frac{\partial(\sum_{t\in\hat{\varpi}_i}(x(t)-\alpha)^T(x(t)-\alpha))}{\partial\alpha}=-2\sum_{t\in\hat{\varpi}_i}(x(t)-\alpha)$$

令倒数为 0，可得最优解为：

$$\alpha=\frac{1}{|\hat{\varpi}_i|}\sum_{t\in\hat{\varpi}_i}x(t)=e_{\hat{\varpi}_i}(x)$$

$$\sum_{i=1}^{k}\rho(\hat{\varpi}_i)=\sum_{i=1}^{k}\sum_{t\in\hat{\varpi}_i}(x(t)-e_{\hat{\varpi}_i}(x))^T(x(t)-e_{\hat{\varpi}_i}(x))\le\sum_{i=1}^{k}\sum_{t\in\hat{\varpi}_i}(x(t)-c_i)^T(x(t)-c_i)=\sum_{i=1}^{k}\rho(\varpi_i)$$

而如果存在其中一个$c_i\neq e_{\hat{\varpi}_i}(x)$，必然有：

$$\sum_{i=1}^{k}\rho(\hat{\varpi}_i)<\sum_{i=1}^{k}\rho(\varpi_i)$$

所以说当每次迭代时的聚类中心发生变化时，优化函数是能够保证下降的；如果聚类中心不再发生变化时，则说明已经到达收敛的局部最优解。

2.2 完成函数 kmeans_clustering()

其中输入变量 data 为 N 行 m 列，每一行为一个数据点，num 表示聚类数目；输出变量 label 为 N 行 1 列，表示对应的数据点属于哪一类（比如属于第一类的点 label 就为 1）

函数思路如下：

① 选取初始中心点，根据分类数目 k，挑选 k 个初始中心点，这里采取随机数生成的方式。

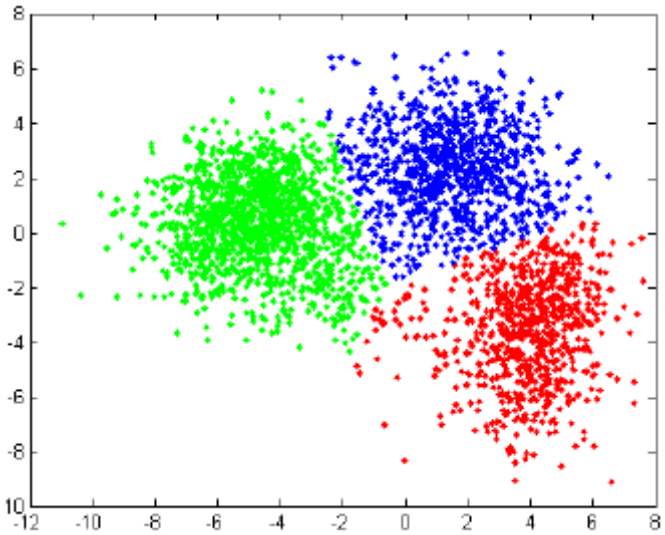
② 逐个利用每个样本修改中心点，对所有样本进行下述计算

a. 将其归入与其最近的中心点所在的类

b. 重新计算该类的中心点，并用新的中心点替换原先中心点。

③ 停止准则：如果上述步骤开始时和结束后的中心点差别很小，停止分类。这里可以利用距离之和来体现差别。

假设 num=3 时，可以利用算法，得到分类如下：



具体代码见源文件。

2.3 增加聚类的类数

聚类数目从 2 类开始逐渐增加，分别进行计算并分析聚类效果，决定最合适的聚类数目并说明理由。

答：在此问题中，由于需要分析最合适的聚类数目，所以需引入轮廓系数作为判断条件，其作为画聚类之后的内聚度和分离度程度，评价聚类效果。用 S(i) 表示：

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

其中：

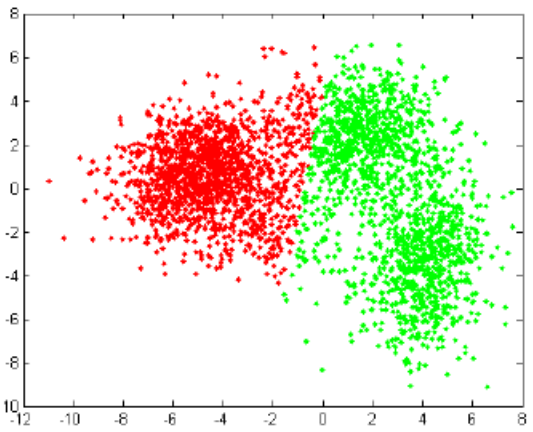
$a(i)$：average（向量 i 到其他所有它属于的类别中的点的距离）

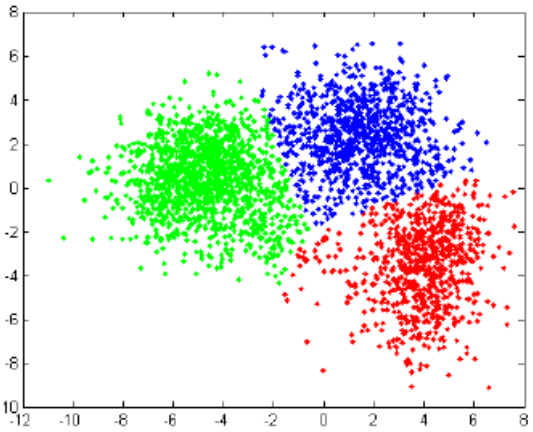
$b(i)$：min（向量 i 到其他每一类别的点的平均距离）

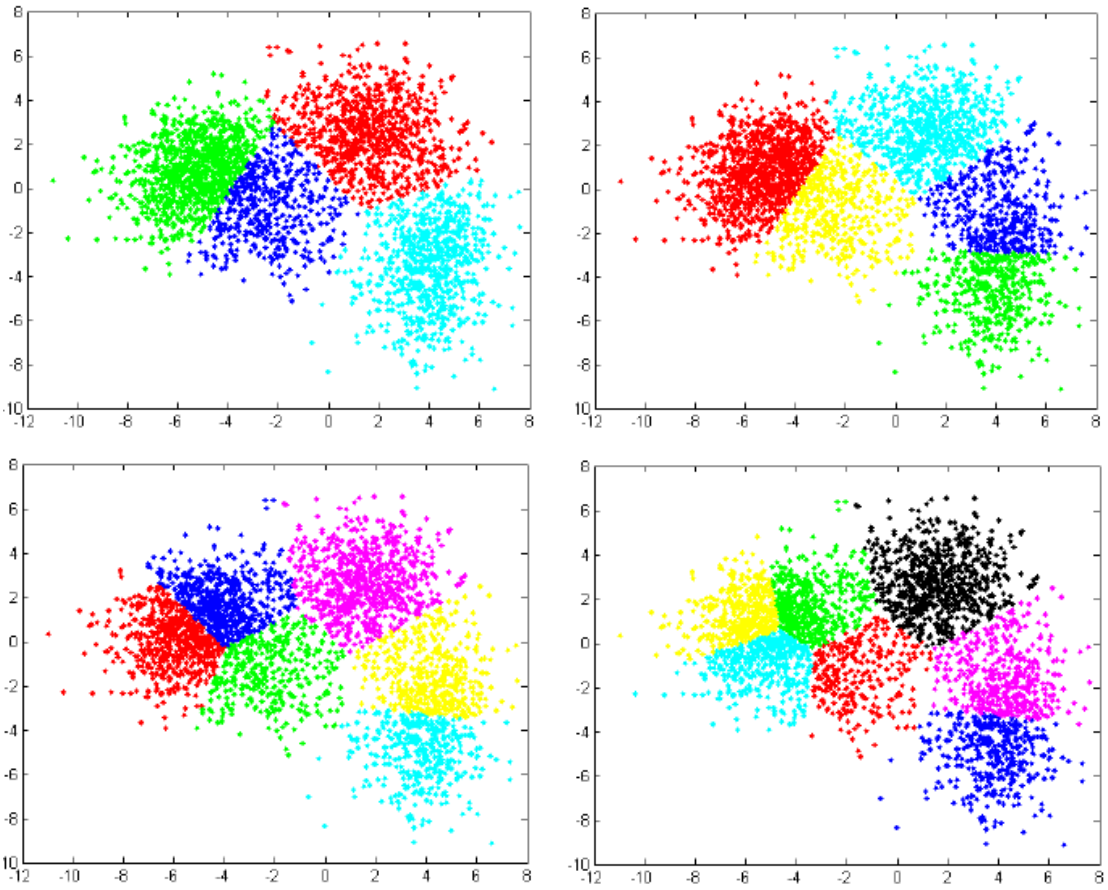
而具体计算时，若有 num 个分类，计算向量 i 到其他(num-1)个分类的点的距离，可以得到(num-1)个平均距离，选出其中最小的即可。

所以轮廓系数的计算方式为，先求出 N 个向量的 S(i)，再取平均，得到总的轮廓系数。而根据定义，可以发现轮廓系数的取值范围是[-1, 1]，当 $b(i)$ 远大于 $a(i)$ ，即类间距离远大于类内距离时，分类效果越好，此时总的轮廓系数接近 1。

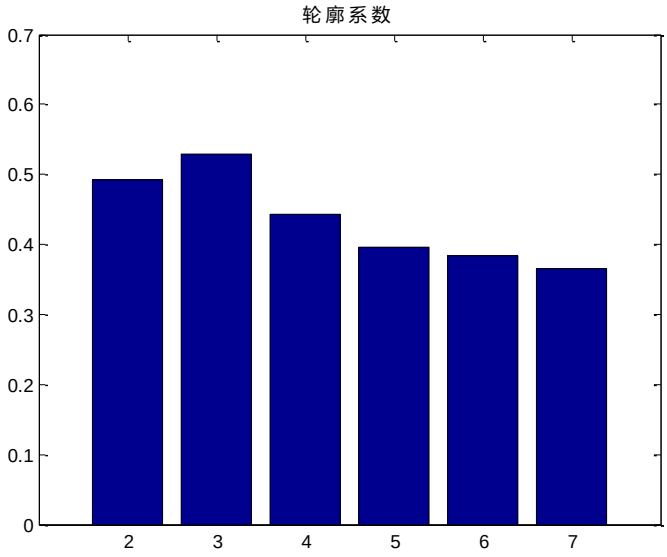
所以这里分别设聚类数 num 取值范围设定为{2, 3, 4, 5, 6, 7}，完成聚类并各自计算轮廓系数。聚类的结果分别如下：







根据前述分析，计算轮廓系数，并以柱状图的形式呈现。



分析：由上述柱状图**可以看出当 num=3 时，轮廓系数最大，聚类效果最好。**更进一步分析，当 num 小时，各类内部的距离较大，聚类不紧密；当 num 大时，各类之间的距离较小，区分不明显。

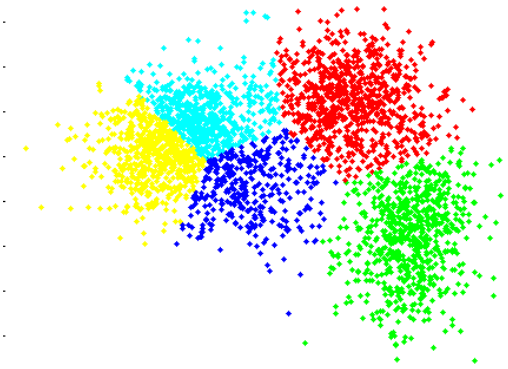
2.4 选择不同的初始点

选择不同的初始点多次实验，观察初始点的选择对最终结果的影响。

答：这里以 num=5 为例，并修改函数，使得其输出初始的中心点。

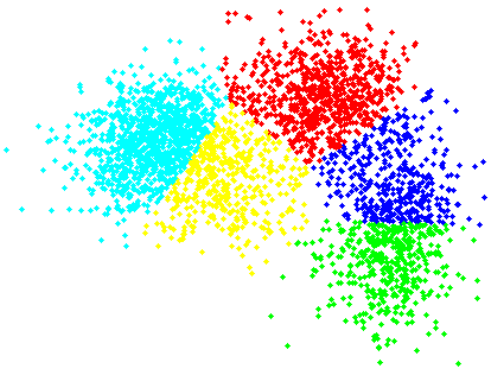
① 第一次测试（左为中心点，右为聚类效果）

	1	2
1	3.6430	4.7309
2	4.0421	-6.1424
3	-3.2087	-1.0324
4	-5.2851	0.1516
5	-5.7875	1.3970



② 第二次测试

	1	2
1	0.8559	5.9969
2	2.4264	0.1707
3	2.0406	2.8111
4	-3.7048	0.7218
5	-6.1730	-1.1020



分析：由两次实验效果不同可知，初始点的选择对最后的聚类效果影响是十分显著的。由于 K-means 方法只能保证收敛到局部最优解，不能保证收敛到全局最优解，而且目标函数无法确定是否是全局上的凸函数，所以可能不同的初始点，会导致函数收敛到不同的局部最优解最终结果有可能有所不同。

2.5 观察耗时

选择不同的数据规模进行实验，计算你的程序耗时，观察耗时与数据规模之间的关系，从中你能得到什么结论？（此题选作）

答：分别选择数据规模 500, 1000, 1500, 2000, 2500, 3000 做多次实验，结果如下：

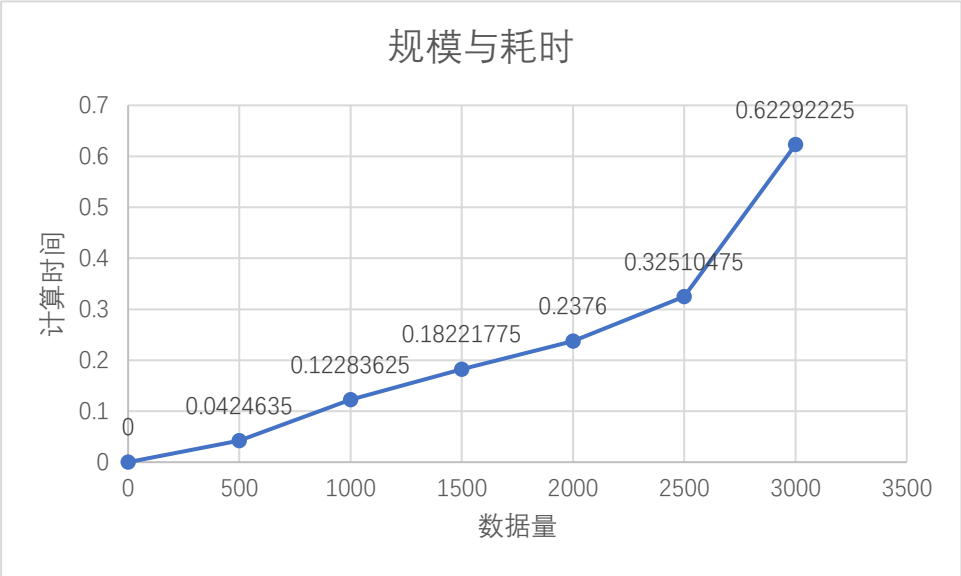
```
>> HW7
数据量为500的耗时为0.030053
数据量为1000的耗时为0.072177
数据量为1500的耗时为0.174008
数据量为2000的耗时为0.202233
数据量为2500的耗时为0.340695
数据量为3000的耗时为0.391439
```

```
>> HW7
数据量为500的耗时为0.061138
数据量为1000的耗时为0.191715
数据量为1500的耗时为0.231102
数据量为2000的耗时为0.454946
数据量为2500的耗时为0.460590
数据量为3000的耗时为0.579117
```

```
>> HW7
数据量为500的耗时为0.032389
数据量为1000的耗时为0.123306
数据量为1500的耗时为0.132859
数据量为2000的耗时为0.249761
数据量为2500的耗时为0.272439
数据量为3000的耗时为0.496086
```

```
>> HW7
数据量为500的耗时为0.047145
数据量为1000的耗时为0.085413
数据量为1500的耗时为0.222179
数据量为2000的耗时为0.353060
数据量为2500的耗时为0.237565
数据量为3000的耗时为1.325940
```

并对 4 次结果取平均，得到曲线为：



结论：由曲线看到，在规模较小时，计算耗时的增长近似为线性，但规模增大以后有耗时上涨速率超过线性速率的趋势，成为指数增长。

具体分析从初始中心点开始迭代寻找下一组中心点，每迭代一次复杂度为 O(N)，总迭代次数又与 N 成线性正相关，因此复杂度应为 O(N^2)。即与数据规模呈二次关系，从绘制的波形图来看，与实际结果较为相符。

2.6 原始代码

见工程文件，主函数为HW6.m，点击运行即可。其余函数功能可见代码注释。

3. 作业小结

本次作业编写了聚类分析 CA 的实现函数，而且对于聚类程度进行了刻画，使得我对聚类分析有了更深的理解。另外，也对 matlab 作图有了更多的了解，收获很大。