K-means 算法简介

《用 Python 玩转数据》　　by 大壮@NJU

K-means 算法是典型的基于距离的聚类算法，采用距离作为相似性的评价指标，两个对象的距离越近，其相似度就越大。而簇是由距离靠近的对象组成的，因此算法目的是得到紧凑并且独立的簇。

假设要将对象分成 k 个簇，算法过程如下：

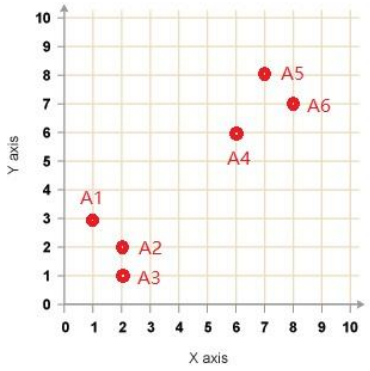
(1) 随机选取任意 k 个对象作为初始聚类的中心（质心，Centroid），初始代表每一个簇；

(2) 对数据集中剩余的每个对象根据它们与各个簇中心的距离将每个对象重新赋给最近的簇；

(3) 重新计算已经得到的各个簇的质心；

(4) 迭代步骤(2)-(3)直至新的质心与原来的质心相等或小于设定的阈值，算法结束。

随意找几个数据简单模拟（借用当年老师教的方法^_^）算法如下：



假设有 6 个点 A1，A2，…，A6：

	X	Y
A1	1	3
A2	2	2
A3	2	1
A4	6	6
A5	7	8
A6	8	7

要聚成 2 类，算法过程如下：

(1) 假设选择 A1 和 A2 为初始质心；

(2) 计算 A3-A6 与 A1 和 A2 的距离，这里用欧氏距离公式 d = sqrt((x1-x2)²+(y1-y2)²)：

	A1	A2
A3	2.24	1
A4	5.83	5.66

A5	6.4	6.4
A6	8.06	7.81

(3) 根据与 A1 和 A2 距离的比较，A3、A4、A6 都离 A2 近，A5 与 A1 和 A2 距离相同，假设 A5 也分到 A2 这一簇，因此形成新的两簇：

簇 1：A1

簇 2：A2，A3，A4，A5，A6

(4) 计算新簇的质心

簇 1 质心：A1

簇 2：新质心“C_temp”计算用每个维度的平均值

((A2.x+A3.x+A4.x+A5.x+A6.x)/5 , (A2.y+A3.y+A4.y+A5.y+A6.y)/5)=(5, 4.8)

	A1	C_temp
A2	1.41	4.1
A3	2.24	4.84
A4	5.83	1.56
A5	6.4	3.77
A6	8.06	3.72

(5) 根据距离数据被分成了新的 2 簇：

簇 1：A1，A2，A3

簇 2：A4，A5，A6

新质心 1 “C_temp1”：((A1.x+A2.x+A3.x)/3 , (A1.y+A2.y+A3.y)/3)=(1.67, 2)

新质心 2 “C_temp2”：((A4.x+A5.x+A6.x)/3 , (A4.y+A5.y+A6.y)/3)=(7, 7)

	C_temp1	C_temp2
A1	1.2	7.21
A2	0.33	7.07
A3	1.05	7.81
A4	5.89	1.41
A5	6.66	1
A6	6.71	1

(6) 依据距离，簇 1 和簇 2 与前一轮一样，收敛，聚类结束，bingo ٩(◕‿◕)۶

簇 1：A1，A2，A3

簇 2：A4，A5，A6

提示：

(1) 在 K-means 算法 k 值通常取决于人的主观经验；

(2) 距离公式常用欧氏距离和余弦相似度公式，前者是根据位置坐标直接计算的，主要体现个体数值特征的差异，而后者更多体现了方向上的差异而不是位置上的，cos θ 越接近 1 个体越相似，可以修正不同度量标准不统一的问题；

(3) K-means 算法获得的是局部最优解，在算法中，初始聚类中心常常是随机选择的，一旦初始值选择的不好，可能无法得到有效的聚类结果。