K-Means

1 问题引入

假设你是一家淘宝店的店主。为了感谢消费者的关顾，你准备年前进行感恩大回馈活动。你想要针对不同的消费群体进行不同的回馈方式。但是根据已有的消费者信息，你并不知道应该把消费者分为几个群体进行回馈，也不知道根据什么特点来对消费者进行划分。

无监督学习：训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质和规律。

聚类：将数据集中的样本划分成为若干个子集，每个子集称为一个簇。

常用的聚类应用场景：如 Web 搜索出来的结果是大量的无标签数据的集合，可以利用聚类算法将大量的网页进行分组，便于搜索结果的查看；面对大量消费者的购买记录，也是无标签的数据信息，可以利用聚类将消费者进行划分，识别出不同的消费群体，从来更好的组织营销的手段。

2 k-means

给定样本集 $D=\{x_1,x_2,...x_m\}$，k-均值算法针对聚类所得簇划分 $C=\{C_1,C_2,...C_k\}$ 最小平方误差：

$$E=\sum_{i=1}^{k}\sum_{x\in C_i}\| x-\mu_i \|_2^2$$

其中 μ_i 为簇 C_i 的均值向量。

2.1 步骤

（1）选取 k 个初始聚类中心；

（2）计算每个数据点到聚类中心的距离，将数据点分配给离它最近的聚类中心。

（3）重新计算每个类的聚类中心，计算方法：聚类中所有点的均值即为新的聚类中心；

（4）重复第（2）（3）步，直至收敛，收敛条件：没有或最少数目数据点被重新分配给不同聚类；没有或最少数目聚类中心发生变化；或迭代到指定数目。

2.2 对噪声点敏感

2.2.1 问题表现

噪声点很大程度上会影响均值。上述的例子，增加一个噪声点。

例子：(1，2，3，8，9， 10，31)按照 k 为 2 来进行聚类。

最终聚类结果：(1,2,3,8,9,10)、(31)。效果显然不好。

2.2.2 解决

①k-中心点算法

因为一个具有很大极端值的对象会扭曲数据分布。那么我们可以考虑新的簇中心不选择均值而是选择簇内的某个对象，只要使总的代价降低就可以。

选用簇中位置最中心的对象，试图对 n 个对象给出 k 个划分；代表对象也被称为是中心点，其他对象则被称为非代表对象；最初随机选择 k 个对象作为中心点，该算法反复地用非代表对象来代替代表对象，试图找出更好的中心点，以改进聚类的质量；在每次迭代中，所有可能的对象对被分析，每个对中的一个对象是中心点，而另一个是非代表对象。对可能的各种组合，估算聚类结果的质量；一个对象O_j可以被使最大平方-误差值减少的对象代替；在一次迭代中产生的最佳对象集合成为下次迭代的中心点。

输入：

K：结果簇的个数

D：包含 n 个对象的数据集合

输出：k 个簇的集合

方法：

(1) 从 D 中随机选择 k 个对象作为初始的代表对象或种子；

(2) Repeat

(3) 讲每个剩余的对象分配到最近的代表对象所代表的簇；

(4) 随机的选择一个非代表对象O_{random}

(5) 计算用O_{random}代替代表对象O_j的总代价 S；

(6) If S<0，thenO_{random}替代O_j，形成新的 k 个代表对象的集合；

(7)until 不发生变化

举例：

数据散点图

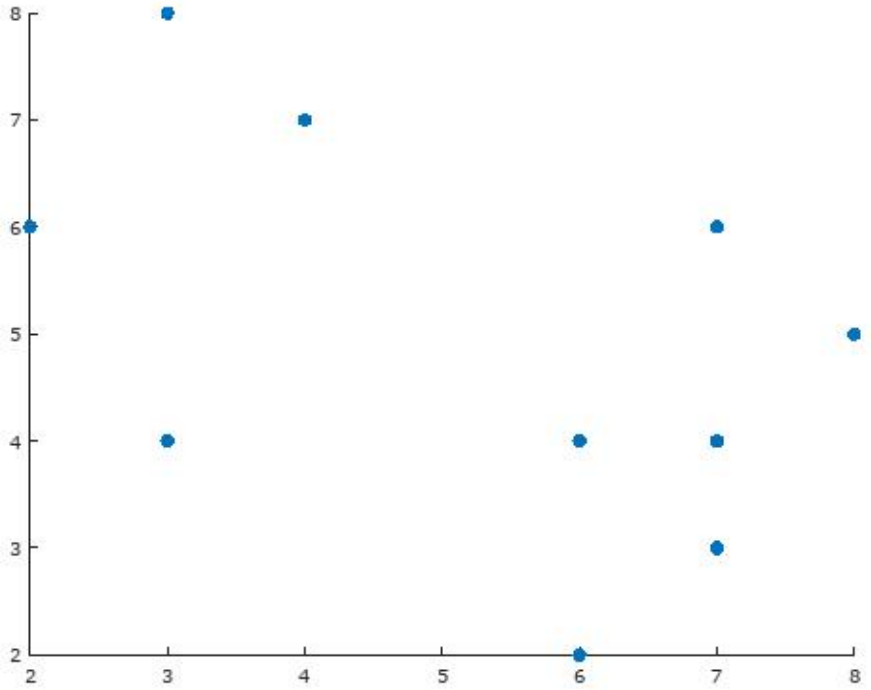

图 2.1

选择 C1 =(3,4) 和 C2=(7，3)两个中心点计算 cost

Data point		Distance to	
i	x_i	C1=(3,4)	C2=(7,3)
1	(2 , 6)	3	8
2	(3 , 4)	0	5
3	(3 , 8)	4	9
4	(4 , 7)	4	7
5	(6 , 2)	5	2
6	(6 , 4)	3	2
7	(7 , 3)	5	0
8	(7 , 4)	4	1
9	(8 , 5)	6	3
10	(7 , 6)	6	3
Cost		11	11

选非中心点(7,4)来替换 C2=(7,3)

Data point		Distance to	
i	x_i	C1=(3,4)	C2=(7,4)
1	(2 , 6)	3	7
2	(3 , 4)	0	4
3	(3 , 8)	4	8
4	(4 , 7)	4	6
5	(6 , 2)	5	3
6	(6 , 4)	3	1
7	(7 , 3)	5	1
8	(7 , 4)	4	0
9	(8 , 5)	6	2
10	(7 , 6)	6	2
Cost		11	9

2.3 初始点的选择

2.3.1 问题表现

初始点选择不同可能会影响最终的聚类结果如图 2.1 和 2.2 所示。

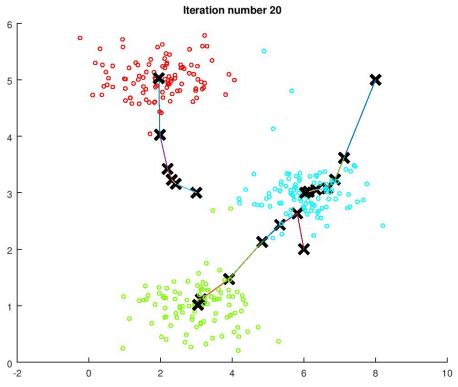

图 2.2 效果好

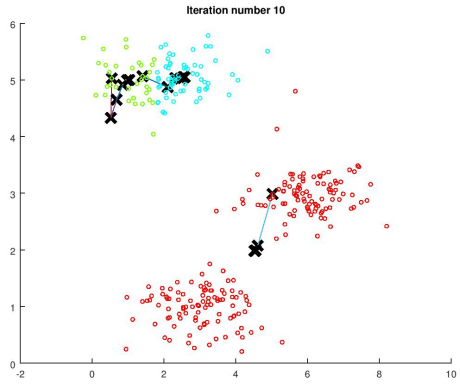

图 2.3 效果差

2.3.2 解决

①k-means++

主要思想：离当前已有聚类中心较远的点有更大的概率被选为下一个聚类中心。（初始点之间的距离尽可能的远）

步骤：（初始点的选择）

- 步骤一：随机选取一个样本作为第一个聚类中心 c1；
- 步骤二：
  - 计算每个样本与当前已有类聚中心最短距离（即与最近一个聚类中心的距离），用 D(x)表示；
  - 这个值越大，表示被选取作为聚类中心的概率较大；
  - 最后，用轮盘法选出下一个聚类中心；
- 步骤三：重复步骤二，知道选出 k 个聚类中心。

举例：

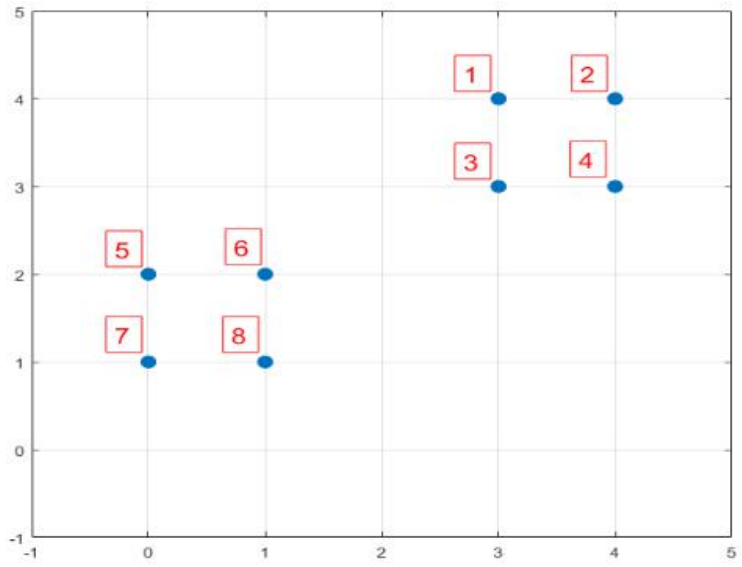

图 2.4

假设 6 号点被选择为第一个初始聚类中心，那在进行步骤二时每个样本的 D(x)和被选择为第二个聚类中心的概率如下表所示：

序号	①	②	③	④	⑤	⑥	⑦	⑧
$D(x)$	$2\sqrt{2}$	$\sqrt{13}$	$\sqrt{5}$	$\sqrt{10}$	1	0	$\sqrt{2}$	1
$D(x)^2$	8	13	5	10	1	0	2	1
$P(x)$	0.2	0.325	0.125	0.25	0.025	0	0.05	0.025
Sum	0.2	0.525	0.65	0.9	0.925	0.925	0.975	1

图 2.5

当已经存在多个中心点时，选取下一个点的 D(x)取离已有中心点最近的距离。

②多次随机初始化

2.4 K 值的选择

2.4.1 问题表现

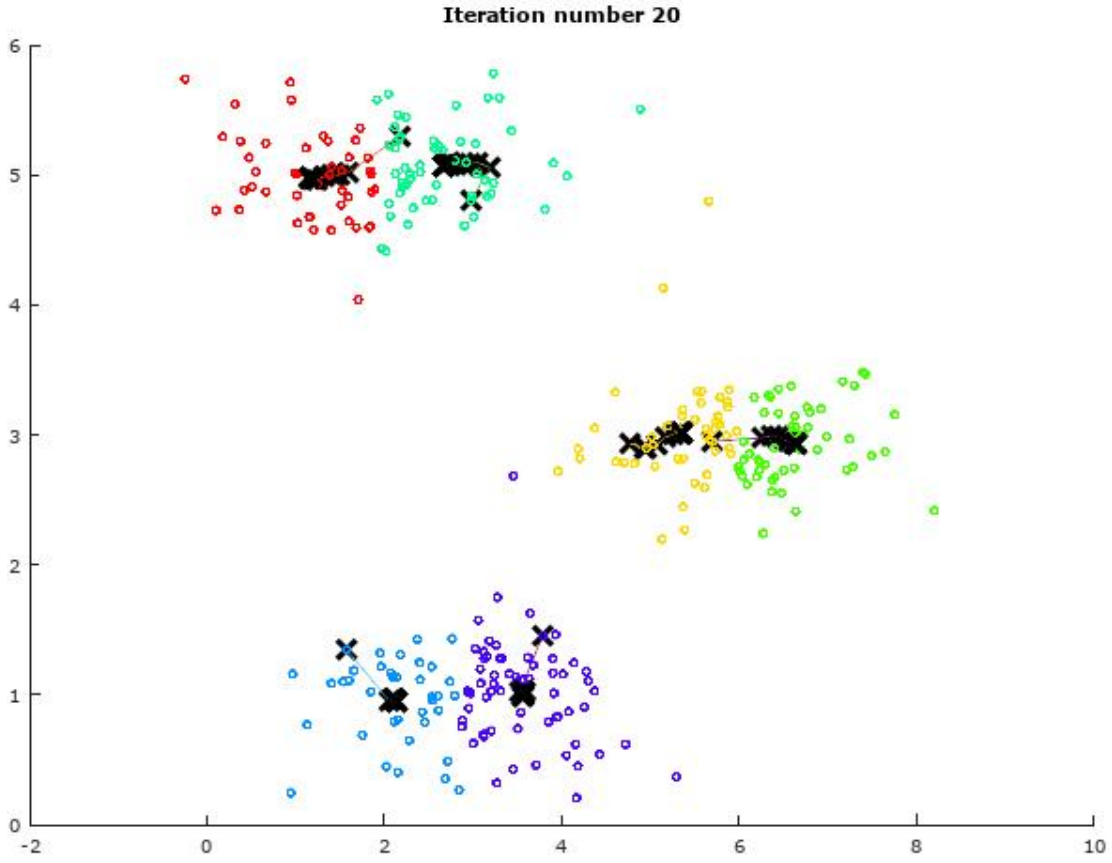

图 2.6

2.4.1 解决

①肘部法则

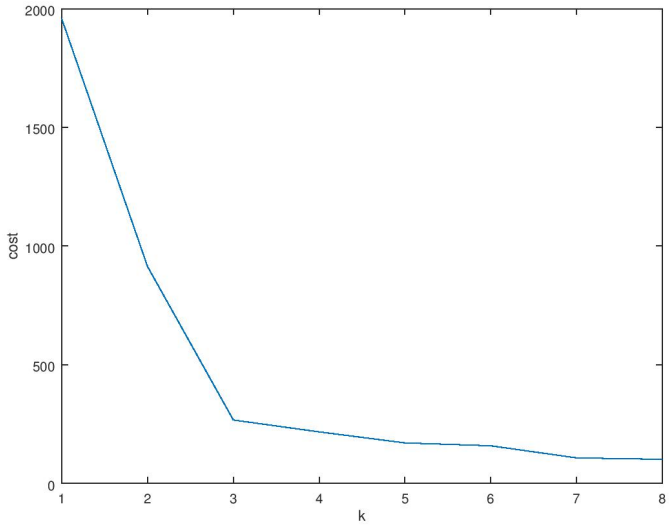

图 2.7

②Canopy 算法

Canopy 算法的优势在于不用指定 k 值，解决了 k-means 需要人为指定 k 值的问题，它是一种基于 k-means 算法的一种优化的聚类方法，用于 k-means 之前的粗聚类。Canopy 聚类在第一阶段选择简单、计算代价较低的方法计算对象相似性，将相似的对象放在一个子集中，这个子集被叫做 Canopy，通过一系列计算得到若干 Canopy，Canopy 之间可以是重叠的，但不会存在某个对象不属于任何 Canopy 的情况，可以把这一阶段看作数据预处理。

步骤：

（1）设样本集合为 S，确定两个阈值 t1 和 t2，且 t1>t2，t1 和 t2 的值通过交叉验证得到。

（2）任取一个样本点 p，作为一个 Canopy，记为 C，从 S 中移除 p。

（3）计算 S 中所有点到 p 的距离 dist

（4）若 dist<t1，则将相应点归到 C,作为弱关联。

（5）若 dist<t2，则将相应点移出 S，作为强关联。

（6）重复（2）~（5），直至 S 为空。

效果图：

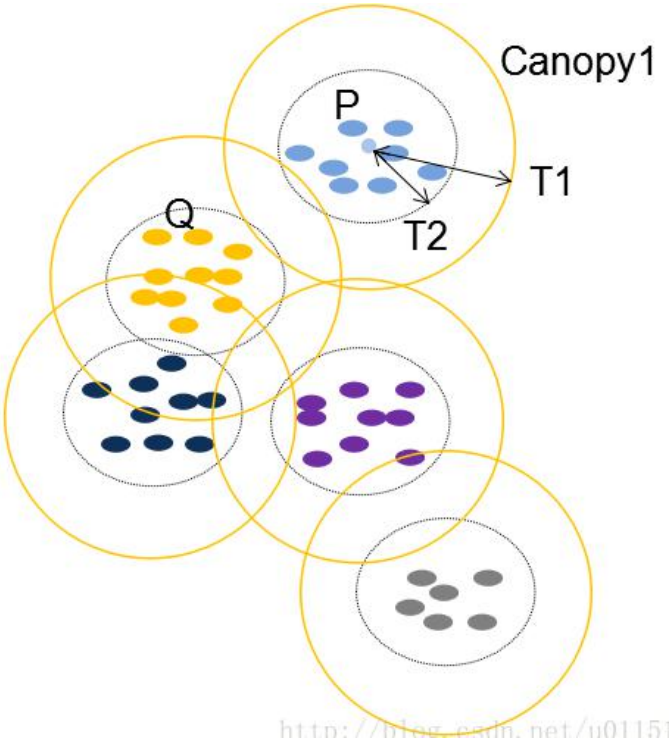

图 2.8

3 其他

(1)k-means 算法本身的存在的缺陷是聚类出来的簇多是球形的。

(2)性能度量：

①外部指标：通过参考模型来度量聚类效果，常用的有 Rand 指数等（机器学习 P.198）

②内部指标：其中之一就是簇内的紧密度，即可以通过 E 来衡量。第二是簇之间的分离度。簇内紧密度高，簇之间分离度高，才是较优的结果。

(3)标称属性(非数值)处理：非数值没有办法进行距离度量，可以用 VDM 的方法对离散值进行距离度量(机器学习 P.200)