5.3 K-均值算法要点

- （1）选定某种距离作为数据样本间的相似性度量
- k-means聚类算法不适合处理离散型属性，对连续型属性比较适合。因此在计算数据样本之间的距离时，可以根据实际需要选择欧式距离、曼哈顿距离或者明考斯距离中的一种来作为算法的相似性度量，其中最常用的是欧式距离。

5.3 K-均值算法要点

- （2）评价聚类性能的准则函数
- k-means聚类算法使用误差平方和准则函数来评价聚类性能。
- 给定数据集X，其中只包含描述属性，不包含类别属性。假设X包含k个聚类子集X1,X2,…XK；各个聚类子集中的样本数量分别为n1，n2,…,nk；各个聚类子集的均值代表点（也称聚类中心）分别为m1，m2,…,mk。
- 则误差平方和准则函数公式为：

$$E = \sum_{i=1}^{k} \sum_{p \in X_i} \left\| p - m_i \right\|^2$$

5.3 K-均值算法要点

- （3）相似度的计算根据一个簇中对象的平均值来进行。
  - 1）将所有对象随机分配到k个非空的簇中。
  - 2）计算每个簇的平均值，并用该平均值代表相应的簇。
  - 3）根据每个对象与各个簇中心的距离，分配给最近的簇。
  - 4）然后转2），重新计算每个簇的平均值。这个过程不断重复直到满足某个准则函数才停止 。

【例2.7】

对表2.6中的数据进行K-means聚类分析。

用于K-means算法的数据集

表2.6 用于K-means算法的数据集

Instance	x	y
1	1.0	1.0
2	2.0	1.5
3	4.0	3.5
4	5.0	4.5
5	3.5	5

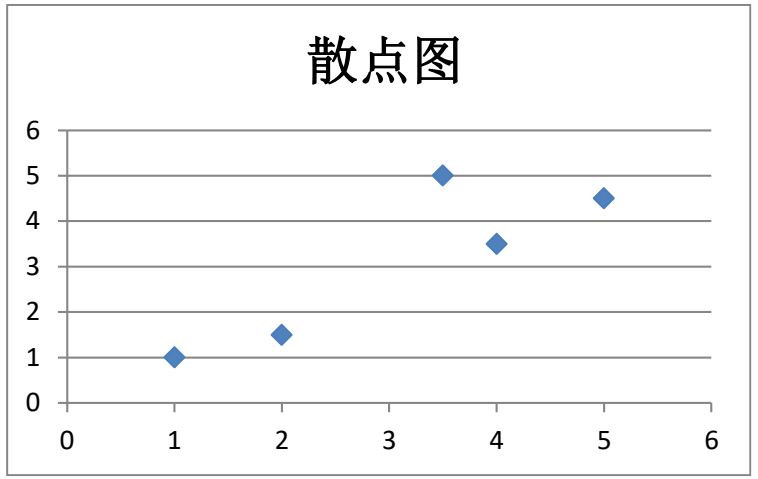

图2.18 表2.6中数据的坐标映射

步骤

（1）设置 K 值为2。

（2）任意选择两个点分别作为两个簇的初始簇中心。假设选择实例1作为第1个簇中心，实例2作为第2个簇中心。

（3）使用式2.9，计算其余实例与两个簇中心的简单欧氏距离（Euclidean Distance），结果如表2.7所示。

第一次到第三次迭代中实例与簇之间的简单欧氏距离

表2.7 第一次到第三次迭代中实例与簇之间的简单欧氏距离

簇中心 C_1 = (1.0,1.0) 和C_2=(2.0,1.5)			簇中心 C_1 = (1.0,1.0) 和 C_2=(3.625,3.625)		簇中心 C_1 = (1.5,1.25) 和 C_2=(4.17,4.33)	
Instance	C_1	C_2	C_1	C_2	C_1	C_2
1	0	1.12	0	3.71	0.56	4.60
2	1.12	0	1.12	2.68	0.56	3.57
3	3.91	2.83	3.91	0.40	3.36	0.85
4	5.32	4.24	5.32	1.63	4.78	0.85
5	4.72	3.81	4.72	1.38	4.25	0.95

步骤

（4）重新计算新的簇中心。

- 对于簇1簇中心不变，即C_1 = (1.0,1.0)。
- 对于簇2：x = (2.0+4.0+5.0+3.5) / 4 = 3.625，y = (1.5+3.5+4.5+5) / 4 = 3.625。 （第2、3、4、5个实例）
- 得到新的簇中心C_1= (1.0,1.0) 和 C_2= (3.625,3.625)，因为簇中心发生了变化，算法必须执行第二次迭代，重复步骤（3）。
- 第二次迭代之后的结果导致了簇的变化：{1,2}和{3,4,5}。

Instance	x	y
1	1.0	1.0
2	2.0	1.5
3	4.0	3.5
4	5.0	4.5
5	3.5	5

第一次到第三次迭代中实例与簇之间的简单欧氏距离

表2.7 第一次到第三次迭代中实例与簇之间的简单欧氏距离

簇中心 C_1 = (1.0,1.0) 和C_2=(2.0,1.5)			簇中心 C_1 = (1.0,1.0) 和 C_2=(3.625,3.625)		簇中心 C_1 = (1.5,1.25) 和 C_2=(4.17,4.33)	
Instance	C_1	C_2	C_1	C_2	C_1	C_2
1	0	1.12	0	3.71	0.56	4.60
2	1.12	0	1.12	2.68	0.56	3.57
3	3.91	2.83	3.91	0.40	3.36	0.85
4	5.32	4.24	5.32	1.63	4.78	0.85
5	4.72	3.81	4.72	1.38	4.25	0.95

步骤

（5）重新计算每个簇中心。

- 对于簇1：x = (1.0+2.0) / 2= 1.5，y = (1.0+1.5) / 2 = 1.25。
- 对于簇2：x = (2.0+5.0+3.5) / 3= 4.17，y = (3.5+4.5+5) / 3 = 4.33。

- 这次迭代后簇中心再次改变。因此，该过程继续进行第三次迭代，结果形成{1,2}和{3,4,5}两个簇，与第二次迭代后形成的簇完全一样，若继续计算新簇中心的话，簇中心的值一定不变，至此，算法结束。

结果

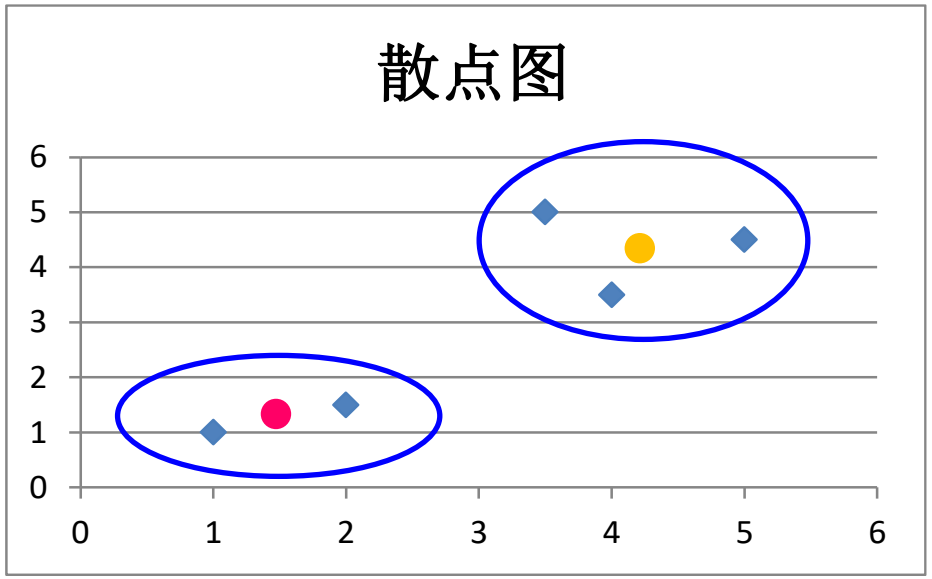

图2.19 表2.6中数据的聚类结果

K-means算法的最优聚类通常为——簇中所有实例与簇中心的误差平方和最小的聚类。

K-Means的细节问题

- 1. K值怎么定？我怎么知道应该几类？
  - 答：这个没有确定的做法，分几类主要取决于个人的经验与感觉，通常的做法是多尝试几个K值，看分成几类的结果更好解释，更符合分析目的等。
  - 或者可以把各种K值算出的SSE（即误差平方和）做比较，取最小的SSE的K值。

K-Means的细节问题

- 2. 初始的K个质心怎么选?
  - 答：最常用的方法是随机选，初始质心的选取对最终聚类结果有影响，因此算法一定要多执行几次，哪个结果更reasonable，就用哪个结果。
  - 当然也有一些优化的方法，第一种是选择彼此距离最远的点，具体来说就是先选第一个点，然后选离第一个点最远的当第二个点，然后选第三个点，第三个点到第一、第二两点的距离之和最小，以此类推。
  - 第二种是先根据其他聚类算法（如层次聚类）得到聚类结果，从结果中每个分类选一个点。

K-Means的细节问题

- 3. K-Means会不会陷入一直选质心的过程，永远停不下来？
  - 答：不会，数学证明K-Means一定会收敛，大致思路是利用SSE的概念（也就是误差平方和），即每个点到自身所归属质心的距离的平方和，这个平方和是一个函数，然后能够证明这个函数是可以最终收敛的函数。

- 4. 判断每个点归属哪个质心的距离怎么算？
  - 答：第一种，欧几里德距离。第二种，余弦相似度。

K-Means的细节问题

- 例子：歌手大赛，三个评委给三个歌手打分，第一个评委的打分（10，8，9） 第二个评委的打分（4，3，2），第三个评委的打分（8，9，10）。
  - 如果采用余弦相似度来看每个评委的差异，虽然每个评委对同一个选手的评分不一样，但第一、第二两个评委对这四位歌手实力的排序是一样的，只是第二个评委有更高的评判标准，说明第一、第二个评委对音乐的品味上是一致的。
  - 因此，用余弦相似度来看，第一、第二个评委为一类人，第三个评委为另外一类。
  - 如果采用欧氏距离， 第一和第三个评委的欧氏距离更近，就分成一类人了，但其实不太合理，因为他们对于四位选手的排名都是完全颠倒的。
- 总之，如果注重数值本身的差异，就应该用欧氏距离，如果注重的是上例中的这种差异，就要用余弦相似度来计算。

K-Means的细节问题

- 5. 各属性的单位、量纲要一致
- 比如X的单位是米，Y也是米，那么距离算出来的单位还是米，是有意义的。但是如果X是米，Y是吨，用距离公式计算就会出现“米的平方”加上“吨的平方”再开平方，最后算出的东西没有数学意义，这就有问题了。
- 还有，即使X和Y单位一致，但是如果数据中X整体都比较小，比如都是1到10之间的数，Y很大，比如都是1000以上的数，那么，在计算距离的时候Y起到的作用就比X大很多，X对于距离的影响几乎可以忽略，这也有问题。
- 因此，如果选择欧氏距离计算距离，数据集又出现了上面所述的情况，就一定要进行数据的标准化，即将数据按比例缩放，使之落入一个小的特定区间。去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行计算和比较。

K-Means的细节问题

- 6. 每一轮迭代如何选出新的质心?
  - 答：各个维度的算术平均，比如（X1，Y1，Z1）、（X2，Y2，Z2）、（X3，Y3，Z3），那就新质心就是（X1+X2+X3）/3，（Y1+Y2+Y3）/3，（Z1，Z2，Z3）/3，这里要注意，新质心不一定是实际的一个数据点。

- 7. 关于离群值?
  - 答：离群值就是远离整体的，非常异常、非常特殊的数据点，在聚类之前应该将这些“极大”“极小”之类的离群数据都去掉，否则会对于聚类的结果有影响。但是，离群值往往自身就很有分析的价值，可以把离群值单独作为一类来分析。

K-means算法小结

- 优势
  - 非常受欢迎的算法，容易理解，实现简单。
  - 对处理大数据集，该算法是相对可伸缩和高效率的。
  - 当结果簇是密集的，而簇与簇之间区别明显时，它的效果较好。

K-means算法小结

- 局限性

（1）在簇的平均值被定义的情况下才能使用，因此只能处理数值型数据，若数据集中有分类类型的属性，要么将该属性删除，要么将其转换成等价的数值数据。

（2）算法开始前，需要随机选择*K*值作为初始的簇个数（带有随意性，错误的选择将影响聚类效果）。通常选择不同的*K*值进行重复实验，期望找到最佳*K*值。

（3）对初值敏感，对于不同的初始值，可能会导致不同结果。

（4）它对于“躁声”和孤立点数据是敏感的，少量的该类数据能够对平均值产生极大的影响。

（5）对于聚类贡献不大的属性可能会对聚类效果造成影响。在聚类之前需对属性进行选择。

（6）簇的解释困难。可使用有指导的挖掘工具对无指导聚类算法所形成簇的性质作进一步的解释。