数据挖掘第六次作业

李晨昊 2017011466

2020 年 5 月 27 日

目录

1 K-means 聚类

给定下列 13 个数据点：

(1,3); (1,2); (2,1); (2,2); (2,3); (3,2); (5,3); (4,3); (4,5); (5,4); (5,5); (6,4); (6,5)

使用 K-means 算法对它们进行聚类。令 k=2，初始中心点为 (0,4) 和 (6,5)，写出聚类过程。

Iter 1:

- (1, 3)：距离 (0, 4) 为 1.414，距离 (6, 5) 为 5.385，选 (0, 4)
- (1, 2)：距离 (0, 4) 为 2.236，距离 (6, 5) 为 5.831，选 (0, 4)
- (2, 1)：距离 (0, 4) 为 3.606，距离 (6, 5) 为 5.657，选 (0, 4)
- (2, 2)：距离 (0, 4) 为 2.828，距离 (6, 5) 为 5.000，选 (0, 4)
- (2, 3)：距离 (0, 4) 为 2.236，距离 (6, 5) 为 4.472，选 (0, 4)
- (3, 2)：距离 (0, 4) 为 3.606，距离 (6, 5) 为 4.243，选 (0, 4)
- (5, 3)：距离 (0, 4) 为 5.099，距离 (6, 5) 为 2.236，选 (6, 5)

- (4, 3)：距离 (0, 4) 为 4.123，距离 (6, 5) 为 2.828，选 (6, 5)
- (4, 5)：距离 (0, 4) 为 4.123，距离 (6, 5) 为 2.000，选 (6, 5)
- (5, 4)：距离 (0, 4) 为 5.000，距离 (6, 5) 为 1.414，选 (6, 5)
- (5, 5)：距离 (0, 4) 为 5.099，距离 (6, 5) 为 1.000，选 (6, 5)
- (6, 4)：距离 (0, 4) 为 6.000，距离 (6, 5) 为 1.000，选 (6, 5)
- (6, 5)：距离 (0, 4) 为 6.083，距离 (6, 5) 为 0.000，选 (6, 5)

聚为两类，分别为 {(1, 3), (1, 2), (2, 1), (2, 2), (2, 3), (3, 2)} 和 {(5, 3), (4, 3), (4, 5), (5, 4), (5, 5), (6, 4), (6, 5)}，中心点分别为 (1.833, 2.167) 和 (5，4.143)。

Iter 2:

- (1, 3)：距离 (1.833, 2.167) 为 1.179，距离 (5, 4.143) 为 4.160，选 (1.833, 2.167)
- (1, 2)：距离 (1.833, 2.167) 为 0.850，距离 (5, 4.143) 为 4.538，选 (1.833, 2.167)
- (2, 1)：距离 (1.833, 2.167) 为 1.179，距离 (5, 4.143) 为 4.345，选 (1.833, 2.167)
- (2, 2)：距离 (1.833, 2.167) 为 0.236，距离 (5, 4.143) 为 3.687，选 (1.833, 2.167)
- (2, 3)：距离 (1.833, 2.167) 为 0.850，距离 (5, 4.143) 为 3.210，选 (1.833, 2.167)
- (3, 2)：距离 (1.833, 2.167) 为 1.179，距离 (5, 4.143) 为 2.931，选 (1.833, 2.167)
- (5, 3)：距离 (1.833, 2.167) 为 3.274，距离 (5, 4.143) 为 1.143，选 (5, 4.143)
- (4, 3)：距离 (1.833, 2.167) 为 2.321，距离 (5, 4.143) 为 1.519，选 (5, 4.143)
- (4, 5)：距离 (1.833, 2.167) 为 3.567，距离 (5, 4.143) 为 1.317，选 (5, 4.143)
- (5, 4)：距离 (1.833, 2.167) 为 3.659，距离 (5, 4.143) 为 0.143，选 (5, 4.143)
- (5, 5)：距离 (1.833, 2.167) 为 4.249，距离 (5, 4.143) 为 0.857，选 (5, 4.143)
- (6, 4)：距离 (1.833, 2.167) 为 4.552，距离 (5, 4.143) 为 1.010，选 (5, 4.143)
- (6, 5)：距离 (1.833, 2.167) 为 5.039，距离 (5, 4.143) 为 1.317，选 (5, 4.143)

聚类结果与前一次迭代相同，迭代中止。

综上，聚类结果为 {(1, 3), (1, 2), (2, 1), (2, 2), (2, 3), (3, 2)} 和 {(5, 3), (4, 3), (4, 5), (5, 4), (5, 5), (6, 4), (6, 5)}。

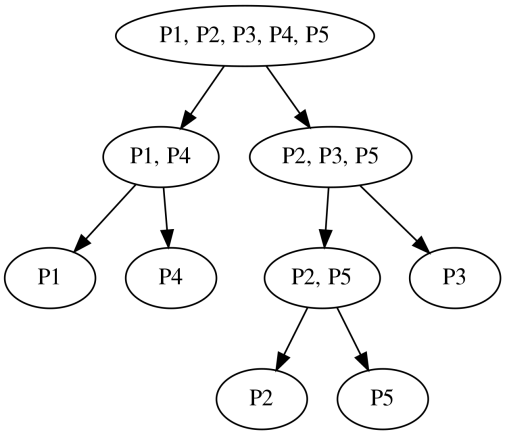
2 层次聚类

使用表中的相似度矩阵进行 Min 聚类和 Max 聚类，分别绘制树状图显示结果。树状图应当清楚地显示合并的次序。

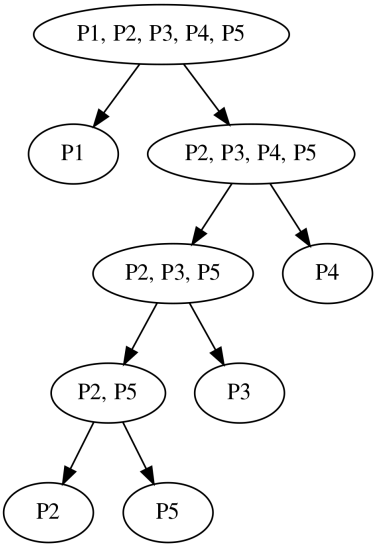
相似度矩阵

	P1	P2	P3	P4	P5
P1	1.00	0.10	0.41	0.55	0.35
P2	0.10	1.00	0.64	0.47	0.98
P3	0.41	0.64	1.00	0.44	0.85
P4	0.55	0.47	0.44	1.00	0.76
P5	0.35	0.98	0.85	0.76	1.00

- Min



- Max



3 比较 K-means 和 DBSCAN

为了简化比较，假定对于 K-Means 和 DBSCAN 都没有距离的限制，并且 DBSCAN 总是将于若干个核心点相关联的边界点指派到最近的核心点。至少写出四点。（提示：可以从复杂度、离群点、参数等方面考虑。）

1. K-means 的时间复杂度为 $O(tkn)$，其中 n 是对象个数，k 是簇的个数，t 是迭代次数，一般来说有 $t, k \ll n$；DBSCAN 最坏情况下时间复杂度为 $O(n^2)$，其中 n 是对象个数。这样来看一般 K-means 更高效一些.

2. K-means 受数据噪音和离群点影响较大，DBSCAN 则更加鲁棒。

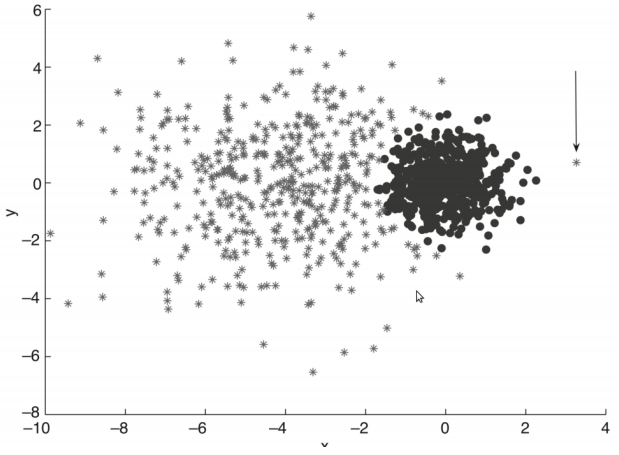
3. K-means 需要用户指定簇的个数作为参数，DBSCAN 则不需要，而是运行中自动确定簇的个数。

4. K-means 很难处理非凸形状的簇，DBSCAN 可以发现任意形状的簇。

4 比较 EM 聚类和 K-Means

下图显示具有两个簇的二维点集的聚类。左边的簇（点用星号标记）多少有点散开，而右边的簇（点用圆标记）是紧凑的。在紧凑簇的右边有一个单独的点（用箭头指出）属于散开的簇。该簇的中心比紧凑簇的中心远得多。解释为什么用 EM 聚类是可能的，但是用 K-Means 聚类不可能。

具有两个不同密度的簇的二维点集的 EM 聚类



这个聚类结果可能是 EM 聚类生成的，因为 EM 聚类基于高斯分布 $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$，因为左边的簇更分散，所以左边的簇对应的高斯分布的 σ 大于右边的簇的 σ。因此即使右边的单独的点距离右边的簇更近（即 $(x-\mu_{左})^2 > (x-\mu_{右})^2$），仍有可能有 $\frac{(x-\mu_{左})^2}{2\mu_{左}^2} < \frac{(x-\mu_{右})^2}{2\mu_{右}^2}$，即它属于左边的簇的概率仍然可能更高一些。

这个聚类结果不可能是 K-Means 聚类生成的，因为右边的单独的点距离右边的簇的中心比距离自身所在簇的中心更近，只要再进行一次迭代，最后一次迭代的时候不可能将它划分到左边的簇，因此这不可能是 K-Means 聚类的最终结果。

5 如何选择合适的聚类算法

在确定使用哪种类型的聚类算法时，需要考虑各种各样的因素。简单分析选择合适的聚类算法时，涉及对哪些问题的考虑。至少写出四点。

1. 时间和空间效率。例如 K-Means 算法简单且效率较高，如果用它就能达到不错的效果(即数据集中没有 K-Means 难以处理的哪几个特征)，那么可以优先使用它，而无需采用更复杂的算法。或者如果数据规模很大，那么有些复杂度较高的算法即使效果好，也无法在实际工程中使用。

2. 鲁棒性，包括算法对数据中的噪音和参数设置的敏感性。例如 K-Means 受噪音影响较大，而且很容易受到初始聚类的影响，一些种类的层次聚类也受噪音影响较大，而 DBSCAN 的鲁棒性则更好。如果实际数据中有较多的噪音，就应该选用鲁棒性更好的算法。

3. 数据的特征。不同的算法对数据特征有不同的要求，例如 K-Means 要求数据必须有均值的概念，所以很难处理标签数据，层次聚类只要求数据有相似度的概念。如果实际数据不具备某一算法必要的特征，那么就很难应用它。

4. 数据集的特征，也就是簇的形状和大小等特征。例如 K-Means 和一些种类的层次聚类很难处理非凸形状的簇，而 DBSCAN 可以处理不同大小和形状的簇。这一点一般来说很难事先知道，很多时候我们都是看着聚类的结果去讨论数据集的特征的，所以只能多尝试几种聚类算法，根据结果来调整。