K-means算法

主要内容

- ✓K-means算法
- ✓算法实例
- ✓算法优缺点

K-means算法概述

K-means算法， 也被称为*k*-平均或*k*-均值算法，是一种得到最广泛使用的聚类算法。它是将各个聚类子集内的所有数据样本的均值作为该聚类的代表点，算法的主要思想是通过迭代过程把数据集划分为不同的类别，使得评价聚类性能的准则函数达到最优（平均误差准则函数E），从而使生成的每个聚类（又称簇）内紧凑，类间独立。

聚类与分类的区别

- 聚类(clustering)是指根据“物以类聚”的原理，将本身没有类别的样本聚集成不同的组，这样的一组数据对象的集合叫做簇，并且对每一个这样的簇进行描述的过程。
- 在分类（ classification ）中，对于目标数据库中存在哪些类是知道的，要做的就是将每一条记录分别属于哪一类标记出来。
- 聚类分析也称无监督学习， 因为和分类学习相比，聚类的样本没有标记，需要由聚类学习算法来自动确定。聚类分析是研究如何在没有训练的条件下把样本划分为若干类。

欧氏距离

- 假设给定的数据集 $X=\{x_m \mid m=1,2,...,total\}$，X中的样本用d个描述属性$A_1, A_2 \cdots A_d$ （维度）来表示。
- 数据样本$x_i=(x_{i1}, x_{i2}, \cdots x_{id})$， $x_j=(x_{j1}, x_{j2}, \cdots x_{jd})$其中，$x_{i1}, x_{i2}, \cdots x_{id}$和$x_{j1}, x_{j2}, \cdots x_{jd}$分别是样本$x_i$和$x_j$对应d个描述属性$A_1, A_2, \cdots A_d$的具体取值。
- 样本xi和xj之间的相似度通常用它们之间的距离$d(x_i, x_j)$来表示，距离越小，样本x_i和x_j越相似，差异度越小；距离越大，样本x_i和x_j越不相似，差异度越大。

欧式距离公式如下：

$$d(x_i, x_j)=\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2}$$

平均误差准则函数

- **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$X_1, X_2, \cdots X_K$；各个聚类子集中的样本数量分别为n_1，$n_2, \cdots, n_k$;各个聚类子集的均值代表点（也称聚类中心）分别为m_1，$m_2, \cdots, m_k$。

- 误差平方和准则函数公式为：

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

K-means算法过程

算法 *k*-means算法

输入：簇的数目*k*和包含*n*个对象的数据库。

输出：*k*个簇，使平方误差准则最小。

算法步骤：

- 1.为每个聚类确定一个初始聚类中心，这样就有K 个 初始聚类中心。
- 2.将样本集中的样本按照最小距离原则分配到最邻近聚类
- 3.使用每个聚类中的样本均值作为新的聚类中心。
- 4.重复步骤2.3直到聚类中心不再变化。
- 5.结束，得到K个聚类

算法实例

O	x	y
1	0	2
2	0	0
3	1.5	0
4	5	0
5	5	2

数据对象集合**S**见表**1**，作为一个聚类分析的二维样本，要求的簇的数量**k=2**。

(1)选择 $O_1(0,2)$ ，$O_2(0,0)$ 为初始的簇中心，即 $M_1=O_1=(0,2)$ ， $M_2=O_2=(0,0)$ 。

(2)对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。

对 O_3 ：

$$d(M_1,O_3)=\sqrt{(0-1.5)^2+(2-0)^2}=2.5$$

$$d(M_2,O_3)=\sqrt{(0-1.5)^2+(0-0)^2}=1.5$$

显然 $d(M_2,O_3)\le d(M_1,O_3)$ ， 故将 O_3 分配给 C_2

O	x	y
1	0	2
2	0	0
3	1.5	0
4	5	0
5	5	2

- 对于 O_4 ： $d(M_1,O_4)=\sqrt{(0-5)^2+(2-0)^2}=\sqrt{29}$

$$d(M_2,O_4)=\sqrt{(0-5)^2+(0-0)^2}=5$$

- 因为 $d(M_2,O_4)\le d(M_1,O_4)$ 所以将 O_4 分配给 C_2
- 对于 O_5： $d(M_1,O_5)=\sqrt{(0-5)^2+(2-2)^2}=5$

$$d(M_2,O_5)=\sqrt{(0-5)^2+(0-2)^2}=\sqrt{29}$$

- 因为 $d(M_1,O_5)\le d(M_2,O_5)$ 所以将 O_5 分配给 C_1
- 更新，得到新簇 $C_1=\{O_1,O_5\}$ 和 $C_2=\{O_2,O_3,O_4\}$
- 计算平方误差准则，单个方差为

$$E_1=\left[(0-0)^2+(2-2)^2\right]+\left[(0-5)^2+(2-2)^2\right]=25$$

$M_1=O_1=(0,2)$

$E_2=27.25$ $M_2=O_2=(0,0)$

O	x	y
1	0	2
2	0	0
3	1.5	0
4	5	0
5	5	2

总体平均方差是：$E = E_1 + E_2 = 25 + 27.25 = 52.25$

（3）计算新的簇的中心。

$$M_1 = ((0+5)/2, (2+2)/2) = (2.5, 2)$$

$$M_2 = ((0+1.5+5)/3, (0+0+0)/3) = (2.17, 0)$$

重复（2）和（3），得到O_1分配给C_1；O_2分配给C_2，O_3分配给C_2，O_4分配给C_2，O_5分配给C_1。更新，得到新簇 $C_1 = \{O_1, O_5\}$ 和 $C_2 = \{O_2, O_3, O_4\}$ 。中心为 $M_1 = (2.5, 2)$ ，$M_2 = (2.17, 0)$ 。

单个方差分别为

$$E_1 = [(0-2.5)^2 + (2-2)^2] + [(2.5-5)^2 + (2-2)^2] = 12.5 \quad E_2 = 13.15$$

总体平均误差是：$E = E_1 + E_2 = 12.5 + 13.15 = 25.65$

由上可以看出，第一次迭代后，总体平均误差值**52.25~25.65**，显著减小。由于在两次迭代中，簇中心不变，所以停止迭代过程，算法停止。

K-means算法的优点分析

■主要优点：

- 是解决聚类问题的一种经典算法，简单、快速。
- 对处理大数据集，该算法是相对可伸缩和高效率的。
- 因为它的复杂度是0 (n k t) , 其中, n 是所有对象的数目, k 是簇的数目, t 是迭代的次数。通常k < <n 且t < <n 。
- 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算法的缺点分析

■主要缺点：

- 在簇的平均值被定义的情况下才能使用，这对于处理符号属性的数据不适用。
- 必须事先给出k（要生成的簇的数目），而且对初值敏感，对于不同的初始值，可能会导致不同结果。经常发生得到次优划分的情况。解决方法是多次尝试不同的初始值。
- 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。

K-means算法总结

- K-means 算法属于聚类分析方法中一种基本的且应用最广泛的划分算法；
- 它是一种已知聚类类别数的聚类算法。指定类别数为K，对样本集合进行聚类，聚类的结果由K 个聚类中心来表达；
- 基于给定的聚类目标函数（或者说是聚类效果判别准则），算法采用迭代更新的方法，每一次迭代过程都是向目标函数值减小的方向进行，最终的聚类结果使目标函数值取得极小值，达到较优的聚类效果。
- 使用平均误差准则函数E作为聚类结果好坏的衡量标准之一，保证了算法运行结果的可靠性和有效性。

谢谢观看！