

### 算法 1: Apriori 关联规则

输入: 事务集

输出: 强关联规则

$k \leftarrow 1$ ;

扫描事务集中的每个事务, 所有不重复的项目都是候选 1 项集;

计算所有候选 1 项集的支持度, 和最小支持度对比得到频繁 1 项集;

**while** 存在频繁  $k$  项集 **do**

    // 组合要求: (1) 组合后项目个数为  $k+1$  (2) 重复的组合只保留一个  
    由频繁  $k$  项集两两组合得到候选  $k+1$  项集;

    利用定理对候选  $k+1$  项集剪枝, 删掉所有不可能频繁的  $k+1$  项集

    计算候选  $k+1$  项集的支持度, 得到频繁  $k+1$  项集;

**if** 不存在频繁  $k+1$  项集 **then**

        | **break**;

**end if**

    所有频繁  $k$  项集  $\leftarrow$  所有频繁  $k+1$  项集;

$k \leftarrow k+1$ ;

**end while**

依次获取频繁  $k$  项集的所有真子集, 生成关联规则

计算所有关联规则的置信度, 返回所有强关联规则

制表位把竖线做出来: <https://zhuanlan.zhihu.com/p/660791698>

## 算法 2: Kmeans 聚类分析

**输入:** 各个样本的坐标  $x^i$ , 簇的数量  $k$

**输出:** 聚好的各个类

随机选择  $k$  个点的坐标作为初始聚类中心

**while** True **do**

    计算每个样本到各个聚类中心的距离, 将其分配到最近聚类中心所在簇;

    忽略聚类中心, 重新计算各个簇的聚类中心;

**if** 各个聚类中心都没有发生改变 **then**

**break**;

**end if**

**end while**

**return** 聚好的各个类