

### 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>

尝试: 0.5 1.5

## 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

### 算法 3：模拟退火算法

**输入：** 初始温度  $T_0$ ，终止温度  $T_f$ ，马尔科夫链长度  $L$

**输出：** 整个搜索过程遇到的最好解  $x_{best}$

$x_{cur} \leftarrow$  问题的一个初始解；

$x_{best} \leftarrow x_{cur}$ ；

$T_{cur} \leftarrow T_0$ ；

**while**  $T_{cur} \geq T_f$  **do**

    // 恒温过程，不断做局部搜索

**for** iteration  $\leftarrow 1$  to  $L$  **do**

$x_{new} \leftarrow$  使用邻域操作得到  $x_{cur}$  邻域内的一个解；

        // 如果  $x_{new}$  非常好，能量越低约好，直接作为  $x_{cur}$

**if**  $x_{new}$  与  $x_{cur}$  的能量差为负数 **then**

$x_{cur} \leftarrow x_{new}$ ；

**else**

            按 metropolis 公式计算的概率将  $x_{cur}$  设置为  $x_{new}$ ；

**end if**

        // 完成一次邻域移动，检查是否到了一个很好的点上

**if**  $x_{cur}$  比  $x_{best}$  的能量小 **then**

$x_{best} \leftarrow x_{cur}$ ；

**end if**

**end for**

    // 恒温过程结束，开始降温，这里使用指数降温公式

$T_{cur} = \alpha \times T_{cur}$ ；

**end while**

#### 算法 4：禁忌搜索算法

**输入：** 邻域元素个数  $n_0$ ，最大迭代次数  $N$ ，禁忌表长度  $L$

**输出：** 整个搜索过程遇到的最好解  $x_{best}$

$x_{cur} \leftarrow$  问题的一个初始解；

$x_{best} \leftarrow x_{cur}$ ；

**for**  $i \leftarrow 1$  to  $N$  **do**

$\text{move\_candidate}, x\_candidates \leftarrow$  获取  $n_0$  个具体的邻域操作及其对应的邻域解；

**for each** move in  $\text{move\_candidates}$  **do**

**if** move 在禁忌表中并且对应的解好于  $x_{best}$  **then**

            // 应用特赦准则

$x_{cur} \leftarrow$  move 对应的  $x$ ；

$x_{best} \leftarrow$  move 对应的  $x$ ；

**break**；

**end if**

**else**

        // 没有使用特赦准则确定性的接受  $x\_candidates$  里最好的作为  $x_{cur}$

$x_{cur} \leftarrow x\_candidates$  中不在禁忌表中的最好的解，如果都在禁忌表，就不变；

        把对应 move 放入禁忌表中；

        检查禁忌表长度，先进先出，只禁忌  $L$  个 move；

**end for**

**end for**

**注：** 考试的时候注意，现代优化那本书认为这个算法的参数包括（1）禁忌表（2）邻域结构（3）特赦准则（4）禁忌准则