关联规则挖掘

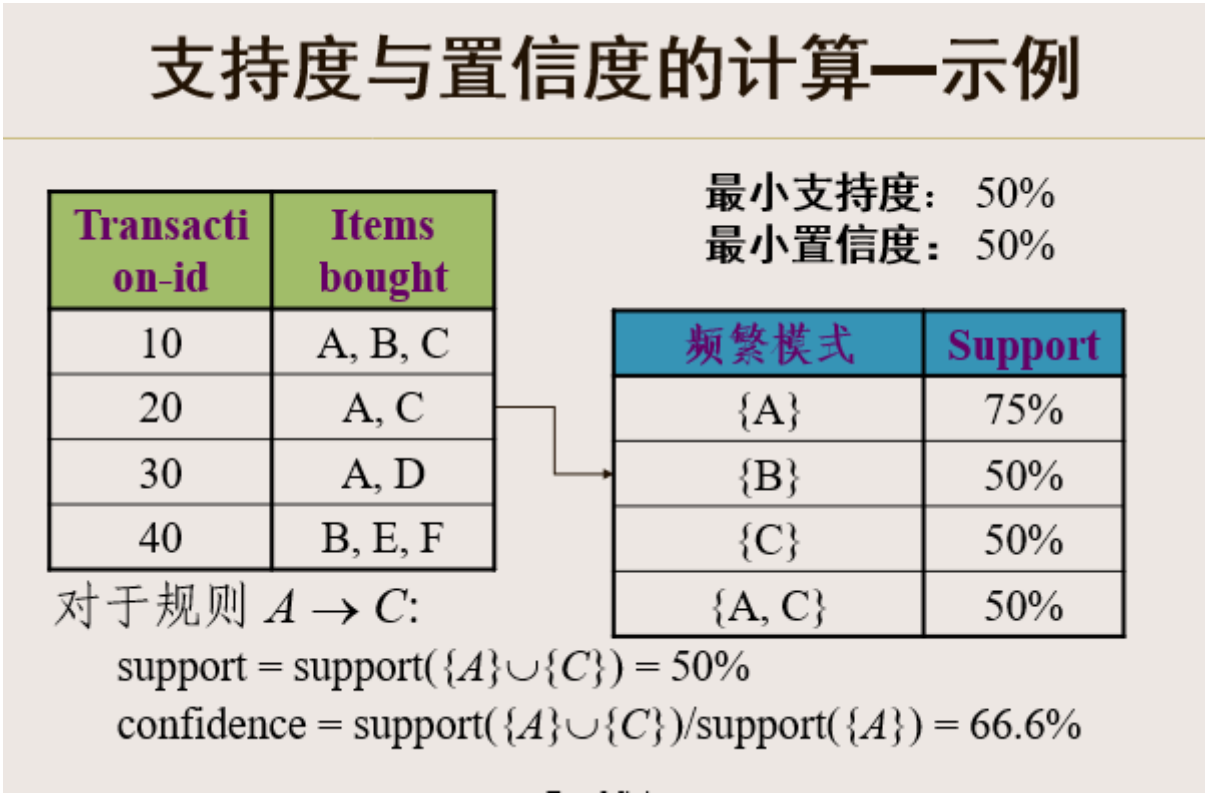
基本概念

支持度的计算

$$Support(X \rightarrow Y) = P(X \cup Y)$$
$$= \frac{\text{项集}\{X \cup Y\}\text{的支持度计数}}{\text{事务表中总的事务数}}$$
$$= \frac{\text{事务表中包含项集}\{X \cup Y\}\text{的事务数}}{\text{事务表中总的事务数}}$$

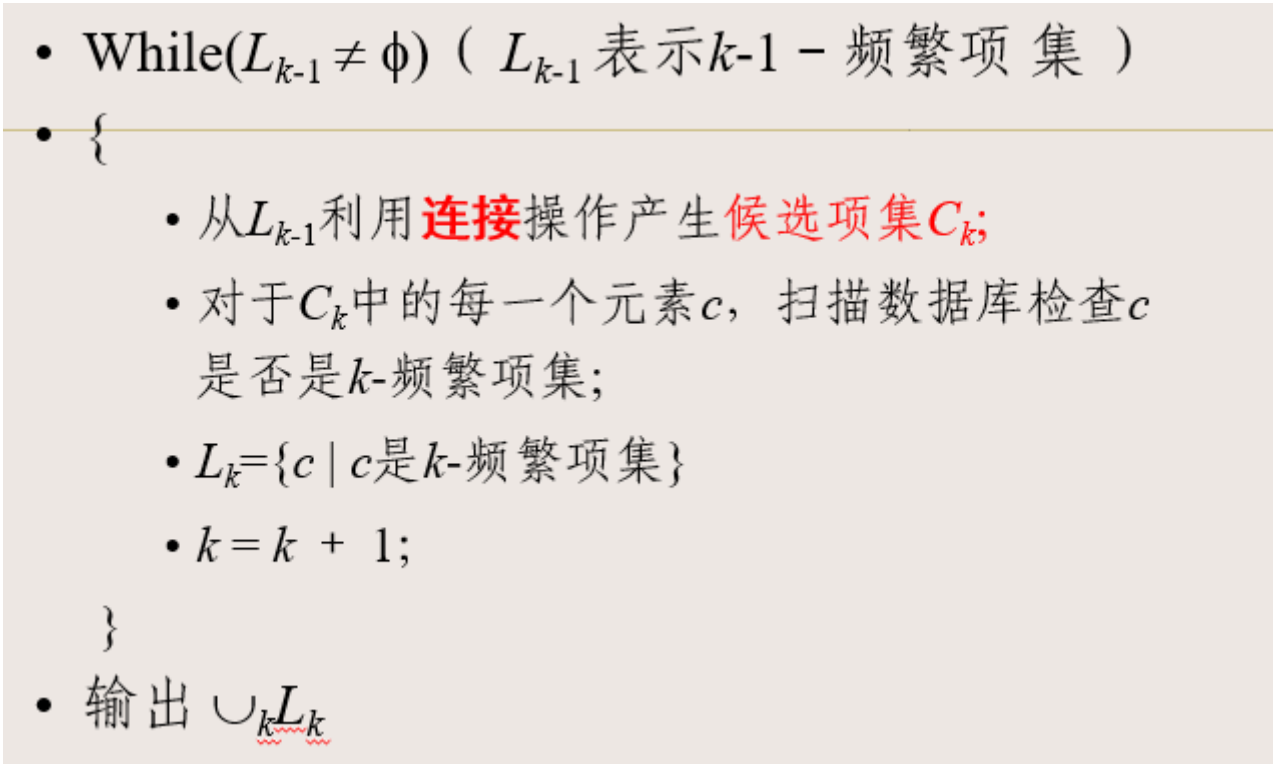
置信度的计算

$$Confidence(X \rightarrow Y) = P(Y \mid X)$$
$$= \frac{P(X \cup Y)}{P(X)}$$
$$= \frac{\text{项集}\{X \cup Y\}\text{的支持度计数}}{\text{项集}\{X\}\text{的支持度计数}}$$
$$= \frac{\text{事务表中包含项集}\{X \cup Y\}\text{的事务数}}{\text{事务表中包含项集}\{X\}\text{的事务数}}$$



Apriori算法

Apriori裁剪原理： 对于任意项集，如果它不是频繁集，则它的任何超集不用产生/测试!
算法流程：

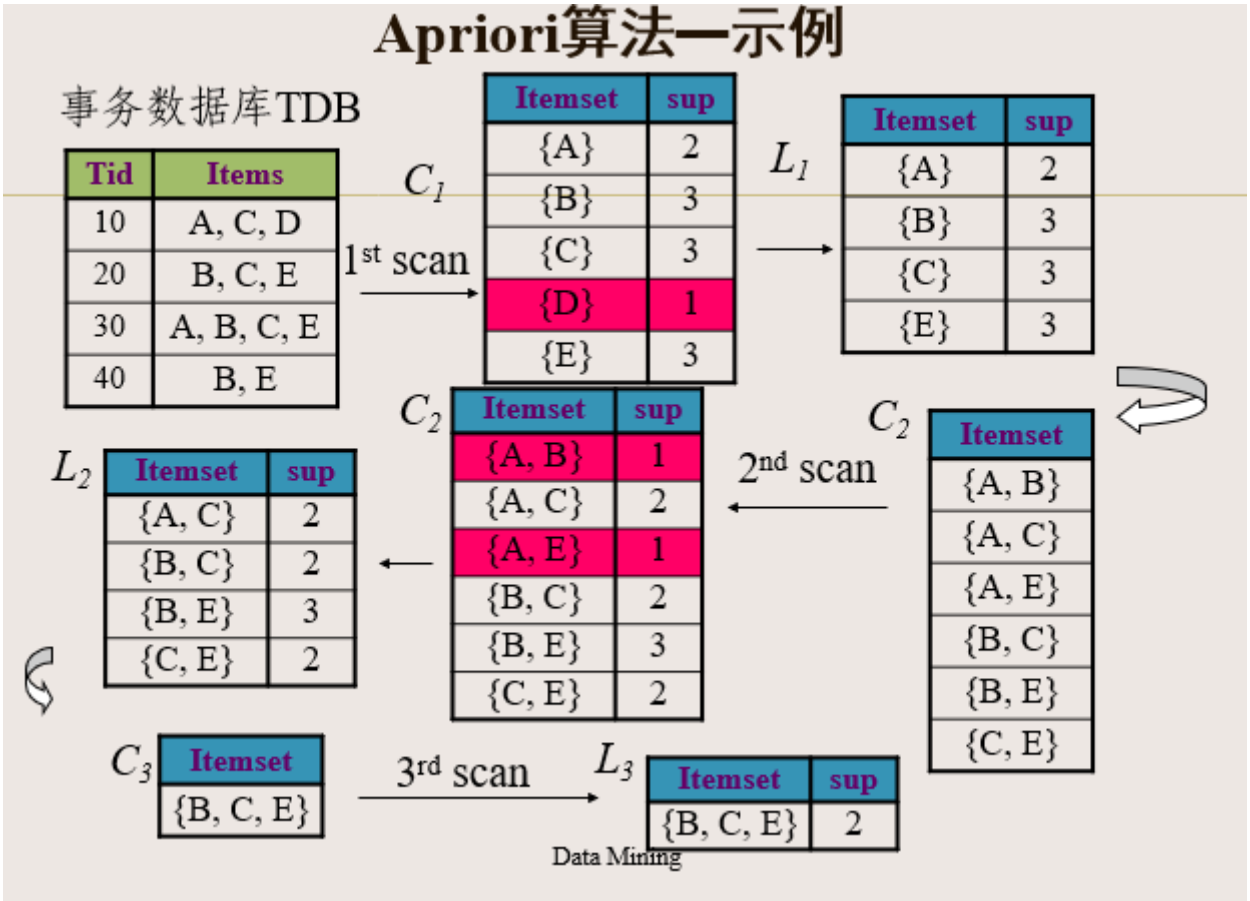


关于连接操作：

连接操作

- 对于两个k-项集$\{I_1, I_2, \ldots, I_{k-1}, I_k\}$，$\{I_1', I_2', \ldots, I_{k-1}', I_k'\}$，能够连接产生一个$k+1$－项集，当且仅当$I_i=I_i'$，其中$i \in \{1, 2, \ldots, k-1\}$且$I_k \neq I_k'$，连接结果为($k+1$项集)：
  － $\{I_1, I_2, \ldots, I_{k-1}, I_k, I_k'\}$
- 如{A, C, D}与{A, C, E}连接产生{A, C, D, E}；而{A, C, D}与{A, B, D}不能连接。

一个例子：



Apriori算法存在问题：

1. 多次扫描数据库
2. 产生大量的候选集合

FP-Tree算法

https://blog.csdn.net/kisslotus/article/details/80328045

FP-tree 算法的优点

1. FP-tree 算法只需对事务数据库进行二次扫描；
2. 避免产生大量候选集；

FP-tree 算法的缺点

1. 要递归生成条件数据库和条件 FP-tree，所以内存开销大；
2. 只能用于挖掘单维的布尔关联规则；

多维关联规则挖掘

多维关联规则：规则中有两个以上的谓词。
例如：
Age(X, “30到40”)∧Income(X, “4万 - 6万”) → Buys(X, “computer”)