第三次作业答案

Answer1

置信度(confidence)：判断关联规则是否为强关联规则的指标，关联规则的置信度越高则表明该关联规则越值得信任，一般通过条件概率计算关联规则$X \rightarrow Y$的置信度：

$$\mathrm{Conf}(I_1 \rightarrow I_2) = \mathrm{P}(I_1|I_2) = \frac{\mathrm{Sup}(I_1 \cup I_2)}{\mathrm{Sup}(X)}$$

其中I_1和I_2是频繁项集，$\mathrm{Sup}(I_1 \cup I_2)$是包含$I_1$和$I_2$的事务数，而$\mathrm{Sup}(I_1)$是包含项集$I_1$的事务数，$\mathrm{Conf}(I_1 \rightarrow I_2)$是同时包含$I_1$和$I_2$的事务数量占包含$I_1$的事务数量的百分比，该式在已知频繁项集时可直接用于发现强关联规则。

Answer2

Apriori 算法：首先从数据库删除支持度少于minsup的项，得到频繁1-项集的集合L_1；接着将L_1与自身进行连接产生长度为 2 的频繁项集的候选集C_2，通过对其进行剪枝操作，删除C_2中支持度少于minsup的项集得到L_2；迭代并重复该过程，直到无法再生产新的频繁项集。在每轮迭代过程中，新的频繁模式候选项集都由前一次迭代挖掘的频繁项集的集合连接自身产生。

优点：

1) 采用逐层搜索的迭代方法，算法简单明了，易于实现。
2) 适合稀疏数据集的关联规则挖掘，也就是频繁项目集的长度稍小的数据集。

缺点：

1) 对数据库的扫描次数过多。
2) 可能产生大量的候选项集。
3) 在频繁项目集长度变大的情况下,运算时间显著增加,不适用于大数据集。

FP-Growth 算法：首先扫描数据库得到所有频繁项并按支持度降序排列；之后以空节点作为根节点建立 FP 树：对数据库中每个事务创建分支，其中沿共同前缀的每个结点支持度计数加一，为前缀之后的项创建结点和链接；最后以项头表的结构自底依次向上地遍历 FP 树，根据频繁项寻找对应的条件模式基和条件 FP 树，递归挖掘频繁项集，直到 FP 树中没有元素为止。

优点：

1) 与 Apriori 算法相比，FP-Growth 算法无需重复扫描数据库，搜索空间的大小得到了显著压缩，因此该算法对于长频繁模式的挖掘具有较高的效率。
2) 不需要生成候选集。

缺点：

1) 内存开销大。
2) 只能用于挖掘单维的布尔关联规则。

Answer3

已知最小支持度为 2。扫描数据库得到所有的频繁项集。

候选 1-项集 C1

项集	支持度计数
{A}	4
{B}	4
{C}	4
{D}	5
{E}	4

频繁 1-项集 L1

项集	支持度计数
{A}	4
{B}	4
{C}	4
{D}	5
{E}	4

候选 2-项集 C2

项集	支持度计数	项集	支持度计数
{A,B}	3	{B,D}	4
{A,C}	2	{B,E}	2
{A,D}	3	{C,D}	3
{A,E}	3	{C,E}	3
{B,C}	2	{D,E}	3

频繁 2-项集 L2

项集	支持度计数	项集	支持度计数
{A,B}	3	{B,D}	4
{A,C}	2	{B,E}	2
{A,D}	3	{C,D}	3
{A,E}	3	{C,E}	3
{B,C}	2	{D,E}	3

候选 3-项集 C3

项集	支持度计数	项集	支持度计数
{A,B,C}	1	{A,D,E}	2
{A,B,D}	3	{B,C,D}	2
{A,B,E}	2	{B,C,E}	1
{A,C,D}	1	{B,D,E}	2
{A,C,E}	2	{C,D,E}	2

频繁 3-项集 L3

项集	支持度计数	项集	支持度计数
{A,B,D}	3	{B,D,E}	2
{A,B,E}	2	{C,D,E}	2

{A,C,E}	2		
{A,D,E}	2		
{B,C,D}	2		

候选 4-项集 C4

项集	支持度计数
{A,B,D,E}	2

频繁 4-项集 L4

项集	支持度计数
{A,B,D,E}	2

根据所有频繁项集产生所有的强关联规则（最小置信度为 50%）：

L2 产生的所有强关联规则

关联规则	置信度	关联规则	置信度
{A}→{B}	75%	{B}→{D}	100%
{B}→{A}	75%	{D}→{B}	80%
{A}→{C}	50%	{B}→{E}	50%
{C}→{A}	50%	{E}→{B}	50%
{A}→{D}	75%	{C}→{D}	75%
{D}→{A}	60%	{D}→{C}	60%
{A}→{E}	75%	{C}→{E}	75%
{E}→{A}	75%	{E}→{C}	75%
{B}→{C}	50%	{D}→{E}	60%
{C}→{B}	50%	{E}→{D}	75%

L3 产生的所有强关联规则

关联规则	置信度	关联规则	置信度	关联规则	置信度	关联规则	置信度
{A}→{B,D}	75%	{A}→{C,E}	50%	{B}→{C,D}	50%	{C}→{D,E}	50%
{B,D}→{A}	75%	{C,E}→{A}	66.7%	{C,D}→{B}	66.7%	{D,E}→{C}	66.7%
{B}→{A,D}	75%	{C}→{A,E}	50%	{C}→{B,D}	50%	~~{D}→{C,E}~~	~~40%~~
{A,D}→{B}	100%	{A,E}→{C}	66.7%	{B,D}→{C}	50%	{C,E}→{D}	66.7%
{D}→{A,B}	60%	{E}→{A,C}	50%	~~{D}→{B,C}~~	~~40%~~	{E}→{C,D}	50%
{A,B}→{D}	100%	{A,C}→{E}	100%	{B,C}→{D}	100	{C,D}→{E}	66.7%
关联规则	置信度	关联规则	置信度	关联规则	置信度		
{A}→{B,E}	50%	{A}→{D,E}	50%	{B}→{D,E}	50%		
{B,E}→{A}	100%	{D,E}→{A}	66.7%	{D,E}→{B}	66.7%		
{B}→{A,E}	50%	~~{D}→{A,E}~~	~~40%~~	~~{D}→{B,E}~~	~~40%~~		
{A,E}→{B}	66.7%	{A,E}→{D}	66.7%	{B,E}→{D}	100%		
{E}→{A,B}	50%	{E}→{A,D}	50%	{E}→{B,D}	50%		
{A,B}→{E}	66.7%	{A,D}→{E}	66.7%	{B,D}→{E}	50%		

L4 产生的所有强关联规则

关联规则	置信度	关联规则	置信度
{A}→{B,D,E}	50%	{A,B}→{D,E}	66.7%
{B,D,E}→{A}	100%	{D,E}→{A,B}	66.7%
{B}→{A,D,E}	50%	{A,D}→{B,E}	66.7%
{A,D,E}→{B}	100%	{B,E}→{A,D}	100%

~~{D}→{A,B,E}~~	~~40%~~	{A,E}→{B,D}	66.7%
{A,B,E}→{D}	100%	{B,D}→{A,E}	50%
{E}→{A,B,D}	50%		
{A,B,D}→{E}	66.7%		

表中除{D}→{A,E}，{D}→{B,C}，{D}→{B,E}，{D}→{C,E}，{D}→{A,B,E}外，均为强关联规则。

Answer4

（1）con({牙刷}→{防晒霜}) = σ({牙刷，防晒霜})/σ({牙刷}) = 1500/6000 =25%

con({牙刷，防晒霜}→{凉鞋}) = σ({牙刷，防晒霜，凉鞋})/σ({牙刷，防晒霜}) = 600/1500 =40%

（2）I({牙刷}→{防晒霜}) = sup({牙刷}→{防晒霜})/(sup({牙刷})×sup({防晒霜})) = (N×σ ({牙刷}→{防晒霜})/(σ({牙刷})×σ({防晒霜})) = (10000×1500) / (6000×5000) = 0.5 < 1

所以{牙刷}和{防晒霜}是负相关。

（3）$Lift$({牙刷}，{太阳镜}) = I({牙刷}→{太阳镜}) = sup({牙刷}→{太阳镜})/(sup({牙刷})×sup({太阳镜})) = (N×σ ({牙刷}→{太阳镜})/(σ({牙刷})×σ({太阳镜})) = (10000×250) / (6000×2000) = 0.208

Answer5

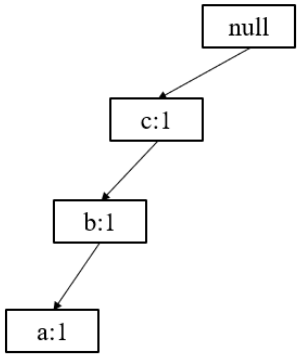
扫描数据库中对各项计数，按支持度递减降序对频繁项排序

L={{c:5},{b:4},{d:3},{e:3},{a:2}}

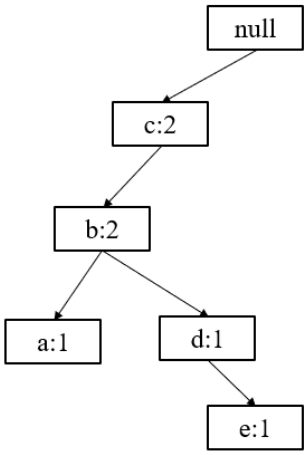
去除频繁项并重新排序

TID	Itemset
1	c，b，a
2	c，b，d，e
3	c，e，a
4	c，b，d
5	c，b，d，e

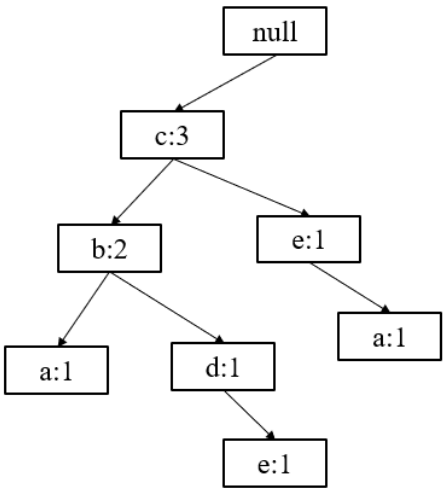
扫描第一个事务：



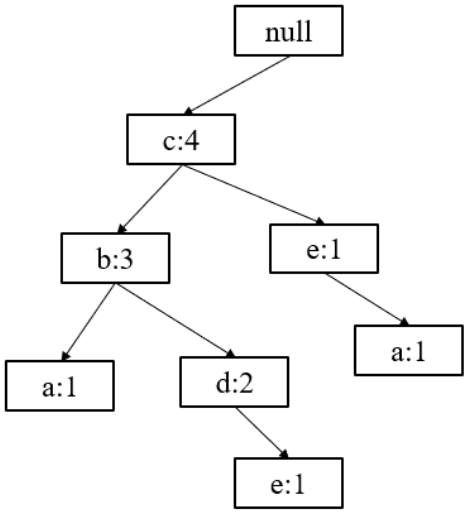
扫描第二个事务：



扫描第三个事务：



扫描第四个事务：



扫描第五个事务: