【例2.5】

将表2.3作为数据集，使用Apriori算法进行关联分析，产生描述网络购买行为的关联规则。

表2.3 网络购物交易记录表

表2.3 网络购物交易记录表

序号	Book	Sneaker	Earphone	DVD	Juice
1	1	1	1	1	1
2	1	1	1	1	0
3	0	1	1	0	0
4	0	1	0	1	1
5	0	0	1	1	0
6	1	0	1	1	0
7	1	0	1	1	1
8	0	1	0	1	1
9	0	0	1	1	1
10	1	0	0	0	1

步骤

（1）设置支持度阈值为50%，创建第一个条目集表，包含单项条目。

表2.4 网络购物行为关联分析条目表1

条目集	条目个数	符合支持度要求	结果
Book = 1			
Sneaker = 1			
Earphone = 1			
DVD = 1			
Juice = 1			
Book = 0			
Sneaker = 0			
Earphone = 0			
DVD = 0			
Juice = 0			

表2.3 网络购物交易记录表

表2.3 网络购物交易记录表

序号	Book	Sneaker	Earphone	DVD	Juice
1	1	1	1	1	1
2	1	1	1	1	0
3	0	1	1	0	0
4	0	1	0	1	1
5	0	0	1	1	0
6	1	0	1	1	0
7	1	0	1	1	1
8	0	1	0	1	1
9	0	0	1	1	1
10	1	0	0	0	1

步骤

（2）设置支持度阈值为40%，创建第二个条目集表，包含双项条目

表2.5 网络购物行为关联分析条目表2

条目集	条目个数	符合支持度要求	结果

步骤

（3）仍将支持度阈值设置为40%，使用双项条目表中的“属性-值”组合生成三项条目集，有两条条目。

- Book =1 & Earphone = 1& DVD = 1
- Sneaker =0 & Earphone = 1 & DVD = 1

（4）再次将支持度阈值设置为40%，以三项条目集为基础，生成四项条目集，没有符合支持度要求的条目，条目集生成工作结束。

表2.3 网络购物交易记录表

表2.3 网络购物交易记录表

序号	Book	Sneaker	Earphone	DVD	Juice
1	1	1	1	1	1
2	1	1	1	1	0
3	0	1	1	0	0
4	0	1	0	1	1
5	0	0	1	1	0
6	1	0	1	1	0
7	1	0	1	1	1
8	0	1	0	1	1
9	0	0	1	1	1
10	1	0	0	0	1

步骤

（5）以生成的条目集为基础创建关联规则。

- 首先设置置信度阈值为80%；
- 然后从双项和三项条目集表中生成关联规则；
- 最后，所有不满足置信度阈值的规则将被删除。
- **以双项条目集中的第一条条目生成的两条规则——**
  - IF Book =1 THEN Earphone = 1 （ ）
  - IF Earphone = 1 THEN Book =1（ ）
- **以三项条目集中的第一条条目生成的三条规则——**
  - IF Book =1 & Earphone = 1 THEN DVD = 1（ ）
  - IF Book =1 & DVD = 1 THEN Earphone = 1（ ）
  - IF Earphone = 1 & DVD = 1 THEN Book =1（ ）

【例2.6】

将表2.4作为数据集，使用Apriori算法进行关联分析，产生描述网络购买行为的关联规则。

步骤

设最小支持度计数为2。

表2.4 网络购物行为关联分析条目表

事务	所购商品ID列表	事务	所购商品ID列表
T1	I1，I2，I5	T6	I2，I3
T2	I2，I4	T7	I1，I3
T3	I2，I3	T8	I1，I2，I3，I5
T4	I1，I2，I4	T9	I1，I2，I3
T5	I1，I3		

步骤

（1）第一次扫描，对每一候选1项集计数，确定频繁1-项集的集合L1 。

事务	所购商品ID列表	事务	所购商品ID列表
T1	I1，I2，I5	T6	I2，I3
T2	I2，I4	T7	I1，I3
T3	I2，I3	T8	I1，I2，I3，I5
T4	I1，I2，I4	T9	I1，I2，I3
T5	I1，I3		

C1

项集	支持度计数
{I1}	
{I2}	
{I3}	
{I4}	
{I5}	

选取大于最小支持度计数的项目集候选1项计数 →

L1

项集	支持度计数
{I1}	
{I2}	
{I3}	
{I4}	
{I5}	

步骤

L1中各项两两任意组合

（2）第二次扫描，使用L1∞L1产生候选2-项集的集合C2，基于候选集C2计算支持度，以确定频繁2-项集的集合L2。

L1

项集	支持度计数
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

由L1产生C2 →

C2

项集

在剪枝步没有候选从C2中删除，因为这些候选的每个子集都是频繁的。

步骤

（2）第二次扫描，使用L1∞L1产生候选2-项集的集合C2，基于候选集C2计算支持度，以确定频繁2-项集的集合L2。

由L1产生C2 →

C2

项集
{I1，I2}
{I1，I3}
{I1，I4}
{I1，I5}
{I2，I3}
{I2，I4}
{I2，I5}
{I3，I4}
{I3，I5}
{I4，I5}

扫描D，对每一个候选2-项集计数 →

C2

项集	支持度计数
{I1，I2}	
{I1，I3}	
{I1，I4}	
{I1，I5}	
{I2，I3}	
{I2，I4}	
{I2，I5}	
{I3，I4}	
{I3，I5}	
{I4，I5}	

选取大于最小支持度的项目集 →

L2

项集	支持度计数
{I1，I2}	
{I1，I3}	
{I1，I5}	
{I2，I3}	
{I2，I4}	
{I2，I5}	

注意：{I1，I4}、{I3，I4}、{I3，I5}、{I4，I5}的支持度低于阈值，故而舍弃，在随后的迭代中，如果出现它们的组合形式将不被考虑。

步骤

（3）第三次扫描，使用L2∞L2产生候选3-项集的集合C3，基于候选集C3计算支持度，以确定频繁3-项集的集合L3 。

发现规律了吗?

在频繁项集为K的元素上找频繁项集为K+1的元素的方法是：在频繁项集为K的项目（每行记录）中，假如共有N行，两两组合，满足两两中前K-1个元素相同，只后一个元素要求前一条记录的商品名称小于后一条记录的商品名称，这样是为了避免重复组合，求它们的并集得到长度为K+1的准频繁项集。

步骤

使用L2∞L2产生候选3-项集的集合C3。

L2

项集	支持度计数
{I1，I2}	4
{I1，I3}	4
{I1，I5}	2
{I2，I3}	4
{I2，I4}	2
{I2，I5}	2

连接C3=
L2∞L2 →

C3

项集
{I1，I2，I3}
{I1，I2，I5}
{I1，I3，I5}
{I2，I3，I4}
{I2，I3，I5}
{I2，I4，I5}

步骤

使用Apriori剪枝：频繁项集的所有非空子集也必须是频繁的。

例如 {I1,I3,I5}的2-项集分别是：{I1,I3}，{I1,I5}，{I3,I5}。

{I3,I5}是非频繁的，因此从C3中删除{I1,I3,I5}。

剪枝后的C3为：

C3

项集
{I1，I2，I3}
{I1，I2，I5}
{I1，I3，I5}
{I2，I3，I4}
{I2，I3，I5}
{I2，I4，I5}

剪枝C3 →

准L3

项集
{I1,I2,I3}
{I1,I2,I5}

步骤

针对准频繁项集计算支持度，得到频繁项集是：

准L3

项集
{I1,I2,I3}
{I1,I2,I5}

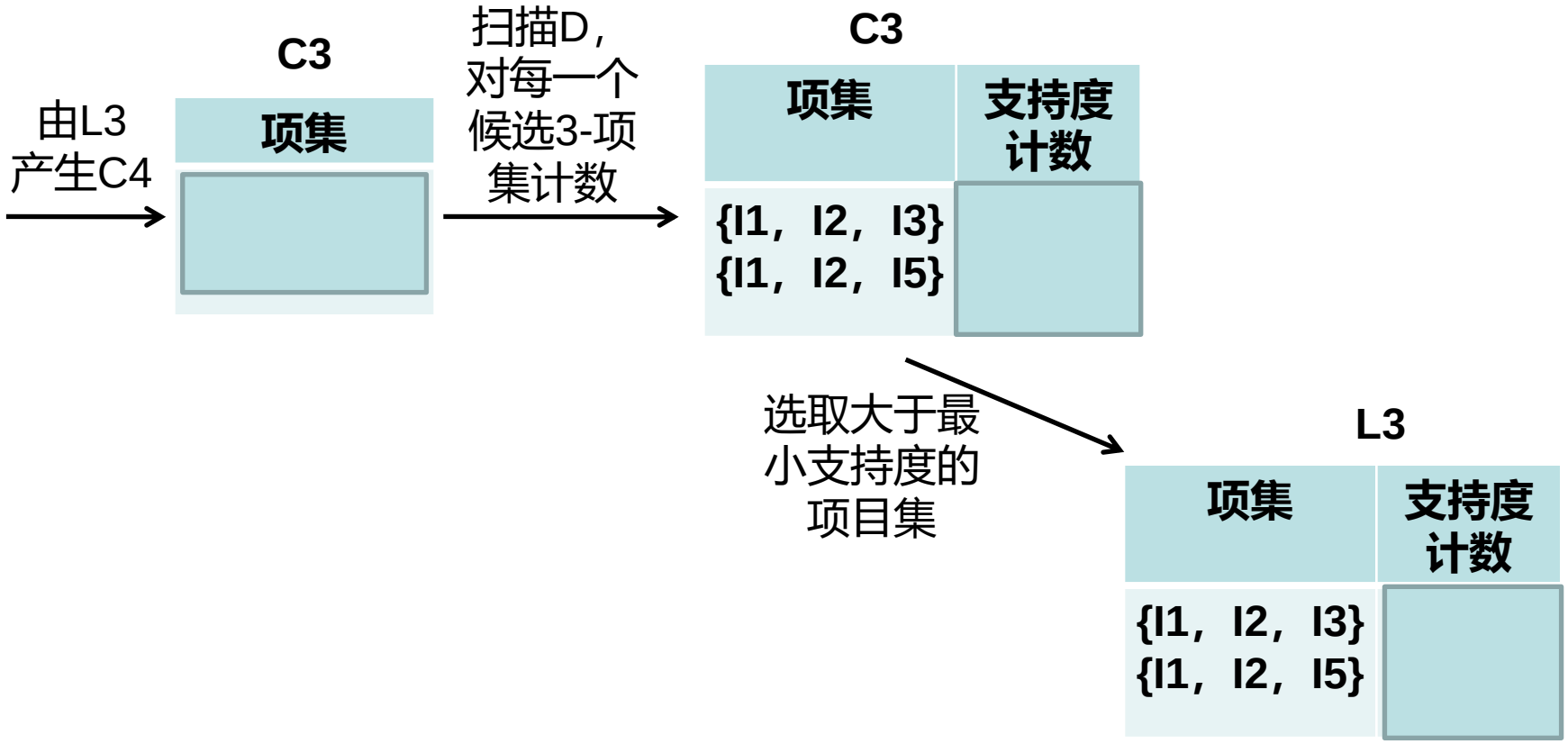
计算支持度，选取大于最小支持度的项目集 →

L3

项集	支持度计数
{I1,I2,I3}	2
{I1,I2,I5}	2

步骤

（3）第三次扫描，使用L2∞L2产生候选3-项集的集合C3，基于候选集C3计算支持度，以确定频繁3-项集的集合L3 。



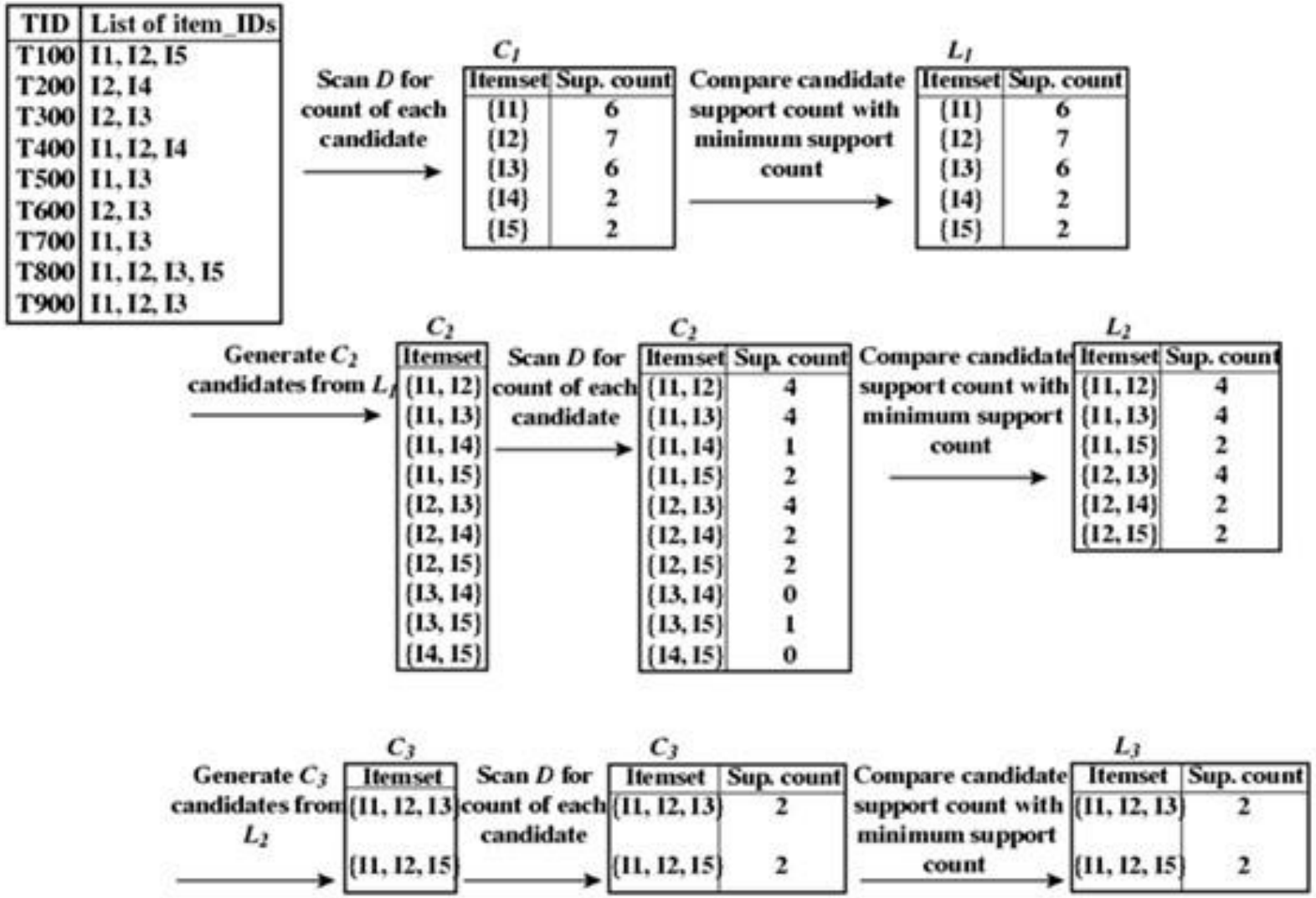
步骤

（4）第四次扫描，使用L3∞L3产生候选4-项集的集合C4，基于候选集C4计算支持度，以确定频繁4-项集的集合L4 。

L3∞L3产生候选4-项集的集合C4={{L1，L2，L3，L5}}，保留吗?

因为它的子集{L2，L3，L5}不是频繁的，所以这个项集被删除。因此C4为空集，算法终止。

步骤


TID | List of item_IDs
T100 | I1, I2, I5
T200 | I2, I4
T300 | I2, I3
T400 | I1, I2, I4
T500 | I1, I3
T600 | I2, I3
T700 | I1, I3
T800 | I1, I2, I3, I5
T900 | I1, I2, I3
Scan D for count of each candidate
C1
Itemset | Sup. count
{I1} | 6
{I2} | 7
{I3} | 6
{I4} | 2
{I5} | 2
Compare candidate support count with minimum support count
L1
Itemset | Sup. count
{I1} | 6
{I2} | 7
{I3} | 6
{I4} | 2
{I5} | 2
Generate C2 candidates from L1
C2
Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}
Scan D for count of each candidate
C2
Itemset | Sup. count
{I1, I2} | 4
{I1, I3} | 4
{I1, I4} | 1
{I1, I5} | 2
{I2, I3} | 4
{I2, I4} | 2
{I2, I5} | 2
{I3, I4} | 0
{I3, I5} | 1
{I4, I5} | 0
Compare candidate support count with minimum support count
L2
Itemset | Sup. count
{I1, I2} | 4
{I1, I3} | 4
{I1, I5} | 2
{I2, I3} | 4
{I2, I4} | 2
{I2, I5} | 2
Generate C3 candidates from L2
C3
Itemset
{I1, I2, I3}
{I1, I2, I5}
Scan D for count of each candidate
C3
Itemset | Sup. count
{I1, I2, I3} | 2
{I1, I2, I5} | 2
Compare candidate support count with minimum support count
L3
Itemset | Sup. count
{I1, I2, I3} | 2
{I1, I2, I5} | 2

Apriori分析购物行为的关联规则-1

我们看到，数据库存储的数据格式，会员100购买了 1 3 4三种商品，那么对应的集合形式如右边的图所示。那么基于候选集C1，我们得到频繁项集L1，如下图所示，在此表格中{4}的支持度为1，而我们设定的支持度为2。支持度大于或者等于指定的支持度的最小阈值就成为L1了，这里{4}没有成为L1的一员。因此，我们认定包含4的其他项集都不可能是频繁项集，后续就不再对其进行判断了。

Database

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$\overline{C}_1$

TID	Set-of-Itemsets
100	
200	
300	
400	

Apriori分析购物行为的关联规则-1

此时我们看到L1是符合最低支持度的标准的，那么下一次迭代我们依据L1产生C2（4就不再被考虑了），此时的候选集如右图所示C2（依据L1*L1的组合方式）确立。C2的每个集合得到的支持度对应在我们原始数据组合的计数，如下图左所示。

L_1

Itemset	Support

C_2

Itemset	Support

Apriori分析购物行为的关联规则-1

此时，第二次迭代发现了{1 2} {1 5}的支持度只有1，低于阈值，故而舍弃，那么在随后的迭代中，如果出现{1 2} {1 5}的组合形式将不被考虑。

$\overline{C}_2$

TID	Set-of-Itemsets
100	{ {1 3} }
200	{ {2 3}, {2 5}, {3 5} }
300	{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }
400	{ {2 5} }

L_2

Itemset	Support

Apriori分析购物行为的关联规则-1

如图，由L2得到候选集C3，那么这次迭代中的{1 2 3} { 1 3 5}哪去了?

如刚才所言，{1 2} {1 5}的组合形式将不被考虑，因为这两个项集不可能成为频繁项集L3，此时L4不能构成候选集L4，即停止。

C_3

Itemset	Support
{2 3 5}	2

$\overline{C}_3$

TID	Set-of-Itemsets
200	{ {2 3 5} }
300	{ {2 3 5} }

L_3

Itemset	Support
{2 3 5}	2

Apriori分析购物行为的关联规则-1

如果用一句化解释上述的过程，就是不断通过Lk的自身连接，形成候选集，然后在进行剪枝，除掉无用的部分。

Apriori的关联规则是在频繁项集基础上产生的，进而这可以保证这些规则的支持度达到指定的水平，具有普遍性和令人信服的水平。

Apriori分析购物行为的关联规则-2

Transaction ID	Items Purchased
1	(orange juice, soda)
2	(milk, orange juice, window cleaner)
3	(orange juice, detergent, soda)
4	(window cleaner, soda)
5	(soda, potato chips)

Assume minimum support s=30% and minimum confidence c=60%. orange juice → soda is discovered because

1) orange juice and soda occur together in 2 transactions (i.e., support = 40%) and,

2) in all transactions containing orange juice (i.e., 3 transactions), 66.67% also contain soda (i.e., confidence = 66.67%).

Apriori分析购物行为的关联规则-2

Step 1: Generate large 1-itemsets

All 1-itemsets (C_1)

1-itemset	Support
Orange juice	60%
Soda	80%
Milk	20%
Window Cleaner	40%
Detergent	20%
Potato chips	20%

Large 1-itemsets (L_1)

1-itemset	Support
Orange juice	60%
Soda	80%
Window Cleaner	40%

Step 2: Combine candidate 2-itemsets

Candidate 2-itemsets (C_2)

2-itemset	Support
{Orange juice, Soda}	
{Orange juice, Window Cleaner}	
{Soda, Window Cleaner}	

Apriori分析购物行为的关联规则-2

Step 3: Generate large 3-itemsets

Candidate 2-itemsets (C_2)

2-itemset	Support
{Orange juice, Soda}	40%
{Orange juice, Window Cleaner}	20%
{Soda, Window Cleaner}	20%

Large 2-itemsets (L_2)

2-itemset	Support
{Orange juice, Soda}	40%

Step 4: Construct association rules from large 2- itemsets

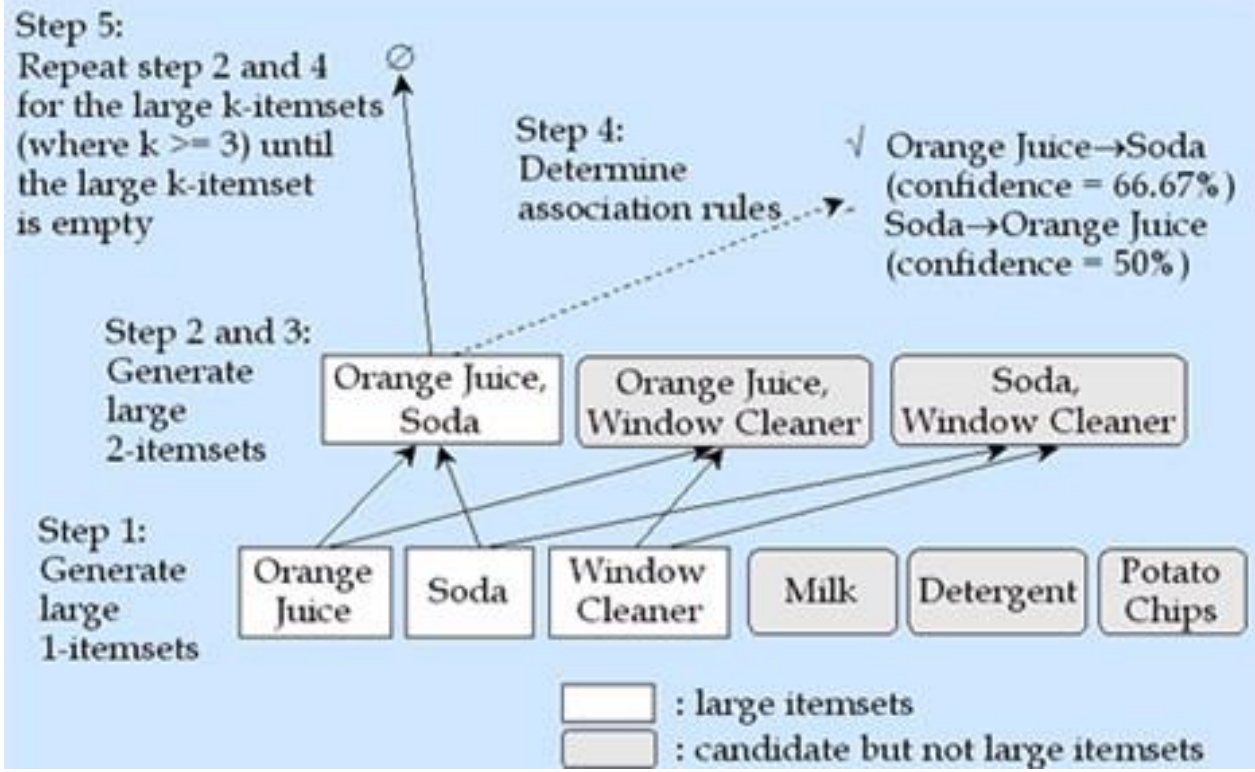
Orange juice→Soda (confidence =66.67%)
Soda→Orange juice (confidence =50%)

An association rule is generated:
Orange juice→Soda (confidence =66.67%)

Apriori分析购物行为的关联规则-2

Step 5: Repeat step 2 for combining candidate 3- itemsets

The large 2-itemset contains only {Orange juice, Soda}, resulting in an empty candidate 3-itemset. Thus, the association rule analysis terminates at this point.



Apriori分析购物行为的关联规则-3