第3章

# 关联规则模型与应用

【内容提要】 关联规则模型算法作为快速知识发现的重要算法之一，最早流行于大型数据库中快速发现知识的应用中，后来在多个领域得到应用和拓宽研究。本章重点描述的内容是：①关联规则形成的相关理论，如候选项集、项集、强项集、支持度、可信度、规则产生式；②关联分析算法的逻辑步骤；③关联规则的应用。

【学习要求】 了解 Apriori(即快速发现知识)算法的核心内容，能够通过一个实际小样本应用例子的手算过程，来掌握产生规则的全部模拟过程。同时，通过对关联分析算法的应用，密切关注关联规则的创新应用。

## 3.1 关联规则的解释、理论与相关术语

关联规则(Association Rules)的含义是指在大型的数据库系统中，迅速找出各事物之间潜在的、有价值的关联，用规则表示出来，经过推理、积累形成知识后，得出重要的相关联的结论，从而为当前市场经济提供准确的决策手段。

关联规则的应用已经比较广泛，早在 20 世纪 90 年代率先应用于条形码中，使大型零售商品的合理组织的复杂问题成为现实，也在决策领域到通信报警等系统得到应用；到了 21 世纪初，关联分析普遍用于医疗中的病因分析和诊断；在交通运输领域中用于安全运输的相关因素分析、道路交通安全的路段事故成因和风险分析；2010 年至今，专家们还将研究和应用的重点集中在大数据的分析的应用研究中，如基于并行计算环境的词频统计、基于 Hadoop 环境下的并行关联分析算法的应用、交通肇事逃逸案的关联分析、传染病或疑难病因的关联分析等相关分析及应用研究，目前关联规则的应用研究已经渗透到多个研究领域。

### 3.1.1 关联规则的解释

关联规则的研究和应用是数据挖掘中最活跃和比较深入的分支，目前，已经提出了许多关联规则挖掘的理论和算法。最为著名的是 R.Agrawal 等学者提出的 Apriori 及其改进算法。

在 Apriori 算法中提到：为了发现有意义的关联规则，需要给定两个阈值：最小支持度(Minimum Support，min-sup)和最小可信度(Minimum Confidence，min-conf)。min-sup 和 min-conf 的解释是：①挖掘出的关联规则必须满足用户

规定的最小支持度，它表示了一组项目关联在一起需要满足的最低联系程度；②挖掘出的关联规则也必须满足用户规定的最小可信度，它反映了一个关联规则的最低可靠度。因此，求关联规则的最终目的在于：可以快速从大型数据库中挖掘出同时满足最小支持度和最小可信度的关联规则即知识。另外，规则作为知识的最直接的、最简单的表达形式，这也是多数科学研究者选用关联规则挖掘知识的原因。

### 3.1.2　关联规则的理论及相关术语

关联规则的基础理论与方法概括如下。

(1) 项目集格空间理论。

(2) 模糊理论。

(3) 快速查询。

(4) 散列存储。

(5) 层次树的数据结构。

(6) 规则的表示与形成、知识的产生与描述方法。

#### 1. 项集或候选项集

项集是 Item Set 的统称，一般简称为 Items；候选项集是 Candidate Item Set 的统称，一般简称为 C。

项集 Items＝{Item1，Item2，…，Item$m$}；TR 是事物的集合，TR⊂Items，并且 TR 是一个{0，1}属性的集合。

例如，表 3.1 给出了一个事务数据库例子。

其中，表 3.1 中的 item1，item2，…，item5 分别代表事务 1，事务 2，……，事务 5。也就是说，表 3.1 中的“行”代表数据库中的一条记录(在数据结构中也叫作数据元素，它是数据的基本单位)；表 3.1 中的“列”代表数据库的属性(数据库即二维表中的列)或者叫作字段(在数据结构中也叫作数据项，它是数据的最小单位，即不能再分割的数据)。

表 3.1　事务数据库(DB)例子

| Tid | Item1 | Item2 | Item3 | Item4 | Item5 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 1 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 1 | 0 | 1 | 0 | 0 |
| 8 | 1 | 1 | 1 | 0 | 1 |
| 9 | 1 | 1 | 1 | 0 | 0 |

方便起见，约定将表 3.1 中的事务 1(项集 1 即 item1)简称为 $I_1$，将表 3.1 中的事务 2

(项集 2 即 item2)简称为 $I_2$,……,将表 3.1 中的事务 5(项集 5 即 item5)简称为 $I_5$。如果将表 3.1 增加 1 列,其单元格内容存放每行包含的项集总数,例如,第一行中含有三个 1,即第 1 行的第 1 列即 $I_1$,第 1 行的第 2 列即 $I_2$,第 1 行的第 5 列即 $I_5$,因此,该行的项集总数是 $\{I_1,I_2,I_5\}$,共三个项集。其他行按照该计算方法得出结果如表 3.2 所示。

表 3.2 事务数据库例子对应的项集总数的例子

| Tid | 事务 1 $I_1$ Item1 | 事务 2 $I_2$ Item2 | 事务 3 $I_3$ Item3 | 事务 4 $I_4$ Item4 | 事务 5 $I_5$ Item5 | (每行包含的项集总数) Items |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | $\{I_1,I_2,I_5\}$ |
| 2 | 0 | 1 | 0 | 1 | 0 | $\{I_2,I_4\}$ |
| 3 | 0 | 1 | 1 | 0 | 0 | $\{I_2,I_3\}$ |
| 4 | 1 | 1 | 0 | 1 | 0 | $\{I_1,I_2,I_4\}$ |
| 5 | 1 | 0 | 1 | 0 | 0 | $\{I_1,I_3\}$ |
| 6 | 0 | 1 | 1 | 0 | 0 | $\{I_2,I_3\}$ |
| 7 | 1 | 0 | 1 | 0 | 0 | $\{I_1,I_3\}$ |
| 8 | 1 | 1 | 1 | 0 | 1 | $\{I_1,I_2,I_3,I_5\}$ |
| 9 | 1 | 1 | 1 | 0 | 0 | $\{I_1,I_2,I_3\}$ |

集合 k_Item=$\{$Item1, Item2,…, Item$k$$\}$称为 $k$ 项集,或者 $k$ 项候选项集。

例如,表 3.2 的第 5 行及行编号 Tid-5,对应的项集总数为 Items=$\{I_1,I_3\}$,即 $k=2$,属于长度为 2 的项集,可以表示为 2_Item=$\{$Item$i$, Item$j$ $\}$,其中,$k\geqslant i$,$j\geqslant 1$。

假设 DB 包含 $m$ 个属性($A$, $B$,…, $M$);1 项集即长度为 1 的项集 1_Item=$\{\{A\},\{B\},\cdots,\{M\}\}$,共有 $m$ 个候选项集;2 项集即长度为 2 的项集 2_Item=$\{\{A,B\},\{A,C\},\cdots,\{A,M\},\{B,C\},\cdots,\{B,M\},\{C,D\},\cdots,\{L,M\}\}$,共有$[m\times(m-1)/2]$个项集;以此类推,$m$ 项集 m_Item=$\{A,B,C,\cdots,M\}$,只有 1 个项集。

### 2. 支持度

支持度 sup 指的是某条规则的前件或后件对应的支持数与记录总数的百分比。

假设 $A$ 的支持度是 sup($A$),sup($A$)=$|\{TR\,|TR\supseteq A\}|/|n\,|$;$A\Rightarrow B$ 的支持度 sup($A\Rightarrow B$)=sup($A\cup B$)= $|\{TR\,|\,TR\supseteq A\cup B\}|/|n\,|$,其中,$n$ 是 DB 中的总的记录数目。

### 3. 可信度

规则 $A\Rightarrow B$ 具有可信度 conf($A\Rightarrow B$)表示 DB 中包含 $A$ 的事物同时也包含 $B$ 的百分比,是 $A\cup B$ 的支持度 sup($A\cup B$)与前件 $A$ 的支持度 sup($A$)的百分比。

$$\mathrm{conf}(A\Rightarrow B)=\mathrm{sup}(A\cup B)/\mathrm{sup}(A)$$

### 4. 强项集和非频繁项集

如果某 $k$ 项候选项集的支持度大于或等于所设定的最小支持度阈值,则称该 $k$ 项候选项集为 $k$ 项强项集(**Large k-itemset**)或者 $k$ 项频繁项集(Frequent k-itemset)。

同时,对于支持度小于最小支持度的 $k$ 项候选项集称为 $k$ 项非频繁项集。

**定理(频繁项集的反单调性)**：设 $A$，$B$ 是数据集 DB 中的项集，若 $A$ 包含 $B$，则 $A$ 的支持度大于 $B$ 的支持度；若 $A$ 包含于 $B$，且 $A$ 是非频繁项集，则 $B$ 也是非频繁项集；若 $A$ 包含于 $B$，且 $B$ 是频繁项集，则 $A$ 也是频繁项集。

**5. 产生关联规则**

若 $A$，$B$ 为项集，$A \subset \text{Item}$，$B \subset \text{Item}$ 并且 $A \cap B = \varnothing$，一个关联规则是形如 $A \Rightarrow B$ 的蕴含式。

(1) 当前关联规则算法普遍基于 Support-Confidence 模型。

支持度是项集中包含 $A$ 和 $B$ 的记录数与所有记录数之比，描述了 $A$ 和 $B$ 这两个物品集的并集 $C$ 在所有的事务中出现的概率，能够说明规则的有用性。

(2) 规则 $A \Rightarrow B$ 在项集中的可信度，是指在出现了物品集 $A$ 的事务 $T$ 中，物品集 $B$ 也同时出现的概率，能够说明规则的确定性。产生关联规则，即是从强项集中产生关联规则。

在最小可信度的条件门槛下，若强项集的可信度满足最小可信度，称此 $k$ 项强项集为关联规则。

例如，若$\{A, B\}$为 2 项强项集，同时 conf($A \Rightarrow B$)大于或等于最小可信度，即 sup($A \cup B$)$\geqslant$min_sup 且 conf($A \Rightarrow B$)$\geqslant$min_conf，则称 $A \Rightarrow B$ 为关联规则。

### 3.1.3　Apriori 算法介绍

R.Agrawal 等学者在 1993 年设计了一个 Apriori 算法，它是一种最有影响力的挖掘布尔关联规则频繁项集的算法。其核心是基于两阶段的频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。该算法将关联规则挖掘分解为两个子问题：①找出存在于事务数据库中所有的频繁项目集，即那些支持度大于用户给定支持度阈值的项目集；②在找出的频繁项目集的基础上产生强关联规则，即产生那些支持度和可信度分别大于或等于用户给定的支持度和可信度阈值的关联规则。

在上述两步中，第二步相对容易些，因为它只需要在已经找出的频繁项目集的基础上列出所有可能的关联规则，同时，满足支持度和可信度阈值要求的规则被认为是有趣的关联规则。但由于所有的关联规则都是在频繁项目集的基础上产生的，已经满足了支持度阈值的要求，只需要考虑可信度阈值的要求，只有那些大于用户给定的最小可信度的规则才被留下来。第一个步骤是挖掘关联规则的关键步骤，挖掘关联规则的总体性能由第一个步骤决定，因此，所有挖掘关联规则的算法都是着重于研究第一个步骤。

Apriori 算法在寻找频繁项集时，利用了频繁项集的向下封闭性(反单调性)，即频繁项集的子集必须是频繁项集，采用逐层搜索的迭代方法，由候选项集生成频繁项集，最终由频繁项集得到关联规则，这些操作主要是由连接和剪枝来完成。下面是 Apriori 算法的伪代码。

---

**Begin**

　//算法的初始化(为各个参与运算的参数赋初值等描述)

　$L_1$ = {Large 1−itemsets}　　//扫描所有事务，计算每项出现次数，产生频繁项集长度为

　　//1 的项集集合即 1-项集集合 $L_1$

**for**　(**k**=2; $\mathbf{L_{k-1}} \neq \varnothing$; **k**++)　**do**　//进行迭代循环，根据前一次的 $L_{k-1}$ 得到频繁 k-项集集合 $L_k$

```
begin
  Ck′=join(Lkm,Lkn)          //join 对每两个有 k−1 个共同项目的长度为 k 的模式 Lkm 和 Lkn 进行连接
  Ck=prune(Ck′)              //prune 根据频繁项集的反单调性,对 Ck′进行剪枝,得到 Ck
  Ck= apriori−gen(Lk−1)      //产生 k 项候选项集 Ck
    for all transactions t∈D do   //扫描数据库一遍
      begin
        Ct=subset(Ck,t)      //确定每个事务 t 所含 k-候选项集的 subset(Ck,t)
      for all candidates c∈ Ct do
          c.count++          //对候选项集的计数存放在 hash 表中
      end
    Lk={c∈Ct| c.count≥ min_sup}  //删除候选项集中小于最小支持度的,得到 k-频繁项集 Lk
end
  for all subset s⊆Lk        //对于每个频繁项集 Lk,产生 Lk 的所有非空子集 s
    If conf(s⇒ Lk−s )>=min_conf  //可信度大于最小可信度的强项集为关联规则
      Then Output ( s⇒ Lk−s)     //由频繁项集产生关联规则
end
end                          //得到所有的关联规则
```

Apriori 算法最大的问题是产生大量的候选项集,可能需要频繁重复扫描大型数据库,非常浪费扫描数据库的时间和大量候选项的存储空间,因此为候选项集合理分配内存,实现对大型数据库系统快速扫描的技术和方法是提高管理规则效率的重要途径。面向大型数据库,从海量数据中高效提取关联规则是非常重要的。

## 3.2 关联规则举例

**例 3.1** Apriori 关联规则方法的实例。

通过关联规则分析受过高等教育与性别、工资收入、职业、年龄等之间的潜在关联。给出一个简单的数据库的例子,如表 3.3 所示。

表 3.3 一个简单的数据库的例子

| 记录号 | 性别 | 年龄 | 学历 | 职业 | 收入 |
|---|---|---|---|---|---|
| 100 | 男 | 46 | 博士 | 高校教师 | 7500 |
| 200 | 女 | 32 | 硕士 | 高校教师 | 6500 |
| 300 | 男 | 35 | 学士 | 技术员 | 4900 |
| 400 | 男 | 40 | 硕士 | 高校教师 | 6000 |
| 500 | 男 | 37 | 博士 | 高校教师 | 7000 |
| 600 | 男 | 25 | 学士 | 技术员 | 4000 |

### 1. 将实际的 DBS 问题转换成对应的逻辑值

对性别二元化(1:男,2:女);对年龄离散化(若年龄≥40,3:40 岁以上;若年龄<40,

4：40 岁以下)；对是否受过研究生教育学历离散化(若学历为博士或者硕士，5：高学历；若学历为本科和本科以下，6：低学历)；对职业进行二元化处理(7：高校教师，8：技术员)；对收入进行二元化处理(若每月平均收入大于 5000 元，9：收入 5000 元以上；若每月平均收入小于 5000 元，10：收入 5000 元以下)。通过以上的数据规约，表 3.4 给出了与表 3.3 相对应的逻辑表格。

表 3.4　数据库对应的逻辑库

| 记录号 | 性别 | | 年龄 | | 学历 | | 职业 | | 收入 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 100 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 200 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 300 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 400 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 500 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 600 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

用关联规则算法找出表 3.4 中各属性之间有价值的、潜在的关联的信息即规则，希望最终可以获得高等教育与工资、性别与职业、职务与工资等属性之间的关联。经过检索逻辑库(参见表 3.4)得到每条记录中各个 Item 的取值，如表 3.5 所示。

表 3.5　数据库中记录的属性项取值集合

| 记　录　号 | 项　　集 |
|---|---|
| 100 | 1，3，5，7，9 |
| 200 | 2，4，5，7，9 |
| 300 | 1，4，6，8，10 |
| 400 | 1，3，5，7，9 |
| 500 | 1，4，5，7，9 |
| 600 | 1，4，6，8，10 |

**2. 设最小支持度 min_sup＝0.5，最小置信度 min_conf＝0.7，求得关联规则**

通过数据库查询(参见表 3.5)得到 $k$ 项候选集和 $k$ 项强项集($L_k$)及关联规则。

(1) 求 1 项集和 1 项强项集，如表 3.6 所示。

表 3.6　1 项集和 1 项强项集

| Item | Sum | sup($I$) | $L_1$ |
|---|---|---|---|
| {1} | 5 | 5/6 | √ |
| {2} | 1 | 1/6 | |
| {3} | 2 | 2/6 | |

续表

| Item | Sum | sup($I$) | $L_1$ |
|---|---|---|---|
| {4} | 4 | 4/6 | √ |
| {5} | 4 | 4/6 | √ |
| {6} | 2 | 2/6 | |
| {7} | 4 | 4/6 | √ |
| {8} | 2 | 2/6 | |
| {9} | 4 | 4/6 | √ |
| {10} | 2 | 2/6 | |

所以,1 项强项集 $L_1=\{\{1\},\{4\},\{5\},\{7\},\{9\}\}$。

(2) 通过 1 项强项集得到 2 项候选集,再计算 2 项集的支持度得到 2 项强项集,如表 3.7 所示。

表 3.7 2 项集和 2 项强项集

| Items | Sum | sup($I_m \cup I_n$) | $L_2$ |
|---|---|---|---|
| {1, 4} | 3 | 3/6 | √ |
| {1, 5} | 3 | 3/6 | √ |
| {1, 7} | 3 | 3/6 | √ |
| {1, 9} | 3 | 3/6 | √ |
| {4, 5} | 2 | 2/6 | |
| {4, 7} | 2 | 2/6 | |
| {4, 9} | 2 | 2/6 | |
| {5, 7} | 4 | 4/6 | √ |
| {5, 9} | 4 | 4/6 | √ |
| {7, 9} | 4 | 4/6 | √ |

所以,2 项强项集 $L_2=\{\{1,4\},\{1,5\},\{1,7\},\{1,9\},\{5,7\},\{5,9\},\{7,9\}\}$。

(3) 通过 1 项(即长度为 $k=1$)强项集的支持度 sup($A$),来计算 2 项(即长度为 $k=2$)强项集的可信度 $\text{conf}(I_m \Rightarrow I_n) = \text{sup}(I_m \cup I_n)/\text{sup}(I_m)$,得到 2 项(即规则长度为 $k=2$)关联规则,如表 3.8 所示。

表 3.8 2 项强项集的可信度和 2 项关联规则

| Items | $I_m$(前件) | $I_n$(后件) | sup($I_m \cup I_n$) | sup($I_m$) | conf($I_m \Rightarrow I_n$) | 2 项关联规则 |
|---|---|---|---|---|---|---|
| {1, 4} | 1 | 4 | 3/6 | 5/6 | 3/5 | |
| | 4 | 1 | 3/6 | 4/6 | 3/4 | √ |

续表

| Items | $I_m$（前件） | $I_n$（后件） | sup($I_m \cup I_n$) | sup($I_m$) | conf($I_m \Rightarrow I_n$) | 2 项关联规则 |
|---|---|---|---|---|---|---|
| {1, 5} | 1 | 5 | 3/6 | 5/6 | 3/5 | |
| | 5 | 1 | 3/6 | 4/6 | 3/4 | √ |
| {1, 7} | 1 | 7 | 3/6 | 5/6 | 3/5 | |
| | 7 | 1 | 3/6 | 4/6 | 3/4 | √ |
| {1, 9} | 1 | 9 | 3/6 | 5/6 | 3/5 | |
| | 9 | 1 | 3/6 | 4/6 | 3/4 | √ |
| {5, 7} | 5 | 7 | 4/6 | 4/6 | 1 | √ |
| | 7 | 5 | 4/6 | 4/6 | 1 | √ |
| {5, 9} | 5 | 9 | 4/6 | 4/6 | 1 | √ |
| | 9 | 5 | 4/6 | 4/6 | 1 | √ |
| {7, 9} | 7 | 9 | 4/6 | 4/6 | 1 | √ |
| | 9 | 7 | 4/6 | 4/6 | 1 | √ |

产生的 2 项关联规则为 $I(4)\Rightarrow I(1)$，$I(5)\Rightarrow I(1)$，$I(7)\Rightarrow I(1)$，$I(9)\Rightarrow I(1)$，$I(5)\Rightarrow I(7)$，$I(7)\Rightarrow I(5)$，$I(5)\Rightarrow I(9)$，$I(9)\Rightarrow I(5)$，$I(7)\Rightarrow I(9)$，$I(9)\Rightarrow I(7)$。

（4）通过 2 项强项集得到 3 项候选集，再计算 3 项集的支持度得到 3 项强项集，如表 3.9 所示。

表 3.9　3 项集和 3 项强项集

| Items | Sum | sup($I_m \cup I_n \cup I_p$) | $L_3$ |
|---|---|---|---|
| {1, 4, 5} | 1 | 1/6 | |
| {1, 4, 7} | 1 | 1/6 | |
| {1, 4, 9} | 1 | 1/6 | |
| {1, 5, 7} | 3 | 3/6 | √ |
| {1, 5, 9} | 3 | 3/6 | √ |
| {1, 7, 9} | 3 | 3/6 | √ |
| {5, 7, 9} | 4 | 4/6 | √ |

所以，3 项强项集 $L_3=\{\{1,5,7\},\{1,5,9\},\{1,7,9\},\{5,7,9\}\}$。

（5）计算 3 项强项集的可信度，得到 3 项关联规则，如表 3.10 所示。

表 3.10　3 项强项集的可信度和 3 项关联规则

| Items | $I_m$（前件） | $I_n$（后件） | sup($I_m$) | conf($I_m \Rightarrow I_n$) | 3 项关联规则 |
|---|---|---|---|---|---|
| {1, 5, 7} sup($I_m \cup I_n$)=3/6 | 1 | 5, 7 | 5/6 | 3/5 | |
| | 5 | 1, 7 | 4/6 | 3/4 | √ |

续表

| Items | $I_m$(前件) | $I_n$(后件) | sup($I_m$) | conf($I_m \Rightarrow I_n$) | 3 项关联规则 |
|---|---|---|---|---|---|
| {1, 5, 7} sup($I_m \cup I_n$)=3/6 | 7 | 1, 5 | 4/6 | 3/4 | √ |
| | 1, 5 | 7 | 3/6 | 1 | √ |
| | 1, 7 | 5 | 3/6 | 1 | √ |
| | 5, 7 | 1 | 4/6 | 3/4 | √ |
| {1, 5, 9} sup($I_m \cup I_n$)=3/6 | 1 | 5, 9 | 5/6 | 3/5 | |
| | 5 | 1, 9 | 4/6 | 3/4 | √ |
| | 9 | 1, 5 | 4/6 | 3/4 | √ |
| | 1, 5 | 9 | 3/6 | 1 | √ |
| | 1, 9 | 5 | 3/6 | 1 | √ |
| | 5, 9 | 1 | 4/6 | 3/4 | √ |
| {1, 7, 9} sup($I_m \cup I_n$)=3/6 | 1 | 7, 9 | 5/6 | 3/5 | |
| | 7 | 1, 9 | 4/6 | 3/4 | √ |
| | 9 | 1, 7 | 4/6 | 3/4 | √ |
| | 1, 7 | 9 | 3/6 | 1 | √ |
| | 1, 9 | 7 | 3/6 | 1 | √ |
| | 7, 9 | 1 | 4/6 | 3/4 | √ |
| {5, 7, 9} sup($I_m \cup I_n$)=4/6 | 5 | 7, 9 | 4/6 | 1 | √ |
| | 7 | 5, 9 | 4/6 | 1 | √ |
| | 9 | 5, 7 | 4/6 | 1 | √ |
| | 5, 7 | 9 | 4/6 | 1 | √ |
| | 5, 9 | 7 | 4/6 | 1 | √ |
| | 7, 9 | 5 | 4/6 | 1 | √ |

产生的 3 项关联规则(即长度为 3 的关联规则)为 $I(5)\Rightarrow I(1,7)$, $I(7)\Rightarrow I(1,5)$, $I(1,5)\Rightarrow I(7)$, $I(1,7)\Rightarrow I(5)$, $I(5,7)\Rightarrow I(1)$, $I(5)\Rightarrow I(1,9)$, $I(9)\Rightarrow I(1,5)$, $I(1,5)\Rightarrow I(9)$, $I(1,9)\Rightarrow I(5)$, $I(5,9)\Rightarrow I(1)$, $I(7)\Rightarrow I(1,9)$, $I(9)\Rightarrow I(1,7)$, $I(1,7)\Rightarrow I(9)$, $I(1,9)\Rightarrow I(7)$, $I(7,9)\Rightarrow I(1)$, $I(5)\Rightarrow I(7,9)$, $I(7)\Rightarrow I(5,9)$, $I(9)\Rightarrow I(5,7)$, $I(5,7)\Rightarrow I(9)$, $I(5,9)\Rightarrow I(7)$, $I(7,9)\Rightarrow I(5)$。

(6) 通过 3 项强项集 $L_3=\{\{1,5,7\},\{1,5,9\},\{1,7,9\},\{5,7,9\}\}$,来计算长度为 4 的关联规则即 4 项集,4 项集只有一个{1,5,7,9},如表 3.11 所示。

表 3.11 4 项集和 4 项强项集

| Items | Sum | sup($I_m \cup I_n \cup I_p$) | $L_4$ |
|---|---|---|---|
| {1, 5, 7, 9} | 3 | 3/6 | √ |

(7) 计算 4 项强项集的可信度，得到 4 项关联规则，如表 3.12 所示。

表 3.12　计算 4 项强项集的可信度和 4 项关联规则

| Items | $I_m$（前件） | $I_n$（后件） | sup($I_m$) | conf($I_m \Rightarrow I_n$) | 4 项关联规则 |
|---|---|---|---|---|---|
| {1, 5, 7, 9}<br>sup($I_m \cup I_n$)=3/6 | 1 | 5, 7, 9 | 5/6 | 3/5 | |
| | 5 | 1, 7, 9 | 4/6 | 3/4 | √ |
| | 7 | 1, 5, 9 | 4/6 | 3/4 | √ |
| | 9 | 1, 5, 7 | 4/6 | 3/4 | √ |
| | 1, 5 | 7, 9 | 3/6 | 1 | √ |
| | 1, 7 | 5, 9 | 3/6 | 1 | √ |
| | 1, 9 | 5, 7 | 3/6 | 1 | √ |
| | 5, 7 | 1, 9 | 4/6 | 3/4 | √ |
| | 5, 9 | 1, 7 | 4/6 | 3/4 | √ |
| | 7, 9 | 1, 5 | 4/6 | 3/4 | √ |
| | 1, 5, 7 | 9 | 4/6 | 3/4 | √ |
| | 1, 5, 9 | 7 | 3/6 | 1 | √ |
| | 1, 7, 9 | 5 | 3/6 | 1 | √ |
| | 5, 7, 9 | 1 | 4/6 | 3/4 | √ |

产生的 4 项关联规则为 $I(5)\Rightarrow I(1,7,9)$，$I(7)\Rightarrow I(1,5,9)$，$I(9)\Rightarrow I(1,5,7)$，$I(1,5)\Rightarrow I(7,9)$，$I(1,7)\Rightarrow I(5,9)$，$I(1,9)\Rightarrow I(5,7)$，$I(5,7)\Rightarrow I(1,9)$，$I(5,9)\Rightarrow I(1,7)$，$I(7,9)\Rightarrow I(1,5)$，$I(1,5,7)\Rightarrow I(9)$，$I(1,5,9)\Rightarrow I(7)$，$I(1,7,9)\Rightarrow I(5)$，$I(5,7,9)\Rightarrow I(1)$。

(8) 对获得的关联规则进行解释和可视化处理。

也就是将已经规约离散化的数据返回到原始的含义，进行有含义的解释，使得使用关联规则的用户知道以上计算过程所得到的结论代表的实际含义。下面对得到的部分关联规则的含义加以说明。

(1) $I(7)\Rightarrow I(9)$表示：在最小支持度为 0.5 和最小可信度为 0.7 的水平下，一名高校教师⇒月收入大于 5000 元。

(2) $I(5)\Rightarrow I(1,7)$表示：在最小支持度为 0.5 和最小可信度为 0.7 的水平下，有博士和硕士学位的⇒性别为男士并且可以成为一名高校教师。

(3) $I(1, 5, 7)\Rightarrow I(9)$表示：在最小支持度为 0.5 和最小可信度为 0.7 的水平下，性别为男士并且有博士和硕士学位的并且是一名高校教师⇒月收入大于 5000 元。

从上述结果得出，高等教育与性别、高等教育与工资、大学教师与性别、职业与工资、高工资与教育、高等教育与年龄等的潜在关联。

若将上例的最小支持度改为 0.3，候选项集、强项集以及关联规则会否发生变化呢？我们通过以下计算加以说明。

**3. 设最小支持度 min_sup=0.3，最小置信度 min_conf=0.7，求得关联规则**

(1) 求 1 项集和 1 项强项集，如表 3.13 所示。

表 3.13　1 项集和 1 项强项集

| Item | Sum | sup($I$) | $L_1$ |
|---|---|---|---|
| {1} | 5 | 5/6 | √ |
| {2} | 1 | 1/6 | |
| {3} | 2 | 2/6 | √ |
| {4} | 4 | 4/6 | √ |
| {5} | 4 | 4/6 | √ |
| {6} | 2 | 2/6 | √ |
| {7} | 4 | 4/6 | √ |
| {8} | 2 | 2/6 | √ |
| {9} | 4 | 4/6 | √ |
| {10} | 2 | 2/6 | √ |

所以,1 项强项集 $L_1=\{\{1\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\}\}$。

(2) 通过 1 项强项集得到 2 项候选集,再计算 2 项集的支持度得到 2 项强项集,如表 3.14 所示。

表 3.14　2 项集和 2 项强项集

| Items | Sum | sup($I_m \cup I_n$) | $L_2$ | Items | Sum | sup($I_m \cup I_n$) | $L_2$ |
|---|---|---|---|---|---|---|---|
| {1, 3} | 2 | 2/6 | √ | {3, 10} | 0 | 0/6 | |
| {1, 4} | 3 | 3/6 | √ | {4, 5} | 2 | 2/6 | √ |
| {1, 5} | 3 | 3/6 | √ | {4, 6} | 2 | 2/6 | √ |
| {1, 6} | 2 | 2/6 | √ | {4, 7} | 2 | 2/6 | √ |
| {1, 7} | 3 | 3/6 | √ | {4, 8} | 2 | 2/6 | √ |
| {1, 8} | 2 | 2/6 | √ | {4, 9} | 2 | 2/6 | √ |
| {1, 9} | 3 | 3/6 | √ | {4, 10} | 2 | 2/6 | √ |
| {1, 10} | 2 | 2/6 | √ | {5, 6} | 0 | 0/6 | |
| {3, 4} | 0 | 0/6 | | {5, 7} | 4 | 4/6 | √ |
| {3, 5} | 2 | 2/6 | √ | {5, 8} | 0 | 0/6 | |
| {3, 6} | 0 | 0/6 | | {5, 9} | 4 | 4/6 | √ |
| {3, 7} | 2 | 2/6 | √ | {5, 10} | 0 | 0/6 | |
| {3, 8} | 0 | 0/6 | | {6, 7} | 0 | 0/6 | |
| {3, 9} | 2 | 2/6 | √ | {6, 8} | 2 | 2/6 | √ |
| {6, 9} | 0 | 0/6 | | {7, 10} | 0 | 0/6 | |
| {6, 10} | 2 | 2/6 | √ | {8, 9} | 0 | 0/6 | |

续表

| Items | Sum | sup($I_m \cup I_n$) | $L_2$ | Items | Sum | sup($I_m \cup I_n$) | $L_2$ |
|---|---|---|---|---|---|---|---|
| {7, 8} | 0 | 0/6 | | {8, 10} | 2 | 2/6 | √ |
| {7, 9} | 4 | 4/6 | √ | {9, 10} | 0 | 0/6 | |

所以，2 项强项集 $L_2$ = {{1,3},{1,4},{1,5},{1,6},{1,7},{1,8},{1,9},{1,10},{3,5},{3,7},{3,9},{4,5},{4,6},{4,7},{4,8},{4,9},{4,10},{5,7},{5,9},{6,8},{6,10},{7,9},{8,10}}。

（3）计算 2 项强项集的可信度，得到 2 项关联规则，如表 3.15 所示。

表 3.15　2 项强项集的可信度和 2 项关联规则

| Items | sup($I_m \cup I_n$) | sup($I_m$) | sup($I_n$) | conf($I_m \Rightarrow I_n$) | 2 项关联规则 |
|---|---|---|---|---|---|
| {1, 3} | 2/6 | 5/6 | 2/6 | 2/5 | |
| {1, 4} | 3/6 | 5/6 | 4/6 | 3/5 | |
| {1, 5} | 3/6 | 5/6 | 4/6 | 3/5 | |
| {1, 6} | 2/6 | 5/6 | 2/6 | 2/5 | |
| {1, 7} | 3/6 | 5/6 | 4/6 | 3/5 | |
| {1, 8} | 2/6 | 5/6 | 2/6 | 2/5 | |
| {1, 9} | 3/6 | 5/6 | 4/6 | 3/5 | |
| {1, 10} | 2/6 | 5/6 | 2/6 | 2/5 | |
| {3, 5} | 2/6 | 2/6 | 4/6 | 1 | √ |
| {3, 7} | 2/6 | 2/6 | 4/6 | 1 | √ |
| {3, 9} | 2/6 | 2/6 | 4/6 | 1 | √ |
| {4, 5} | 2/6 | 4/6 | 4/6 | 1/2 | |
| {4, 6} | 2/6 | 4/6 | 2/6 | 1/2 | |
| {4, 7} | 2/6 | 4/6 | 4/6 | 1/2 | |
| {4, 8} | 2/6 | 4/6 | 2/6 | 1/2 | |
| {4, 9} | 2/6 | 4/6 | 4/6 | 1/2 | |
| {4, 10} | 2/6 | 4/6 | 2/6 | 1/2 | |
| {5, 7} | 4/6 | 4/6 | 4/6 | 1 | √ |
| {5, 9} | 4/6 | 4/6 | 4/6 | 1 | √ |
| {6, 8} | 2/6 | 2/6 | 2/6 | 1 | √ |
| {6, 10} | 2/6 | 2/6 | 2/6 | 1 | √ |
| {7, 9} | 4/6 | 4/6 | 4/6 | 1 | √ |
| {8, 10} | 2/6 | 2/6 | 2/6 | 1 | √ |

产生的关联规则为 $I(3)\Rightarrow I(5)$,$I(3)\Rightarrow I(7)$,$I(3)\Rightarrow I(9)$,$I(5)\Rightarrow I(7)$,$I(5)\Rightarrow I(9)$,$I(6)\Rightarrow I(8)$,$I(6)\Rightarrow I(10)$,$I(7)\Rightarrow I(9)$,$I(8)\Rightarrow I(10)$。

同理,按照上述算法,可以求出 3 项、4 项候选集、强项集和关联规则等,在此不再做详细计算。通过这两个例子可以发现,设定不同的最小支持度,相应求出的强项集也会发生变化,产生的关联规则也将有差异。

#### 4. 使用 Python 语言实现关联规则算法

当最小支持度 min_sup=0.5,最小置信度 min_conf=0.7 时,实现本实例的 Python 示例代码如下。

Python 示例代码:

```
(1)  #加载数据
(2)  def loadDataSet():
(3)      return [["男", "40 岁以上", "高学历","高校教师", "收入 5000 以上"],
(4)              ["女", "40 岁以下", "高学历", "高校教师", "收入 5000 以上"],
(5)              ["男", "40 岁以下", "低学历", "非高校教师", "收入 5000 以下"],
(6)              ["男", "40 岁以上", "高学历", "高校教师", "收入 5000 以上"],
(7)              ["男", "40 岁以下", "高学历", "高校教师", "收入 5000 以上"],
(8)              ["男", "40 岁以下", "低学历", "非高校教师", "收入 5000 以下"]]
(9)  #生成候选集
(10) def createC1(dataSet):
(11)     C1=[]
(12)     for transaction in dataSet:
(13)         for item in transaction:
(14)             if not [item] in C1:
(15)                 C1.append([item])
(16)     C1.sort()
(17)     return list(map(frozenset,C1))
(18) #扫描数据,返回频繁项集和支持度
(19) def scanD(D,CK,minSupport):
(20)     ssCnt = {}
(21)     for tid in D:
(22)         for can in CK:
(23)             if can.issubset(tid):
(24)                 if not can in ssCnt:ssCnt[can]=1
(25)                 else:ssCnt[can]+=1
(26)     numItems = float(len(D))
(27)     retList = []
(28)     supportData={}
(29)     for key in ssCnt:
(30)         support = ssCnt[key]/numItems
(31)         if support>=minSupport:
(32)             retList.insert(0,key)
(33)         supportData[key]=support
(34)     return retList,supportData
(35) #频繁项集两两组合
```

```
(36)  def aprioriGen(Lk,k):
(37)      retList=[]
(38)      lenLk = len(Lk)
(39)      for i in range(lenLk):
(40)          for j in range(i+1,lenLk):
(41)              L1=list(Lk[i])[:k-2];L2=list(Lk[j])[:k-2]
(42)              L1.sort();L2.sort()
(43)              if L1==L2:
(44)                  retList.append(Lk[i]|Lk[j])
(45)      return retList
(46)  #生成频繁项集主函数,返回频繁项集和支持度
(47)  def apriori(dataSet,minSupport):
(48)      C1=createC1(dataSet)
(49)      D=list(map(set,dataSet))
(50)      L1,supportData =scanD(D,C1,minSupport)
(51)      L=[L1]
(52)      k=2
(53)      while(len(L[k-2])>0):
(54)          CK = aprioriGen(L[k-2],k)
(55)          Lk,supK = scanD(D,CK,minSupport)
(56)          supportData.update(supK)
(57)          L.append(Lk)
(58)          k+=1
(59)      return L,supportData
(60)  #规则计算的主函数
(61)  def generateRules(L,supportData,minConf):
(62)      bigRuleList = [] #关联规则
(63)      for i in range(1,len(L)):
(64)          for freqSet in L[i]:
(65)              H1 = [frozenset([item]) for item in freqSet]
(66)              if(i>1):
(67)                  rulesFromConseq(freqSet,H1,supportData,bigRuleList,minConf)
                                                    #如果有频繁 2 项集以上,计算规则
(68)              else:
(69)                  calcConf(freqSet,H1,supportData,bigRuleList,minConf)
                                                #如果只有频繁 1 项集,直接计算最小置信度
(70)      return bigRuleList
(71)  #大于最小支持度的规则进行输出
(72)  def calcConf(freqSet,H,supportData,brl,minConf):
(73)      prunedH=[]
(74)      for conseq in H:
(75)          conf = supportData[freqSet]/supportData[freqSet-conseq]
(76)          if conf>=minConf:
(77)              print (freqSet-conseq,'--->',conseq,'置信度:',conf)
(78)              brl.append((freqSet-conseq,conseq,conf))
(79)              prunedH.append(conseq)
```

```
(80)        return prunedH
(81)    #生成规则
(82)    def rulesFromConseq(freqSet,H,supportData,brl,minConf):
(83)        m = len(H[0])
(84)        if (len(freqSet)>(m+1)):
(85)            Hmp1 = aprioriGen(H,m+1)
(86)            Hmp1 = calcConf(freqSet,Hmp1,supportData,brl,minConf)
(87)            if(len(Hmp1)>1):
(88)                rulesFromConseq(freqSet,Hmp1,supportData,brl,minConf)
(89)    #主函数
(90)    if __name__=='__main__':
(91)        dataSet=loadDataSet()
(92)        L,supportData=apriori(dataSet,minSupport=0.5)
(93)        rules = generateRules(L,supportData,minConf=0.7)
```

运行结果如下：

```
frozenset({'40岁以下'}) ---> frozenset({'男'}) 置信度: 0.75
frozenset({'收入5000以上'}) ---> frozenset({'男'}) 置信度: 0.75
frozenset({'收入5000以上'}) ---> frozenset({'高学历'}) 置信度: 1.0
frozenset({'高学历'}) ---> frozenset({'收入5000以上'}) 置信度: 1.0
frozenset({'高学历'}) ---> frozenset({'男'}) 置信度: 0.75
frozenset({'收入5000以上'}) ---> frozenset({'高校教师'}) 置信度: 1.0
frozenset({'高校教师'}) ---> frozenset({'收入5000以上'}) 置信度: 1.0
frozenset({'高校教师'}) ---> frozenset({'男'}) 置信度: 0.75
frozenset({'高学历'}) ---> frozenset({'高校教师'}) 置信度: 1.0
frozenset({'高校教师'}) ---> frozenset({'高学历'}) 置信度: 1.0
frozenset({'高学历'}) ---> frozenset({'高校教师', '男'}) 置信度: 0.75
frozenset({'高校教师'}) ---> frozenset({'高学历', '男'}) 置信度: 0.75
frozenset({'收入5000以上'}) ---> frozenset({'高校教师', '高学历'}) 置信度: 1.0
frozenset({'高学历'}) ---> frozenset({'高校教师', '收入5000以上'}) 置信度: 1.0
frozenset({'高校教师'}) ---> frozenset({'高学历', '收入5000以上'}) 置信度: 1.0
frozenset({'收入5000以上'}) ---> frozenset({'高校教师', '男'}) 置信度: 0.75
frozenset({'高校教师'}) ---> frozenset({'收入5000以上', '男'}) 置信度: 0.75
frozenset({'收入5000以上'}) ---> frozenset({'高学历', '男'}) 置信度: 0.75
frozenset({'高学历'}) ---> frozenset({'收入5000以上', '男'}) 置信度: 0.75
frozenset({'高学历', '收入5000以上'}) ---> frozenset({'高校教师', '男'}) 置信度:
0.75
frozenset({'高学历', '男'}) ---> frozenset({'高校教师', '收入5000以上'}) 置信度:
1.0
frozenset({'收入5000以上', '男'}) ---> frozenset({'高校教师', '高学历'}) 置信度:
1.0
frozenset({'高校教师', '高学历'}) ---> frozenset({'收入5000以上', '男'}) 置信度:
0.75
frozenset({'高校教师', '收入5000以上'}) ---> frozenset({'高学历', '男'}) 置信度:
0.75
frozenset({'高校教师', '男'}) ---> frozenset({'高学历', '收入5000以上'}) 置信度:
1.0
frozenset({'高学历'}) ---> frozenset({'高校教师', '收入5000以上', '男'}) 置信度:
0.75
frozenset({'收入5000以上'}) ---> frozenset({'高校教师', '高学历', '男'}) 置信度:
0.75
```

```
frozenset({'高校教师'}) ---> frozenset({'收入 5000 以上', '高学历', '男'}) 置信度:
0.75
```

代码运行结果如上，其中，frozenset 为 Python 语言中的“冻结集合”数据类型，冻结集合中的元素代表相应项集。例如，“frozenset({'40 岁以下'}) －－－＞ frozenset({'男'}) 置信度：0.75”代表年龄＜40，在置信度为 75％的情况下可以推出其性别为男性；“frozenset({'高校教师', '高学历'}) －－－＞ frozenset({'收入 5000 以上', '男'}) 置信度：0.5”代表学历为博士或者硕士，同时为高校教师，在置信度为 75％的情况下可以推出在其性别为男性，同时工资在 5000 元以上。

**例 3.2**　Apriori 算法举例。

表 3.16 给出了每个客户(记录 ID 编号)交易各类商品的记录情况。

**表 3.16　每个客户交易各类商品的记录情况**

| 交易 ID | 商品 *A* | 商品 *B* | 商品 *C* | 商品 *D* | 商品 *E* | 商品 *F* |
|---|---|---|---|---|---|---|
| 1000 | *A* | | *C* | | | |
| 2000 | *A* | *B* | *C* | | | |
| 4000 | *A* | | | *D* | | |
| 5000 | | *B* | | | *E* | *F* |

假设表 3.16 交易商品对应的项集、满足最小支持度与最小可信度为 50％的频繁项集分别如表 3.17 和表 3.18 所示。

**表 3.17　购买商品(对应的项集)**

| 交易 ID | 购买商品(对应的项集) |
|---|---|
| 1000 | *A*，*C* |
| 2000 | *A*，*B*，*C* |
| 4000 | *A*，*D* |
| 5000 | *B*，*E*，*F* |

**表 3.18　频繁项集和支持度**

| 频 繁 项 集 | 支　持　度 |
|---|---|
| {A} | sup(*A*)＝3/4＝75％ |
| {B} | sup(*B*)＝2/4＝50％ |
| {C} | sup(*C*)＝2/4＝50％ |
| {A，C} | sup(*A*，*C*)＝2/4＝50％ |

对表 3.17 的解释如下。

(1) 由于交易总的记录为 4，因此 $n=4$。

(2) 由于在“购买商品”属性中，*A* 商品有 3 次，*B* 和 *C* 商品分别有 2 次，{*A*，*C*}即同时

购买 $A$,$C$ 商品也是 2 次,因此,$\{A\}$的支持度=3/4=0.75=75%,$\{B\}$的支持度=2/4=0.5=50%…。

(3) 由于事先已经(假设)定义好其支持数为 2,即用 2 除以(总的记录数)4 得 50%为最小支持度,后面的计算规则中只有满足其最小支持度才有可能产生规则即知识。

对于 $A$,$C$ 而言,$\sup(\{A,C\})=2/4=50\%$。

因此,$\text{conf}(\{A,C\})=\sup(\{A,C\})/\sup(A)=50\%/75\%=66.66\%$。

**例 3.3** 现有 $A$,$B$,$C$,$D$,$E$ 五种商品的交易记录表如表 3.19 所示,试找出三种商品关联销售情况($K=3$),假设最小支持度≥50%。

**表 3.19 五种商品的交易记录表对应的项集**

| 交易 ID | 商品代码(对应的项集) |
|---|---|
| 100 | $A$,$C$,$D$ |
| 200 | $A$,$C$,$E$ |
| 300 | $A$,$B$,$C$,$E$ |
| 400 | $B$,$E$ |

解答:该问题的实质是要求我们运用表 3.19 中的已知数据,求出其长度为 3 的强项集即 $L_3$。

可以根据表 3.19 的已知条件,分别计算其 $K=1(L_1)$,$K=2(L_2)$和 $K=3(L_3)$的结果。

当 $K=1$ 时,对应的项集与候选项集如表 3.20 所示。

**表 3.20 当 $K=1$ 时,对应的项集与候选项集**

| $K=1$(长度为 1 的候选项集) | 候选项集 | 支 持 度 |
|---|---|---|
| C1 | $\{A\}$ | 2/4=0.5=50% |
| C1 | $\{B\}$ | 3/4=0.75 |
| C1 | $\{C\}$ | 3/4=0.75 |
| C1 | $\{D\}$ | 1/4=0.25==25%<br>因支持度<50%,将其剔除 |
| C1 | $\{E\}$ | 3/4=0.75 |

通过表 3.20 满足条件的结果,再求出当 $K=1$ 时,其支持度≥50%对应的强项集,如表 3.21 所示。

**表 3.21 当 $K=1$ 时,其支持度≥50%对应的强项集**

| $K=1$(长度为 1 的候选项集) | 候选项集 | 支 持 度 |
|---|---|---|
| C1 | $\{A\}$ | 2/4=0.5=50% |
| C1 | $\{B\}$ | 3/4=0.75 |
| C1 | $\{C\}$ | 3/4=0.75 |
| C1 | $\{E\}$ | 3/4=0.75 |

再根据表 3.21 的结果作为计算 $K=2$ 的基础条件，求出 $K=2$ 对应的支持度，如表 3.22 所示。

表 3.22　当 $K=2$ 时，其候选项集对应的支持度

| $K=2$（长度为 2 的候选项集） | 候选项集 | 支持度 |
|---|---|---|
| C2 | $\{A,B\}$--1 | 1/4=0.25=25% |
| C2 | $\{A,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{A,D\}$--1 | 1/4=0.25=25% |
| C2 | $\{B,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{B,E\}$--3 | 3/4=0.75=75% |
| C2 | $\{C,D\}$--1 | 1/4=0.25=25% |
| C2 | $\{C,E\}$--2 | 2/4=0.5=50% |

将表 3.22 中不满足最小支持度＜50%的候选项集剔除，如表 3.23 所示。

表 3.23　将不满足最小支持度的候选项集剔除

| $K=2$（长度为 2 的候选项集） | 候选项集 | 支持度 |
|---|---|---|
| C2 | $\{A,B\}$--1 | 1/4=0.25=25%<br>因支持度＜50%，将其剔除 |
| C2 | $\{A,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{A,D\}$--1 | 1/4=0.25=25%<br>因支持度＜50%，将其剔除 |
| C2 | $\{B,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{B,E\}$--3 | 3/4=0.75=75% |
| C2 | $\{C,D\}$--1 | 1/4=0.25=25%<br>因支持度＜50%，将其剔除 |
| C2 | $\{C,E\}$--2 | 2/4=0.5=50% |

剔除不满足最小支持度后的情况如表 3.24 所示。

表 3.24　剔除不满足最小支持度后的情况

| $K=2$（长度为 2 的项集） | 项集 | 支　持　度 |
|---|---|---|
| C2 | $\{A,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{B,C\}$--2 | 2/4=0.5=50% |
| C2 | $\{B,E\}$--3 | 3/4=0.75=75% |
| C2 | $\{C,E\}$--2 | 2/4=0.5=50% |

因此，根据表 3.24 计算 $K=2$ 对应的强项集 $L_2$ 结果如表 3.25 所示。

表 3.25　$K=2$ 对应的强项集 $L_2$

| $K=2$（长度为 2 的强项集 $L_2$） | 强项集 | 支　持　度 |
|---|---|---|
| $L_2$ | $\{A,C\}$--2 | 2/4=0.5=50% |
| $L_2$ | $\{B,C\}$--2 | 2/4=0.5=50% |

续表

| $K=2$(长度为 2 的强项集 $L_2$) | 强项集 | 支　持　度 |
|---|---|---|
| $L_2$ | $\{B,E\}$--3 | 3/4=0.75=75% |
| $L_2$ | $\{C,E\}$--2 | 2/4=0.5=50% |

即将表 3.25 缩写成 $L_2$ 的表示形式,如表 3.26 所示。

表 3.26　将表 3.25 缩写成 $L_2$ 的表示形式

| | | |
|---|---|---|
| $L_2$ | $\{A,C\}$ | 50% |
| | $\{B,C\}$ | 50% |
| | $\{B,E\}$ | 75% |
| | $\{C,E\}$ | 50% |

再按照表 3.26 计算 $K=3$ 的强项集,如表 3.27 所示。

表 3.27　$K=3$ 对应的强项集 $L_3$

| $K=3$(长度为 3 的强项集) | 强项集 | 支　持　度 |
|---|---|---|
| C3 | $\{A,B,C\}$--1 | sup($A,B,C$)=1/4=0.25=25% |
| C3 | $\{A,C,E\}$--1 | sup($A,B,C$)=1/4=0.25=25% |
| C3 | $\{B,C,E\}$--2 | sup($A,B,C$)=2/4=0.5=50%<br>**满足 $K=3$ 的强项集条件 |

表 3.27 缩写成 $L_3$ 的表示形式,如表 3.28 所示。

表 3.28　长度为 3 的强项集 $L_3$

| | | |
|---|---|---|
| $L_3$ | $\{B,C,E\}$ | 50% |

归纳起来,求解表 3.19 的三种关联销售情况(即求 $K=3$ 的强项集)的整个计算过程,可以用图 3.1 表示。

五种商品的交易记录表

| 交易ID | 商品代码$A$ | 商品代码$B$ | 商品代码$C$ | 商品代码$D$ | 商品代码$E$ |
|---|---|---|---|---|---|
| 100 | $A$ | | $C$ | $D$ | |
| 200 | | $B$ | $C$ | | $E$ |
| 300 | $A$ | $B$ | $C$ | | $E$ |
| 400 | | $B$ | | | $E$ |

| | | |
|---|---|---|
| $L_3$ | $\{B,C,E\}$ | 50% |
| $C_3$ | $\{A,B,C\}$ | sup($A,B,C$)=1/4=0.25=25% |
| | $\{A,C,E\}$ | sup($A,C,E$)=1/4=0.25=25% |
| | $\{B,C,E\}$ | sup($B,C,E$)=2/4=0.5=50% |

根据已知的五种商品的交易记录表,求出所有满足$K$=3的候选项集,计算如下:

(1) ($A,C$) + ($B,C$) ⟶ ($A,B,C$)　仅存在1项

(2) ($A,C$) + ($B,E$) ⟶ 超过3项,不满足$K$=3的条件

(3) ($A,C$) + ($C,E$) ⟶ ($A,C,E$)　仅存在1项

(4) ($B,C$) + ($B,E$) ⟶ ($B,C,E$)　存在2项

(5) ($B,C$) + ($C,E$) ⟶ ($B,C,E$)　存在2项

(6) ($B,E$) + ($C,E$) ⟶ ($B,C,E$)　存在2项

图 3.1　按照表 3.19 计算 $K=3$ 的强项集 $L_3$ 的图示

基于 Hadoop 的词频统计方法

## 3.3　关联规则的创新应用研究

### 3.3.1　基于 Hadoop 环境下 Map-Reduce 流程的应用

Hadoop 环境下执行 Map-Reduce 流程的一个例子，如图 3.2 所示。

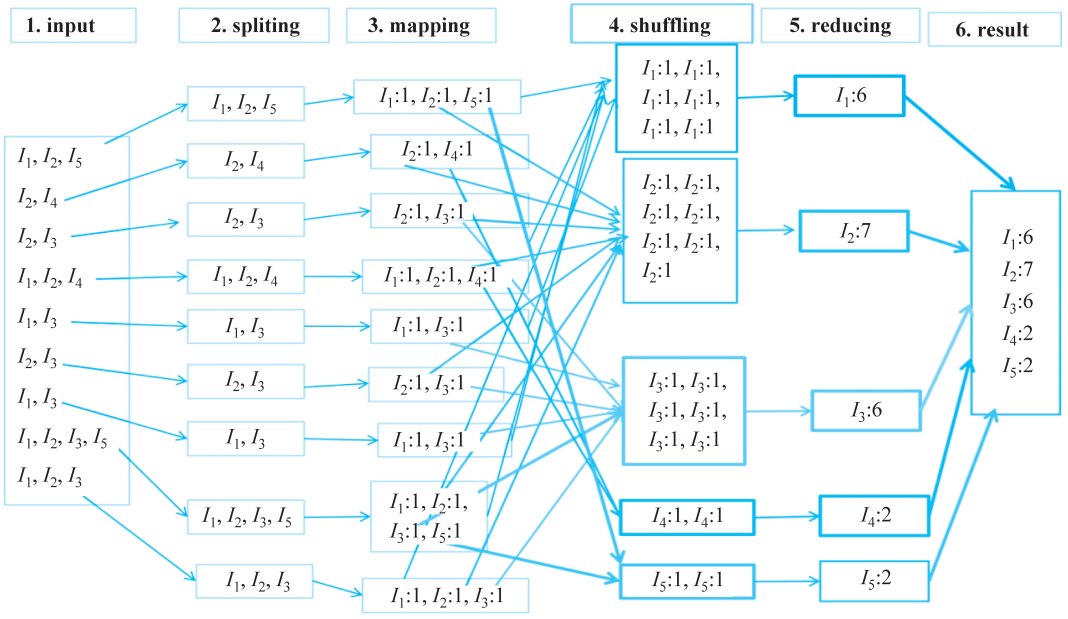


图 3.2　Hadoop 环境下执行 Map-Reduce 流程的一个例子

在图 3.2 的机制下，可以完成基于 Hadoop 环境下的并行词频统计，例如，可以将 $I_1$ 到 $I_5$ 称为一个文本，假设是由英语单词构成的文本（语句或词组，或英语词组等内容），英语文本内容是 $I_1$ =“traffic”，$I_2$ =“big”，$I_3$ =“data”，$I_4$ =“text”，$I_5$ =“mining”；经过 Hadoop 环境下 Map-Reduce 处理流程（统计单词）后，其输出结果分别为：$I_1$ =“traffic”（总数为 6），$I_2$ =“big”（总数为 7），$I_3$ =“data”（总数为 6），$I_4$ =“text”（总数为 2），$I_5$ =“mining”（总数为 2）。

通过 Hadoop 机制下的词频统计例子，可以得知：Map-Reduce 是目前处理并行计算的非常优秀的解决方法。也可以运用这种机制来解决其他相关领域的并行统计机制的算法，例如，通过对网络环境下的某领域的热点词和热点话题内容的统计与分析，来判断物流企业的管理与决策领域关注的重点业务需求如何；通过某大型超市商品种类的新需求来决定并及时调整进货的数量。

### 3.3.2　基于 FP-tree 的频繁项集的挖掘方法

韩家伟等学者在 2000 年提出了分而治之策略的 FP-tree 递归增长频繁项集。具体的方法是：①对每个项（item）生成其条件模式库，再作为其条件 FP-tree；②对每个新生成的条件 FP-tree，重复执行其步骤；③如果 FP-tree 为空，或者仅含有一个路径（此路径对应的每个子路径都是频繁项集）则结束。

下面通过表 3.29 的举例来解释构造 FP-tree 的过程。