第八章 关联分析

【学习目标】

- 理解关联规则原理
- 掌握频繁模式的方法
- 掌握Apriori算法

关联分析概念

- 关联分析是一种简单、实用的分析技术，就是发现存在于大量数据集中的关联性或相关性，从而描述了一个事物中某些属性同时出现的规律和模式。
- 关联分析是从大量数据中发现项集之间有趣的关联和相关联系。关联分析的一个典型例子是购物篮分析。该过程通过发现顾客放入其购物篮中的不同商品之间的联系，分析顾客的购买习惯。通过了解哪些商品频繁地被顾客同时购买，这种关联的发现可以帮助零售商制定营销策略。其他的应用还包括价目表设计、商品促销、商品的排放和基于购买模式的顾客划分。
- 可从数据库中关联分析出形如“由于某些事件的发生而引起另外一些事件的发生”之类的规则。如“67%的顾客在购买啤酒的同时也会购买尿布”，因此通过合理的啤酒和尿布的货架摆放或捆绑销售可提高超市的服务质量和效益。又如“‘C语言’课程优秀的同学，在学习‘数据结构’时为优秀的可能性达88%”，那么就可以通过强化“C语言”的学习来提高教学效果。

8.1 频繁模式与关联规则

TID	Items
001	Cola, Egg, Ham
002	Cola, Diaper, Beer
003	Cola, Diaper, Beer, Ham
004	Diaper, Beer

如表所示是一个超市几名顾客的交易信息。
TID代表交易流水号，Items代表一次交易的商品。
对这个数据集进行关联分析，可以找出关联规则{Diaper}→{Beer}。它代表的意义是：购买了Diaper的顾客会购买Beer。这个关系不是必然的，但是可能性很大，这就已经足够用来辅助商家调整Diaper和Beer的摆放位置了，例如摆放在相近的位置，进行捆绑促销来提高销售量。

关联分析的关键概念

1．事务（Transaction）：每一条交易称为一个事务，例如表8-1中的数据集就包含四个事务。事务用t_i表示，其中i是交易号。交易数据库D是交易的集合，即：

$$D = \{t_1, t_2, \cdots, t_m\}$$

2.项(Item)：交易的每一个物品称为一个项，例如Cola、Egg等。

3.项集(Item set)：包含零个或多个项的集合叫做项集，例如{Cola, Egg, Ham}。项集用I表示，

$$I = \{i_1, i_2, \cdots, i_n\}$$

。一个项集包含项的个数称为该项集的长度。

4.k−项集：包含k个项的项集叫做k-项集，例如{Cola}叫做1-项集，{Cola, Egg}叫做2-项集。

- 5.支持度计数(count(X))：一个项集X在数据库D中出现的次数，等于包含该项集的交易个数。若X={cola}，则count(X)=3。又如{Diaper, Beer}出现在事务 002、003和004中，所以它的支持度计数是3。

6.支持度（Support）：支持度计数除于总的事务数,记为Support（X）。

例如上例中总的事务数为4，{Diaper, Beer}的支持度计数为3，所以它的支持度是3÷4=75%。
说明有75%的人同时买了Diaper和Beer。

$$\sup port(X) = \frac{count(X)}{|D|} \times 100\%$$

7.频繁项集：支持度大于或等于某个阈值（又称为最小支持度（Minimum Support，minsup））的项集就叫做频繁项集。例如阈值设为50%时，因为{Diaper, Beer}的支持度是75%，所以它是频繁项集。

TID	Items
001	Cola, Egg, Ham
002	Cola, Diaper, Beer
003	Cola, Diaper, Beer, Ham
004	Diaper, Beer

- 8.置信度：项集 的支持度除以项集X的支持度，定义如下：
- $$confidence(X \rightarrow Y) = \frac{\sup\ port(X \rightarrow Y)}{\sup\ port(X)} \times 100\%$$
- 对于规则{Diaper}→{Beer}，{Diaper, Beer}的支持度计数除于{Diaper}的支持度计数，为这个规则的置信度。例如规则{Diaper}→{Beer}的置信度为3÷3=100%。说明买了Diaper的人100%也买了Beer。

例子：某超市2021年共交易1200W笔，其中，可乐购买次数400W笔，购买可乐又购买了薯片是300W笔，顾客购买可乐又购买面包有100W笔。面包购买：500W笔；薯片购买600万笔。

- support(可乐）= 400/1200　support(面包) = 500/1200,　support(薯片) = 600/1200
- Support(可乐，薯片) = 300/1200
- Support(可乐，面包) = 100/1200
- Confidence(可乐，薯片) = 300/1200 / 400/1200 = 3/4
- Confidence(可乐，面包) = 100/1200 / 400/1200 = 1/3
- Confidence(薯片，可乐) = 300/1200 / 600/1200 =1/2
- Confidence(面包，可乐) = 100/1200 / 500/1200 = = 1/5

8.1.3关联规则的度量

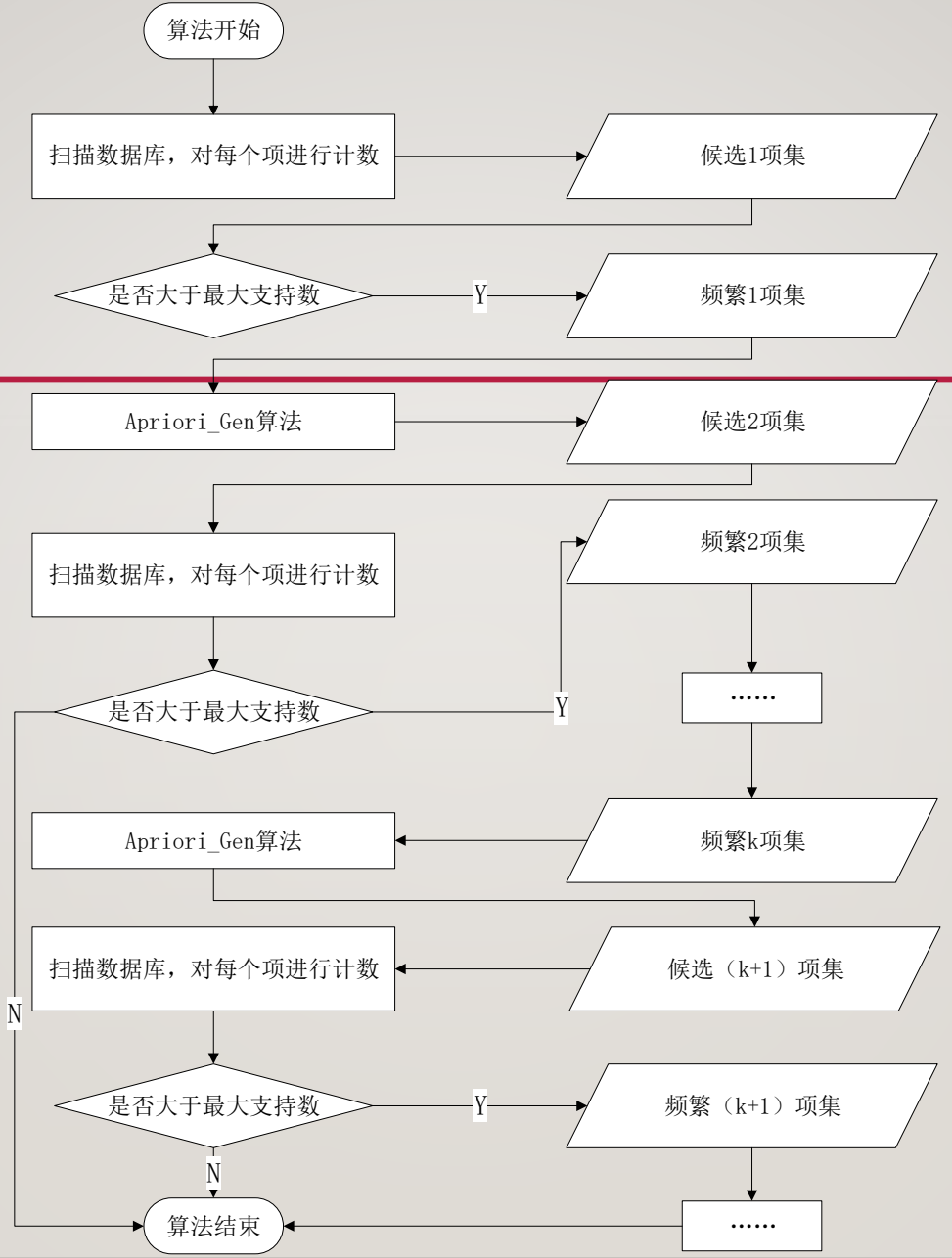
- 前面介绍的关联规则的定义中涉及两个参数：支持度和置信度。这两种度量是描述一条关联规则是否有意义的常用度量。但是在有些情况下，仅仅根据这两个度量发现的规则可能具有误导性。例如，如果利用关联分析的方法分析修计算机原理课程的学生的基本信息以及所得成绩之间的关系，可能得到这样一条关联规则：专业=计算机→成绩=良（53%，72.6%），这使人感觉计算机专业的学生更易得“良”。但是实际上总人数中得“良”的比例为75%，也就是说如果是计算机专业的学生，得“良”的可能性反而降低了。那么，我们如何能避免发现此类关联规则呢？

例子：

- 例：在所分析的10000个事务中,6000个事务包含计算机游戏,7500个包含游戏机游戏,4000个事务同时包含两者。
- 关联规则（计算机游戏，游戏机游戏） 支持度为0.4，看似很高，但其实这个关联规则是一个误导。
- 在用户购买了计算机游戏后有 （4000÷6000）=0.667 的概率的去购买游戏机游戏？**怎么算出来的？**，而在没有任何前提条件时，用户反而有（7500÷10000）=0.75的概率去购买游戏机游戏，也就是说设置了购买计算机游戏这样的条件反而会降低用户去购买游戏机游戏的概率，所以计算机游戏和游戏机游戏是相斥的。
- 在理论上把 0.667除以0.75称为提升度， 一般在数据挖掘中当提升度大于3时,我们才承认挖掘出的关联规则是有价值的。

8.2.1 APRIORI算法

- Apriori算法是一种挖掘关联规则的频繁项集算法，其核心思想是通过候选集生成和情节的向下封闭检测两个阶段来挖掘频繁项集。而且算法已经被广泛的应用到商业、网络安全等各个领域。
- 该算法的基本思想是：首先找出所有的频繁集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用找到的频繁集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递归的方法。


算法开始
扫描数据库，对每个项进行计数
候选1项集
是否大于最大支持数
Y
频繁1项集
Apriori_Gen算法
候选2项集
扫描数据库，对每个项进行计数
频繁2项集
是否大于最大支持数
Y
……
Apriori_Gen算法
频繁k项集
扫描数据库，对每个项进行计数
候选（k+1）项集
N
是否大于最大支持数
Y
频繁（k+1）项集
N
算法结束
……

例子：

下面举例说明该算法的运行过程： 假设有一个数据库D，其中有4个事务记录

设置最小支持度miSup=2,算法运行的过程如下：

TID	Items
T1	I1,I3,I4
T2	I2,I3,I5
T3	I1,I2,I3,I5
T4	I2,I5

扫描D，对每个候选项进行支持度计数得到表C1：

TID	Items
T1	I1,I3,I4
T2	I2,I3,I5
T3	I1,I2,I3,I5
T4	I2,I5

项集	支持度计数
{I1}	2
{I2}	3
{I3}	3
{I4}	1
{I5}	3

表C1