

一种挖掘频繁项的新方法

陈冰, 张化祥

(山东师范大学信息科学与工程学院, 山东 济南 250014)

摘要:介绍了关联规则挖掘的情况, 然后对关联规则挖掘算法进行分析, 并在此分析的基础上对经典的 Apriori 算法作出了进一步的改进, 从而提出了这种改进的关联规则挖掘算法——Apriori-New 算法。Apriori-New 算法只需对数据库扫描一次, 并在扫描过程中通过不断将被标记为频繁项的项集提取出来, 最终找出所有的频繁项集。通过一个简单的实例说明了该算法的扫描过程, 从而体现了该 Apriori-New 算法的效率及其所具有的实用性。

关键词:数据挖掘; 关联规则; Apriori 算法

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2008)08-0118-03

A New Method of Mining Frequent-Item

CHEN Bing, ZHANG Hua-xiang

(College of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract: Provides a survey of the study in association rule generation. And then makes an analysis of the algorithm of association rule generation. On the basis of the analysis, the classical algorithm of Apriori is analyzed. Meanwhile the algorithm is making a further modification. Then an improved Apriori algorithm of Apriori-New is proposed. Due to its advantage of scanning DB only once, during the process of scanning, the frequent-items are marked and selected. In the end, all of the frequent-items can be found. A simple example is used to show the process of scanning. Then the new algorithm - Apriori-New is proposed with high efficiency and certain practical significance.

Key words: data mining; association rules; Apriori algorithm

0 引言

数据挖掘^[1,2]是从大量数据中提取或“挖掘”知识, 也称数据库中的知识发现 (Knowledge Discovery in Database, KDD), 具体是指从大型数据库或数据仓库中提取人们感兴趣的知识, 而且这些知识是隐含的、事先未知的潜在有用信息, 提取的知识一般可表示为概念 (Concepts), 规则 (Rules), 规律 (Regularities), 模式 (Patterns) 等形式。即: 数据挖掘是一个利用各种分析工具在海量数据中发现模型和数据间关系的过程, 这些模型和关系可以用来作出预测。对于数据挖掘技术的研究已引起了国际人工智能和数据库等领域专家与学者的广泛关注, 这其中在事务数据库中挖掘关联规则是数据挖掘领域中的一个非常重要的研究课题。关联规

则是美国 IBM Almaden research center 的 Rabesh Agrawal 等人于 1993 年首先提出的, 最近几年在数据挖掘研究领域对关联规则挖掘的研究开展得比较积极和深入。

1 关联规则分析

关联规则发现的一个典型的例子就是购物篮分析^[3]。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。这是关联规则挖掘的最初形式。

设 $I = \{i_1, i_2, \dots, i_m\}$ 是由 m 个不同的项目组成的集合, 给定一个事务^[4]数据库 D , 其中的每一个事务 T 是 I 中一组项目的集合, 即 $T \subseteq I$, T 有一个唯一的标识符 TID。若项集 $A \subseteq I$ 且 $A \subseteq T$, 则事务 T 包含项集^[5] A 。一条关联规则就是形如 $A \Rightarrow B$ 的蕴涵式, 其中 $A \subseteq I, B \subseteq I, A \cap B = \emptyset$ 。相关规则 $A \Rightarrow B$ 成立的条件是:

(1) 它具有支持度 (support)^[5] s , 即事务数据库 D

收稿日期: 2007-11-15

基金项目: 山东省科技攻关计划 (2005GG4210002); 山东省青年科学家科研奖励基金 (2006BS01020); 山东省教育厅科技计划项目 (J07YJ04)

作者简介: 陈冰 (1981-), 女, 山东泰安人, 硕士研究生, 研究方向为数据挖掘、机器学习; 张化祥, 博士, 教授, 研究方向为机器学习、人工智能及 Web 挖掘。

中至少有 $s\%$ 的事务包含 $A \cup B$;

(2) 它具有置信度(confidence)^[5] c , 即在事务数据库 D 中包含 A 的事务至少有 $c\%$ 同时也包含 B 。

如^[6]: Contains(T , computer) $>$ contains(T , software) [$s = 5\%$, $c = 70\%$], 其中 T 表示交易。该规则说明若一笔交易中含有 computer, 则它也含有 software 的可能为 70%, 也就是说顾客如果购买了计算机, 则他同时也购买软件的可能是 70%, 而 $s = 5\%$ 则表明所有的交易中, 有 5% 的交易出现以上情况(同时购买了计算机和软件)。显然, 这样的规则对于货物摆放、促销安排都很有价值。关联规则的挖掘问题就是在事务数据库 D 中找出具有用户给定的最小支持度 minsup 和最小置信度 minconf 的关联规则, 即强关联规则。

因此, 关联规则提取问题可以分为以下两个问题^[7]:

(1) 找出所有满足最小支持度 min-conf 的项集, 称为频繁项集(或者强项集)。

(2) 根据频繁项集, 产生所有大于最小置信度的规则。

目前的研究重点在第一步, 即找出频繁项, 因为第二步相对而言比较容易做, 并且对于不同的项集数目可达 2^m 个, 且数据库规模大, 对所有的项集进行支持度的计算几乎是不可能的。文中对关联规则挖掘中的 Apriori 算法进行了深入研究, 从另一个角度对 Apriori 算法进行了进一步的改进, 使它在扫描数据库的效率上要优先于经典的 Apriori 算法。

2 Apriori 算法

Apriori 算法^[2]是挖掘产生布尔关联规则所需频繁项集的基本算法, 也是一个很有影响的关联规则挖掘算法。该算法利用了一个层次顺序搜索的循环方法来完成频繁项集的挖掘工作, 这一循环方法就是利用 k -项集来产生 $(k+1)$ -项集。具体做法是: 首先扫描数据库找出频繁 1-项集, 记为 L_1 ; 然后对 L_1 做连接^[8]操作生成 C_2 , 扫描数据库从 C_2 中选出 L_2 , 即频繁 2-项集; 不断如此循环下去直到不能再找到更多的频繁 k -项集为止。每挖掘一层 L_k 就需要扫描整个数据库一遍。

为了提高逐层产生频繁项集的效率, 一种称作 Apriori 性质的重要性质用于压缩搜索空间。Apriori 性质: 频繁项集的所有非空子集都必须是频繁项集。将 Apriori 性质用于寻找频繁项集分成两个过程^[1]: 连接和删除。

(1) 连接步骤: 为找 L_k , 通过 L_{k-1} , 与自己连接产

生候选 k -项集的集合, 该候选项集的集合记作 C_k 。

(2) 删除步骤: C_k 的成员可以是也可以不是频繁的, 但所有的频繁 k -项集都包含在 C_k 中。扫描数据库, 确定 C_k 中每个候选的计数, 从而确定 L_k (根据定义, 计数值不小于最小支持度计数的所有候选项是频繁的, 从而属于 L_k)。为了压缩, 可以这样使用 Apriori 性质: 任何非频繁的 $(k-1)$ -项集都不可能是频繁 k -项集的子集。因此, 如果一个候选 k -项集的 $(k-1)$ -子集不在 L_{k-1} 中, 则该候选项也不可能是频繁的, 从而可以在 C_k 中删除。

3 改进的 Apriori 算法——Apriori-New

3.1 Apriori-New 算法思想

Apriori-New 算法每次读取数据库中的一条记录, 从数据库的头部依次读取直至数据库的尾部, 且依次只读一遍。首先读取数据库的第一条记录 T_1 , 根据组成 T_1 的项目, 将这些项目构成 1-项集, 并构造这些 1-项集所有可能的组合(即由 1-项集组成的所有 2-项集、3-项集……), 此时并标记这些项集获得的覆盖量计数为 1。接着读取数据库的第二条记录 T_2 , 根据记录 T_2 包含的所有 1-项集构造可能的组合, 如果组合后得到的某个 k -项集:

① 在之前的组合中未出现, 则标记其获得的覆盖量计数为 1;

② 在之前的组合中已出现, 且已被标记为频繁项集, 则跳过该项集考察下一个项集;

③ 属于其它情形(即该项集在之前的组合中已出现, 但不满足最小支持度阈值 minsup), 则其对应的覆盖量计数加 1。

同样的方法来处理数据库的第三条记录 T_3 、第四条记录 T_4 ……直到数据库的最后一条记录 T_n 。

算法: Apriori-New 算法

输入: 数据库 D ; 最小支持度阈值 minsup。

输出: Answer = D 中的所有频繁项集。

方法:

Answer: = {};

For $i = 1$ To $|D|$ do

begin

为第 i 个记录 T_i 包含的所有 1-项集构造可能的组合 $\{A_{i2}, \dots, A_{in}\}$;

其中, A_{i2} 表示 2-项集, A_{in} 表示 n -项集(即由 T_i 包含的项目所组成的最大项), 由于 1-项集对于生成关联规则没有意义, 而且每扫描一个记录都要存储这些 1-项集将占用相当的空间, 所以不必记录这些 1

—项集而只记录由该 1—项集构造的所有组合(2—项集、3—项集……)。

```

For j = 1 To n do
begin
  对每个  $A_{ij}$ 
    如果  $A_{ij}$  第一次出现, 则为其生成一个计数器;
    如果  $A_{ij} \in \text{Answer}$  则 continue;
    否则:
       $A_{ij}$  对应的计数器加 1;
end;
如果  $A_{ij}$  对应的计数器满足不小于 minsup 则
  Answer = Answer  $\cup \{A_{ij}\}$ ;
end;

```

最后输出所得到的所有频繁项集 Answer 即可。

3.2 Apriori - New 算法示例

有一数据库 $D^{[1]}$, 其中有 6 个事务记录, 假设最小支持度为 2。运用 Apriori - New 算法得到关联规则见图 1。

记录 T_1 所有的项集: $I_1; I_2; I_3;$
 $I_1, I_2; I_1, I_3; I_2, I_3; I_1, I_2, I_3。$

记录 T_2 所有的项集: $I_2, I_3; I_2,$
 $I_3。$

记录 T_3 所有的项集: $I_1, I_4; I_5;$
 $I_1, I_4; I_1, I_5; I_4, I_5; I_1, I_4, I_5。$

记录 T_4 所有的项集: $I_2; I_6; I_2,$
 $I_6。$

记录 T_5 所有的项集: $I_1; I_2; I_3, I_5; I_1, I_2; I_1, I_3;$
 $I_1, I_5; I_2, I_3; I_2, I_5; I_3, I_5; I_1, I_2, I_3; I_1, I_2, I_5; I_1, I_3,$
 $I_5; I_2, I_3, I_5; I_1, I_2, I_3, I_5。$

记录 T_6 所有的项集: $I_1; I_3, I_1, I_3。$

扫描每一个记录后所得项集的过程如图 1 所示。

因此, 最后得到的最大频繁项为 $\{I_1, I_2, I_3\}$ 。

得到满足最少支持度 minsup 的频繁项集 Answer 后, 分别将项集转换成满足最少置信度 minconf 的规则或规则集。即完成了数据库 D 的关联规则的挖掘。

4 结束语

Apriori - New 算法的优越性:

该算法只需扫描数据库一次, 并且在扫描过程中通过不断将被标记为频繁项的项集提取出来, 从而大

TID	Items
T_1	I_1, I_2, I_3
T_2	I_2, I_3
T_3	I_1, I_4, I_5
T_4	I_2, I_6
T_5	I_1, I_2, I_3, I_5
T_6	I_1, I_3

Item set	覆盖量计数
I_1, I_2	1
I_1, I_3	1
I_2, I_3	1
I_1, I_2, I_3	1
Answer:={};	

Item set	覆盖量计数
I_1, I_2	1
I_1, I_3	1
I_2, I_3	2
I_1, I_2, I_3	1
Answer:={ $\{I_2, I_3\}$ };	

Item set	覆盖量计数
I_1, I_2	1
I_1, I_3	1
I_1, I_4	1
I_1, I_5	1
I_2, I_3	2
I_4, I_5	1
I_1, I_2, I_3	1
I_1, I_4, I_5	1
Answer:={ $\{I_2, I_3\}$ };	

Item set	覆盖量计数
I_1, I_2	1
I_1, I_3	1
I_1, I_4	1
I_1, I_5	1
I_2, I_3	2
I_2, I_6	1
I_4, I_5	1
I_1, I_2, I_3	1
I_1, I_4, I_5	1
Answer:={ $\{I_2, I_3\}$ };	

Item set	覆盖量计数
I_1, I_2	2
I_1, I_3	2
I_1, I_4	1
I_1, I_5	2
I_2, I_3	2
I_2, I_5	1
I_2, I_6	1
I_3, I_5	1
I_4, I_5	1
I_1, I_2, I_3	2
I_1, I_2, I_5	1
I_1, I_3, I_5	1
I_1, I_4, I_5	1
I_2, I_3, I_5	1
I_1, I_2, I_3, I_5	1
Answer:={ $\{I_1, I_2\}, \{I_1, I_3\},$ $\{I_1, I_5\}, \{I_2, I_3\}, \{I_1, I_2, I_3\}$ };	

结果同扫描 T_5 后的输出

图 1 利用 Apriori - New 算法提取关联规则的示例
 大地降低了搜索的开销。当数据库的记录非常多而每条记录中包含字段值(1—项集)又较少时, 该算法的优越性就更为明显。因此, 该算法具有一定的实用性。

参考文献:

- [1] 冯兴杰, 周 淳. Apriori 算法的改进[J]. 计算机工程, 2005, 31: 171-172.
- [2] 高伟峰. 数据挖掘中关联规则的研究及应用[D]. 武汉: 武汉理工大学, 2006.
- [3] 唐 敏. 关联规则挖掘算法在超市销售分析中的应用[J]. 计算机科学, 2006, 33(2): 149-150.
- [4] Han J, Pei J, Yin Y. Mining Frequent Patterns without Candidate Generation: A Frequent - Pattern Tree Approach[J]. Data Mining and Knowledge Discovery, 2004, 8(1): 53-87.

(下转第 125 页)

件的路由进行管理。这样既方便对构件和连接件进行统一的形式化描述,同时也有利于复合接口交互协议的推导(端口 Ψ 的规格、交互协议与构件 A0 端口 P01 的完全一致)。

规则 6 设端口 α 属于复合构件 A0,端口 α' 属于其成员构件 A1,则端口 α 映射到端口 α' 的必要条件是端口 α 的所有通道集都需要映射到端口 α' 的通道集上,即 $(E_a^I \rightarrow E_a'^I) \wedge (E_a^O \rightarrow E_a'^O) \wedge (E_a^{IO} \rightarrow E_a'^{IO})$ 。

在 D-ADL 语言中,为了提高连接规模,引入了端口的映射关系,其实质是通道之间的映射。

规则 7 设端口 α 属于复合构件 A0, α' 属于连接件(复合连接件)C0,则端口 α 和端口 α' 相连的必要条件是端口 α 和 α' 兼容,即 $\alpha \sim \alpha'$ 。

在 D-ADL 中,复合构件的行为分为两种^[4]:一种是表示其处理业务逻辑的计算功能;另一种是用于预定义演化行为,需要显示地表示。根据复合构件的每一个交互活动都需要通过接口来完成,因此,可以将复合构件的端口分为两类:一类端口主要映射到成员构件端口上,属 port 类;另一类端口主要用于演化行为的输入输出,与外界环境进行交互,属 Eport 类。

仍以上述订单支付系统为例进行说明。如图 4 所示,其中复合构件 PayFlat 的端口包括 Ψ 和 μ ,其中 Ψ 映射到端口 P01 上,主要用于传送订单号和支付方式; μ 映射到端口 P32 上,主要用于传送确认信息。它们的 D-ADL 语法描述如下:

```
Port  $\Psi$  is in (A)
Port  $\mu$  is out (B)
Port P01 is in (A')
Port P32 is out (B')
```

其中 A,B,A',B' 是通道集合,A 中的通道与 A' 通道一一对应,B 中的通道与 B' 通道一一对应。

在复合构件中端口的映射描述如下:

```
Configuration is {
Payflat  $\Psi$  rely P01
Payflat  $\mu$  rely P01
.....
}
```

规则 6 和规则 7 进一步定义了端口与通道的参数

类型规则。当多个构件进行组装的时候,除了要满足规则 1、2、3、4、5 外,还需要保证端口之间传送的数据类型是匹配的,这样才能保证传递数据的正确性。在 D-ADL 中的元素类型有多种,其中包括常规类型和体系结构类型^[5],在此不做过多的阐述,因此在端口的连接中主要还需要考虑到传输参数的兼容,包括所传递参数的数量、类型。

3 结束语

组装构件除了要考虑构件功能的组合外,还需要考虑接口的组装。在 D-ADL 框架中定义了构件组装的宏观理论及交互协议的推导,为了进一步完善 D-ADL,文中定义了接口的连接规则,其中定义 1~10 主要为端口连接规则提供必要的支持;规则 1、2、3、4、5 主要从结构上对端口的连接进行约束;规则 6、7 主要从端口本身的类型、参数类型以及参数个数进行一致性约束。整个定义有助于系统的设计与推演,同时也有利于避免系统行为的偏离以及系统死锁。

目前已开发了一套原型系统(SASM),在该原型系统中已经实现了构件与连接件的连接,但还未能完整地将端口的连接规则、构件的行为规约以及接口的交互规约应用到系统中。因此,下一步的研究工作包括:继续完善软件体系结构描述框架,利用体系结构端口连接规则进行构件动态组合失配检测,将端口连接规则应用到原型系统,直至完善整个系统。

参考文献:

- [1] 张世琨,张文娟,杨美清,等.基于软件体系结构的可复用构件制作和组装[J].软件学报,2001,12(9):1351-1359.
- [2] 冯冲,江贺,冯静芳.软件体系结构理论与实践[M].北京:人民邮电出版社,1996:7-8.
- [3] 李长云,李赣生,何频捷.一种形式化的动态体系结构描述语言[J].软件学报,2006,17(6):1349-1359.
- [4] Oreizy P, Gorlick M, Taylor R, et al. An architecture-based approach to self-adaptive software[J]. IEEE Intelligent Systems, 1999, 14(3):54-62.
- [5] 李长云.基于体系结构的软件动态演化研究[D].杭州:浙江大学,2005:26-27.

(上接第 120 页)

- [5] Witten I H, Frank E. 数据挖掘实用机器学习技术[M]. 第 2 版. 北京:机械工业出版社,2006:76-80.
- [6] 王锐,李晶,熊海蕴,等.基于关联规则的 Apriori 算法的可视化实现方法[J]. 计算机工程与设计,2007,28(4):757-759.
- [7] 芦洁,刘志镜.挖掘关联规则中对 Apriori 算法的一个改进[J]. 微电子学与计算机,2006,23(2):10-12.
- [8] 骆嘉伟,王艳,杨涛,等.一种结合完全连接的改进 Apriori 算法[J]. 计算机应用,2006,26(5):1174-1177.