第18卷 第10期
2008年10月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.18 No.10
Oct. 2008

一种改进的最大频繁集发现算法

李景文,刘军锋,闫遂军,邓晓斌
(桂林工学院,广西 桂林 541004)

摘 要:关联规则是数据挖掘的主要技术,而最大频繁集是关联规则挖掘的核心。关联规则发现的准确性与效率的好坏直接决定了发现的知识规则是否适用。阐述了关联规则、频繁集和频繁超集的定义,分析了现有关联规则算法的思想及其不足,然后在概率的基础上引入了期望长,提出了ELMFI算法,最后用实例进行仿真实验并做了比较分析。该算法直接产生期望长度的候选项集并进行验算,试验结果验证了其可行性,发现效率有所提高,能节约大量的系统空间和运算时间。

关键词:关联规则;最大频繁集;期望长;最小支持度;数据挖掘
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2008)10-0113-03

An Improved Algorithm of Maximum Frequent Itemsets

LI Jing-wen, LIU Jun-feng, YAN Sui-jun, DENG Xiao-bin
(Guilin University of Technology, Guilin 541004, China)

Abstract: Association rules is the main technique for data mining, maximum frequent itemset is the key of association rules. The accuracy of the association rules and the quality of the efficiency come to a decision of whether knowledge rule apply or not. First elaborate the definition frequent itemset, frequent super itemset and association rules, analyze the thought and its shortage of the association rules. Then import expectation length based on probablity, put forward ELMFI algorithm. Finally, carry on analysis by using solid examples. This algorithm could economize a great deal of system space and the operation time, suitable for the large database.

Key words: association rules; maximum frequent itemset; expectation length; min_sup; data mining

0 引 言

随着信息技术的飞速发展和信息量呈几何级数的增长,如何有效地利用海量数据成为数据挖掘技术研究的核心和重点。数据挖掘技术不仅能对过去的数据进行查询,而且能够找出过去数据之间的潜在联系,进行更高层次的分析,以便更好地解决决策、预测等问题[1]。关联规则是数据挖掘的主要技术之一,自Rakesh Agrawal等人于1993年提出以来,已在分类设计、交叉购物、附加邮寄、目录设计等方面得到了广泛的应用。在其应用中,核心问题是频繁项目集的发现,Apriori算法是最经典的频繁项目集发现算法。许多学者[2~7]对该算法进行了改进,如DMFIA,FPST,FP-Tree,改进的FP-tree,close,ISS-DM等。这些算法在一定的程度上改善了Apriori算法的性能和效率,但还是不能突破Apriori算法性能的瓶颈问题,仍需要对数据库进行多次扫描或产生大量的候选集,从而消耗了大量的系统空间和运算时间。文中定义了期望长概念,提出了ELMFI算法(Expectation Length Maximum Frequent Itemset)。在此算法中,首先计算最大频繁集的期望长度,然后直接产生期望长度的候选项集进行验算,减少了查找和比较的运算量,节省了系统的存储空间,提高了最大频繁集的发现效率。

收稿日期:2008-01-08
基金项目:国家自然科学基金项目(40574002);广西自然科学基金项目(0448076)
作者简介:李景文(1971-),男,博士,副教授,研究方向为GIS空间分析与决策方法研究。

1 关联规则

1.1 关联规则

关联规则可形式化描述如下[8]:设 $I=(i_1,i_2,\cdots,i_m)$ 是由 m 个不同的属性组成的集合,给定一个数据库 D,其中每一个记录 T 是 I 中一组属性的集合,即 $T\subseteq I$,T 有唯一的标识符TID,若集合 $X\subseteq I$ 且 $X\subseteq T$,则记录 T 包含集合 X,一条关联规则就是一个形如 $X\Rightarrow Y$ 的蕴涵式,其中 $X\subseteq I$,$Y\subseteq I$,而且 $X\cap Y=\varnothing$。关联规则 $X\Rightarrow Y$ 成立条件是:

1)它具有支持度 s,即数据库 D 中至少有 $s\%$ 的记

录包含 $X \cup Y$。

2) 它具有置信度 c(confidence),即在数据库 D 中包含X的记录中至少有$c\%$ 的记录同时包含Y,习惯上将关联规则表示为 $X \Rightarrow Y(s\%, c\%)$,关联规则的挖掘问题就是生成具有用户指定的最小支持度(min_ sup)和最小置信度(min_ conf) 的关联规则。

同时满足最小支持度(min_ sup) 和最小置信度(min_ conf) 的规则称作强规则。项的集合称为项集(itemset)。包含 k 个项的项集称为k 项集。

如果项集满足最小支持度,则称它为频繁项集(frequent_ itemset)。

两个频繁项集 s 和t, 如果 $s \subset t$,则称 t 为s 的频繁超集(frequent_ superset)。

在所有的频繁项集中有一些项集没有频繁超集,被称为最大频繁项集。所有最大频繁项集的集合称为最大频繁集。

挖掘关联规则分两步进行[9]:

1) 找出所有频繁项集。这些项集出现的频繁性至少和预定义的最小支持计数一样。

2) 由频繁项集产生强关联规则。这些规则必须满足最小支持度和最小置信度。

第一步找出所有频繁项集是挖掘关联规则的核心问题。寻求频繁项集的问题可简化为寻求最大频繁集的发现算法。

1.2 最大频繁集的发现算法

随着关联规则的应用和众多学者的研究,现已提出了许多最大频繁集的发现算法,但总的说来,可分为自底向上(bottom_ up)算法和自顶向下(top_ down)算法。

1)自底向上(bottom_ up)算法[10]。

Apriori算法是一种典型的自底向上模式。通常分为两步:

a. 由频繁 k - 项集通过连接、剪枝生成$(k+1)$ - 项候选项集,剪枝时用到了性质(如果一个项集是非频繁的,那么它的所有超集都是非频繁的,不用再进行验证)。

b. 扫描数据库,计算$(k+1)$ - 项候选项集的支持度,生成频繁$(k+1)$ - 项集(k 从 1 到 $|I|$)。

反复执行这两步直至找到最大频繁集。此方法中,每一个频繁项集都在某一次验证中作为候选项被访问过,随着最大频繁项集长度的增大(如接近 $|I|$),验证次数将成指数级增长。

2)自顶向下(top_ down)算法[11]。

把单个的 n 项集($n = |I|$) 作为初始候选项集,通过每一次验证缩短候选项集的长度。当某个 k - 项集被验证为非频繁的,验证$(k-1)$ - 项子集是否是频繁项集。应用这种方法,在得到最大频繁集之前每个非频繁项集都被访问过,随着最大频繁项集的长度减小,验证次数也将呈指数级增长。

2 ELMFI 算法

由上面的分析可以看出,最大频繁项集都是通过候选项集逐层生成的,要么所有的频繁项集都被访问过,要么所有的非频繁项集都被访问过。因而要经过大量的查找和比较运算。然而事实上寻求频繁项集的问题可简化为寻求最大频繁项集的发现算法。只要找到最大频繁项集,也就可通过其子集生成频繁项集。为此,引入期望长来估计最大频繁项集的长度,从而直接产生该长度的侯选项集,进行最大频繁集的发现。

2.1 相关度量

设 $I = (i_1, i_2, \cdots, i_m)$ 是由 m 个不同的属性组成的集合,给定一个数据库 D,其中每一个记录 T 是I 中一组属性的集合,min_ sup 为指定的最小支持度,$|I|$ 为属性集 I 中的属性个数,count(I_i) 为属性 I_i 的个数,count(T) 为数据库 D 中记录个数。

1) 属性的覆盖率:

定义属性 I_i 在事物数据库中出现的概率,即

$$P(I_i) = \frac{\text{count}(I_i)}{\text{count}(T)}$$

2) 属性的总覆盖率:

定义各属性 I_i 在事物数据库中出现的概率之和,即

$$P_{\text{sum}}(I_i) = \sum_{i=1}^{n} p(I_i)$$

3) 属性的最大覆盖率:

定义事物数据库中各属性 I_i 覆盖率的最大值,即

$$P_{\max}(I_i) = \max_{I_i \in I}\{P(I_i)\}$$

4) 属性的最小覆盖率:

定义事物数据库中各属性 I_i 覆盖率的最小值,即

$$P_{\min}(I_i) = \min_{I_i \in I}\{P(I_i)\}$$

5) 期望长:

定义提取出的最大频繁项集的期望长度,即

$$E(L) = \text{INTEGER}\left(\frac{P_{\text{sum}}(I_i)}{(P_{\max}(I_i) + P_{\min}(I_i)) \otimes (|I| \times \text{min\_ sup})}\right)$$

式中 INTEGER 表示取整。其中 $\otimes$ 表示一种关系运算,在此取乘。

2.2 算法描述与分析

ELMFI 算法主要是避免了自底向上的由 1 - 项集产生 2 - 项集,再产生 3 - 项集,直到产生 k - 项集,或者自顶向下地由 n - 项集产生$(n-1)$ - 项集,再产生

$(n-2)$ - 项集,直到产生 k - 项集的这种逐层产生候选项集所带来的大量查找和比较运算。主要流程如下:

Step1: 计算 $P(I_i)$,$P_{sum}(I_i)$,$P_{max}(I_i)$,$P_{min}(I_i)$,$E(L)$;

Step2:生成 $E(L)$ 长度的候选项集;

Step3:根据性质(非频繁集的超集一定是非频繁集)对 Step2 产生的候选项集进行剪枝;

Step4:计算候选项集的支持度,如果大于最小支持度,则转到 Step5;否则转到 Step6;

Step5:利用自底向上算法继续寻找频繁集,直到不满足最小支持度为止;

Step6:利用自顶向下算法继续寻找频繁集,直到不满足最小支持度为止。

假设最大频繁项集长度为 k,其中 $E(L)=j$,自底向上算法、自顶向下算法和 ELMFI 算法所产生的候选项集数量是不同的。

用自底向上算法的候选集数量为 $C_n^1+C_n^2+\cdots+C_n^j+\cdots+C_n^k$;

用自顶向下算法的候选集数量为 $C_n^n+C_n^{n-1}+\cdots+C_n^j+\cdots+C_n^k$;

用 ELMFI 算法的候选集数量为 $C_n^j+C_n^{j+1}+\cdots+C_n^k$,$(j<k)$,或者 $C_n^j+C_n^{j-1}+\cdots+C_n^k$,$(j>k)$。

由上可以看出 ELMFI 算法少产生了 $C_n^1+C_n^2+\cdots+C_n^{j-1}$(或者 $C_n^n+C_n^{n-1}+\cdots+C_n^{j+1}$)个候选项集。

ELMFI 算法(发现频繁集)如下:

输入:数据集 D;最小支持数 min_sup

输出:频繁项目集 L

```
(0)  FOR all attributes I_i∈I DO BEGIN
(1)    FOR all transactions t∈D DO BEGIN
(2)      Calculate(P(I_i));//Calculate()为统计数量的函数
(3)      Calculate(Psum(I_i));
(4)      Calculate(Pmax(I_i));
(5)      Calculate(Pmin(I_i));
(6)      Calculate(E(L));
(7)    END
(8)  END
(9)  C_n = Genarate(L_E(L));//C_n 是 E(L)个元素的候选项集
(10) C_k = Cut_branch(Cl);//C_k 是对 C_n 剪枝后的候选项集
(11) FOR all transactions t∈D DO BEGIN
(12)   C_t = subset(C_k,t); //C_t 是所有 t 包含的候选项集元素
(13) FOR all candidates c∈C_t DO
(14)   c.count++;
(15) END
(16) L_k = {c∈C_k | c.countminsup_count}
(17) IF L_k IS NULL
(18)   Top_down();//Top_down()自顶向下发现最大频繁集的函数
(19) ELSE
(20)   Bottom_up();//Bottom_up()是自底向上发现最大频繁集的函数
(21)   L = ∪L_k;
```

2.3 实例分析

表 1 给出一个样本事物数据库,对它发现最大频繁项集,要求最小支持度 20%。

表 1　样本事物数据库

TID	Itemset	TID	Itemset
001	1	009	24
002	2	010	123
003	3	011	134
004	4	012	1234
005	5	013	1345
006	12	014	2456
007	16	015	12345
008	23		

计算 $P(I_i)$ 如下(见表 2),$P_{sum}(I_i)=36/15$;$P_{max}(I_i)=8/15$;$P_{min}(I_i)=2/15$;$E(L)=3$,生成 3 - 项集,因为属性 6 的支持度小于 20%,所以不予考虑,仅有 5 个属性。产生的候选集共 10 个分别为{123},{124},{125},{134},{135},{145},{234},{235},{245},{345},扫描数据库可得最大频繁集为{123},{134},支持度分别为 20%,26.7%。

表 2　属性覆盖率

I_i	1	2	3	4	5	6
$P(I_i)$	8/15	8/15	7/15	7/15	4/15	2/15

文中用 Bottom_up,Top_down, ELMFI 三种算法针对本实例发现最大频繁项集。对其进行了必要的剪枝后产生的候选集数量进行比较,如图 1 所示,从中可以看出 ELMFI 算法的优势。

3　结束语

ELMFI 算法先计算期望长,然后直接产生最接近长度的候选项集来发现最大频繁集,在此过程中采取从中间向两边拓展的方法,缩减了寻找频繁集的时间,避免了逐层产生候选项集带来的大量运算。最后通过

(下转第 119 页)

$10^{-2}x7$

方差贡献率 1.374%，累计贡献率 100%

前面三个主成分的方差贡献率接近 85%，说明前三个主成分已经包括了全部指标的大部分信息，则这三个综合指标在之后的研究中就可以用来取代原先的七个指标，降低了问题的复杂度。

3.3 模型分析

在第一主成分的表达式中，变量 $x1$、$x4$、$x5$、$x6$ 的系数较大，则这个主成分主要是用来反映短语字段是否以时间词或方位词结尾、是否以“的 + 名词”结构结尾、是否以“的 + 名词”结构结尾并且“的”前面没有动词、是否以“的 + 名词”结构结尾并且“的”前面是用双引号包含的这四个属性的，这反映了在当前考查的语料中，以时间词或方位词结尾的和以“的 + 名词”结构结尾的短语字段占所有短语字段的绝大多数。

在第二主成分的表达式中，变量 $x3$ 和 $x7$ 的系数较大，则这个主成分是用来反映短语字段关于动词和形容词的数量，即从短语字段中的谓语成分的有无来判断是否为短语字段。

在第三主成分的表达式中，变量 $x2$ 的系数较大，并且 $x3$、$x5$、$x6$ 的系数占一定的大小，则这个主成分是用来反映短语字段是否以介词打头并以方位词或名词结构结尾这个特征的。

4 结束语

将数学中多元统计的方法和语言学的理论结合起来，来分析复句中用以判断短语字段的因素，从而找出各因素之间的相关性和对判断的综合影响，降低了问题的复杂性，为今后实现短语字段的自动识别做好基础工作。由于材料的有限性和提取领域的局限性，数据在很大程度上会失真，还存在很多的不足，这也是今后努力完善的方向。

在完善的基础上，针对更多的更全面的语料，利用主成分分析所得的结果，欲利用提取出来的主成分采用聚类的方法[6]对语料库中的短语字段进行分类识别，努力实现短语字段的自动识别，为之后的复句的层次结构和关系标注做好准备工作。

参考文献：

[1] 邢福义. 汉语复句研究[M]. 北京:商务印书馆, 2001.

[2] 邢福义. 汉语语法学[M]. 长春:东北师范大学出版社, 1996.

[3] 罗积玉，邢　瑛. 经济统计分析方法及预测[M]. 北京:清华大学出版社,1987.

[4] 俞士汶. 计算语言学概论[M]. 北京:商务印书馆,2003.

[5] 刘　颖. 计算语言学[M]. 北京:清华大学出版社,2002.

[6] 谷　波，李济洪. 基于 COSA 算法的中文文本聚类[J]. 中文信息学报,2007(6):65－70.

(上接第 115 页)

实例比较分析证实此算法减少了候选集的数量，提高了运算速度，节省了系统空间，在一定程度上突破了传统算法的性能瓶颈，在预测分析和智能决策中具有一定的意义。

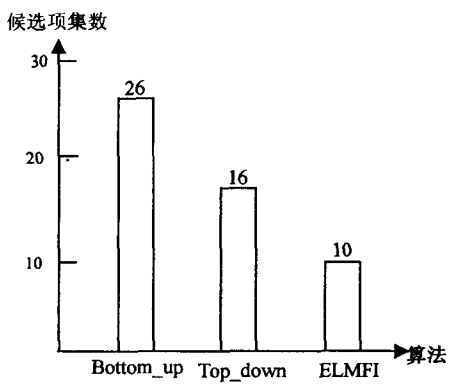

图 1　算法比较

参考文献：

[1] 毛国君，段立娟，王　实，等. 数据挖掘原理与算法[M]. 北京:清华大学出版社,2005.

[2] Kantardzic M. 数据挖掘[M]. 北京:清华大学出版社,2003.

[3] 王　丹，张　浩，陆剑峰. 针对高项频繁集的关联规则改进算法[J]. 计算机工程,2006,24(32):29－31.

[4] 王创新. 关联规则提取中对 Apriori 算法的一种改进[J]. 计算机工程与应用,2004(34):183－185.

[5] 李海军. 数据挖掘在 GIS 中的应用[D]. 北京:北京化工大学,2004.

[6] 邵峰晶，于忠清. 数据挖掘原理与算法[M]. 北京:水利水电出版社,2003.

[7] 章　艳，刘美玲，张师超，等. Apriori 算法的三种优化方法[J]. 计算机工程与应用,2004(36):191－193.

[8] Han J, Kamber M. 数据挖掘:概念与技术[M]. 北京:机械工业出版社,2001.

[9] 宋　雨，赵建利，王保义. 关联规则挖掘中最大频繁集的双向查找算法[J]. 华北电力大学学报,2005,32(2):67－71.

[10] 李清峰，杨路明，张晓峰，等. 数据挖掘中关联规则的一种高效 Apriori 算法[J]. 计算机应用与软件,2004,21(12):84－86.

[11] 刘桂庆，胡学钢，李　凯. CR 一种逆向的关联规则挖掘算法[J]. 微电子学与计算机,2004,21(9):83－86.