计算机科学2002Vol.29№.11

E-FP:一种挖掘多层高维频繁序列的高效算法

E-FP:An Efficient and Affective Method for Multi-level Multi-Dimensional Sequential Frequent Patterns Mining

何光辉　王蔚韬　郭　平　蒋　渝
(重庆大学计算机学院　重庆400044)

Abstract The classic sequential frequent pattern mining algorithms are based on a uniform mining support, either miss interesting patterns of low support or suffer from the bottleneck of pattern generation. In this thesis, we extend FP-growth to attack the problem of multi-level multi-dimensional sequential frequent pattern mining. The experimental result shows that our E-FP is more flexible at capturing desired knowledge than previous studies.

Keywords FP-growth, Sequential pattern, Passing threshold, Output threshold

一、引言

频繁序列挖掘一直是数据挖掘的一个活跃的研究课题。大部分的频繁序列挖掘算法是基于统一的最小支持度,如Apriori算法、SPADE、FP-growth等。但这将会丢失支持度较低的有效集合,或是遇到集合产生的瓶颈。除FP-growth之外,其余算法需要多次扫描数据库。为此,我们将FP-growth加以扩展,使其可以处理多层高维频繁序列。

二、问题的定义

多层高维频繁序列是一个具有广阔前景的研究课题,而且在实际生活中也是具有很大的实用价值。此处,我们先介绍一下相关的概念和符号。

2.1 序列P的支持度定义(sup(p))

$$\sup(p)=\frac{\text{包含 } P \text{ 的事务数目}}{\text{总事务数目}}$$

如果P的支持度满足$\sup(p)>\xi$,(ξ是人为设定的阈值),则序列P是频繁序列。

2.2 事件折叠窗口(Event folding window,记为W)

当一组事件发生在一个指定的时间段内,我们认为是发生在一个相同的事件折叠窗口,具体时间段的长度可以由用户自己决定。

2.3 多层高维频繁序列(multi-level multi-dimensional sequential frequent pattern,记为MMSFP)

在实际的事物数据库中通常包含项信息、地点信息和时间信息。下面我们给出多层高维频繁序列的定义。

定义1(MMSFP) 设事务数据库包含:

1)对每个事件折叠窗口(W),具有形式$\langle W_i,\text{description}_i\rangle$

2)对每个地点(称为'维',Dimension)具有形式$\langle D_j,\text{description}_j\rangle$

3)对每个集合中的项(items)具有形式$\langle A_k,\text{description}_k\rangle$

4)对每条记录I,具有形式$\langle I_m,\text{description}_m\rangle$

5)对一个事务数据库τ,包含一系列形如:$\langle W_i,T_j,\{D_m,\cdots,D_n\},\{A_k,\cdots,A_l\}\rangle$,其中$W_i$是事件折叠窗口的标志,$T_j$是事务的标志,$D_j$是维的标志,$A_k\in\tau$。

频繁序列P是事件折叠窗口、维、项A_k或一系列有联系的项和维及事件折叠窗口,$T_i,\cdots,T_j,\cdots,D_m,\cdots D_n,A_k,\cdots,A_l$。此处$A_k,\cdots,A_l\in\tau$。

三、相关工作

有关经典的频繁序列挖掘算法有Apriori,SPADE,ML_T2L1等。Apriori需要多次扫描数据库,尤其对一个具有n项序列I的数据库,在判断是否为频繁序列时需扫描n遍数据库,从而使时间复杂度高居不下。基于Aprori算法思想上的其他算法,也具有相似的缺点。如Adaptive-Apriori,SPADE的最大缺点是产生的候选集数量过多。ML_T2L1分别对每个概念层次进行挖掘,但ML_T2L1对其概念层次的每一层仍然采用了类似于Apriori的方法,从而时间复杂度较高且ML_T2L1不能处理多维序列。最近,新出现的FP-growth方法可以避免上述的缺点,但FP-growth也是基于统一的支持度的。所以,我们希望对FP-growth进行改进和扩展,使其能使用于产生跨概念层次的频繁序列。

四、扩展的FP-growth (Extend-FP)

本文在对FP-growth算法进行扩展的同时,保留了FP-growth的精髓。如FP树、基于FP树的集合分片、基于划分的分类树和控制策略。

4.1 在设计一个高效的多层高维频繁序列挖掘算法时需考虑的几个关键问题

1.为避免产生类似于Apriori的算法产生的瓶颈——多次扫描数据库。我们采用FP-growth算法来处理长序列、大型数据库以及产生适当的频繁序列候选集,因为无论频繁序列有多长,我们都可以将其压缩为一棵前缀树,然后只需对前缀树进行遍历即可。

2.对维信息和事件折叠窗口信息的处理,我们是将它们也看成项信息。在扫描数据库时,维信息和事件折叠窗口信息与项同样计数。当维和事件折叠窗口的发生频率大于给定的阈值时,也将其认为是频繁的。

3.如前所述,统一的支持度阈值可能会引起产生无用的候选集或支持度较低的有用候选集丢失。为此,我们考察实际生活中的事务数据库特征:

1)通过对实际生活中的数据库的观察可以看出,实际生活中的数据库已经分类:一类是普通性项目(如,电视机,冰箱等);一类是特殊性项目。而特殊性项目又可以分为两种:经常性项目(如,牛奶)和稀少性项目(如,钻戒)。例如在一家大型商场里的1000个事务中,"牛奶"发生200次仍然不能算是频繁的,因为"牛奶"是日常必备用品。而"钻戒"如果发生10次,则

可能认为是频繁的，因为钻戒是奢侈品。

2)支持度规定随着概念层次的不同而改变。对所有项(包括经常性的和特殊性的)设置两个阈值："通过阈值"和"输出阈值"。通常情况下，后者比前者高。例如，一个只包含 k 项的序列的出现频率比"输出阈值"高，则该序列是频繁序列且有机会参加生成(k+1)项序列的序列候选集。如果该序列的出现频率只是比"通过阈值"高，比"输出阈值"低，虽然它不是频繁序列但它仍有机会参加生成包含(k+1)项的序列。

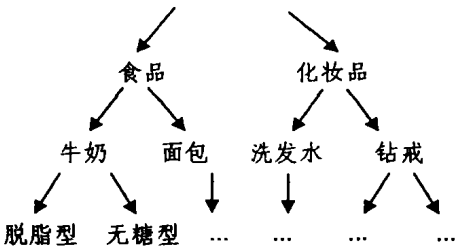

图1 项目概念层次图

例1 在事务数据库中，我们将其分为三层，其层次关系如图1。

表1 项目数量表

项目名称	数量
面包	60
洗发水	20
钻戒	2
牛奶	500

对本例子我们有以下说明：

·对一个单独的维或事件折叠窗口，我们有"维/事件折叠窗口"相应的"通过阈值"和"输出阈值"。

表2 阈值设定表

		通过阈值	输出阈值
普通项		15	25
特殊项	经常项目	300	400
	稀少项目	2	4

·对长度为 k(k>2)的集合，如果其所有项目都是普通的，则对此序列是否为频繁序列的判定可用长度为 k 的序列的"通过阈值"和"输出阈值"。通常情况下，k 值越大，对应的"通过/输出阈值"越小。

表3 项目阈值通过情况表

	通过阈值	输出阈值	频集
面包	√	√	√
洗发水	√	×	×
奶粉	√	√	√
钻戒	√	×	×

·如果长度为 k 的序列，包含特殊项(如奶粉、钻戒)我们将其发生的频率与长度为 k 的特殊项对应的"通过/输出阈值"相比较。如果特殊序列的项包含经常项(如"牛奶")和稀少项(如"钻戒")，则我们优先选择最低的阈值进行判定。

4.2 扩展的 FP-growth(E-growth)算法

Step1 扫描一次数据库，对每个事件折叠窗口、维和项进行计数，并将它们与相应的"通过阈值"和"输出阈值"作比较，从而判断是否为频繁序列以及能否参加下一次频繁序列的产生。

Step2 利用 FP-growth 的前缀树结构建立 FP 树，详细的方法请看文[4]。

Step3 挖掘 FP 树产生多层高维频繁序列。频繁序列可能有以下7种形式：

$\langle item_i,\cdots,item_j\rangle$；$\langle dim_i,\cdots,dim_j\rangle$，$\langle W_i,\cdots,W_j\rangle$；$\langle item_i,\cdots,item_k, dim_l,\cdots,dim_j\rangle$；$\langle item_i,\cdots,item_k, W_l,\cdots,W_j\rangle$；$\langle W_i,\cdots,W_k, dim_l,\cdots,dim_j\rangle$；$\langle item_i,\cdots,item_k, W_l,\cdots,W_j, dim_q,\cdots,dim_j\rangle$。

例2 设有一个系列事务如表4，其概念层次树如图2所示。我们采用 ML_T2L1算法的数据预处理模式，将时间窗口、维信息和项目用数字进行编码，结果如表5。

表4 事务项目表

TID	事件折叠窗口	销售地点	项目
100	50	南岸	{海尔台式机，海信纯平彩电，Epson 激光打印机，电脑配件}
200	60	江北	{台式机，电脑配件}
300	50	南岸	{海尔台式机，Epson 激光打印机}
400	60	南岸	{台式机，电脑配件}
500	50	江北	{纯平彩电}

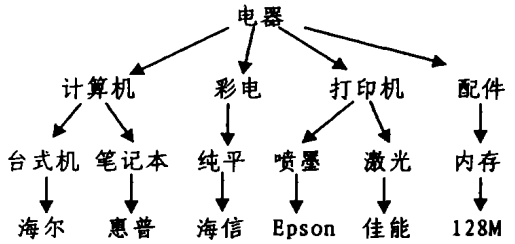

图2 项目概念层次树

在第一步中，我们扫描事务数据库一次，计数出每个项、维和时间折叠窗口的数目。将其数量及相应的阈值列在表6，7，8中。由表8中可以看出，基于从第一次数据库扫描得到的数据，我们可以得到以下信息：除纯平彩电(发生频率低于"通过阈值")外，其余项、维和事件折叠窗口都可以参加下一次频集产生；海信纯平彩电不是频繁序列，但可以参加下一次频繁序列的产生。即在第二步 FP 树中有9个节点。

根据文[4]的证明可知，事务数据库的所有符合条件的数据都可以通过编码加入 FP 树成为树的节点。

表6 项目编码表

TID	时间折叠窗口	销售地点	项目代码
100	50	3	{111,211,321,4}
200	60	4	{11,4}
300	50	3	{111,321}
400	60	4	{11,4}
500	50	4	{21}

表7 设定阈值

	通过阈值	输出阈值
较高层	2	3
中间层	2	2
较低层	1	2

表8 项目数量及相应阈值表

	数量	通过阈值	输出阈值
海尔台式机	2	1	2
纯平彩电	1	2	2
海信纯平彩电	1	1	2
佳能激光打印机	2	1	2
台式机	2	1	2
配件	3	2	3
南岸	3	2	2
江北	2	2	2
W1	3	2	2
W2	2	2	2

五、实验和性能评价

所有实验都是在800MHz，Intel Celeron256M 内存的台式机上进行，程序用 Borland Dephi5.0编写。事务总数据为1980条(随机产生)，平均事务长度为27，平均最大频繁序列长度为12。实验结果如图3所示。E-FP 方法与 ML-T2L1相比较有以下优点：

·只需要扫描数据库两次，而 ML-T2L1和 Apriori 却需要根据序列的长度确定。如果序列长度为 n，则需要扫描 n 次数据库。从而大大地节约了运行时间。

·E_FP 方法能够产生跨层的频繁序列，而 ML-T2L1算法不能产生。

·E_FP 方法可以处理多层高维的序列，而 ML-T2L1只能处理多层序列。

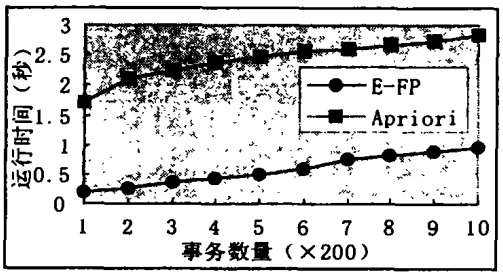

图3 实验结果比较

小结 本文针对多层高维频繁序列挖掘给出了一种新方法。由于该方法采用了 FP-growth 的结构，只需扫描三次数据库，从而大大节约了运行时间。同时我们又对 FP-growth 的统一支持度进行了改进，从而避免了丢失一些有用信息。实验结果表明，此方法优于以往的方法。下一步我们将考虑 FP 树的更新问题。

参 考 文 献

1 Agarwal R，et al. A tree projection algorithm for generation of frequent itemsets. In Journal of Parallel and Distributed Computing，2000

2 Agarwal R，Srikant S. Mining sequential patterns. In：Proc. 1995 Int. Conf. Data Engineering (ICDE'95)，Taipei，Taiwan，March 1995. 3～14

3 Han J，Fu Y. Discovery of multi-level association rules from large database. In：Proc. 1995 Int. Conf. Very large DataBases (VLDB'95)，Zurich，Switzerland，Sep. 1995. 420～431

4 Han J，Pei J，Yin Y. Mining frequents without Candidate Generation. In：Proc. 2000 ACM-SIGMOD Int. Conf. On Mangement of Data(SIGMOD'00)，dallas，TX，May 2000. 1～12

5 Liu B，Hsu W，Ma Y. Mining association rules with multiple minimum supports. In：Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining(KDD'99)，San diego，CA，Aug. 1999. 337～341

6 Pei J，Han J，Mao R. CLOSET：An Efficient Algorithm for Mining Frequent Closed Itemsets. In：Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery(DMKD'00)，Dallas，TX，May 2000. 11～20

7 Pei J，Han J，et al. PrefixSpan：Mining Sequential Patterns Efficiently by Prefix-Projected Pattern BibliographyGrowth. In：Proc. 2001 Int. Conf. Data Engineering (ICDE'01)，Heidelberg Germany，April 2001

8 Srikant R，Agrawal R. Mining generalized association rules. In：Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)，Zurich，Switzerland，Sept. 1995. 407～419

9 Wang K，He Y，Han J. Mining Frequent Itemsets Using Support Constraints. In：Proc. 2000 Int. Conf. Very Largo Data Bases (VLDB'00)，Cairo，Egypt，Sep. 2000. 43～52

(上接第49页)

结论 本文以粗集理论为基础给出了一种有效可行的对动态知识信息系统进行规则约简的算法。实验结果证明该算法是灵活可行有效的。由于该算法易于编程实现，适于处理大型知识系统。进一步的工作将是把该方法用于知识发现、数据挖掘的研究与应用方面。

参 考 文 献

1 Pawlk Z. Rough Sets：Theoretical Aspects of Reasoning about Data. Dordrecht：Kluwer Publishers，1991

2 Pawlk Z，et al. Rough Sets. Communications of the ACM，1995，38(11)：89～95

3 Pawlk Z. Rough set theory and its application to data analysis. Cybernetics and Systems，1998，29(9)：661～668

4 Ziarko W. Introduction to the special issue on rough sets and knowledge discovery. International Journal of Computational Intelligence，1995，11(2)：223～226

5 曾黄麟. 粗集理论及其应用. 重庆：重庆大学出版社，1996

6 刘请. Rough 集及 Rough 推理. 北京：科学出版社，2001