基于频繁链表的频繁集的挖掘算法

袁鼎荣　张师超
（广西师范大学数学与计算机科学学院　桂林541004）

An Algorithm of Mining Frequent Set Based on Frequent Link

YUAN Ding-Rong　ZHANG Shi-Chao
(College of Mathematics & Computer Science, Guangxi Normal University, Guilin 541004)

Abstract The problem of mining frequent set is a key issue in data mining. In this paper, a new method of mining frequent set based on the frequent link is proposed. The algorithm constructs alternate frequent link from the transaction, the alternate link is yielded by adding up the alternate frequent link which constructed by scanning the transaction database in proper order. The frequent link that comprises all the information is constructed with the frequent node which is selected according requirement. Our algorithm need to scan the transaction database only once and easy supervises the change of frequent set in order to guarantee the right of association rule.

Keywords Frequent link, Frequent set, Transaction database, Data mining

自从1989年提出KDD以来，关联规则的挖掘一直是人工智能及数据库领域关注的焦点，尤其是项目决策者渴求的制胜法宝。挖掘关联规则的前提是频繁集的挖掘，目前典型的频繁集挖掘算法以Appriori算法为代表。在Appriori算法的基础上提出了一些可行的方法，所有这些算法不外乎达到两个目的：①在穷举的基础上，设法删除对关联规则不太有效的频繁集，减少候选频繁集的数量，达到提高挖掘算法性能的目的。②直接挖掘最大频繁集，以最大频繁集为基础挖掘感兴趣的、有效的关联规则。但所有这些算法都以静态的事物数据库为基础，记录的是过去的数据，反映的是过去的规则，对新问题、新规则不能及时挖掘，供决策者快速作出反映；对特定时期、特定范围的关联规则也无能为力。且几乎所有的算法均以多次扫描数据库为代价，若数据库记录变化、支持度改变，新的频繁集的产生几乎从头开始。这样，既不经济实惠，也适应不了变化的客观世界（比如百货公司每日的交易数据），直接影响到频繁集的变化，进而影响所挖掘出的关联规则的正确与否。挖掘最大频繁集的方法也只是记录最大频繁集的有关信息，对于挖掘关联规则时所需频繁子集的有关信息无从保证。为此，我们提出一种基于频繁链表的频繁集挖掘算法，该算法不仅能及时反映频繁集的变化，记录所有频繁集的频繁信息，而且只需扫描一次事物数据库，进而为决策者提供及时准确的关联规则。

1　引言

设 $I=\{i_1,i_2,i_3\cdots,i_n\}$ 为项目集；$D=\{T_1,T_2,T_3\cdots,T_n\}$ 为事物数据库，$T_i\subseteq I,i\in\{1,2,3,\cdots,n\}$；对于给定的项集 A 和事物 T，若有 $A\subseteq T$，我们说事物 T 包涵了项集 A；对于规则 $x\Rightarrow y$，如果有 $x\subseteq I$、$y\subseteq I$、$x\cap y=\phi$，事物数据库 D 中包含 x 的事物有 $c\%$ 也包含 y，则称规则 $x\Rightarrow y$ 的可信度为 c；若 D 中有 $s\%$ 的事物包含 x 的同时也包含 y，即包含 $x\cup y$，则称规则的支持率为 s，支持率不小于 minsup 的项目集称为频繁集。关联规则的挖掘就是事物数据库中发现满足 $\sup(x\cup y)\geqslant$ minsup，confidence≥minconfidence 形式为 $x\Rightarrow y$，其中 $x\subseteq I$、$y\subseteq I$、$x\cap y=\phi$ 的规则。其中 $x\cup y$ 称为频繁集，$\text{confidence}(x\cup y)=p(x\cup y)/p(x)$ 称为规则的可信度，$\sup(x\cup y)=|x\cup y|/|D|$ 为规则的支持度。频繁集有如下性质：①若 A 是频繁集，任意子集是频繁集。②若 A 是非频繁集时，A 的任意超集是非频繁集。

任意超集是非频繁集的频繁集，是最大频繁集，记为MF。由于 $MF=|MF|/|D|\geqslant$minsup，$|D|$ 是变化的，MF 也就动态发展。原来挖掘结果为频繁集的，有可能成为非频繁集；原来为非频繁集的，完全有可能成为频繁集。若不及时反映这样的频繁集、新的关联规则无从发现。下面我们建立的挖掘算法既能及时发现新的频繁集，又能记录所有满足要求的频繁集的频率信息。

2　频繁链的设计与构造

链是由 n 个结点首尾连结而成的有限集合，我们构造频繁链结点如下：

Items	frequent	next

结点的 items 域为候选频繁项集，frequent 域为候选频繁集的支持度。若有 frequent/$|D|\geqslant$minsup，则项集为频繁集。next 为指针域，指向以下链结点：由于 $D=\{T_1,T_2,T_3,\cdots,T_n\}$，$T_i\subseteq I$，$I=\{i_1,i_2,i_3,\cdots,i_n\}$，因此事物 T 是由 K 个项构成的 K 项事物，它的任意项的任意组合都可能成为频繁集。例如：事物 $T_1=\{a,b,c,d\}$，T 的任意子集 $\{a,b\}$，$\{a,c\}$，$\{a,d\}$，$\{b,c\}$，$\{b,d\}$，$\{c,d\}$，$\{a,b,c\}$，$\{a,b,d\}$，$\{a,c,d\}$，$\{b,c,d\}$，$\{a,b,c,d\}$ 都可能成为频繁集，我们用作候选频繁集构成候选频繁结点产生2-项、3-项、4-项候选频繁链，且结点按 items 域的字典降序排列。

对另一3-项事物 $T_2=\{a,b,f\}$，又可产生2-项、3-项候选频繁链，然后两事物的频繁链相应归并为 L_2,L_3,L_4 依次类推。扫描事物数据库 $D=\{T_1,T_2,T_3,\cdots,T_n\}$ 依次得到各事物的候选频繁链归并得 D 的候选频繁链 $L_2,L_3,L_4,\cdots,L_K$，其中 K 为事物数据库中最长事物的长度。遍历各类链表，选取满足 frequent/$|D|>$minsup 的结点构成新链，成为该类的频繁链。链表中结点的 items 域即为所需发现的频繁集。若事物数据库新增 P 项事物，则只需扫描新增的事物，依次取得新增的事物各项频繁链并归并到各候选频繁链 $L_2,L_3,\cdots$ 中。这样，事物数据库的变化对频繁集的影响及时得到反映，对整个数据库项记录只需扫描一次，就记录下各候选频繁集的频繁信息，并可依 minsup 的变化，容易筛选得到所需的相应于频

繁集的频繁链表。频繁链表具有如下性质：

①各项频繁链结点按 items 域字典降低排列；

②高项频繁链结点的 items 域的任意子集在低项频繁链中有对应的频繁结点；

③低项链结点的 frequent 大于、等于高项链相应超集结点的 frequent 域。

根据频繁链的结构我们给出建立频繁链的算法。

3 FL-Generation 算法

本算法可分为三大过程：首先根据事物项 T 生成候选频繁链 $L_{T\text{-}K}$，然后将 $L_{T\text{-}K}$ 归并入候选频繁链 L_K，最后将候选频繁链经删减演化成频繁链，则该频繁链包含了用于关联规则挖掘的所有信息。

3.1 事物 T 的候选频繁链生成算法 $L_{T\text{-}K}$

输入：事物 $T=\{i_{j1},i_{j2},i_{j3},\cdots,i_{jk}\}$，$T\subseteq I$，且 $i_{j1},i_{j2},i_{j3},\cdots,i_{jk}$ 按字典降序排列。
输出：不同项长的候选频繁链 $L_{T\text{-}K}$（K 是小于等于 $|T|$ 的整数）。
方法：$K=2$；
while $k\leqslant|T|$ do
初始 $L_{T\text{-}K}$
for $i=1$ to $c_{|T|}$ do
N=new node；
N.items←依次从事物 T 中取 k 项组合；
N.frequent=1；
Insert($L_{T\text{-}K}$,N)；
$K++$；
End while.

3.2 将事物 T 产生的候选频繁链 $L_{T\text{-}K}$ 归并入候选频繁链 L_K 算法

输入：L_k，$L_{T\text{-}K}$
输出：L_K
方法：p1=$L_{T\text{-}K}$；
p2=L_K；
if *p1.items = *p2.items then {*p2.frequent++；p1=p1.next；p2=p2.next}
if *p1.items < *p2.items then p2 = p2.next
if *p1.items > *p2.items then{p1所指结点插入 p2所指结点之前；p1=p1.next；}
output L_k

3.3 候选频繁链删减演化成频繁链的算法

遍历各项频繁链 L_K（K 是小于事物最大长度的整数），删除满足 frequent/$|D|$<minsup 的结点，将各项 L_K 归并成频繁链表 FL。

4 应用举例

设 D 是来自百货店的拥有6个事物的给定数据库；D′是同一百货店另一时段产生的新的交易事物数据库，数据记录如下：

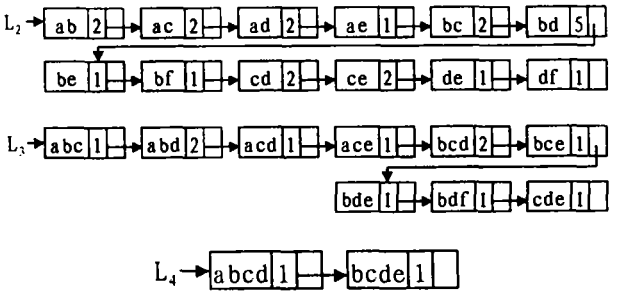
事物数据库 D

ID	Items
T1	ABD
T2	ABCD
T3	BD
T4	BCDE
T5	ACE
T6	BDF

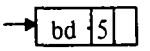
事物数据库 D′

ID	Items
T1	AEF
T2	CF
T3	BCF
T4	ABCDEF

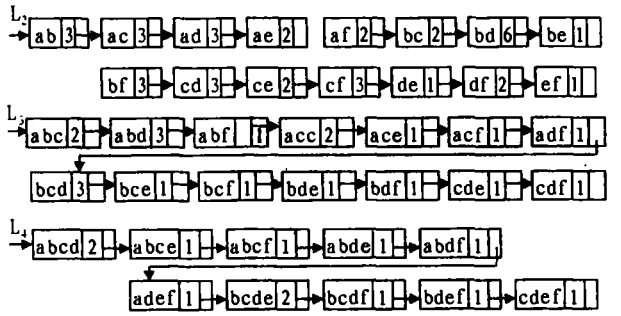
运用我们建立的频繁链表算法扫描 D，获取各事物的候选频繁链，归并后得：



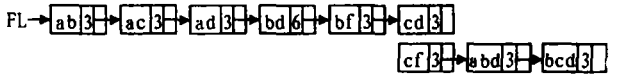
在候选频繁链的基础上，若指定 minsup=3，则得 D 的频繁链 FL 为：



若事物数据库在 D 的基础上增加 D′，则只需扫描 D′中的记录，形成候选频繁链，并归并到 D 的候选频繁链，得：



在候选频繁链的基础上，若指定 minsup=3，则得 D∪D′的频繁链 FL 为：



比较事物数据库 D 和 D∪D′，若指定 minsup=3，则频繁集有显著的变化；若指定 minsup=2，频繁集随着 D 的变化也有较大的变化。

小结 本文提出了一种全新的通过频繁链表挖掘频繁集的算法，该算法只需扫描数据库一次，并且能方便地监控变化的事物数据库，及时挖掘新环境下的频繁集，以确保关联规则的正确性。频繁链表中存储着所有频繁集的信息，为关联规则的挖掘提供了极大的方便。当然，该算法的候选集数量较穷举法有所减少，但还没有得到非常有效的控制。若对候选频繁链表采取一定的剪切技术，基于频繁链表的频繁集的挖掘算法不愧是一种理想的挖掘方法。

参 考 文 献

1 Lin, Dao-I, Kedem Z M. Pincer-Search: a new algorithm for discovering the maximun frequent set. In: Schek. H. J., Saltor, F., Ramos, I., et al, eds. Proc. of the 6th European Conf. on Extending Database Technology. Heidelberg: Springer-verlag, 1998. 105～119

2 Fayyad U, Stolorz P. Data mining and KDD: Promise and challenges. Future Generation Computer Systems, 1997, 13: 99～115

3 Shen Li, Shen Hong, Cheng Ling. New algorithms for efficient mining of association rules [J]. Information Sciences, 1999, 118 (4): 251～268

4 路松峰，卢正鼎. 快速开采最大频繁项目集. 软件学报，12(2)：293～297

5 王晓峰，王天然. 基于双空间的频繁项挖掘方法. 计算机科学，2002，29(4)：55～59

6 苏毅娟，严小卫. 一种改进的频繁集挖掘方法. 广西师范大学学报(自然科学版)，2001，19(3)：22～26