一种有效的不确定数据概率频繁项集挖掘算法 *

刘立新[1], 张晓琳[1], 毛伊敏[2]
(1. 内蒙古科技大学 信息工程学院, 内蒙古 包头 014010; 2. 中南大学 信息科学与工程学院, 长沙 410083)

摘 要: 针对 PFIM 算法中频繁概率计算方法的局限性,且挖掘时需要多次扫描数据库和生成大量候选集的不足,提出 EPFIM(efficient probabilistic frequent itemset mining) 算法。新提出的频繁概率计算方法能适应数据流等项集的概率发生变化时的情况;通过不确定数据库存储在概率矩阵中,以及利用项集的有序性和逐步删除无用事物来提高挖掘效率。理论分析和实验结果证明了 EPFIM 算法的性能更优。

关键词: 不确定数据; 可能世界; 期望支持度; 概率频繁项集
中图分类号: TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2012)03-0841-03
doi:10.3969/j.issn.1001-3695.2012.03.010

Efficient mining probabilistic frequent itemset in uncertain databases

LIU Li-xin[1], ZHANG Xiao-lin[1], MAO Yi-min[2]
(1. *School of Information Engineering, Inner Mongolia University of Science & Technology, Baotou Inner Mongolia* 014010, *China*; 2. *School of Information Science & Engineering, Central South University, Changsha* 410083, *China*)

Abstract: The way to calculate the frequentness probability in PFIM limited its applications, it needed to scan the database for many times and generated a large number of candidate sets. This paper proposed a new algorithm named EPFIM. First, the new method of calculating the frequentness probability made it easier to update frequentness probability of itemset, and could be adapted in more situations. Second, it used uncertain probability matrix to store the database in order to scan database less. In addition, the sequence of items and deleting unwanted transactions gradually improved efficiency of mining. Theoretical analysis andexperimental results show EPFIM performances better.

Key words: uncertain databases; possible word; expected support; probabilistic frequent itemset

频繁项集是关联规则、相关分析、序列模式、显露模式等许多重要数据挖掘任务的基本步骤,是数据挖掘中一个基础研究内容。传统数据频繁项集挖掘过程中,事物包含的项集是确定的,然而在现实世界中普遍存在不确定数据,如在经济、军事、物流、金融、电信、传感器网络等领域,不确定性数据扮演关键的角色[1]。由于数据的不确定性对挖掘结果产生了不可忽视的影响,不确定性数据挖掘算法越来越引起人们的关注。

目前,根据采用的模型,不确定数据频繁项集挖掘分为基于期望支持度模型和基于概率频繁模式模型两类。期望支持度是指项集在各事物中出现的概率和,其期望支持度大于指定的支持度阈值为频繁项集。采用期望支持度模型的算法有 U-Apriori 算法[2],它基于 Apriori 算法挖掘频繁项集,并使用数据修整技术剪掉原始数据集中低概率出现的项来提高效率;但鉴于修整阈值确难以确定,Chui 等人对 U-Apriori 进行改进,提出 UCP-Apriori 算法[3],并提出交替递减剪枝技术;UF-growth 算法[4]拓展了 FP-tree,每个节点除了存储项名称外还存储项的期望支持度和同一项在不同事物中以相同概率出现的次数,算法分为构建 UF-tree 和在 UF-tree 中挖掘两个过程。UD-FP-tree 算法[5]也通过拓展 FP-tree 挖掘频繁项集,是基于期望支持度模型研究特定领域项集的概率值计算方法。但是基于期望支持度模型的方法并没有考虑到支持度的概率分布情况,从而丢失了项集的支持度的置信度[6]。概率频繁模式挖掘(PFIM)是指在给定不确定数据库中,给定最小支持度和用户定义的频繁概率阈值,挖掘支持度大于最小支持度且频繁概率大于用户定义的阈值的项集。此模型能正确地反映项集支持度的置信度,由 Bernecker 等首先在 PFIM 算法[6]中提出。PFIM 算法采用递推方法计算项集的频繁概率,基于 Apriori 算法,连接阶段产生新的候选项集,剪枝阶段计算频繁概率和提取概率频繁项集。本文针对 PFIM 算法的缺陷,提出了 EPFIM 算法。

1 相关描述与问题定义

1.1 不确定数据模型

不确定性数据的种类较多,如关系型数据、半结构化数据、流数据、移动对象数据等。针对不确定数据建模也有很多种方法,存在许多与数据类型紧密相关的数据模型,其中最通用的模型是可能世界模型(possible world model),且其他模型最终都可以转换为可能世界模型[1]。

定义 1 不确定项集(uncertain itemset)。在事物 t_i 中,任意项集 x 以一定的概率出现,用$(x, P(x \in t_i))$表示。其中,x 表示该不确定项集的值,$P(x \in t_i)$表示其对应的概率值。

定义 2 不确定事物(uncertain transaction)。事物 t_i 中包含不确定项集,则事物 t_i 称为不确定事物。包含不确定事物

收稿日期: 2011-08-28; **修回日期**: 2011-10-09 **基金项目**: 国家自然科学基金资助项目(61163015);教育部"春晖计划"基金资助项目(Z2009-1-01024)

作者简介:刘立新(1983-),女(满族),内蒙通辽人,助教,CCF 会员,主要研究方向为数据挖掘、不确定数据管理(99liulixin@163.com);张晓琳(1966-),女,内蒙古包头人,教授,博士,主要研究方向为数据库理论与技术、信息安全;毛伊敏(1972-),女,新疆人,副教授,博士研究生,主要研究方向为数据分析、数据挖掘.

的数据库称为不确定数据库 T。

定义 3 可能世界实例。可能世界模型从一个不确定性数据库中演化出很多确定的数据库实例,称为可能世界实例。每一个可能世界实例由确定的事物组成。不确定项集 x 在 t_i 发生的概率为 $P(x\in t_i)$,此概率 $P(x\in t_i)$ 可以产生两个可能世界实例,一个实例是 x 存在 t_i 中,另一个实例是 x 不存在于 t_i 中。各元组的任一合法组合均构成一个可能世界实例 w。

每一个实例 w 有一个概率 $P(w)$,如果不确定事物是互相独立的,则 $P(w)$ 等于实例内的元组概率乘积与实例外的元组概率乘积,$P(w)=\prod_{t\in T}(\prod_{x\in t}P(x\in t)\times\prod_{x\notin t}(1-P(x\in t)))$,且所有可能世界实例的发生概率之和为 1。

由定义可知,事物数量和事物中不确定项集数量决定可能世界实例的数量。如果仅考虑事物中不确定项集互相独立的情况,可能世界实例的数量是不确定数据库规模的指数倍,这是不确定数据管理技术面临的最大难点[7]。

1.2 概率频繁模式挖掘

在确定数据的频繁模式挖掘中,频繁项集是指支持度大于给定最小支持度阈值的项集。然而在不确定数据中,事物中项集的支持度是不确定的,不能仅由一个值来表示,而应由离散的概率分布函数决定,概率频繁模式挖掘解决了此问题。

定义 4 概率频繁项集。在不确定数据库 T 中,项集是频繁的概率大于给定的阈值 τ,此项集称为概率频繁项集;概率频繁项集挖掘就是在不确定数据库中挖掘支持度大于最小支持度 minSup,且其概率大于用户定义的阈值 τ 的项集。

定义 5 支持度概率。在不确定数据库 T 和它的可能世界 W 中,项集 x 的支持度概率 $P_i(x)$ 是指项集 x 的支持度是 i 的概率。

$$P_i(x)=\sum_{w_j\in W;(S(x,w_j)=i)}P(w_j)$$

其中,$S(x,w_j)=i$ 表示在可能世界实例 w_j 中项集 x 的支持度为 i。项集 x 的支持度概率 $P_i(x)$,i 的取值是 $0\leqslant i\leqslant|T|$,则有 $\sum_{0\leqslant i\leqslant|T|}P_i(x)=1$。支持度概率是以概率函数分布的,叫做概率分布函数。同理,$P_{\geqslant i}(x)=\sum_{w_j\in W;(S(x,w_j)\geqslant i)}P(w_j)$ 表示项集的支持度大于 i 的概率。此支持度概率可转换为 $P_{\geqslant i}(x)=\sum_{S\subseteq T,|S|\geqslant i}(\prod_{t\in S}P(x\subseteq t)\cdot\prod_{t\in T-S}(1-P(x\subseteq t)))$。

定义 6 频繁概率。不确定数据库 T 中的不确定项集 x,$P_{\geqslant i}(x)$ 表示 x 的支持度至少是 i 的概率,即 $P_{\geqslant i}(x)=\sum_{k=i}^{|T|}P_k(x)$。如果给定最小支持度 $\text{minSup}\in\{0,\cdots,|T|\}$,那么项集 x 的频繁概率为 $P_{\geqslant\text{minSup}}(x)$,它是指项集 x 的支持度至少是 minSup 的概率。

定义 7 概率矩阵。不确定数据库 T,有 n 个事物,m 个不同的项集,经 $f:T\to R$ 转换为概率矩阵 R。其中,$R=f(T)=(r_{ij})_{n\times m}(i=1,2,\cdots,n;j=1,2,\cdots,m)$,这里 r_{ij} 是指项集 I_j 在 T_i 中出现的概率。

2 算法描述

2.1 数据存储

在确定数据库的频繁项集挖掘中,把数据存入 0-1 矩阵中,转换为在矩阵中的挖掘,并且利用向量运算提高效率。本文借鉴此思想,把不确定数据库中的概率信息存入矩阵,得到概率矩阵;同时,为了提高挖掘时的效率,还增加事物长度列 T_length[ ] 和标志列 flag[ ]。此方法的优点在于:利用 Apriori 算法求解概率频繁项集时,转换为利用概率矩阵求解,避免多次扫描数据库时的 I/O 操作;可以利用向量的卷积及内积运算求解项集的频繁概率,频繁概率的求解与分析见 2.2 节。

根据定义 7,数据库 T 经过一次扫描后,就可以在 f 的作用下映射成 $n\times m$ 概率矩阵 R,同时可以得到 T_length[i] 的内容。表 1 给出一个不确定数据库实例,表 2 给出其对应的概率矩阵和 T_length[i] 值。

表 1 不确定数据库实例

tid	transaction				
t_1	(a,0.8)	(b,0.2)	(d,0.5)	(f,1.0)	
t_2	(b,0.1)	(c,0.7)	(d,1.0)	(e,1.0)	(g,0.1)
t_3	(a,0.5)	(d,0.2)	(f,0.5)	(g,1.0)	
t_4	(d,0.8)	(e,0.2)	(g,0.9)		
t_5	(c,1.0)	(d,0.5)	(f,0.8)	(g,1.0)	
t_6	(a,1.0)	(b,0.2)	(c,0.1)		

表 2 不确定数据库实例对应的概率矩阵

tid	a	b	c	d	e	f	g	T_length[ ]
t_1	0.8	0.2	0	0.5	0	1.0	0	4
t_2	0	0.1	0.7	1.0	1.0	0	0.1	5
t_3	0.5	0	0	0.2	0	0.5	1.0	4
t_4	0	0	0	0.8	0.2	0	0.9	3
t_5	0	0	1.0	0.5	0	0.8	1.0	4
t_6	1.0	0.2	0.1	0	0	0	0	3

2.2 有效计算频繁概率

PFIM 算法采用了文献[8]中概率 top-k 查询的方法,利用递推思想来重叠计算项集的频繁概率;文献[7]中也应用了此方法计算概率。若项集 X 在 j 个事物中至少出现 i 次的概率 $P_{\geqslant i,j}(X)$,则

$$P_{\geqslant i,j}(X)=\sum_{S\subseteq T_j;|S|\geqslant i}(\prod_{t\in S}P(X\subseteq t)\cdot\prod_{t\in T_j-S}(1-P(X\subseteq t)))$$

进一步划分为子问题,得公式:

$$P_{\geqslant i,j}(X)=P_{\geqslant i-1,j-1}(X)\cdot P(X\subseteq t_j)+P_{\geqslant i,j-1}(X)\cdot(1-P_j(X\subseteq t_j))$$

因为有 $P_{\geqslant 0,j}=1$ 和 $P_{\geqslant i,j}=0(i>j)$,所以,可以通过反复递推叠加运算求得项集 X 的频繁概率。

采用递推的方法计算频繁概率不适合用于项集概率发生变化时的情况。本文采用向量的卷积方法求解项集的频繁概率。考虑到对于 i 个事物和多项式 $F^i=\prod_{t\in\{t_1,\cdots,t_i\}}((1-P(x\in t))+P(x\in t)x)=\sum_{j\in\{0,\cdots,i\}}c_jx^j$,在此多项式展开式中,$x^j$ 的系数 c_j 就是项集 x 在 i 个事物中发生 j 次的概率值,即项集 x 在 i 个事物中的支持度是 j 的概率值。

根据卷积的原理,可以用多个向量的卷积来求 c_j 的值,即 $(1-P(x\in t_1),P(x\in t_1))$,$(1-P(x\in t_2),P(x\in t_2))$,$\cdots$,$(1-P(x\in t_n),P(x\in t_n))$ 的卷积运算。

与 PFIM 算法中的频繁概率计算相比较,此方法更适合于项集的频繁概率发生变化时。例如在数据流中,旧事物流出窗口时用解卷积除去其对项集的频繁概率的影响,新事物流入窗口时用卷积增加其对项集的频繁概率的影响,但对于 PFIM 算法,则需要重新开始递推计算新的频繁概率值。以项集 D 为例,应用卷积方法计算项集 D 的频繁概率,图 1 给出了项集 D 的概率分布函数,图 2 给出了项集 D 的频繁概率。

2.3 挖掘优化策略

PFIM 算法中在用 Apriori 算法求概率频繁项集时,时间开销主要在生成候选概率频繁 k-项集和候选概率频繁项集的频繁概率计算过程。确定数据库 Apriori 算法通过项集的顺序性

来提高连接时候选概率频繁 k-项集的生成效率[9],此方法在不确定数据库仍然有效。此外,本文通过将数据存储于矩阵,并压缩未来迭代扫描的事物数目(即逐步压缩概率矩阵)的方法减少候选概率频繁项集的频繁概率计算时需要扫描的概率矩阵的数据量。

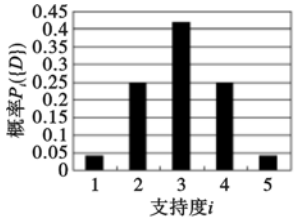

图1　项集D的支持度概率分布

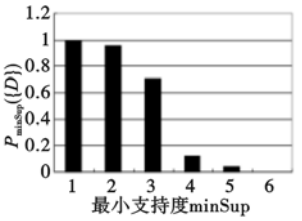

图2　项集D的频繁概率

性质 1[6]　概率频繁项集的子集都是概率频繁的,非概率频繁项集的超集是非概率频繁的。

性质 2[10]　不包含任何频繁 k-项集的事物,不可能包含任何频繁($k+1$)-项集。

基于以上性质,可以通过如下策略进行优化:

策略 1　不确定数据库中,任意 k-项集的支持度概率与规模小于 k 的事物无关,因此可以删除规模小于 k 的事物,用 flag[ ]标记。

策略 2　当一个事物不包含长度为 k 的概率频繁项集时,则必然不包含长度为 $k+1$ 的概率频繁项集,因此可以在生成 $k+1$-概率频繁项集之前先删除这样的事物,用 flag[ ]标记,以减少下次扫描的时间。

通过使用策略 1 和 2 对概率矩阵进行压缩,在确定候选 k-概率频繁项集的概率支持度的同时,删除数据库中事物项不大于 k 的事物和事物中不包含任何候选 k-项集的事物。通过多轮扫描矩阵,大量事物被删除,从而节省了扫描时间。而原算法每次要彻底扫描整个原始数据库,这消耗了大量的时间。

2.4　概率频繁项集挖掘算法

算法把不确定数据的概率信息存储在矩阵中;采用 Apriori 挖掘概率频繁项集,并采用优化策略提高效率。

算法　EPFIM

输入:不确定数据库;最小支持度计数阈值 min_sup;概率阈值 τ。
输出:不确定数据库中的概率频繁项集。
a)扫描不确定数据库,转换为概率矩阵;
T_length[j]数组记录相应事物长度;flag[i]数组作为标志列,初始值为全 0;
b)计算每个 1-项集的频繁概率,得到 1-概率频繁项集,除去非 1-概率频繁项集;
for 每个事物 T_j do
　if(T_length[j] = =1) flag[j] =1; //不再参与后续运算
　k =2;
c)由(k-1)-概率频繁项集得到候选 k-概率频繁项集;
d)for 每个事物 T_i do
{if (T_length[i] < =k) Flag[i] =1;
　if(k-概率频繁项集 $\notin T_i$) Flag[i] =1; }
使用内积和卷积计算候选 k-概率频繁项集的频繁概率,生成 k-概率频繁项集;
e)if (k-概率频繁项集不为空)
　k ++; 转到 c);

3　实验结果与分析

实验环境为 Pentium 2.0 GHz CPU,1 GB 内存、Windows XP 操作系统的 PC 机。C ++语言实现了 PFIM 和 EPFIM 算法,并在 VC ++ 6.0 环境中运行。实验数据采用真实的实验数据集 accidents(http://fimi.cs.helsinki.fi/data/),并设定实验中项集 item 存在于事物中的概率服从(0-1)均匀分布。

实验 1　取最小支持度为 0.1,频繁概率为 0.9,测试随着不确定数据事物大小的改变,运行时间的变化情况。由图 3 得出,随着需要处理的事物数增加,EPFIM 和 PFIM 算法运行时间均增加。但是 EPFIM 比 PFIM 运行速度快,因为 PFIM 算法需要多次扫描数据库,而 EPFIM 算法是扫描概率矩阵,且挖掘时采用的优化策略也提高了挖掘效率。

实验 2　频繁概率为固定值 0.9,最小支持度是变化的,测试随着最小支持度改变,运行时间的变化情况。由图 4 得出在支持度较小时,两算法的运行时间都较长,因为要处理大量概率频繁项集;随着支持度增加,两算法的运行时间都减少。这是因为支持度增加,符合条件的概率频繁项集减少,处理时间变短,最后趋于不变。

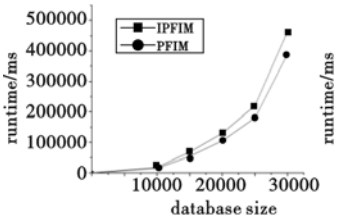

图3　事物逐渐增大时算法运行时间

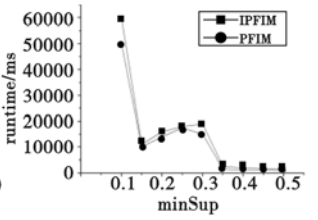

图4　最小支持度变化时算法运行时间

4　结束语

不确定数据频繁模式挖掘中,首先由 PFMI 算法提出概率频繁模式模型能正确地反映项集支持度的置信度。本文提出的 EPFIM 算法采用概率矩阵记录不确定数据库,转换为在概率矩阵中挖掘,减少数据库扫描次数;采用向量的内积和卷积的方法求项集的频繁概率,与 PFIM 中相比能更好地支持数据流等项集概率变化时的情况;挖掘时采用项集的有序性和逐步删除无用事物的优化策略来减少产生候选集的数量。理论分析和实验结果证明了 PFMI 算法的性能。

参考文献:

[1] 周傲英,金澈清,王国仁. 不确定性数据管理技术研究综述[J]. 计算机学报,2009,32(1):1-16.

[2] CHUI C K, KAO Ben, HUNG E. Mining frequent itemsets from uncertain data[C]//Proc of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin: Springer-Verlag, 2007: 47-58.

[3] CHUI C K, KAO Ben. A detrimental approach for mining frequent itemsets from uncertain data[C]//Proc of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin: Springer-Verlag,2008:64-75.

[4] LEUNG C K S, CARMICHAEL C L, HAO Bo-yu. Efficient mining of frequent patterns from uncertain data[C]//Proc of the 17th IEEE International Conference on Data Mining Workshops. 2007:489-494.

[5] 高聪,申德荣,于戈. 一种基于不确定数据的挖掘频繁集方法[J]. 计算机研究与发展,2008,45(z1): 71-76.

[6] BERNECKER T, KRIEGEL H P, RENZ M, *et al*. Probabilistic frequent itemset mining in uncertain databases[C]//Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM Press,2009:119-127.

[7] 王爽,杨广明,朱志良. 基于不确定数据的频繁项查询算法[J]. 东北大学学报,2011, 32(3):344-347.

[8] YI Ke, LI Fei-fei, KOLLIOS, *et al*. Efficient processing of top-k queries in uncertain databases[C]//Proc of the 24th International Conference on Data Engineering. Washington DC:IEEE Computer Society,2009:1406-1408.

[9] WITTEN I H, FRANK E. Data mining: practical machine tools and techniques[M]. 北京:机械工业出版社,2006:202-204.

[10] HAN Jia-wei, KAMBER M. 数据挖掘概念与技术[M]. 范明,孟小峰,译. 北京:机械工业出版社,2007:155 -156.