高效隐私保护频繁模式挖掘算法研究

程舒通[1,2] 徐从富[1] 但红卫[1]

(浙江大学计算机科学与技术学院 杭州 310027)[1] (杭州广播电视大学信息工程学院 杭州 310012)[2]

摘 要 阐述了隐私保护数据挖掘的目标,即在获取有效的数据挖掘结果的同时,满足用户对隐私保护的要求。针对个体用户及组织用户的隐私保护,论述了不同的方法,并归纳出隐私保护数据挖掘中所采用的两种主流算法。改进了高效隐私保护关联规则挖掘算法(EMASK)中需要完全的数据库扫描并且进行多次比较操作的弊端,提出了基于粒度计算的高效隐私保护频繁模式挖掘算法(BEMASK)。该算法将关系数据表转换成面向机器的关系模型,数据处理被转换成粒度计算的方式,计算频繁项集变成了计算基本颗粒的交集。特别是数据的垂直Bitmap表示,在保证准确性不降低的情况下,一方面减少了I/O操作的次数,另一方面较大地提高了效率。

关键词 数据挖掘,隐私保护,频繁模式,知识粒度

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.4.039

Research on Efficient Privacy Preserving Frequent Pattern Mining Algorithm

CHENG Shu-tong[1,2] XU Cong-fu[1] DAN Hong-wei[1]

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)[1]

(Department of Information Science and Technology, Hangzhou Radio and TV University, Hangzhou 310012, China)[2]

Abstract This paper elaborated the goal of privacy preserving data mining, that is to satisfy the demand of users for privacy protection as we acquire the mining results of effective data mining. For privacy protection to individual users and group users, this paper discussed different methods, and summed up two main algorithms in data mining of privacy preservation. Since efficient mining associations with secrecy konstraints(EMASK) needs full database scan and many comparison operations, the author came up with efficient mining associations with secrecy konstraints which is based on Bitmap computation(BEMASK). It transforms relational data form into relational model for machine, data processing is converted into granular computing method and calculation of frequent item-sets is turned into computing the intersection set of basic particles. Especially the vertical representation of Bitmap, under the condition of ensuring that accuracy is not reduced, on one hand reduces the number of I/O operations, on the other hand, greatly improves the efficiency.

Keywords Data mining, Privacy preserving, Frequent pattern, Knowledge granularity

1 引言

随着信息与网络技术的飞速发展,数据挖掘在揭示各类隐藏的模式或知识的层面上,也相应地暴露了它由于人为不正确使用产生的缺点,使得一些个人隐私或者共同隐私变得容易侵犯。隐私问题正在成为一个潜在、公共的误用问题呈现在大众面前,向现代科学发起了挑战,如消费者在使用信用卡、会员卡、医疗保健卡和电子商务网站购物等活动中,个人信息很容易被公司或企业获取,更为严重的是由于用户的无意识行为,数据的二次使用使得隐私数据提高了识别性与解释性,直接导致了用户对个人信息的不可控制。因此,隐私保护和信息安全成为数据挖掘中一个迫切需要解决的问题,越来越引起人们的广泛关注。

2 隐私保护数据挖掘相关研究

2.1 隐私的定义

一般来说,数据挖掘中提到的隐私主要分为两类:个人隐私与共同隐私。与个人相关又不愿被暴露的信息可以称为个人隐私,如身份证信息、医院的就诊记录、薪资等。包含一个群体共同表现出来的同时也不愿被公众所获取的信息或模式,称为共同隐私,如网站中相似的浏览路径、公司薪资分布等。隐私保护数据挖掘的目的是通过在得到有效的数据挖掘的基础上,同时实现个人隐私或共同隐私的保护。

2.2 隐私保护数据挖掘的目标

隐私保护数据挖掘的主要目标是在获取有效的数据挖掘结果的同时,满足用户对隐私保护的要求。一方面,我们需要保护用户敏感的属性和特征信息(Protecting Individual Privacy),在不泄漏用户个人私有属性和特征数据的前提下得到挖掘结果。这种保护可以看成是一种基于记录级上的保护,研究的重点是对事务和项集的保护。另一方面,在商业竞争中,为了不使对手获得有价值的决策依据,我们需要对数据库进行修改,以使得挖掘者不能从中发现有价值的机密,保护数据挖掘和统计分析之后的结果(Protecting Organization Privacy),这在基于组织间或者商业企业间进行数据挖掘和商业

到稿日期:2014-05-20 返修日期:2014-08-21 本文受国家自然科学基金(61272303),杭州广播电视大学科研课题(HKYYB-2013-1)资助。

程舒通(1976—),男,硕士,副教授,主要研究方向为数据挖掘、人工智能,E-mail:chengshutong@21cn.com;徐从富(1969—),男,博士,副教授,主要研究方向为人工智能、智能CAD、数据挖掘、知识发现、数据融合;但红卫(1981—),男,硕士,主要研究方向为人工智能、数据挖掘。

分析的过程中尤为重要。

(1)保护个体用户隐私(Protecting Individual Privacy):这是一种属性、特征和记录级上的隐私保护。在数据库文件中,类似于ID、姓名、联系方式、家庭住址和个人兴趣等用户具体的敏感属性数据作为用户的隐私应该被保护。基于用户敏感原始数据的隐私保护数据挖掘方法主要使用变换隐藏和随机的方法进行更改,使得真实的属性和记录值不能被其他用户直接存取,同时保证在进行数据挖掘时不能得到足够的信息来对原始数据进行重构,且不能准确估计原始数据具体属性值,以此保护用户的私有数据和属性不被泄漏。

(2)保护组织用户隐私(Protecting Organization Privacy):这是一种知识和模式级上的隐私保护,其要求不仅仅是保护原始数据的属性和特征不被泄漏,而且一些重要的策略模式和数据挖掘之后的结果同样不能泄漏,在商业领域,这些模式被认为是能够提升企业竞争力的重要知识和商业秘密,为此,这些私有模式和知识同样也必须得到很好的保护。基于组织用户隐私保护的数据挖掘方法主要使用数据采样和清洗等方法来对原始数据库进行变换,以保证数据集中重要的统计属性不能被其他用户得到,保护数据挖掘之后的模式和知识不被泄漏,以此保护用户的隐私。

本质上,这两个方面是相互关联、相互照应的,所使用的途径很多是相同的:一方面,为了保护敏感的数据,需要对数据库记录内容进行修改和处理,使得敏感数据不被他人直接获取;另一方面,因为无法猜度用户所采用的数据挖掘方法,所以我们需要修改原始数据,使得挖掘出这种敏感知识的概率大大降低,以此隐藏希望保护的知识。

目前,在隐私保护数据挖掘中所采用的算法主要分为以下两类。

(1)基于随机的方法(Randomization-Based Techniques):根据具体应用的需要对原始数据库中的数据记录进行模糊化处理,同时保持数据的统计特性,将经过处理的数据库进行数据挖掘,通过对数据的原始统计特性的估计来得到较为准确的处理结果,同时又不泄漏用户的原始数据。利用这样的方法,一方面可以保护数据的隐私不被泄漏,另一方面保证了数据的可用性,这是一种数据扰乱的方法[1,2]。

(2)基于安全多方计算技术的方法(Secure Multi-Party-Based Techniques):在多个结点之间进行协同工作和数据挖掘时,为了保护用户的隐私,每个结点只提供经过加密处理之后的转换数据,各个站点经过协同计算完成数据挖掘工作,使得其它站点无法从这些经过加密的数据和最后的结果中对其它站点的原始数据进行推测[3,4]。

3 粒度计算概述

20世纪80年代初,由Z. Pawlak带领的团队提出了粗糙理论,从一种全新的角度审视知识,认为知识是有粒度的,颗粒状的知识导致了知识表示的粗糙性[5]。Zadeh在讨论模糊信息粒度理论时,提出粒度计算的概念,认为粒度计算是模糊信息粒度理论的超集,覆盖了所有有关粒度的理论、方法论、技术和工具的研究[6]。

近些年来,粒度计算倡导对现实问题多粒度、多角度、多层次的理解,对知识的描述更具备科学性,从而成为数据挖掘领域重要的研究课题,引起人们广泛的关注与探讨,许多粒度计算的模型和方法被提出并得到深入的研究[7,8]。

梳理国内相关文献,不少专家针也对粒度计算提出了新的想法与改进意见,力求通过这种较新的智能计算的理论和方法来处理不精确、不完全、不一致、不可靠和不确定的知识。

王磊等从矩阵的视角探讨知识粒度粗糙度和属性重要度等概念的计算方法,揭示出知识粒度与等价关系矩阵之间的关系,在提出知识粒层次结构的基础上,进一步探讨了属性增删时知识粒度的变化规律[9]。

项海飞提出了一种基于互信息粒度的相对约简模型,该模型利用互信息度量决策系统中的条件属性,将互信息对属性的度量映射到布尔矩阵,并能得到完备的相对约简结果[10]。

4 BEMASK算法

本文的主要工作是研究基于粒度计算的高效隐私保护频繁模式挖掘,提出了BEMASK算法。该算法对S. Agrawal等提出的EMASK算法进行了改进,用粒度方式进行隐私保护频繁模式挖掘,关系数据表被转换成面向机器的关系模型,利用这种模型,数据处理被转换成粒度计算的方式,计算频繁项集变成了计算基本颗粒的交集,以此在保证准确性不降低的情况下较大地提高了效率。

4.1 EMASK算法简介

S. J. Rizvi和J. R. Haritsa于2002年提出了隐私保护关联规则挖掘算法(Mining Associations with Secrecy Konstraints,MASK),这种算法主要基于"market-basket"事务数据库,以购物篮数据为原型,采用了著名的Apriori算法,先产生候选k-项集,再产生k-频繁项集,最后生成强关联规则[11]。其由于转换方法简单,从心理学角度满足了顾客隐私保持的需求。

但是,这种方法在计算中引起了两个问题:

(1)数据库的稠密。因为在转变时1和0以同样的概率进行翻转,造成了数据库中错误的1产生,导致了数据库的密度显著变大。为此,在扭曲的数据库中计算项集需要更多的处理,导致了效率的降低。

(2)计数方式低效。虽然在MASK算法中利用了高效的计算项集的优化方法,但是这种计算方法同样需要对数据库中的多个项集进行计算,导致了效率的下降。

为此,S. Agrawal、V. Krishnan和J. R. Haritsa提出了EMASK(Efficient MASK)算法[12],这种算法与MASK算法的不同在于:

(1)在数据库转变时,1和0分别以概率p和q进行转换(见图1),例如,在1项集的计算时,$M=\begin{bmatrix} p & 1-q \\ 1-p & q \end{bmatrix}$;

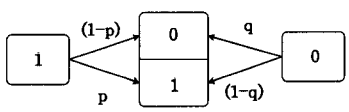

图1 原始数据变换概率

(2)在项集计数时,本文用到了集合计算的方法。给定全集U、子集A和B,$N(\overline{A}\cap B)=N(B)-N(A\cap B)$,这里,$N$表示元素的数量。利用这种集合计算方式,可以简化项集计数。例如,对二项集进行计算,为了计算原有数据库中的11的支持度,我们需要计算变换后数据库的00,01,10和11。而利

用这种集合计数方式，我们只需要对 $N(A\cap B)$进行计数，同时利用已经得到的结果 $N(A)$和 $N(B)$，就可以分别求出 $N(\overline{A}\cap B)$，$N(A\cap\overline{B})$和 $N(\overline{A}\cap\overline{B})$，加快了计算的效率。

虽然 EMASK 算法提高了计算的效率，但是，该算法所依据的 Apriori 算法需要完全的数据库扫描并且进行多次比较操作，即使利用了 hash-table 的结构，也需要较高的花费。但由于其减少了因计算失真所产生的各个环节的时间与空间消耗，在效率上稍做改进，仍然不失为一种好的隐私保护数据挖掘算法。

为此，本文用粒度计算的思想对 EMASK 算法进行改进，利用粒度方式将关系数据表转换成面向机器的关系模型。利用这种模型，数据处理被转换成粒度计算的方式，计算频繁项集变成了计算基本颗粒的交集。特别地，为了有效地计算项集的支持度，本文算法使用了数据的垂直 Bitmap 表示(vertical bitmap)。利用这种方法在保证准确性不降低的情况下对 EMASK 算法进行了改进，一方面减少了 I/O 操作的次数，另一方面较大地提高了效率。

4.2 数据表示形式

本文利用 Bitmap 技术来表示粒度，关系数据表中的每个属性被创建成一个 Bitmap，表中的每条记录在 Bitmap 中有一个唯一的位移偏置。如果记录中存在这个属性，Bitmap 中的对应位被表示成 1，否则被表示成 0。它利用位操作(AND 或 OR)的方法来改进计数方式，提高了算法的效率，目前Bitmap 技术已经被广泛应用于多个领域，特别是在关联规则挖掘中，Bitmap 技术能够提高计算效率。

下面给出一个利用 Bitmap 来实现频繁模式挖掘的例子。设 X 和 Y 为一个属性表的两个相互独立的属性。如果 $X\cup Y$ 的支持度大于 $s\%$，那么 2-项集(X,Y)就是频繁的。

表 1 是所采用的交易列表，一共存在 8 个事务，交易列表中存在 11 个不同的属性，设支持度为 37.5%，也就是说出现 3 次的项就是频繁项。

表 1 交易列表

T_ID	Items	T_ID	Items
01	1 2 3 5 10	05	1 2 3
02	3 4 5 10 11	06	4 5
03	4 5 6 7 10 11	07	10 11
04	1 5 6 8 10	08	6 7 8 9

下文表示 Bitmap 执行的大致步骤：

(1)首先把交易列表中不同属性的数据进行规范化。因为这里有 11 个属性，所以需要 11 个名称，因此，这张表就被划分成 11 个小颗粒。这些规范的名称都是用二进制的形式表示。而“位表示”代表这些属性在这些交易中间的存在情况。所以，对于如上的交易列表，表 2 给出其规范化的形式。

表 2 交易表的规范名称表示

Item Names	List Representation	位表示
1	{ID1,ID4,ID5}	(10011000)
2	{ID1,ID5}	(10001000)
3	{ID1,ID2,ID5}	(11001000)
4	{ID2,ID3,ID6}	(01100100)
5	{ID1,ID2,ID3,ID4,ID6}	(11110100)
6	{ID3,ID4,ID8}	(00110001)
7	{ID3,ID8}	(00100001)
8	{ID4,ID8}	(00010001)
9	{ID8}	(00000001)
10	{ID1,ID2,ID3,ID4,ID7}	(11110010)
11	{ID2,ID3,ID7}	(01100010)

(2)选出长度为 1 的项集。因为规范名称的每个属性表示为一行，表 3 列出 1-项集的记数情况。

表 3 频繁 1-项集的计数

列表	规范的计数方式	计数	是否频繁项集
1	(10011000)	3	Y
2	(10001000)	2	N
3	(11001000)	3	Y
4	(01100100)	3	Y
5	(11110100)	5	Y
6	(00110001)	3	Y
7	(00100001)	2	N
8	(00010001)	2	N
9	(00000001)	1	N
10	(11110010)	5	Y
11	(01100010)	3	Y

(3)接着，利用与 Apriori 类似的方法生成候选 k-项集$(k\geqslant 2)$。

(4)对这些 $k+1$ 项候选集进行检查，看它们是否满足最小的支持度。

(5)重复第(3)和第(4)步，直到没有 $k+1$ 项集产生。

(6)频繁集就是长度为 $1,2,3,\cdots,n$ 的满足最少支持度的项集。

4.3 BEMASK 算法基本流程结构

通过上文对 Bitmap 技术的讨论可以看到，在进行基于 Bitmap 运算时，首先需要将数据表文件进行规范化处理。也就是需要先将数据库文件转换成以行列转换的文件，这是 BEMASK 与 EMASK 算法不同的地方之一。

对于小型数据库文件，可以一次性地将数据库文件转化成 Bitmap 规范化文件存储在内存中，然后直接计算。但是对于大型数据库文件，这种转换方法是行不通的，所以需要先将 Bitmap 文件存储到磁盘上面，然后在每次进行计算时分次和分块地进行读取。本文将在下文的论述中具体讨论这个问题。

BEMASK 与 EMASK 的另一点重要区别在于支持度的计算方式，利用 Bitmap 技术进行支持度计数用到了位操作的方法，节省了计算时间，提高了效率。

本文提出的 BEMASK 方法为了利用 EMASK 中已有的隐私保护方式，将数据库进行扭曲之后才将其用 Bitmap 的形式来表示粒度，生成规范化文件。其它过程与 EMASK 过程基本相似，BEMASK 算法的大致过程如下(见图 2)。

(1)首先把原始数据库文件进行扭曲，然后把扭曲后的数据(按记录顺序排列)文件转化成规范化的 Bitmap 文件。

(2)计算 1-项集的支持度。然后通过 EMASK 中用到的推断方法对原始数据的支持度进行估计。剪除不满足最少支持度的 1-项集，生成 1-频繁项集。

(3)产生 $k+1(k\geqslant 2)$项的候选项集：

i. 如果长度为 k 的两个项中前 $k-1$ 个项相同，那么我们可以产生一个长度 $k+1$ 的候选项。

ii. 在这些得到的 $k+1$ 候选项集中，如果其 k-项子集都是频繁项集，把这个 $k+1$ 项候选项集保留，如果不是，把它删掉。

(4)对产生的 $k+1$ 项候选集进行检查，对其支持度进行计算，在这里，通过简单的“AND”操作就可以对项集进行计数，然后对原始数据的支持度进行估计，得到其支持度，去除不满足最少支持度的项集。

(5)重复第(3)和第(4)步，只到没有 $k+1$ 项集产生。

(6)频繁集就是长度为 $1,2,3,\cdots,n$ 的满足最少支持度的项集。

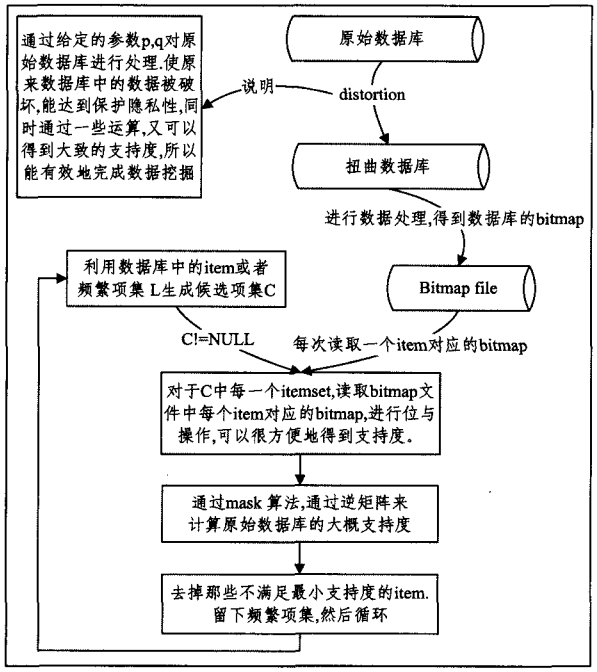

图 2　BEMASK 算法示意图

该算法在第(1)、(2)、(4)步与 EMASK 算法有区别,在第(1)步中,BEMASK 算法多了一个将数据库文件转换成Bitmap文件的过程,这将比 EMASK 多耗费一定的时间。在第(2)步及第(4)步中,在支持度的计算方式上也不相同,在 EMASK 中我们读取一条交易记录,然后进行处理。而在 BEMASK 中,每次读取一个 Bitmap 文件,然后进行计数处理,这样可以减少 I/O 次数。同时因为在 Bitmap 中,我们采用位操作"AND"来计算支持度,这样就更能提高效率。具体的效果比较将在后面的实验结果和分析中给出。

4.4　BEMASK 计数方法与 EMASK 计数方法的比较

本文主要讨论 k-项频繁项集的计数操作。

(1)BEMASK 的计数大致算法过程如下:

设 C_k 为 k-项候选项集,c 为 C_k 中的一个 k-项集。

i. 读取一个 Bitmap 文件。

ii. 遍历 C_k 中的所有的 c,$support(c)=Bitmap(B_{1c})+Bitmap(B_{2c})+\cdots+Bitmap(B_{mc})$,对于 B_{ij},i 表示第 i 个 Bitmap,j 表示具体的项集。

iii. 重复第 i、ii 两步,直到读取所有的 Bitmap 文件。

(2)对应的 EMASK 计数方式为:

i. 读取数据库文件中的一条记录。

ii. 遍历 C_k 中的所有的 c,如果 c 中属性在这条记录中存在,那么 support(X_k)++。

iii. 重复第 i、ii 两步,直到读取数据库文件的最后一条记录。

由计数方法可知,BEMASK 算法采用的是位"AND"的操作方式,所以对支持度的计算速度会很快,在实验结果中将会看到明显的效率改进。同时在 EMASK 的计数方式中,对于 2-项以上的计数,所有的候选项都是存放在 hashtree 中。对于 EMASK,因为每读取一条记录后,我们都需要遍历一次 hashtree,这样遍历的次数将达到 T_NO(数据库长度次数)。而利用 BEMASK 算法,我们只需要遍历 I_NO(最大的属性条数)次。一般来说,T_NO 将会是 I_NO 的上千倍,因此 BEMASK 算法也减少了对 hastree 遍历的时间。图 3 为 BEMASK 与 EMASK 计数方式对照图,从图中可以更加清楚地看到 k-项($k>2$)项集支持度计算方法的异同。

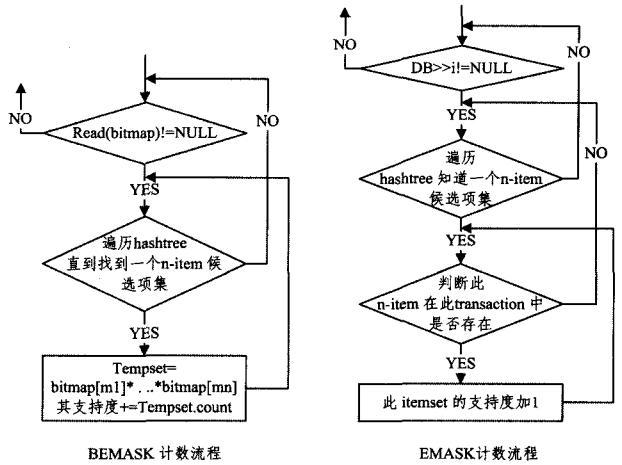

图 3　BEMASK 与 EMASK 计数方式对照

4.5　BEMASK 与 EMASK 的算法时间复杂度比较

随着计算机存储技术的发展,内存的容量变得越来越大,相比在空间开销上而言,算法分析需要更多地考虑提高算法的时间效率。

MASK 算法采用高斯消元法来求解逆矩阵,其计算时间复杂度为 $O(k^3)$,k 为矩阵的维数,改进的 EMASK 算法为了提高计算效率,将概率变换矩阵 M 由 $k=2^n$ 降到 $n+1$ 阶,从而计算 M^{-1} 的时间复杂度由 $O(k^3)$ 降到 $O(n^4)$。本文提出的 BEMASK 在扫描数据库时,仍然认为概率变换矩阵 M 的阶数是 2^n,设矩阵的维数为 k,由于采用 Bitmap 粒度计算方式,求 M^{-1} 的时间复杂度仅为 $O(k)$,从而提高了计算效率。

4.6　实验结果和分析

本文使用 C++在 gcc2.95 环境下实现了 EMASK、BEMASK 算法。在实验中,使用 Cygwin 在 CPU 为 Intel C4 1.7G,Memory 为 512M,操作系统为 Windows XP 的电脑上运行了 EMASK、BEMASK、Apriori 等程序。实验所用数据全部是由 IBM 数据生成器(IBM Almaden generator)生成。

为了验证 EMASK、BEMASK 算法在不同数据库及支持度下的效率对比的稳定情况,并兼顾没有进行隐私保护过程的 Apriori 算法,本文采用了两个固定的数据库 T25I4D1MN1K和 T50I8D1MN1K。实验分为两部分:

(1)测试 T25I4D1MN1K 数据库数据的 EMASK、BEMASK 和 Apriori 算法对比;

(2)测试 T50I8D1MN1K 数据库数据的 EMASK、BEMASK 算法对比。

表 4　BEMASK、EMASK 和 Apriori 算法执行时间记录

数据集	扭曲参数		支持度	Apriori	EMASK	BEMASK
	p	q	(%)			
T25I4D1MN1K	0.5	0.97	0.25	1156	104276	4415
			0.5	520	15570	1351
			0.75	359	2876	1102
			1	299	547	858
			1.25	270	252	796
			1.5	248	166	792

表 4 为基于 T25I4D1MN1K 数据库的 EMASK、BEMASK 和 Apriori 3 种算法在不同支持度(0.25%、0.5%、0.75%、1.0%、1.25%、1.5%)情况下的处理时间对比。图 4

为 EMASK、BEMASK 和 Apriori 3 种算法在处理同一种数据库不同支持度情况下的时间比较图。图 5 为 BEMASK 相对于 EMASK 的效率随着支持度的变化趋势图。

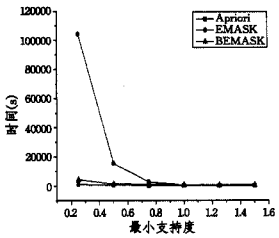

图 4 基于 T25I4D1MN1K 数据库的 3 种算法的时间消耗

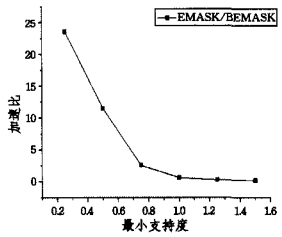

图 5 基于 T25I4D1MN1K 数据库的 EMASK 与 BEMASK 执行时间比

表 5 为基于 T50I8D1MN1K 数据库的 EMASK、BEMASK 两种算法在不同支持度(0.25%、0.5%、0.75%、1.0%、1.25%、1.5%)情况下的处理时间对比。图 6 为 EMASK 和 BEMASK 两种算法在处理同一种数据库不同支持度情况下的时间比较图。图 7 为 BEMASK 相对于 EMASK 的效率随着支持度的变化趋势图。

表 5 BEMASK 与 EMASK 算法执行时间记录

数据集	扭曲参数		支持度	EMASK	BEMASK
	p	q	(%)		
T50I8D1MN1K	0.5	0.97	1.0	67901	2075
			1.5	23417	995
			2.0	10069	621
			2.5	4048	549
			3.0	1196	435

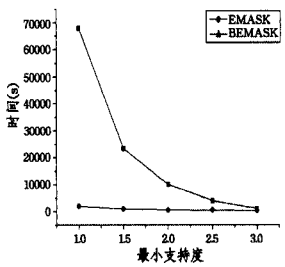

图 6 基于 T50I8D500KN1K 数据库的 EMASK 与 BEMASK 的时间消耗

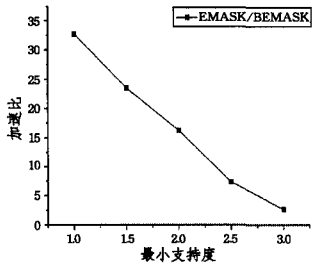

图 7 基于 T50I8D500KN1K 数据库的 EMASK 与 BEMASK 执行时间比

通过实验的结果我们可以得出如下结论：

(1)BEMASK 算法的效率相对于 EMASK 来说有很大幅度的提高。最小支持度越小，BEMASK 消耗的时间与 EMASK 消耗的时间差距越大。

(2)当数据库变得稠密后，BEMASK 的优势更加明显。

(3)由于在 BEMASK 算法中增加了一个将数据库文件转化成 Bitmap 文件的过程，因此最小支持度大于一定点时，3-项以上的频繁集大幅度减少，这样使用 Bitmap 技术来提高速度的优势就得不到明显的体现。

(4)由于增加了隐私保护过程，因此 BEMASK 算法比没有进行隐私保护过程的 Apriori 算法效率稍低。

结束语 隐私保护数据挖掘一直是数据挖掘领域一个重要的研究方向。本文针对 EMASK 算法所依据的 Apriori 算法需要完全的数据库扫描并且进行多次比较操作来降低效率的弊端，提出了 BEMASK 算法，该算法利用粒度方式将关系数据表转换成面向机器的关系模型，将数据处理转换成粒度计算的方式，计算频繁项集变成了计算基本颗粒的交集。实验证明，在保证准确性不降低的情况下，相对 EMASK 算法，算法中所采用的数据垂直 Bitmap 表示(vertical Bitmap)减少了 I/O 操作的次数，效率得到了较大的提高。

参 考 文 献

[1] 王艳，乐嘉锦，孙捷，等. 网络用户行为的隐私保护数据挖掘方法[J]. 计算机工程与应用，2012，48(13)：138-143

[2] 马进，李锋，李建华. 分布式数据挖掘中基于扰乱的隐私保护方法[J]. 浙江大学学报：工学版，2010，44(2)：276-282

[3] 张鹏，童云海，唐世渭，等. 一种有效的隐私保护关联规则挖掘方法[J]. 软件学报，2006，17(8)：1764-1774

[4] 孙茂华. 安全多方计算及其应用研究[D]. 北京：北京邮电大学，2013

[5] Pawlak Z, Grzymala-Busses J, Slowinski R, et al. Rough sets [J]. Communications of the ACM, 1995, 38(11): 88-95

[6] Zadeh L A. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems [J]. Soft Computing, 1998, 2(1): 23-25

[7] Yao Yi-yu. The Art of Granular Computing[C]// Proc of the International Conference on Rough Sets and Emerging Intelligent Systems Paradigms, 2007. Warsaw, Poland, 2007: 101-112

[8] Chen Hong-mei, Li Tian-rui, Ruan Da, et al. A rough-set based incremental approach for updating approximations under dynamic maintenance environments [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(2): 274-284

[9] 王磊，李天瑞. 一种基于矩阵的知识粒度计算方法模式[J]. 模式识别与人工智能，2013，26(5)：447-453

[10] 项海飞. 基于互信息粒度的相对约简的矩阵计算方法[J]. 西南师范大学学报：自然科学版，2014，39(3)：60-64

[11] Rizvi S J, Haritsa J R. Maintaining Data Privacy in Association Rule Mining[C]// Proc of the 28th Intl Conf on Very Large Data Bases (VLDB), 2002. Hong Kong, China, 2002: 682-693

[12] Agrawal S, Krishnan V, Haritsa J R. On Addressing Efficiency Concerns in Privacy-preserving Mining [C] // Proc of 9th Intl Conf on Database Systems for Advanced Applications (DASFAA), 2004. Jeju Island, Korea, 2004: 113-124

(上接第 189 页)

[9] Griffiths T L, Steyvers M. Finding scientific topics [J]. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(1): 5228-5235

[10] Chang C-C, Lin C-J. LIBSVM: A library for support vector machines[J]. ACM Trans. Intell. Syst. Technol., 2011, 2(3): 1-27

[11] Hall M, Frank E, Holmes G, et al. The WEKA data mining software: an update[J]. SIGKDD Explor. Newsl., 2009, 11(1): 10-18

[12] Wu S, Hofman J M, Mason W A, et al. Who says what to whom on twitter[C]// Proceedings of the international conference on World Wide Web (WWW). 2011: 705-714

[13] Diggle P. A kernel method for smoothing point process data[J]. Applied Statistics, 1985, 34(2): 138-147