ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.11, November 2007, pp.2782−2799
DOI: 10.1360/jos182782

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

数据库中的知识隐藏*

郭宇红[1+], 童云海[2], 唐世渭[1,2], 杨冬青[1]

[1](北京大学 计算机科学技术系,北京 100871)

[2](北京大学 视觉与听觉信息处理国家重点实验室,北京 100871)

Knowledge Hiding in Database

GUO Yu-Hong[1+], TONG Yun-Hai[2], TANG Shi-Wei[1,2], YANG Dong-Qing[1]

[1](Department of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](State Key Laboratory of Machine Perception, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62757756, E-mail: yhguo@pku.edu.cn, http://www.pku.edu.cn

Guo YH, Tong YH, Tang SW, Yang DQ. Knowledge hiding in database. *Journal of Software*, 2007,18(11): 2782−2799. http://www.jos.org.cn/1000-9825/18/2782.htm

Abstract: Motivated by the multiple requirements of data sharing, privacy preserving and knowledge discovery, privacy preserving data mining (PPDM) has become the research hotspot in data mining and information security fields. Two main problem are addressed in PPDM: One is the protection of sensitive raw data; the other is the protection of sensitive knowledge contained in the data, which is also called knowledge hiding in database (KHD). This paper gives a survey on the current KHD techniques. It first introduces the background in which KHD appears. Then it mainly presents the techniques on sensitive association rule hiding and classification rule hiding. Evaluation of KHD methods is discussed after that. Finally, it points out three future research directions of KHD: Design of measure function based on target distance in data modification techniques, inverse frequent set mining in data reconstruction techniques and design of general KHD method based on data sampling.

Key words: knowledge hiding; KHD (knowledge hiding in database); sensitive rule; privacy preserving; inverse mining

摘 要: 伴随着数据共享、隐私保护、知识发现等多重需求而产生的PPDM(privacy preserving data mining),成为数据挖掘和信息安全领域近几年来的研究热点.PPDM中主要考虑两个层面的问题:一是敏感数据的隐藏与保护;二是数据中蕴涵的敏感知识的隐藏与保护(knowledge hiding in database,简称KHD).对目前的KHD技术进行分类和综述.首先介绍KHD产生的背景,然后着重讨论敏感关联规则隐藏技术和分类规则隐藏技术,接着探讨KHD方法的评估指标,最后归结出KHD后续研究的3个方向:数据修改技巧中基于目标距离的优化测度函数设计、数据重构技巧中的反向频繁项集挖掘以及基于数据抽样技巧的通用知识隐藏方法设计.

关键词: 知识隐藏;KHD(knowledge hiding in database);敏感规则;隐私保护;反向挖掘

中图法分类号: TP311 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant No.60403041 (国家自然科学基金)
Received 2007-01-10; Accepted 2007-05-10

数据库中的知识发现(knowledge discovery in database,简称 KDD)技术——数据挖掘基于海量数据抽取出新颖、潜在有用的知识,目前已经成为一种有效的分析决策手段,在企、事业中得到广泛应用.数据的共享可为企业带来许多益处,但在共享数据时又担心隐私数据和数据中蕴藏的敏感知识会外流给竞争对手知道,进而暴露个人隐私并影响到自己公司的利益.伴随着数据共享、隐私保护、知识发现等多重需求而产生的隐私保护数据挖掘(privacy preserving data mining,简称 PPDM),受到国内外各著名高校、科研机构和工业届的广泛关注,并成为数据挖掘和信息安全领域近几年来新的研究方向和研究热点.

PPDM 中主要考虑两个层面的问题:一是敏感原始数据的保护,比如身份证号、地址、收入等微数据的保护;二是数据中蕴藏的敏感知识的保护,敏感知识是指使用数据挖掘算法从数据中挖掘到的敏感规则.在数据挖掘背景下,前者要解决的问题是如何在不精确访问个体数据的情况下,仍能运用挖掘算法得到正确的挖掘结果;而后者要解决的问题是如何保护数据中的敏感知识不被恶意用户通过数据挖掘算法发现,以防止机密知识的泄露和基于知识的恶意推理.目前,PPDM 中这两个层面的问题都有大量的研究成果呈现,尤其是第 1 个层面的问题,即面向个体隐私数据保护的挖掘问题,近几年在 SIGMOD,VLDB,KDD,PODS,ICDE,ICDM 等相关的高水平国际会议上均有高质量的论文发表.文献[1]对这方面的工作进行了综述.第 2 个层面的问题,即数据库中的敏感规则隐藏问题,也称为数据库中的知识隐藏(knowledge hiding in database,简称 KHD)[2]问题,自 1999 年在文献[3]被明确提出以来,在 ICDM,CIKM,PAKDD,DaWaK 等国际会议以及 TKDE,IJBIDM 等期刊上也呈现了大量成果,然而目前国内和国际上还没有将 KHD 的发展情况、核心技术和研究成果作一个整体上的介绍.鉴于 KHD 在数据隐私、知识保护等安全领域的重要性,为了捕捉 KHD 的发展动态,对 KHD 研究有一个总体上的把握,促进国内迅速跟上国际研究的步伐,综述这方面的工作十分有意义.

本文在分析国内外相关研究工作的基础上,结合国家自然科学基金课题——"面向隐私保护的数据挖掘",对 KHD 技术进行综述.本文第 1 节介绍数据库中知识隐藏问题产生的背景、动机和应用场景,并对问题的定义进行具体描述.第 2 节详述一系列的知识隐藏方法,包括数据变换法、数据阻塞法、数据重构法、数据抽样法等.第 3 节讨论 KHD 方法的评估.最后总结全文,并展望未来的研究工作.

1 知识隐藏问题

1.1 问题的产生

从数据库中抽取知识,很多年以来一直都是 KDD 研究人员努力要实现的目标.一方面,到现在为止,这一问题已经在研究领域和工业界得到了很好的理解,相关的研究已经相对成熟;另一方面,KDD 技术所带来的信息安全方面的影响,直到最近几年才被给予关注和考虑.

知识发现作为对数据库安全的威胁第一次是在文献[4]中被提出的,该文强调了对数据库中所发现的个人敏感信息的公开限制问题,即对个人隐私信息的保护问题.当时很多研究人员认为,KDD 所发现的模式通常是关于团体而非特定的个人,而团体是不涉及隐私的.因此,刚开始人们对于数据挖掘所引发的隐私泄露等安全问题并未给予足够的重视.随着社会生活的日益开放以及数据采集、数据分发、知识发现、互联网等技术的蓬勃发展,获取数据、信息和知识的过程变得越来越容易,随之而产生的隐私数据、机密信息、敏感知识的泄露与保护等问题也变得越来越突出.Clifton 等人在文献[5]中进一步分析了数据挖掘所带来的信息安全和隐私方面的问题,并针对数据库中敏感规则隐藏问题产生的动机提出了一个情景,展示了商业团体如何使用各种数据挖掘技巧,利用所获得的数据获取机密知识从而在商业中获得优势.该情景有利于我们了解数据库中敏感规则隐藏问题的重要性,具体的情景如下:

假设我们是连锁购物中心 BigMart 采购部门的主管,我们与供应商 Dedtrees Paper 之间有一个协议:如果我们愿意让 Dedtrees 读取顾客购买记录,则 Dedtrees 愿意以较优惠的价格供应我们其公司的产品.我们接受了这项协议,而 Dedtrees 使用关联规则挖掘工具对数据库做分析,他们发现,通常购买脱脂牛奶的人也购买其竞争对手 Green Paper 的产品.于是,Dedtrees 公司就进行促销策略——如果购买脱脂牛奶也一起购买 Dedtrees 产品将获得 50 美分的折扣.Dedtrees 利用这项优惠促销活动,严重冲击了 Green Paper 的销售量,并因此掌握了市场主

动权.当下一次我们再和 Dedtrees Paper 协商的时候,他们因为竞争对手(Green Paper)的减少,不愿意再提供我们优惠的价格,使得我们在和供应商的关系上陷入被动.

上述情景暗示了在一种商业情景下,由于数据中敏感知识的无意泄露,导致了购物中心和供应商之间利益格局的变化,显示了数据拥有者在将数据进行共享和交换以前,隐藏掉数据中蕴藏的敏感知识是非常重要的.保护一些关键的知识可以帮助团体有效保持竞争的优势,让企业或团体在共享数据获取共同的最大利益之余,能够保护好自己的核心商业机密.图 1 给出了敏感规则隐藏问题产生的更为常见的一种情景:多个处于竞争位置的公司合作进行关联规则挖掘,需要共享各自的数据;同时,各个公司都不希望各自的战略性模式被挖掘出来,泄露给竞争对手.基于这样的考虑,各个公司在将自己的数据共享和发布之前,需要应用敏感规则隐藏方法,对共享数据库中的敏感规则进行隐藏,将敏感规则隐藏后的数据库作为共享数据库发布和使用.

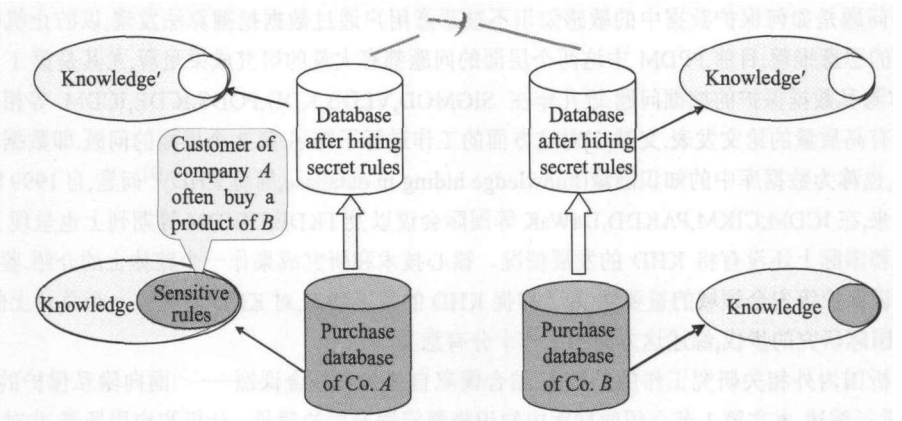

Fig.1 A scenario of sensitive rule hiding

图 1 敏感规则隐藏问题产生的一个情景

上述两个情景从敏感规则自身的重要性角度出发,说明了数据库中敏感规则隐藏问题的重要性.敏感规则隐藏的另一个较为特殊的应用需求就是,通过隐藏规则,阻止基于规则的恶意推理所导致的隐私数据泄露.文献[6,7]正是基于这种动机开展研究的.

总之,敏感规则隐藏问题产生的原因在于数据共享和信息安全之间的矛盾:一方面,出于公司合作需要,需要将数据共享和交换;另一方面,出于自身信息和知识安全上的考虑,需要对其中的隐私数据和敏感知识进行隐藏和保护.数据共享和知识隐藏这一双重需求,使得针对敏感规则隐藏问题的方法,一方面要能隐藏掉数据中的敏感规则,做到发布的数据是安全的,另一方面要能保证数据的正常使用,做到发布的数据是可用的.

1.2 问题描述

概括地讲,数据库中的敏感规则隐藏是指通过一定的方法,将数据库中能够通过数据挖掘算法发现的敏感规则隐藏、保护起来的过程.具体描述如下:给定数据库 D、需要保护的敏感规则集合 R_h,如何将 D 转换为 D',使得敏感规则集 R_h 在 D' 中不能被直接通过挖掘算法发现出来.

由于通过数据挖掘算法发现的规则通常称为知识,所以数据库中的规则隐藏问题也称作数据库中的知识隐藏问题.数据库中的知识隐藏,即 KHD.这一术语首次出现在文献[2]中.该文献给出了一个相对通用的 KHD 方法架构,所描述的 KHD 过程主要包括以下 5 个步骤:第 1 步,确定数据中需要隐藏的敏感知识;第 2 步,确定能够发现敏感知识的挖掘算法;第 3 步,制定安全策略;第 4 步,对数据进行清洗;第 5 步,生成描述 KHD 过程最终结果的报告,报告中包括关于清洗后数据完整性的详细信息.在 KHD 架构下,敏感规则隐藏问题不强调数据中的个体隐私,即假定数据库中的个体数据不需要保护;相反地,从数据库所能挖掘出的敏感规则需要保护,这些规则对于战略决策极为重要,必须得到保护和隐藏.

KHD 是一种有效的方法架构,但当针对不同的应用场景、不同的数据,对不同的敏感知识进行隐藏时,还需

要采取不同的策略,以达到较好的知识隐藏效果和较高的数据可用性.目前,已有很多技术用于实现包括关联规则、分类规则、序列模式等在内的各种敏感规则隐藏方法,本文第 2 节将详细介绍这些方法.

2 现有知识隐藏方法

如图 2 所示,我们从敏感知识的种类、隐藏方式两个不同的角度对现有的知识隐藏方法进行分类.

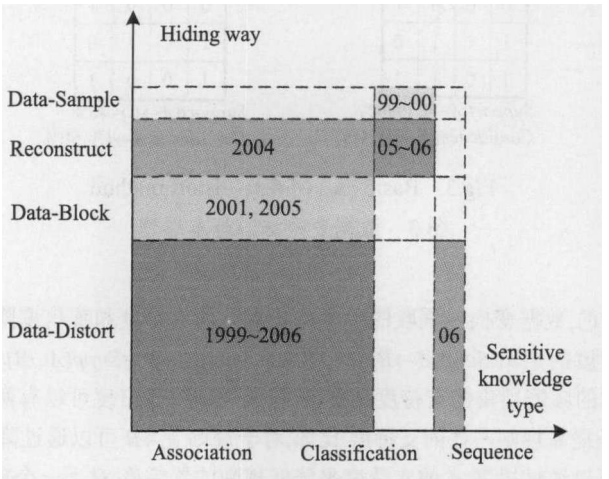

Fig.2 Classification of current knowledge hiding methods

图 2 现有知识隐藏方法分类

- 敏感知识的种类

根据要隐藏的知识种类,可以分为关联规则隐藏方法[8-28]、分类规则隐藏方法[29-32]、序列模式隐藏方法[33]、聚类隐藏方法等.目前研究最多的是关联规则隐藏方法.

- 隐藏方式

根据具体的隐藏方式,当前的研究方法可以分为数据变换法[8-17,19-24]、数据阻塞法[25-27]、数据重构法[28-30]和数据抽样法[31,32].目前研究最多的是数据变换法.

图 1 中不同的规则种类、不同的隐藏方式将平面分成了 12 个不同的区域,着色的区域表示有相应的成果,空白区域表示尚无成果,区域的大小大体代表了成果的多少,相应成果的发表时间在各个区域内进行了标记.

2.1 关联规则的隐藏

2.1.1 数据变换法

2.1.1.1 基本思想

数据变换法的基本思想是,对于原始数据库中的敏感事务,通过删除项或增加项的方式,使敏感规则的支持度或置信度降低到某个阈值以下.若把原始数据库用布尔矩阵表示,矩阵的每一行代表一个事务,每一行中的"1"代表对应列所指的项出现在该事务中,"0"表示对应的项不出现在该事务中,则数据变换法通过将敏感事务中的 1 变成 0 或 0 变成 1 的方式对原始数据库作修改,使敏感规则的支持度或置信度降低,达到被隐藏的目的.

如图 3 所示,规则 $A \rightarrow C$ 在原始数据库中的支持度和置信度分别为 80%和 100%,将第 2 个和第 5 个事务中的项 C 删除后(图中原来的 1 变成 0),$A \rightarrow C$ 的支持度和置信度分别降低到 40%和 50%.如果支持度和置信度安全阈值为 60%,规则 $A \rightarrow C$ 将不再出现在变换后的数据库中,从而在新数据库中得到隐藏和保护.在运用数据变换法对敏感关联规则隐藏时,包括以下 4 个步骤:

① 根据敏感规则,找出待修改的候选事务集;

② 在候选事务集窗口中选择要修改的候选事务;

③ 对选定的候选事务进行修改;

④ 重复①~③步,直到敏感规则的支持度或置信度降低到用户设定的某个安全阈值以下为止.

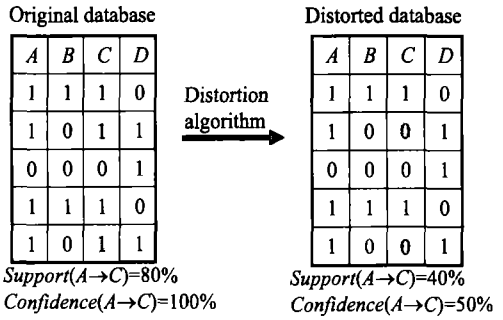

Fig.3 Basic idea of data-distort method

图 3 数据变换法的基本思想

2.1.1.2 典型算法

要达到规则隐藏的目的,数据变换法采取的手段是将规则的支持度和置信度降低到某个安全阈值以下.根据关联规则支持度和置信度的定义:$Supp(A\rightarrow B)=|A\cup B|/|D|$,$Conf(A\rightarrow B)=Supp(A\cup B)/Supp(A)$,降低规则的支持度可以通过降低生成该规则的频繁项集的支持度来实现.降低规则的置信度可以有两种方式:一是降低规则对应的频繁项集的支持度,二是提高规则左件的支持度.比如,对于规则 $A\rightarrow B$,可以通过降低 $A\cup B$ 的支持度来降低规则的支持度和置信度,也可以通过提高 A 的支持度来降低规则的置信度.对于一个项集来说,降低支持度可采取删除项的方式,而提高支持度则可采取添加项的方式,这使得数据变换法在对数据清洗变换时有多种选择.同时,由于给定一个敏感规则集和待清洗的原始数据库,选定最优的数据进行最有效的清洗是一个 NP 难题[3],因此,贪心法、启发式等清洗策略被广泛使用,以便合理的时间内寻求一个近似最优解.根据数据变换对于规则隐藏不同手段(是降低支持度还是降低置信度)的选择、贪心法中不同的优化测度的选择以及不同的启发式策略的选择,呈现了许多算法.在具体介绍这些算法之前,先给出几个要用到的重要概念:

- 敏感事务(sensitive transaction)[11].设 T 是数据库 D 中的一个事务集合,R_h 是从 T 中挖掘到的敏感规则集合,S_T 称为敏感事务集,当且仅当 S_T 是 T 的子集,由且仅由 S_T 推导出敏感规则.包含在 S_T 中的事务称为敏感事务.
- 冲突度(conflict degree)[11].事务所支持的敏感规则的数目.设 T 是数据库 D 中的一个事务集合,R_h 是从 T 中挖掘到的敏感规则集合,对于任一事务 $t\in T$,其冲突度计算如下:对于任一 $r_h\in R_h$,如果 t 在支持该规则的敏感事务集中,t 的冲突度加 1,直至遍历过所有的敏感规则.
- 牺牲项(victim item)[10].在基于数据清洗的敏感规则保护方法中,对于待修改的敏感事务,通常采取选择一项将其移去或者改变取值从而改变敏感规则支持度的方法,称选定的项为牺牲项.
- 公开度(disclosure threshold)[10].敏感规则隐藏的目标是实现规则隐藏与知识发现之间的一个平衡,公开度用来衡量对敏感规则的保护程度,由用户来定义.对于任一敏感规则,若公开度为 0%,则说明该规则要得到完全的保护,即无论支持度和置信度取什么阈值,该规则均不会被发现;若公开度为 100%,那么相应规则无须保护,对用户完全公开.公开度通过控制清洗的敏感事务比例来控制敏感规则的隐藏效果.

(1) 第 1 组算法:Algo1a/Algo1b/Algo2a/Algo2b/Algo2c

第 1 组算法是由 E.Dasseni,V.S. Verykios,A.K. Elmagarmid 等人于 2001 年提出的.这些算法基于以下几个假设和共同点:

a) 敏感规则之间互不相交,即不同的规则没有交集项;

b) 每次只选 1 条敏感规则进行隐藏;

c) 对支持度、置信度的降低每次只降低一个单位,即对一个事务进行一次修改后,对应的项集支持数或加 1,或减 1;

d) 支持度降低到最小支持度阈值以下,或置信度降低到最小置信度阈值以下,就认为规则得到了隐藏.

Algo1a 算法[8].该算法要隐藏的是一组敏感规则,对于每一条敏感规则,增加规则左件支持度,直到规则置信度降低到最小置信度阈值以下为止.其中,待修改的候选事务集由部分支持规则左件的事务组成,算法每次选择包含规则左件中项数最多的事务作修改,修改的方法是向其中添加所有出现在规则左件而不出现在事务中的项,修改后的事务将由部分支持规则左件变成完全支持规则左件.通过"选择包含规则左件中项数最多的事务作修改",该算法试图使为了增加规则左件支持数而额外添加的项尽可能地少,以便原数据库尽可能少作改动.

Algo1b 算法[8].该算法要隐藏的是一组敏感规则,对于每一条敏感规则,降低规则右件支持度,直到规则的支持度降低到最小支持度阈值以下,或置信度降低到最小置信度阈值以下为止.其中,待修改的候选事务集由支持整条规则的事务组成,算法每次选择最短的事务作修改,修改的方法是删除规则右件中的一个项.准确地说,假定规则右件是 r_r,算法产生 r_r 的所有含有$(|r_r|-1)$个元素的子集,然后选择支持度最高的子集中的第 1 个项,作为要删除的牺牲项,以使该项被删除后对其他$(|r_r|-1)$-项集的支持度的影响尽可能地小.通过"选择最短的事务作修改",该算法试图使由该项被删除而给其他项集造成的影响尽可能地小.

Algo2a 算法[8].该算法要隐藏的是一组敏感规则,对于每一条敏感规则,降低产生规则的大项集的支持度,直到规则的支持度降低到最小支持度阈值以下或置信度降低到最小置信度阈值以下为止.其中,待修改的候选事务集由支持整条规则的事务组成,算法每次选择最短的事务作修改,修改的方法是删除规则中的一个项.准确地说,假定规则是 r,算法产生 r 的所有包含$(|r|-1)$个元素的子集,然后选择支持度最低的子集中的第 1 个项作为要删除的牺牲项,以使该项被删除后对其他$(|r|-1)$-项集的支持度的影响尽可能地小.通过"选择最短的事务作修改",该算法试图使由该项被删除而给其他项集造成的影响尽可能地小.

Algo2b 算法[9].该算法要隐藏的是一组大项集,即频繁项集,算法先按大项集的长度和支持度进行排序,最长的支持度最高的大项集被优先隐藏,对于每一个要隐藏的大项集,降低大项集的支持度,直到大项集的支持度降低到最小支持度阈值以下为止.其中,待修改的候选事务集由支持大项集的事务组成,算法每次选择最短的事务作修改,修改的方法是删除大项集中支持度最高的项.通过"选择最短的事务作修改"和"选择支持度最高的项作牺牲项"这两条启发式规则,该算法试图使由该项被删除而给其他项集造成的影响尽可能地小.

Algo2c 算法[9].该算法要隐藏的是一组大项集,即频繁项集,对于每一个要隐藏的大项集,算法降低大项集的支持度,直到大项集的支持度降低到最小支持度阈值以下为止.其中,待修改的候选事务集由支持大项集的事务组成,算法采取随机轮转的方式来选择要修改的事务和确定哪个项作为牺牲项被删除.具体地讲,如果要隐藏的大项集 Z 中项的一个随机排列为 $I_0,I_1,...,I_{(n-1)}$;支持 Z 的事务集合 T_Z 中事务的随机排列为 $T_0,T_1,...,T_{(m-1)}$,算法在第 1 步选择事务 T_0 作为要修改的事务,选择 I_0 作为牺牲项,将 I_0 从 T_0 删除;第 2 步分别选择 T_1 和 I_1,将 I_1 作为牺牲项从 T_1 删除;依次轮转,直到项集 Z 的支持度降低到最小支持度阈值以下为止.直觉上,随机轮转法采取的是一种"公平"的思想,每一个候选事务和每一个候选牺牲项机会均等,以随机轮转的方式,试图减少某个项被过度隐藏(over-killed)而带来较大的副作用的机会.

第 1 组 5 种算法的特点、设计思想的比较见表 1.

Table 1 Comparison of the first group algorithms of data-distort method

表 1 数据变换法第 1 组算法比较

Algorithms	Select transaction	Select victim item	Hiding means	Add/Delete item	Hide rule/ large itemset
Algo1a	Short transaction first	Items in r_l but not included the transaction	Increase support of rule's left side r_l	Add item	Hide rule
Algo1b	Short first	First item of $(\|r_r\|-1)$-itemset with maximum support	Decrease support of rule's right side r_r	Delete item	Hide rule
Algo2a	Short first	First item of $(\|r_r\|-1)$-itemset with minimum support	Decrease support of rule	Delete item	Hide rule
Algo2b	Short first	Item with maximum support	Decrease support of large itemset	Delete item	Hide itemset
Algo2c	Random order	Random order	Decrease support of large itemset	Delete item	Hide itemset

(2) 第 2 组算法:Naive/MinFIA/MaxFIA/IGA/RRA/RA/SWA

第 2 组的 7 种算法由 Oliveira 等人在 2002 年~2003 年提出,与第 1 组算法的区别在于:

(a) 通过引入冲突度,第 2 组算法能够很好地处理规则间有交集项的敏感规则集;而第 1 组算法只能处理规则间相互独立没有交集的敏感规则集合.

(b) 通过引入公开度,第 2 组算法控制要修改的敏感事务比例,在相同的公开度阈值下,不同的规则根据原数据库中支持该规则的敏感事务总数的不同,所要清洗的事务总数是不同的,但所占敏感事务总数的比例相同,公开度并不能直接控制隐藏失败率,只能控制隐藏的大体效果;第 1 组算法则通过最小支持度阈值来控制要修改的事务总数,在相同的最小支持度阈值条件下,不同的规则清洗后所剩余的敏感事务数是相同的,最小支持度阈值可以直接控制隐藏失败率.

(c) 通过引入倒排文件,第 2 组算法最多只需扫描数据库两次,加速了清洗过程;而第 1 组算法每隐藏一条规则就需要扫描整个数据库 1 次.

(d) 第 2 组算法只采取降低规则支持度的方式隐藏规则,对于事务的修改只删除项,而不增加项;第 1 组算法则通过降低规则支持度、置信度两种方式隐藏规则,对事务的修改可以删除项,也可以增加项.

Naive 算法[10].其主要思想是,根据公开度确定需要修改的敏感事务数;然后选择冲突度最小的几个敏感事务,对于选定的敏感事务,将敏感规则中的所有项从中去除.如果一个敏感事务包含且仅包含敏感规则中的项,那么保留数据库中出现频度最大的项,以保持数据库中的事务数不变.这里所应用的启发式规则是:冲突度越小,修改该事务给其他敏感规则带来的影响越小.

MinFIA(minimum frequency item algorithm)算法[10].其基本思想是,根据公开度确定需要修改的敏感事务数;然后选择冲突度最小的几个敏感事务,去除敏感规则所包含的支持度最小的项.

MaxFIA(maximum frequency item algorithm)算法[10].其基本思想是,根据公开度确定需要修改的敏感事务数;然后选择冲突度最小的几个敏感事务,去除敏感规则所包含的支持度最大的项.

IGA(item grouping algorithm)算法[10].该算法的特殊之处在于牺牲项的选择上.其基本思想是,一个敏感事务的冲突度越大,修改该事务,就有越多的敏感规则支持度降低进而被隐藏,这样就可以修改尽可能少的事务而将敏感规则隐藏,从而减少对数据及非敏感规则的影响.算法首先对敏感规则进行分类,分类的原则是要满足同一类中的敏感规则包含相同项,然后选择这些公共项中支持度最小的项作为该类的标识项.需要说明的是,根据前面的分类,一个敏感规则可能包含在多个类中,这时就需要解决重叠问题,方法是,将已分好的敏感规则类按照包含规则的数目进行降序排序,敏感规则类两两比较,如果两类中有相同的规则,则将该规则从小类中删除;若两类大小相同,则将规则从标识项支持度较小的类中去除.对于任一敏感规则,将其所在类的标识项作为牺牲项,根据公开度,选择冲突度最大的几个敏感事务,删除牺牲项.

RRA(round robin algorithm)算法[11].其基本思想是,根据公开度确定需要修改的敏感事务数;然后选择冲突度最大的前几个敏感事务,采取随机轮转的方式选取一个牺牲项从事务中移除.随机轮转是指对第 1 条规则选取第 1 个项作为牺牲项,对第 2 条规则选取第 2 个项作为牺牲项,对第 i 条规则选取第 $i \bmod k$ 个项作为牺牲项(k 为第 i 条规则所包含的项的个数),依次轮转.

RA(random algorithm)算法[11].其基本思想是,根据公开度确定需要修改的敏感事务数;然后选择冲突度最大的前几个敏感事务,随机选取一个牺牲项从事务中移除.

SWA(sliding window algorithm)算法[12].算法每次扫描 K 个事务(K 为窗口大小),在 K 个事务窗口中进行清洗.对于任一敏感规则,其敏感事务的牺牲项选为规则中出现频率最高的数据项,如果数据项的出现频率均为 1,则随机选择牺牲项;然后根据敏感规则的公开度阈值,确定需要清洗的事务数,选择最短的几个事务作修改.当公开度取为 0 时,执行完算法,检查一个敏感事务是否不需要清洗多次,其目的是减少信息的丢失,具体执行如下:当对敏感规则 r_1 的敏感事务 t 完成清洗后,检查 t 是否也在另一敏感规则 r_2 的敏感事务集中,如果是,并且已经选定的牺牲项也包含在 r_2 中,则将 t 从 r_2 的敏感事务集中移去.SWA 算法的一个特点是每一条规则的公开度阈值可以不同,使得安全度控制比较灵活;另一个特点是滑动窗口的运用使得算法不必一次将所有的事务都读

入内存,从而使算法能够处理大规模的数据库.

第 2 组 7 种算法的特点、设计思想及时间代价的比较见表 2.其中,n 为敏感规则的数目;N 为数据库中的事务数;ψ为公开度;m 为需要修改的敏感事务数,对于任一敏感规则 r_h,$m=SensitiveTranSets(r_h)\times(1-\psi)$.

Table 2 Comparison of the second group algorithms of data-distort method

表 2 数据变换法第 2 组算法比较

Algorithms	Select transaction	Select victim item	Time complexity
Naive	Top m transactions with smallest conflict degree	All items in r_h	$O(n.\text{Log}N)$
MinFIA	Top m transactions with smallest conflict degree	Item in r_h with minimum support	$O(n.\text{Log}N)$
MaxFIA	Top m transactions with smallest conflict degree	Item in r_h with maximum support	$O(n.\text{Log}N)$
IGA	Top m transactions with largest conflict degree	Item of group class label	$O(n.\text{Log}N)$
RRA	Top m transactions with largest conflict degree	The i-th item in r_h	$O(n.\text{Log}N)$
RA	Top m transactions with largest conflict degree	Random	$O(n.\text{Log}N)$
SWA	The m-shortest transactions	Item in r_h with maximum support	$O(n.\text{Log}K)$

(3) 其他算法

前面介绍的第 1 组、第 2 组算法都是基于启发式思想的数据清洗变换方法.启发式方法的优点是操作简单,计算代价小,只需根据启发式规则直接选择敏感事务和牺牲项修改即可.其缺点是,算法是在定性规则而非定量指标的引导下进行的,规则隐藏的实际效果只有等算法执行完才能得以验证和了解.最近两年出现的方法逐渐将定量评估引入算法的执行过程,使得隐藏的效果由定性引导转变为定量控制,主要包括文献[13]提出的贪心法、文献[14,15]提出的基于整数规划的最优化方法、文献[16,17]提出的基于清洗矩阵的数据转换法.这些方法的共同点都是对敏感项集而非敏感关联规则进行隐藏,下面简要介绍这些方法.

贪心法.文献[13]提出了基于项集格边界的贪心算法,用于隐藏敏感项集.其特点是在隐藏过程中的每一步,通过测度函数,估算当前状态下删除一个项对非敏感频繁项集边界的影响.算法选取对边界影响最小的事务和项进行清洗,以保证非敏感频繁项集尽可能小地受到影响.其中,测度函数的设计比较巧妙,对于支持度较高的项集赋予较大权重的被影响因子,且权重随清洗过程支持度的变化而动态调整,其目的是平衡对各个非敏感频繁项集造成的负面影响,使修改后的数据库中非敏感频繁项集的数量和相对频繁度得到最大程度地保留.

整数规划法.文献[14]提出了用于隐藏敏感项集的全局最优化方法.针对所有的敏感项集,该方法通过整数规划一次挑选出需要清洗的所有候选事务,其目标是使要清洗修改的事务数最少.文献[15]提出了基于项集格边界和整数规划的全局最优化方法,整数规划目标方程为删除的项数之和最小.目标约束分为两组,一组由所有敏感项集的支持度都小于阈值的约束构成,另一组由所有的非敏感频繁项集的支持度都不小于阈值的约束构成,其目标是:所有的敏感项集都被隐藏;所有的非敏感频繁项集都不被隐藏;为达到前两个目标所需删除的项最少.项集格边界的应用,使约束仅局限在项集格中处于上下边界的项集中,从而大量减少了约束方程数目.

清洗矩阵转换法.文献[16]提出了一种新颖的数据变换方法.该方法基于对敏感项集和非敏感频繁项集的观察,构造出一个清洗矩阵 S,然后将原始数据库事务矩阵 D 乘以 S,得到清洗后的数据库 D'.该方法的重点在于清洗矩阵 S 的构造,初始情况下 S 为单位矩阵.将非对角线上的元素 S_{ij} 设为−1,则可破坏项 i 和项 j 的关联,从而降低项 i 和项 j 所组成的频繁二项集的支持数;而将非对角线元素 S_{ij} 设为 1,则可保持项 i 和项 j 的关联.根据不同的隐藏策略,文献[16]提出了 3 种清洗矩阵设置方法:第 1 种方法称为隐藏优先(hidden-first,简称 HF),其思想是优先考虑敏感项集的隐藏,而先不顾及对非敏感项集的错误隐藏,该方法将造成非敏感频繁项集丢失;第 2 种方法称为非隐藏优先(non-hidden-first,简称 NHF),其思想是优先考虑非敏感项集不被隐藏,在此前提下照顾敏感项集的隐藏,该方法将造成一些敏感项集隐藏失败;第 3 种方法称为完全隐藏最小副作用法(hiding sensitive patterns completely with minimum side effect on non-sensitive patterns,简称 HPCME),其思想是组合前两种方法的优点,在所有敏感项集都被成功隐藏的前提下,使非敏感项集被错误隐藏的数量最小.文献[17]在文献[16]的基础上对清洗矩阵的构造进行了一些改进.该方法在对一个敏感频繁项集进行隐藏时,至少要对该项集的 1 个子 2-项集进行隐藏,其目的是防止文献[18]所讨论的向前推理攻击.

此外,文献[19]综合了第 2 组算法中的 IGA/SWA 等算法,而文献[20]则对关联规则隐藏中的不同启发式策

略进行了比较.文献[21]提出了类似于Algo1b的关联规则隐藏算法PDA(priority-based distortion algorithm)算法和WSDA(weight-based sorting distortion algorithm)算法.其中,PDA算法在隐藏过程的每一步,对每一个可能的候选事务和牺牲项组合计算其将造成的正常规则丢失数目,根据计算结果选择最小正常规则丢失数目的候选事务和牺牲项进行清洗.该算法隐藏效果好,但计算代价大;而WSDA算法仅在候选事务选择上进行计算评估,选择会造成正常规则丢失较少的事务进行清洗,事务中牺牲项则随机选取.该算法计算代价相对较小,且能取得较好的隐藏效果.文献[22]在Algo2b的基础上引入最小副作用评估函数,预先计算出一个敏感项集在哪个项删除后可能产生最小的副作用.文献[23]则假定要隐藏的项(而非项集)给定,通过删除或增加项降低规则置信度来隐藏所有以敏感项作为右件的规则.该方法适用范围较窄.

2.1.2 数据阻塞法

2.1.2.1 基本思想

数据阻塞法通过向原始数据库引入不确定的"?",而非改变数据库使数据库失真的方式,来隐藏数据库中的敏感规则.通过引入不确定的问号,原来规则的支持度和置信度从一个确定值变成了不确定的支持度区间和置信度区间.若把原始数据库用布尔矩阵表示,则数据阻塞法的思想是通过将敏感事务中的"1"变成"?"或"0"变成"?"的方式来对原始数据库修改,使敏感规则的支持度或置信度落入一个不确定性区间,达到隐藏的目的.

如图4所示,规则$A\rightarrow C$在原始数据库中的置信度为100%,将第2个事务中的项C和第3个事务中的项A阻塞后,$A\rightarrow C$的置信度落入了一个不确定性区间[60%,100%],只要设定的最小置信度安全阈值大于此区间下界,就认为规则$A\rightarrow C$在新数据库中得到了隐藏保护.在运用数据阻塞法对敏感关联规则隐藏时,包括以下4个步骤:

① 根据敏感规则,找出待阻塞的候选事务集;

② 在候选事务集窗口中,选择要阻塞的候选事务;

③ 选择一定的策略,对选定的候选事务进行阻塞;

④ 重复①~③步,直到敏感规则的支持度区间下界或置信度区间下界降低到用户设定的安全阈值以下为止.

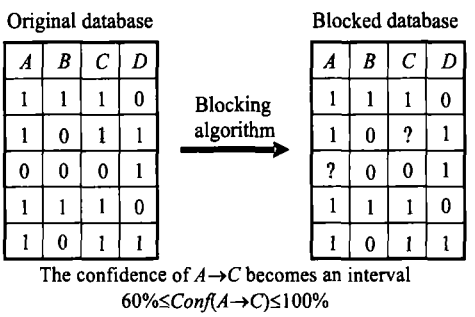

Fig.4 Basic idea of data-block method

图4 数据阻塞法的基本思想

2.1.2.2 典型算法

在数据阻塞法中,规则的支持度和置信度由于引入问号而具有一定的不确定性,使得单一取值的支持度和置信度变成了不确定的区间.对于项集A,其支持度从单一取值变成了支持度区间[min$sup(A)$,max$sup(A)$],其中,min$sup(A)$表示A中所有项对应的取值均为"1"的事务所占的比例,而max$sup(A)$表示A中所有项对应的取值为"1"或者"?"的事务所占的比例.而对于规则$A\rightarrow B$,置信度也从一个简单的取值变成了置信度区间:

$$[\min conf(A\rightarrow B),\ \max conf(A\rightarrow B)],$$

其中,min$conf(A\rightarrow B)$=min$sup(A\cup B)$/max$sup(A)$,max$conf(A\rightarrow B)$=max$sup(A\cup B)$/min$sup(A)$.

当没有未知值时,项集支持度区间的上下界相等,规则置信度区间的上下界也相等.随着未知值"?"的不断加入,区间的上下界开始分离,规则的不确定性程度也随之增加,当规则支持度区间下界低于最小支持度阈值

(minimum support threshold,简称 MST)或置信度区间下界低于最小置信度阈值(minimum confidence threshold,简称 MCT)时,就认为规则得到了隐藏.有 3 种方法实现规则 $A\rightarrow B$ 的隐藏:一是降低 min$sup(A\cup B)$的取值,以降低规则的支持度区间下界;二是降低 min$sup(A\cup B)$的取值,以降低规则的置信度区间下界;三是提高 max$sup(A)$的取值,以降低规则的置信度区间下界.下面分别介绍这 3 种方法.

(1) 降低支持度隐藏规则的 GIH(generating itemsets hiding)算法[25]

通过降低生成关联规则的项集的支持度区间下界,使其小于 *MST-SM*(*SM* 为安全边界)的敏感规则隐藏算法称作 GIH.设需要隐藏的敏感规则的集合为 R_h,生成其中敏感规则的频繁项集集合为 L_h.首先,把 L_h 中的项集按长度和支持度区间下界降序排列,然后从最长的项集开始隐藏.针对某一个项集,算法优先处理具有最大支持度区间下界的项,并从最短的事务开始阻塞它.整个算法执行过程如下:设 Z 是要隐藏的项集,先将 Z 中的项按其支持度区间下界降序排列,再把支持 Z 的事务集 T_Z 中的事务按长度升序排列.在每一步中,选择 Z 中具有最大支持度区间下界的项 i,并用"?"代替长度最短的事务中的项 i,这一过程将重复执行直到 Z 的支持度区间下界小于 *MST-SM*.在阻塞完某个事务中的一个项后,算法将更新 L_h 中其余项集的支持度区间下界以及支持它们的事务列表.算法选择支持度区间下界最大的项优先阻塞,是因为有较多的事务支持它们,从而能够降低由于阻塞这些项对其他项集所产生的负面影响;选择最短的事务,则是因为其中包含的项集数较少,同样是为了降低由于阻塞这些项对其他项集所产生的负面影响.

(2) 降低置信度隐藏规则的 CR(confidence reduction)算法[25]

CR 算法通过降低生成敏感规则 r 的项集的支持度来降低规则的置信度区间下界,从而达到隐藏规则 r 的目的.与前面介绍的 GIH 算法不同,CR 算法只选择规则的右件,也就是 $A\rightarrow B$ 中属于 B 的项来进行阻塞.这是因为,如果用未知值"?"代替属于规则 r 的左件 l_r 中的项,将会使得 l_r 的支持度区间下界 min$sup(l_r)$取值减小,从而导致规则 r 的置信度区间上界 max$conf(r)$增大,这与规则隐藏过程的目标,即要降低敏感规则的置信度取值是相冲突的.隐藏算法将执行到 min$sup(r)\leq$*MST-SM* 或 min$conf(r)\leq$*MCT-SM* 时结束.算法首先生成支持规则 r 的事务集 T_r,然后对 T_r 中的事务按长度升序排列.基于启发式的方式,CR 算法仍将选择具有最大支持度区间下界的项替换成"?",并从最短的事务开始处理,原因与前面的 GIH 算法相同.

(3) 降低置信度隐藏规则的 CR2 算法[25]

CR2 算法将用"?"代替规则 r 的左件 l_r 中取值为"0"的项,以此来提高 max$sup(l_r)$的值,进而降低规则 r 的置信度区间下界 min$conf(r)$,达到隐藏规则 r 的目的.给定一个规则 r,算法首先生成部分支持规则左件 l_r,但不完全支持规则右件 r_r 的事务集 T_r',并计算出 T_r' 中事务所包含的属于 l_r 的项数.然后,将从包含 l_r 中项数最多的事务 t 开始处理.事务 t 中不支持的 l_r 中的项,也就是相应取值为"0"的项,将被替换成"?",从而提高 l_r 的支持度区间上界 max$sup(l_r)$,降低规则 r 的置信度区间下界 min$conf(r)$.算法将执行到 min$conf(r)\leq$*MCT-SM* 时停止.在该方法中,只考虑那些不完全支持规则右件 r_r 的事务,否则,用"?"代替那些部分支持 l_r 且完全支持 r_r 的事务中,属于 l_r 但取值为"0"的项,会使规则 r 的支持度区间上界 max$sup(r)$提高,进而导致规则 r 的置信度区间上界 max$conf(r)$提高,这是不希望看到的.选择部分支持规则左件 l_r 且支持 l_r 中项数最多的事务,最好的情况就是一个事务恰好支持 l_r 中的$|l_r|-1$ 个项,只有一个项取值可被替换成"?".这使得对数据库的改动较小,从而降低对其他规则的负面影响.

近年来出现的基于数据阻塞方法进行关联规则隐藏的算法有文献[26]提出的 MCR 算法、MCR2 算法,分别是在 CR 算法和 CR2 算法的基础上引入量化测度评估函数,在每一步选择信息损失最小的阻塞方式,以减小新数据库和原数据库的差异,保证新数据库的可用性.同时,这些算法克服了原算法中的隐私泄露问题.文献[27]提出了基于数据阻塞的 ISL 算法和 DSR 算法,用于敏感频繁项集的隐藏.

2.1.3 数据重构法

2.1.3.1 基本思想

相对于数据变换法和数据阻塞法,数据重构法是一种比较新的敏感规则隐藏方法,由 Chen 等人在文献[28]中提出.不同于数据变换法和数据阻塞法通过对原始数据集的变换修改来隐藏规则,数据重构法的基本思想是抛开原始数据集,从原始数据集挖掘出的频繁模式出发,首先对频繁模式进行清洗,然后由清洗后的频繁模式反

向重构出一个新数据集,作为共享数据集.

如图 5 所示[28],数据重构法从原始数据集 $D1$ 挖掘出的频繁模式 $FS1$ 出发,对 $FS1$ 实施清洗算法得到 $FS2$,然后由 $FS2$ 出发反向重构一个新数据集 $D2$,作为共享的数据集发布.其中,在 Chen 等人最先提出的基于模式清洗反向重构方法的雏形中,$FS1$ 是带有支持数的频繁项集集合,从 $FS1$ 可以推出关联规则集合 R,考虑到项集间的包含关系,$FS1$ 就是一个带支持数的频繁项集格.$FS2$ 是清洗后的频繁模式集合,由 $FS2$ 推出的关联规则集合为 $R–R_h$,即除去敏感规则后的非敏感关联规则集.由图 5 可以看出,数据重构法在对敏感关联规则隐藏时,包括两个步骤:

① 模式清洗——应用一定的清洗算法,对项集格修改,产生新的项集格,满足由其产生的关联规则不包括敏感规则,但尽可能多地包括非敏感规则.

② 反向构造数据集——由清洗后的频繁模式反向构造新数据集.

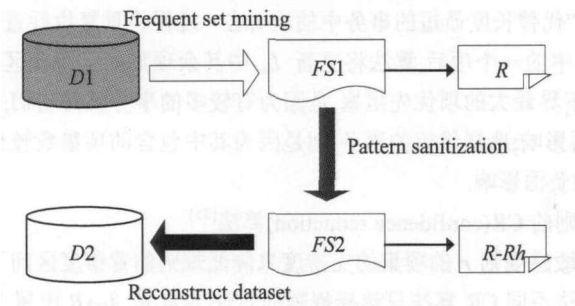

Fig.5 Basic idea of data-reconstruct method

图 5 数据重构法的基本思想

2.1.3.2 典型算法

(1) 项集格清洗算法

$I=\{i_1,i_2,\ldots,i_n\}$是项的集合,项集 X 为 I 的子集,如果 X 包括 k 个元素,则称为 k-项集.I 的一个幂集即 I 的所有子集构成集合 $P(I)$.$D=\{t_1,t_2,\ldots,t_n\}$是事务集.所有可以从 D 中挖掘到的项集及其偏序关系(子集关系)构成了项集空间 $P(I)$上的一个项集格,项集格中的每一项集均与其子集和父集关联,所有项集的支持度构成了格的频率集合.文献[28]提出了一种粗略的项集格清洗算法.算法首先识别出生成敏感关联规则的敏感频繁项集集合,然后根据隐藏策略修改敏感项集的支持数,接着调整项集格中与敏感项集相关的其他项集的支持数,使新项集格满足一致性关系.所谓一致性关系是指项集格各项集的支持数配置对应于一个真实数据集.具体地讲,为了使新项集格满足一致性关系,假定将项集格中的项集 X 的支持数由 $S1$ 降到 $S2$,则项集格中 X 的所有子集的支持数都要随之减少 $S1–S2$,而 X 的所有超集中支持数大于 $S2$ 的项集也要像 X 一样进行清洗,使其支持数降低到 $S2$;相反地,如果将 X 的支持数由 $S1$ 提高到 $S2$,则 X 的所有子集和超集也随之增加 $S2–S1$.算法重复地对项集格中的敏感频繁项集实施清洗(修改其支持数),直到由新项集格产生的关联规则不包括敏感规则为止.

文献[28]给出的项集格清洗算法并没有对新项集格的一致性关系做严格证明,对修改调整项集格的计算代价也没有进行评估,对如何从敏感关联规则识别敏感频繁项集以及如何选择隐藏策略也尚未给出具体的指导.然而,作为一种新的思路,相对于通过数据清洗来隐藏规则,项集格清洗通过对接近关联规则的频繁模式的控制来隐藏规则.直观上,基于模式清洗的方法可以避免数据清洗中大量的数据库扫描和 I/O 操作,同时,可较为直接地控制规则隐藏的效果,而不像启发式的数据清洗方法,隐藏的效果只能通过实验结果来验证.事实上,模式清洗本质上是在数据的知识体现层面进行的清洗,作为一种新的思路,目前还不存在非常成熟的模式清洗算法.可将数据清洗中的一些启发式规则所体现的思想应用到模式清洗中,开发出较为成熟的模式清洗算法.

(2) 反向构造数据集算法

给定频繁项集及其支持度,反向构造数据集,使其满足给定的频繁项集及其支持度约束,并且由其推导出的

其他项集的支持度小于阈值的问题称为反向频繁项集挖掘问题[34]或频繁项集可满足性问题[35].反向频繁项集挖掘是一个 NP 完全问题[34].

目前,对于反向频繁挖掘问题,研究者提出了若干方法.文献[36,37]提出了一种基于线性规划的方法,用以解决近似反向频繁项集挖掘问题.它旨在构造一个事务数据库,该数据库近似满足给定的频繁项集约束.文献[38]提出了基于 FP-tree 的反向频繁项集挖掘方法.借助于设计良好的启发式规则,该方法能从频繁项集快速找到一个近似满足频繁项集约束的 FP-tree,然后由 FP-tree 快速生成近似满足目标约束的数据库.就精确反向频繁项集挖掘而言,Calder 在文献[35]中给出了一种朴素的"产生-测试"方法,以便从给定的频繁项集"猜测"和水平地构造一个数据库,所谓"水平"是指算法逐事务地构造数据库.与"产生-测试"框架下水平地构造数据库相反,文献[39]提出了一种垂直的数据库生成算法,以便垂直地"猜测"和构建一个数据库,所谓"垂直"是指算法一列一列地构造数据库.然而,在"产生-测试"框架下,这两种算法都非常低效,因为它们本质上都属于简单的穷举搜索方法.目前,对于能否找到一种快速、有效的方法来解决反向频繁项集挖掘问题仍有待研究.

特殊情况下,当所有频繁项集和非频繁项集的支持数均已知时,很容易推出原始数据集.文献[28]证明了满项集格(格中所有项集的支持数都已知)和数据集的一一对应关系,并给出了所有项集支持数均已知的项集格反向构造数据集算法,其基本思想是,根据项集格中各项集的支持数,计算各项集的势,项集的势是指数据集中包含且仅包含该项集的事务数.具体地讲,算法从项集格顶层的 k 项集,即最大项集开始,依次计算每一层项集的势直至最底层.由于顶层 k 项集的势等于其支持度本身,其余层项集的势等于其支持度减去父集的势,算法可依次计算 $k-1,k-2,\ldots,1$ 层各项集的势,直到算出所有项集的势为止,从而得到对应的数据集.该算法的前提是项集格中所有项集的支持数均已知.当非频繁项集的支持数未知时,在一些特殊情况下,可利用启发式推导方法,从给定的频繁项集及其支持度估计出非频繁项集的支持度,进而利用该算法推导出原始数据集.

2.2 分类规则的隐藏

2.2.1 数据重构法

Natwichai 在文献[29,30]中提出了基于决策树反向重构数据集的分类规则隐藏方法,如图 6 所示.该方法的大致过程如下:首先基于规则的分类算法用到给定的数据集上获得分类规则;然后去除敏感规则,使用数据拥有者认定的非敏感分类规则构建一个数据生成器,即决策树;最后从决策树重构一个仅包含非敏感分类规则的新数据集.

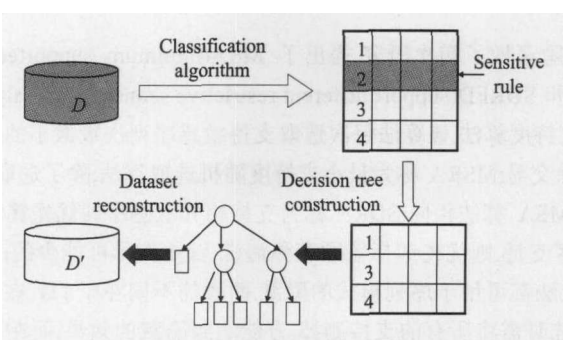

Fig.6 Data reconstruction method based on decision tree

图 6 基于决策树的数据重构法

在基于数据重构的分类规则隐藏方法中,核心的两个步骤是决策树构建算法和数据集重构算法.其中,决策树构建算法由非敏感规则集出发,构建出一棵决策树,而数据集重构算法则由决策树重构出满足决策树特征的目标数据集.

(1) 决策树构建算法

根据决策树构建过程所依据的特征信息的不同,目前有两种不同的决策树构建算法.一种是 Natwichai 最初在文献[29]中提出的基于规则集的决策树构建算法 RDTCA(rule-based decision tree construction algorithm).该

算法构建决策树所依据的特征信息是非敏感规则集.算法仅根据非敏感规则和规则划分数据集的能力构建决策树.另一种是在 RDTCA 基础上,将原始数据集中属性的信息增益特征引入决策树的构建过程.在文献[30]中提出的基于规则和属性信息增益的决策树构建算法 RGDTCA(rule & gain based decision tree construction algorithm)构建决策树依据的特征信息不仅包括非敏感规则,还包括原始数据集中各属性的信息增益.信息增益的引入使得基于 RGDTCA 形成的新数据集比基于 RDTCA 形成的新数据集在可用性上略胜一筹.

(2) 数据集重构算法

文献[29]所采用的由决策树重构数据集算法的基本过程如下:按均匀分布构建每条记录和分配其每一属性值,由决策树的相关路径作引导,根据决策树的终端叶节点将类标签分配到分类属性值中.使用均匀分布的数据生成器的好处在于,决策树中每一路径上重建的记录数可被估计出来.例如,若二元属性"性别"被选择根节点,则重建记录的该属性值中大约有一半是"男"、一半是"女".

2.2.2 数据抽样法

文献[31,32]提出了基于数据抽样的敏感分类规则隐藏方法.与大多数的研究工作围绕已确定的规则的隐藏和保护不同,文献[31,32]的研究工作则抛开要隐藏保护的规则是已知、确定的这种假定.其所要研究的问题是:如果无法确定数据中可能存在什么样的敏感规则需要被隐藏保护,应该如何做.比如,当试图将一些实际数据提供给新系统开发人员作为测试使用时,如果担心数据中会有一些潜在敏感知识被泄露,但无法确认哪一条规则是敏感的,在此情况下,如何将实际数据以一种安全的方式提交给新系统开发人员呢?

针对上述问题,文献[31,32]提出了基于数据抽样的方法来阻止和限制潜在的、不确定的敏感规则的发现.文献[31]研究了抽样大小与挖掘结果误差的函数关系,以便通过限制抽样大小来控制挖掘结果的期望误差,其目标是要为安全管理人员提供一个可用的工具,该工具能够帮助安全管理员决定:

- 给定挖掘结果质量约束,可被允许的样本集的大小;
- 给定样本大小,可被学习到的规则质量如何.

作为基于抽样的规则隐藏方法研究工作的开始,文献[31]研究了分类器的质量和随机抽样的样本大小之间的函数关系,其研究工作表明:只要随机抽取的样本数适度地小,就能排除数据挖掘对数据造成的知识方面的安全威胁,达到隐藏潜在知识的目的.基于抽样的规则隐藏方法的优点在于,它适用于隐藏从数据中发现的各种类型知识,而这种优点同时也造成了该方法的缺点——可能妨碍正常情况下的精确挖掘任务的展开.

2.3 其他类型知识的隐藏

文献[33]对敏感序列的隐藏做了初次研究,提出了 MSA(minimum supported algorithm),MSRA(minimum supported random algorithm)和 SDRFD(support different restrictive sequence first algorithm)这 3 种敏感序列隐藏算法.其中,MSA 称为最小支持度算法,该算法每次选取支持敏感序列次数最小的顾客交易(同一顾客可能支持同一序列多次)作为候选删除交易;MSRA 称为最小支持度随机选取算法,除了选取候选删除交易是以随机方式进行以外,该算法基本上与 MSA 算法相同;SDRF 称为支持相异敏感序列优先算法,其思想是,若多个敏感序列有共同的项,并由同一个顾客支持,则优先选择该顾客作清洗,以便作尽可能少的改动.事实上,关联规则隐藏方法中的很多启发式思想和做法都可用于序列模式的隐藏.两者的不同在于:(1) 在序列模式隐藏中,一位顾客可能会支持某一序列多次,清洗时需将所有的支持删掉,方能达到隐藏的效果,而在关联规则隐藏中,一个事务只支持某一规则一次,清洗操作相对简单;(2) 由于序列模式不涉及置信度,因此只能通过删除项、降低支持度的方式来隐藏序列,而关联规则隐藏则可通过增加项、降低置信度的方式实现.

目前,还没有相关文献对聚类、异常点等模式的隐藏进行研究,主要原因在于有关聚类、异常点等类型知识的挖掘技术本身尚不十分成熟且尚未得到广泛应用,人们一直热衷于寻求数据中的这些模式,相应的安全问题还未引起关注,使得聚类、异常点等类型的知识隐藏本身缺乏需求驱动和应用场景.随着聚类、异常点等知识发现技术本身的日新月异和广泛应用,相应的安全问题和知识隐藏技术将会逐渐得到人们的关注和研究.

3 方法的评估

在敏感规则隐藏方法及相关工具的开发与评估方面,建立恰当的评价标准是很重要的.通常,没有任何一种方法能在各方面都优于其他方法,而只可能是一种方法在某些方面的评测指标比另一种方法要好,其中包括隐藏的效果、执行的效率以及方法的适用性等.因此,向用户提供一系列有效的衡量标准,帮助他们根据数据特征和应用需求选出最合适的规则隐藏技术是很必要的.用于评估敏感规则隐藏方法的评测指标包括以下 3 个方面:

- 效能指标:是指对数据应用某一规则隐藏技术所达到的隐藏效果,包括多少敏感规则隐藏失败、多少非敏感规则丢失、新数据集中新添加了多少虚假规则、新数据集的可用性等.
- 性能指标:是指某一隐藏方法在执行时所花费的时间和空间代价,包括时间性能和空间性能.
- 适用性:是指某一隐藏方法适用于不同类型敏感规则、不同类型数据和不同应用背景的能力.

3.1 效能指标

如图 7 所示,R 代表从原始数据集 D 挖掘到的规则集合;R'代表从实施规则隐藏后的新数据集 D'挖掘到的规则集合;R_h表示根据用户的安全限制,需要隐藏的规则集合,即敏感规则集合;$\sim R_h$表示非敏感规则集合.规则隐藏后可能产生如图 7 中的阴影区域所示的 3 种问题:隐藏失败(hiding failure);规则丢失(rule missing);虚假模式(artificial pattern),又称幽灵规则(ghost rules)[25].图中①表示未被成功隐藏的敏感规则集合;②表示被错误隐藏的非敏感规则集合;③表示原数据集中不存在而在新数据集中突然出现的幽灵规则集合.对于这 3 种问题,分别用隐藏失败率(HF)、规则丢失率(MC)、虚假模式率(AP)这 3 个指标来度量.

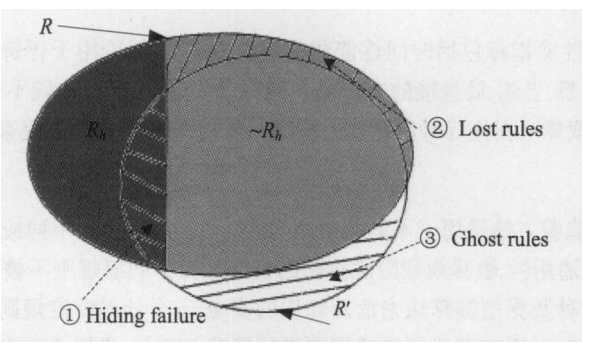

Fig.7 Three problems caused by rule hiding

图 7 规则隐藏带来的 3 类问题

(1) 隐藏失败率

对于隐藏失败率,用新数据中的敏感规则数目占原数据中敏感规则总数的比值来度量,其度量公式为

$$HF=\frac{R_h(D')}{R_h(D)}.$$

其中,$R_h(X)$代表从数据库 X 中挖掘出的敏感规则数目.隐藏失败率实际上度量的是算法对于敏感规则的隐藏保护效果,隐藏失败率越低,隐藏保护效果越好.理想情况下,HF 值为 0%.

(2) 规则丢失率

对于规则丢失率,用丢失的非敏感规则数占原有非敏感规则总数的比值来度量,其度量公式为

$$RM=\frac{\sim R_h(D)-\sim R_h(D')}{\sim R_h(D)}.$$

其中,$\sim R_h(X)$表示由数据库 X 挖掘到的非敏感规则的数目,分子的差值代表新数据与原数据相比丢失的规则数,分母为原数据中的非敏感规则总数.规则丢失率实际上度量的是敏感规则隐藏算法带来的"副作用",即由于隐藏敏感规则而造成非敏感规则的丢失情况,隐藏方法的重要指导原则就是尽量减少这种规则丢失的"副作用".

理想情况下,RM 值为 0%,即丢失的非敏感规则数为 0.

(3) 虚假模式率

对于虚假模式率,用新数据集中新出现的幽灵规则数占新数据集中的规则总数的比值来度量,度量公式为

$$AP=\frac{|R'|-|R\cap R'|}{|R'|}.$$

其中,分子表示新数据集中出现的虚假模式数,由新数据集中的规则总数减去同时出现在新规则集和旧规则集中的规则数表示;分母表示新数据集中的规则总数.虚假模式率度量的也是敏感规则隐藏算法带来的"副作用",即由于隐藏敏感规则而突然出现的幽灵规则.理想情况下,AP 值为 0%,即幽灵规则数为 0.

(4) 数据可用性

敏感规则隐藏方法的另一项效能指标是新数据的可用性,即在对原始数据实施隐藏算法后,生成的新数据集的可用程度.因为要对敏感规则进行隐藏,数据库就势必经过一定的修改,可能插入了一些假信息,也可能阻塞了一些数据的取值,而这些都会造成数据失真和可用性的下降.虽然有些敏感规则隐藏技术,例如数据抽样,并没有修改数据库中存储的数据,但由于其破坏了信息的完整性,所以数据的可用性同样会下降,并且该方法还可看作是对数据库整体进行了修改.通常,对数据库的改变越大,数据库反映的感兴趣的信息范围就越小.

通用的可用性度量可定义为新数据集与原始数据集的差异程度,如数据变动比例、信息损失量等.具体到特定的挖掘背景,挖掘结果越精确,数据的可用性越高.例如,在关联规则应用背景下,可用新规则集的支持度和置信度的变化(包括增加或减少)来度量.对于分类的情况,可用与关联规则挖掘相类似的指标来衡量.对于聚类的情况,在原始数据库和清洗后的数据库中,聚类项之间距离的方差可以作为评价信息损失的基础.

3.2 性能指标

敏感规则隐藏方法的性能指标包括时间性能和空间性能.时间性能用于评价一种敏感规则隐藏算法执行的快慢,可用时间复杂度估算.当然,最直接的方法是在同样的实验环境下,比较不同算法执行的时间.空间性能用于评价一种敏感规则隐藏算法在执行时所花费的内存、磁盘开销,可用空间复杂度估算.

3.3 适用性

适用性用于衡量不同隐藏方法适用于不同类型规则、不同类型数据和不同应用背景的能力,即方法的横向适用性水平,或者说算法的通用性.敏感规则隐藏算法的目的是保护敏感规则不被泄露,在此情况下不能忘记的是,恶意用户会试图通过各种数据挖掘算法来危及知识的安全.一个针对特定规则、特定类型数据、特定应用背景的规则隐藏算法,无法针对所有可能的敏感规则进行隐藏与保护.我们希望有一种算法能适用于不同的规则、不同的数据和不同的应用,对于已知的敏感规则、未知的潜在敏感规则都同时得到较好的隐藏和保护.

4 总结与展望

数据挖掘基于海量数据抽取新颖、有用的知识,在为企业带来价值和便利的同时,也给数据库中的隐私和信息安全带来潜在的威胁.伴随着数据库中知识发现 KDD 技术的日趋成熟,数据库中的知识隐藏 KHD 逐渐受到人们的重视和广泛关注.在数据发布前对敏感知识进行隐藏,可以防止机密知识的泄露和基于知识的恶意推理.

目前,绝大多数的研究工作围绕敏感关联规则的隐藏方法展开.这些方法可以分为 3 类:数据变换法、数据阻塞法和数据重构法.其中,数据变换法和数据阻塞法都是通过直接对原始数据库的清洗和修改来实现规则的隐藏,我们称其为数据清洗的方法.两者的不同点在于,数据变换法通过引入错误信息隐藏规则,而数据阻塞法通过引入不确定性隐藏规则;数据重构法从对频繁模式项集格的清洗出发,通过从清洗后的模式反向重构数据集的方式实现规则的隐藏和数据共享,我们称其为模式清洗的方法.数据清洗和模式清洗各有优、缺点,数据清洗的方法操作简单,但无法直接控制隐藏的效果;模式清洗的方法通过对模式的直接清洗容易控制隐藏的效果,但从清洗后的模式反向重构数据集本身就是一个难题,且反向重构出来的数据集除了频繁项集特征以外,会与

原始数据集其他方面的特征有很大不同.目前,现有方法中的绝大多数属于数据清洗的方法.根据不同的规则隐藏手段、不同的数据修改方式、不同的启发式清洗策略组合产生了很多算法,这些算法都力图在合理、有效的时间内,达到尽可能好的隐藏效果和尽可能高的数据可用性.现有的敏感分类规则的隐藏方法主要是数据抽样法和基于决策树的数据重构方法.其中,数据抽样法适合于要清洗的规则不确定的情况.敏感序列模式的隐藏基本上与敏感频繁项集的隐藏相似,已提出的方法很少.关于这些方法的比较见表 3.

Table 3 Comparison of different methods

表 3 不同方法的比较

	Data sanitization		Pattern sanitization	Sampling
	Data-Distort	Data-Block	Data-Reconstruct	Data-Sample
Privacy breaches	No privacy breaches	Many kinds of privacy breaches	No privacy breaches	No privacy breaches
Algorithm complexity	Simpler	More complicated	Complicated	Complicated
Character of shared database D'	Contain false information	Some uncertainty	Keep some pattern characters of D	Subset of D
Controllability of hiding effect	Indirect, not easy to control		Direct, easy to control	Difficult to control
Applicability	Known association rules, frequent itemsets and sequential patterns		Known association, classification rules	Potential, unknown rules

通过对国内外已有工作的调研、分析和总结,我们归结出 KHD 后续研究的 3 个方向:

- 数据修改技巧中基于目标距离的优化测度函数设计.目前,以数据变换和数据阻塞为代表的数据修改技巧研究成果相对较多,大部分算法采用的是直观、定性的启发式方法.为了尽可能地实现最优化隐藏,如何通过量化指标控制和引导隐藏过程仍有待解决.贪心法中基于目标距离的优化测度函数设计仍有待进一步研究,而伴随其中的是不同应用背景下数据可用性指标的研究.
- 数据重构技巧中的反向频繁项集挖掘.从给定的频繁项集倒推原始数据集的反向频繁项集挖掘[34-39]问题自从 2003 年正式被提出以来已经出现了一些研究成果,但是目前还缺乏十分有效的反向频繁项集挖掘算法.反向频繁项集挖掘在面向隐私数据保护的关联规则挖掘[40]、敏感关联规则的隐藏和关联规则挖掘基准数据集的生成等背景中有着广泛的应用.
- 基于数据抽样技巧的通用知识隐藏方法设计.现有的大多数敏感规则隐藏方法都是针对具体类型且已知确定的敏感规则.直观上,数据抽样技巧可以隐藏各种类型的敏感规则,目前,这方面的研究成果还很少.如何通过数据抽样对其他类型知识(比如聚类、异常点等)隐藏,进而设计出一种通用的知识隐藏方法(一种方法同时能隐藏多种类型的规则),是值得研究的.

References:

[1] Verykios VS, Bertino E, Fovino IN, Provenza LP, Saygin Y, Theodoridis Y. State-of-the-Art in privacy preserving data mining. SIGMOD Record, 2004,33(1):50−57.

[2] Johnsten T, Raghavan V. A methodology for hiding knowledge in databases. In: Clifton C, Estivill-Castro V, eds. Proc. of the IEEE ICDM Workshop on Privacy, Security and Data Mining. Maebashi: Australian Computer Society, 2002. 9−17.

[3] Atallah M, Bertino E, Elmagarmid A, Ibrahim M, Verykios VS. Disclosure limitation of sensitive rules. In: Scheuermann P, ed. Proc. of the IEEE Knowledge and Data Exchange Workshop (KDEX'99). Chicago: IEEE Computer Society, 1999. 45−52.

[4] O'Leary DE. Knowledge discovery as a threat to database security. In: Piatetsky-Shapiro G, Frawley WJ, eds. Knowledge Discovery in Databases. Menlo Park: AAAI Press; Cambridge: MIT Press, 1991. 507−516.

[5] Clifton C, Marks D. Security and privacy implications of data mining. In: Han JW, Lakshmanan LVS, Ng R, eds. Proc. of the ACM SIGMOD Workshop Data Mining and Knowledge Discovery. Vancouver: University of British Columbia, 1996. 15−19.

[6] Chang L, Moskowitz IS. Bayesian methods applied to the database inference problem. In: Jajodia S, ed. Proc. of the 12th Annual IFIP WG 11.3 Working Conf. on Database Security. Deventer: Kluwer Academic Publisher, 1998. 237−251.

[7] Chang L, Moskowitz IS. An integrated framework for database privacy protection. In: Thuraisingham BM, van de Riet RP, Dittrich KR, Tari Z, eds. Proc. of the 14th Annual IFIP WG 11.3 Working Conf. on Database Security. Deventer: Kluwer Academic

Publisher, 2000. 161−172.

[8] Dasseni E, Verykios VS, Elmagarmid A, Bertino E. Hiding association rules by using confidence and support. In: Moskowitz IS, ed. Proc. of the 4th Int'l Information Hiding Workshop (IHW 2001). Berlin: Springer-Verlag, 2001. 369−383.

[9] Verykios VS, Elmagarmid A, Bertino E, Saygin Y, Dasseni E. Association rule hiding. IEEE Trans. on Knowledge and Data Engineering, 2004,16(4):434−447.

[10] Oliveira SRM, Zaïane OR. Privacy preserving frequent itemset mining. In: Clifton C, Estivill-Castro V, eds. Proc. of the IEEE ICDM Workshop on Privacy, Security and Data Mining. Maebashi: Australian Computer Society, 2002. 43−54.

[11] Oliveira SRM, Zaïane OR. Algorithms for balancing privacy and knowledge discovery in association rule mining. In: Desai BC, Ng W, eds. Proc. of the 7th Int'l Database Engineering and Applications Symp. Hong Kong: IEEE Computer Society, 2003. 54−63.

[12] Oliveira SRM, Zaïane OR. Protecting sensitive knowledge by data sanitization. In: Wu XD, Tuzhilin A, eds. Proc. of the 3rd IEEE Int'l Conf. on Data Mining (ICDM 2003). Melbourne: IEEE Computer Society, 2003. 613−616.

[13] Sun X, Yu PS. A border-based approach for hiding sensitive frequent itemsets. In: Han JW, Wah BW, Raghavan V, Wu XD, Rastogi R, eds. Proc. of the 5th IEEE Int'l Conf. on Data Mining (ICDM 2005). Houston: IEEE Computer Society, 2005. 426−433.

[14] Menon S, Sarkar S, Mukherjee S. Maximizing accuracy of shared databases when concealing sensitive patterns. Information Systems Research, 2005,16(3):256−270.

[15] Gkoulalas-Divanis A, Verykios VS. An integer programming approach for frequent itemset hiding. In: Yu PS, Tsotras VJ, Fox EA, Liu B, eds. Proc. of the ACM 15th Conf. on Information and Knowledge Management. Arlington: ACM Press, 2006. 748−757.

[16] Lee G, Chang CY, Chen ALP. Hiding sensitive patterns in association rules mining. In: Wong E, Kanoun K, eds. Proc. of the 28th Int'l Computer Software and Applications Conf. (COMPSAC 2004). Piscataway: IEEE Computer Society, 2004. 424−429.

[17] Wang ET, Lee G, Lin YT. A novel method for protecting sensitive knowledge in association rules mining. In: Chen IR, Ibbett R, Mei H, eds. Proc. of the 29th Annual Int'l Computer Software and Applications Conf. (COMPSAC 2005). Edinburgh: IEEE Computer Society, 2005. 511−516.

[18] Oliveira SRM, Zaïane OR, Saygin Y. Secure association rule sharing. In: Dai H, Srikant R, Zhang C, eds. Proc. of the 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2004). Berlin: Springer-Verlag, 2004. 74−85.

[19] Oliveira SRM, Zaïane OR. A unified framework for protecting sensitive association rules in business collaboration. Int'l Journal of Business Intelligence and Data Mining, 2006,1(3):247−287.

[20] Pontikakis ED, Verykios VS, Theodoridis Y. On the comparison of association rule hiding heuristics. In: Proc. of the 3rd Hellenic Data Management Symp. (HDMS 2004). 2004.

[21] Pontikakis ED, Tsitsonis A, Verykios VS. An experimental study of distortion-based techniques for association rule hiding. In: Farkas C, Samarati P, eds. Proc. of the 18th Annual IFIP WG 11.3 Working Conf. on Data and Applications Security (DBSec 2004). Sitges: Kluwer Academic Publisher, 2004. 325−339.

[22] Zhan JY. A study for association rule hiding using the evaluation of side-effect cost [MS. Thesis]. Tainan: University of Tainan, 2005 (in Chinese with English abstract).

[23] Wang SL, Lee YH, Billis S, Jafari A. Hiding sensitive items in privacy preserving association rule mining. In: Wieringa P, ed. Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetics (SMC 2004). New York: IEEE, 2004. 3239−3244.

[24] Zhang W, Chen Y, Zou HB, Zhou T. Boolean rule hiding algorithm based on inverted file. Computer Engineering, 2005,31(14): 97−98 (in Chinese with English abstract).

[25] Saygin Y, Verykios VS, Clifton C. Using unknowns to prevent discovery of association rules. SIGMOD Record, 2001,30(4):45−54.

[26] Hintoglu AA, İnan A, Saygin Y, Keskinoz M. Suppressing data sets to prevent discovery of association rules. In: Han JW, Wah BW, Raghavan V, Wu XD, Rastogi R, eds. Proc. of the 5th IEEE Int'l Conf. on Data Mining (ICDM 2005). Houston: IEEE Computer Society, 2005. 645−648

[27] Wang SL, Jafari A. Using unknowns for hiding sensitive predictive association rules. In: Zhang D, Khoshgoftaar TM, Shyu ML, eds. Proc. of the IEEE Int'l Conf. on Information Reuse and Integration. IEEE Systems, Man and Cybernetics Society, 2005. 223−228.

[28] Chen X, Orlowska M, Li X. A new framework of privacy preserving data sharing. In: Matwin S, Adams C, Chang LW, Zhan J, eds. Proc. of the IEEE ICDM Workshop on Privacy and Security Aspects of Data Mining. Brighton: IEEE Computer Society, 2004.

47-56.

[29] Natwichai J, Li X, Orlowska M. Hiding classification rules for data sharing with privacy preservation. In: Tjoa AM, Trujillo J, eds. Proc. of the 7th Int'l Conf. on Data Warehousing and Knowledge Discovery. LNCS 3589, Berlin: Springer-Verlag, 2005. 468-477.

[30] Natwichai J, Li X, Orlowska M. A reconstruction-based algorithm for classification rules hiding. In: Dobbie G, Bailey J, eds. Proc. of the 17th Australasian Database Conf. (ADC 2006). Hobart: Australian Computer Society, 2006. 49-58.

[31] Clifton C. Using sample size to limit exposure to data mining. Journal of Computer Security, 2000,8(4):281-307.

[32] Clifton C. Protecting against data mining through samples. In: Atluri V, Hale J, eds. Proc. of the 13th Annual IFIP WG 11.3 Working Conf. on Database Security. Deventer: Kluwer Academic Publisher, 1999. 193-207.

[33] Chen ZX. Privacy preserving of sequential pattern mining [MS. Thesis]. Taizhong: Providence University, 2006 (in Chinese with English abstract).

[34] Mielikainen T. On inverse frequent set mining. In: Clifton C, Du WL, eds. Proc. of the IEEE ICDM Workshop on Privacy Preserving Data Mining. Melbourne: IEEE Computer Society, 2003. 18-23.

[35] Calders T. Computational complexity of itemset frequency satisfiability. In: Deutsch A, ed. Proc. of the 23rd ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS). Paris: ACM Press, 2004. 143-154.

[36] Wu XT, Wu, Y, Wang YG, Li YJ. Privacy-Aware market basket data set generation: A feasible approach for inverse frequent set mining. In: Kargupta H, Srivastava J, Kamath C, Goodman A, eds. Proc. of the 5th SIAM Int'l Conf. on Data Mining. Newport Beach: SIAM, 2005. 103-114.

[37] Wang YG, Wu XT. Approximate inverse frequent itemset mining: Privacy, complexity, and approximation. In: Han JW, Wah BW, Raghavan V, Wu XD, Rastogi R, eds. Proc. of the 5th IEEE Int'l Conf. on Data Mining (ICDM 2005). Houston: IEEE Computer Society, 2005. 482-489.

[38] Guo YH, Tong YH, Tang SW, Yang DQ. A FP-tree-based method for inverse frequent set mining. In: Bell D, Hong J, eds. Proc. of the 23rd British National Conf. on Databases (BNCOD 2006). LNCS 4042, Berlin: Springer-Verlag, 2006. 152-163.

[39] Chen X, Orlowska M. A further study on inverse frequent set mining. In: Li X, Wang S, Dong ZY, eds. Proc. of the 1st Int'l Conf. on Advanced Data Mining and Applications (ADMA). LNCS 3584, Berlin: Springer-Verlag, 2005. 753-760.

[40] Zhang P, Tong YH, Tang SW, Yang DQ, Ma XL. An effective method for privacy preserving association rule mining. Journal of Software, 2006,17(8):1764-1774 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1764.htm

附中文参考文献:

[22] 詹景逸.运用边际效应成本评估之关联法则隐藏演算法研究[硕士学位论文].台南:台南大学,2005.

[24] 张伟,陈芸,邹汉斌,周霆.基于倒排文件的布尔规则隐藏算法.计算机工程,2005,31(14):97-98.

[33] 陈肇勋.序列样式探勘的隐私权保护[硕士学位论文].台中:静宜大学,2006.

[40] 张鹏,童云海,唐世渭,杨冬青,马秀莉.一种有效的隐私保护关联规则挖掘方法.软件学报,2006,17(8):1764-1774. http://www.jos.org.cn/1000-9825/17/1764.htm

郭宇红(1979－),女,河南洛阳人,博士生,主要研究领域为数据挖掘,数据仓库.

童云海(1973－),男,博士,副教授,主要研究领域为数据挖掘,联机分析处理.

唐世渭(1939－),男,教授,博士生导师,CCF高级会员,主要研究领域为数据库,信息系统.

杨冬青(1945－),女,教授,博士生导师,CCF高级会员,主要研究领域为数据库,信息系统.