

基于隐私保护的序列模式挖掘

方炜炜^{1,2} 谢 伟² 黄宏博¹ 夏红科¹

(北京信息科技大学计算中心 北京 100192)¹ (清华大学经济管理学院 北京 100010)²

摘 要 隐私保护是当前数据挖掘领域的一个研究热点,其目标是在不暴露原始数据信息的前提下准确地实现挖掘任务。针对隐私保护序列模式挖掘问题,提出了项集的布尔集合关系概念,设计了基于随机集和扰乱函数对原始序列库进行数据干扰的方法模型,并通过扰乱函数的特性还原出原始序列库的频繁序列模式的真实支持度,完成了在保护原始数据隐私的前提下准确地挖掘出频繁序列模式的任务。理论分析和实验结果表明,该方法模型具有很好的数据隐私保护性、挖掘结果准确性和算法执行高效性。

关键词 序列模式,数据挖掘,隐私保护,数据干扰

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.035

Sequential Pattern Mining Based on Privacy Preserving

FANG Wei-wei^{1,2} XIE Wei² HUANG Hong-bo¹ XIA Hong-ke¹

(Computing Center, Beijing Information Science and Technology University, Beijing 100192, China)¹

(School of Economics and Management, Tsinghua University, Beijing 100010, China)²

Abstract Privacy-preserving is one of the most important topics in data mining. Its' main aim is realizing mining task in the context of uncovering original data information. In this paper, aiming to solve privacy-preserving sequential pattern mining problem, we proposed new concepts about item's Boolean set relationship, and designed data perturbation method based on random set and random function, which can obtain the support of original sequential database. Theoretical analysis and experiment results demonstrate that this method can achieve good performance in terms of privacy preserving, mining quality and efficiency.

Keywords Sequential pattern, Data mining, Privacy preserving, Data perturbation

1 引言

序列模式挖掘^[1]由 Agrawal 和 Skrikant 于 1995 年提出,旨在从具有时间特性的海量数据库中挖掘出有效的、容易理解的和具有时序关系的知识模式。由于该技术在 Web 访问模式、DNA 分析、医疗诊断、购物分析等领域具有广泛的应用前景,现已成为数据挖掘领域的一个重要研究分支。

国内外学者对如何提高序列模式的挖掘效率做了很多研究,并提出了一些有效的挖掘算法。如 Agrawal 和 Skrikant 在文献[1,2]中基于 Apriori 性质,分别提出了 AprioriAll 算法和 GSP 算法,通过先生成频繁序列候选集再按照宽度优先的搜索策略产生频繁序列模式;J Han 和 J Pei 在文献[3,4]中基于前缀投影,分别提出了 FreeSpan 算法和 PrefixSpan 算法,通过投影项集递归划分搜索空间,在投影序列子数据库中挖掘序列模式,减少了搜索空间,提高了算法性能;Guralnik V 在文献[5]中基于树投影技术,提出了分布式计算机的并行挖掘序列模式算法。然而在序列模式的挖掘过程中存在着隐私安全问题,一旦包含有敏感信息的原始数据泄漏给外界,会对数据拥有方的隐私和信息安全构成威胁。

Agrawal 和 Skrikant 于 2000 年在文献[6]中对实现隐私

保护的数据挖掘 PPDM 作了定义:“采用某种措施使数据挖掘过程中使用到的原始隐私数据和产生的中间敏感频繁集及规则等不对外公开,保证数据挖掘过程的可靠性,避免由于信息的泄漏给参与挖掘的主体带来损失”。目前国内外 PPDM 技术的研究成果主要包括关联规则^[7-9]、聚类^[10-12]和决策树分类^[13,14]。有关实现隐私保护的序列模式挖掘技术目前在国内外尚缺乏深入研究。

Rizvi S J 在文献[7]中提出基于随机化应答技术的布尔关联规则挖掘算法 MASK,采用正态分布或高斯分布的随机数据对原始数据库进行数据扰乱,然后基于 Warner 模型对原始数据进行特征重构;文献[8]基于阻塞技术实现敏感关联规则的隐藏,即采用不确定符号“?”代替某些特定的属性值,从而减少真实数据的发布;Vaidya J 在文献[9]中提出基于安全多方计算技术 SMC 的分布式关联规则挖掘方法,在多站点间实现加密数据的统计分析;文献[13]基于随机响应技术分别处理决策树分类和贝叶斯分类的挖掘任务,敏感数据的属性值通过一种应答特定问题的方式,以概率 θ 间接提供给外界。

本文针对隐私保护序列模式挖掘问题,提出了项集的布尔集合关系概念,设计了基于随机集和扰乱函数对原始序列库进行数据干扰的方法模型,并通过扰乱函数的特性还原出

到稿日期:2015-12-10 返修日期:2016-04-28 本文受国家自然科学基金重点项目(60675030),国家自然科学基金项目(60875029),2015 年北京市组织部优秀人才培养项目,2016 年北京教育委员会科技面上项目资助。

方炜炜(1979-),女,博士,副教授,主要研究方向为数据挖掘;谢 伟 男,博士,教授,主要研究方向为技术经济;黄宏博(1976-),男,博士,主要研究方向为人工智能;夏红科(1979-),女,博士,主要研究方向为数据挖掘。

原始序列库的频繁序列模式的真实支持度,完成了在保护原始数据隐私的前提下准确地挖掘出频繁序列模式的任务。

本文第2节阐述了相关定义;第3节给出了问题的描述,并提出了解决问题的总体框架模型;第4节详细阐述了模型中的两个关键技术,即随机集和扰乱函数的设计和频繁序列的真实支持度的计算,并给出了模型的核心算法;第5节从数据的隐私保护质量、频繁序列模式的挖掘质量和算法的执行效率3方面进行理论分析和实验验证;最后总结全文并展望未来。

2 相关定义

定义1 设 $I = \{I_1, I_2, \dots, I_z\}$ 是所有项的集合,集合 $e \subseteq I$ 称为项集。序列 s 是项集的有序列表,记为 $s = \langle e_1 e_2 e_3 \dots e_n \rangle$,其中 $e_j (1 \leq j \leq n)$ 是个项集,称为序列 s 的元素,项序列中的实例数目,即 $l = \sum_{1 \leq j \leq n} |e_j|$ 称为序列的长度,长度为 l 的序列称为 l 序列。

定义2 序列数据库 S 是元组 $\langle SID, s \rangle$ 的集合,其中 SID 是序列 $ID, s = \langle e_1 e_2 e_3 \dots e_n \rangle$ 是一个序列。若序列 $a = \langle a_1 a_2 a_3 \dots a_m \rangle$,存在整数 $1 \leq j_1 < j_2 < \dots < j_m \leq n$,使得 $a_1 \subseteq e_{j_1}, a_2 \subseteq e_{j_2}, \dots, a_m \subseteq e_{j_m}$,则序列 a 称为 s 的子序列, s 称为 a 的超序列,记为 $a \subseteq s$ 。 a 在序列数据库 S 中的支持度是数据库中包含 a 的元组的个数,即 $sup(a) = |\{\langle SID, s \rangle | \langle SID, s \rangle \in S \wedge (a \subseteq s)\}|$ 。

定义3 给定一个正整数 min_sup ,如果序列 a 在序列数据库 S 中的支持度满足 $sup(a) \geq min_sup$,则序列 a 称为频繁序列。

定义4 给定序列数据库 S 和用户指定的最小支持度阈值 min_sup ,序列模式发现的任务就是找出支持度大于或等于 min_sup 的所有序列。

定义5 设 $I = \{I_1, I_2, \dots, I_z\}$ 是所有项的集合,关系 $f: I \rightarrow B = \{b_1, b_2, \dots, b_z\} (b_i \in \{0, 1\})$ 称为项集的布尔集合关系。

定理1 项集 $e = \{I_{j_1}, I_{j_2}, I_{j_3}, \dots, I_{j_{z-1}}, I_{j_m}\} (1 \leq j_1 < j_2 < \dots < j_m \leq z)$ 的布尔集合关系记为 $\langle b_1, b_2, \dots, b_z | b_i = 1 (i \in \{j_1, j_2, \dots, j_m\}) \wedge b_i = 0 (i \notin \{j_1, j_2, \dots, j_m\}) \rangle$ 。

定理2 设序列 $s = \langle e_1 e_2 e_3 \dots e_n \rangle$,序列 $a = \langle a_1 a_2 a_3 \dots a_m \rangle$ 。序列 s 中包含 m 个元素的子序列集合记为 $\{s_i (1 \leq i \leq P_n^m) | s_i = \langle e_{j_1} e_{j_2} \dots e_{j_m} \rangle (1 \leq j_1, j_2, \dots, j_m \leq n)\}$ 。序列 a 的布尔集合关系表示为 p ,子序列 s_i 的布尔集合关系表示为 q_i ,若 $p \wedge q_i = p$,则 $a \subseteq s_i$ 。序列 a 的支持度 $sup(a) = |\{s_i | p \wedge q_i = p\}|$ 。

例1 设数据库 D 中项目集合 I 为 $\{1, 2, 3, 4, 5\}$,8-序列 $s \langle \{1, 2\}, \{2, 3, 4\}, \{2, 4, 5\} \rangle$,序列 a 为 $\langle \{2\}, \{5\} \rangle$ 。序列 s 中包含2个元素的子序列集合为 $\langle \{1, 2\}, \{2, 3, 4\} \rangle, \langle \{2, 3, 4\}, \{2, 4, 5\} \rangle, \langle \{1, 2\}, \{2, 4, 5\} \rangle$,对应的布尔集合关系为 $\langle \langle 11000, 01110 \rangle, \langle 01110, 01011 \rangle, \langle 11000, 01011 \rangle \rangle$,序列 a 的布尔集合关系表示为 $\langle \langle 01000, 00001 \rangle \rangle$,将子序列集合中的元素分别与序列 a 进行布尔集合 \wedge 运算,可求解出序列 a 的超序列为 $\langle \{2, 3, 4\}, \{2, 4, 5\} \rangle$ 和 $\langle \{1, 2\}, \{2, 4, 5\} \rangle$,序列 a 在 s 中的支持度为2。

定义6 设 (U, Σ, P) 是一个概率空间, $(2^W, \sigma(\beta))$ 是另一可测空间,其中 $\beta \subseteq 2^W$,若映射 $R: U \rightarrow 2^W$ 是 $\Sigma \rightarrow \sigma(\beta)$ 可测的,即对于任意 $\alpha \in \sigma(\beta)$ 有 $\{u \in U | R(u) \in \alpha\} \in \Sigma$,则称 R 是一个随机集。

例2 设 $D = (U, I, V, R)$ 是一个信息系统, U 为论域, I 为项目集合, $V = \bigcup_{i \in I} V_i, V_i$ 表示项目 i 是否存在记录中,其值为 $\{0, 1\}, R = \{R_i | i \in I\}, R_i: U \rightarrow V_i$ 为项目描述函数, R 的每一个选择 R_i 可以解释为与信息系统相容的一种特殊的可能的描述。

3 问题描述与架构

本节首先给出序列模式挖掘中隐私保护的问题描述,然后提出解决问题的总体架构。

3.1 问题描述

设项目数据集 $I = \{I_1, I_2, \dots, I_z\}$,序列库 $S = \{s_1, s_2, \dots, s_m\}$ 中任意元素 $s_i = \langle e_1 e_2 e_3 \dots e_n \rangle$,其中项集 $e_j = \{I_{\beta_1}, I_{\beta_2} \dots\} \subseteq \{I_1, I_2, \dots, I_z\}$ 。对于用户设定的最小支持度阈值 min_sup ,若存在序列 $a = \langle a_1 a_2 a_3 \dots a_k \rangle$ 的支持度 $sup(a) \geq min_sup$,则 a 为 S 中的频繁序列模式。实现隐私保护的序列模式挖掘即在不访问原始序列数据库 S 的前提下,准确地挖掘出所有支持度大于 min_sup 的频繁序列。

基于 Apriori 性质的频繁序列模式的产生主要分为3步:1)连接:频繁 $(k-1)$ -序列与频繁 $(k-1)$ -序列连接,产生候选 k -序列;2)剪枝:基于先验原理,剪掉那些非频繁子序列的候选序列;3)计数:扫描序列库 S ,对剪枝后的候选序列计算支持度,识别频繁序列模式。

常规序列模式挖掘方法需要扫描原始序列库 S ,因而容易造成隐私信息的泄露。而现有的应用于关联规则的隐私保护技术^[7-9]不能直接应用于序列模式挖掘,这是因为两者的逻辑结构不相同。关联规则中的每条数据记录仅仅由若干个项目 I 组成,而序列数据库则较为复杂,每条序列由多个具有时间先后顺序的项集 e 组成,且每个项集 e 又再包含若干个项目 I 。

3.2 总体架构

为了解决上述问题,本文提出了一种新的数据扰乱方法,即通过扰乱函数 F 向原序列库 S 添加随机集 R ,以此为基础实现序列模式挖掘的隐私保护方法,总体架构如图1所示。

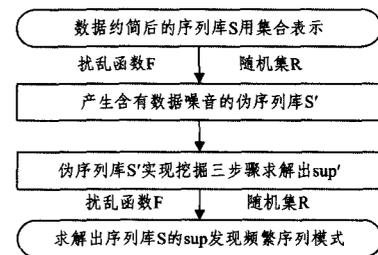


图1 总体架构

模型分为4步实现:1)采用粗糙集理论或主成分分析 PCA 技术实现数据约简,从而减少属性相关性,并用布尔集合形式表示序列库 S ;2)通过扰乱函数 F 和随机集 R 产生伪序列库 S' ;3)在伪序列库 S' 上进行连接、剪枝和计数,求解支持度 sup' ;4)根据函数 F 和随机集 R 特性,重构原序列库 S 的 sup ,从而发现频繁序列模式。

4 序列模式挖掘的隐私保护方法

本文所提出的隐私保护模型架构中,最关键的工作有两点:1)采用随机集和扰乱函数对原始数据库进行数据扰乱;2)计算频繁序列的支持度。

4.1 随机集合扰乱

由于项目数据集 $I = \{I_1, I_2, \dots, I_z\}$ 包含 z 个项目元素, 原数据序列库 $S = \{s_1, s_2, \dots, s_m\}$ 包含 m 个序列元素, 因此产生包含 m 个元素的随机集 $R = \{R_i | R_i = r_{i1} r_{i2} \dots r_{iz}, r_{ij} = \{0, 1\}, 1 \leq i \leq m, 1 \leq j \leq z\}$. 扰乱函数 $F: x' = \bar{x}r + x\bar{r}$, 其中 x 为原数据的布尔形式, r 为随机元素的布尔形式.

对于原数据序列库 S 中的任意序列 $s_i = \langle e_1 e_2 e_3 \dots e_n \rangle (1 \leq i \leq m)$ 中的每个项集 e_j , 即 $\{I_{\beta 1}, I_{\beta 2}, \dots\}$ 经随机集元素 $R_i = \{r_{i1} r_{i2} \dots r_{iz}\}$ 和扰乱函数 F 处理得到扰乱后项集 $e_j' = \{I'_{\beta 1}, I'_{\beta 2}, \dots\}$.

例 3 设项目数据集 $I = \{1, 2, 3, 4, 5\}$, 序列库 $S = \{s_1, s_2\}$, 其中 s_1 为 $\langle \{1, 2\}, \{2, 3, 4\} \rangle$ (其布尔形式为 $\langle 11000, 01110 \rangle$), s_2 为 $\langle \{2, 4, 5\} \rangle$ (其布尔形式为 $\langle 01011 \rangle$). 产生随机集 $R = \{R_1, R_2\}$, 其中 $R_1 = \{01101\}$, $R_2 = \{01101\}$. s_1 中每个项集与 R_1 按函数 F 处理得到 $s_1' = \langle 10101, 00011 \rangle$, s_2 中每个项集与 R_2 按函数 F 处理得到 $s_2' = \langle 00110 \rangle$. 从而获得的扰乱后伪序列库 $S' = \langle \langle \{1, 3, 5\}, \{4, 5\} \rangle, \langle 3, 4 \rangle \rangle$.

将信息系统 D 中的每个序列 s_i 经过扰乱函数 F 和随机集 R 处理后, 得到一个伪装序列 s_i' , 其形式上仍然是与 s_i 长度相同的 0-1 序列, 但其信息内容却被隐藏起来. 由于频繁序列模式的发现是基于所有原始序列的统计信息而不是一个详细的序列, 因此可以通过扰乱函数 F 的特征求解出 k -序列的支持度, 从而挖掘出频繁序列模式.

4.2 计算 1-序列的支持度

设 π 表示原 1-序列 $s_1 = \langle \{I_t\} \rangle$ 的支持度, π' 表示扰乱后的 1-序列 $s_1' = \langle \{I_t'\} \rangle$ 的支持度. 依据扰乱函数 F , 有 $I_t' = \bar{I}_t r + I_t \bar{r}$ 成立. 随机集 R 中任意元素 $r \in \{0, 1\}$, 设 r 取值为 1 的概率为 θ .

$$\begin{aligned} \pi' &= P(I_t' = 1) \\ &= P(I_t = 1) P(\bar{r} = 1) + P(\bar{I}_t = 1) P(r = 1) \\ &= \pi(1-\theta) + (1-\pi)\theta \\ &= \pi(1-2\theta) + \theta \end{aligned}$$

因此, 通过扰乱后 1-序列 s_1' 的支持度 π' 可求解出原始 1-序列 s_1 的支持度 π 为 $(\pi' - \theta) / (1 - 2\theta)$.

4.3 计算 k -序列的支持度

在计算 k -序列的支持度前, 先以 2-序列举例, 然后推理归纳出 k -序列支持度的计算方法. 频繁 1-序列 $\langle \{I_{\beta 1}\} \rangle$ 和 $\langle \{I_{\beta 2}\} \rangle$ 产生候选 2-序列有 3 种情况: $\langle \{I_{\beta 1}\}, \{I_{\beta 2}\} \rangle$, $\langle \{I_{\beta 2}\}, \{I_{\beta 1}\} \rangle$ 和 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$. 以 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$ 为例, 另外两种情况同理. 按照定义 5 及性质 2, 设原始序列库 S 和扰乱后伪序列 S' 分别与 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$ 进行布尔 \wedge 操作的支持度计算, 如表 1 所列.

表 1 序列库与 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$ 进行 \wedge 操作

运算结果	S	S'
包含 $I_{\beta 1}, I_{\beta 2}$	π_{11}	π'_{11}
仅含 $I_{\beta 1}$	π_{10}	π'_{10}
仅含 $I_{\beta 2}$	π_{01}	π'_{01}
不含 $I_{\beta 1}, I_{\beta 2}$	π_{00}	π'_{00}

因为 F 函数 $I_t' = \bar{I}_t r + I_t \bar{r} (1 \leq t \leq z)$, 可得如下等式: $M_{(2)} \cdot \pi_{(2)} = \pi_{(2)'}$, 其中

$$M_{(2)} = \begin{pmatrix} (1-\theta)^2 & (1-\theta)\theta & \theta(1-\theta) & \theta^2 \\ (1-\theta)\theta & (1-\theta)^2 & \theta^2 & \theta(1-\theta) \\ \theta(1-\theta) & \theta^2 & (1-\theta)^2 & (1-\theta)\theta \\ \theta^2 & \theta(1-\theta) & (1-\theta)\theta & (1-\theta)^2 \end{pmatrix}$$

$$\pi_{(2)} = \begin{pmatrix} \pi_{00} \\ \pi_{01} \\ \pi_{10} \\ \pi_{11} \end{pmatrix}, \pi_{(2)'} = \begin{pmatrix} \pi'_{00} \\ \pi'_{01} \\ \pi'_{10} \\ \pi'_{11} \end{pmatrix}$$

通过在伪序列库 S' 上执行如图 1 所示的步骤 3 求解出 $\pi_{(2)'}$, 然后由 $\pi_{(2)} = M_{(2)}^{-1} \cdot \pi_{(2)'}$ 求解出 π_{11} , 即得到 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$ 在原始序列库 S 中的支持度.

同理, 原始序列库 S 和扰乱后序列 S' 分别与候选 k -序列进行布尔 \wedge 操作的支持度计算结果为 $\pi_{(k)}$ 和 $\pi_{(k)'}$, 表示如下:

$$\pi_{(k)} = \begin{pmatrix} \pi_{00\dots 00} \\ \pi_{00\dots 01} \\ \dots \\ \pi_{11\dots 10} \\ \pi_{11\dots 11} \end{pmatrix}, \pi_{(k)'} = \begin{pmatrix} \pi'_{00\dots 00} \\ \pi'_{00\dots 01} \\ \dots \\ \pi'_{11\dots 10} \\ \pi'_{11\dots 11} \end{pmatrix}$$

且满足等式 $M_{(k)} \cdot \pi_{(k)} = \pi_{(k)'}$ (1)

其中 $M_{(k)}$ 表示如下:

$$M_{(k)} = \begin{pmatrix} M_{00} & M_{01} & \dots & M_{0, 2^k-1} \\ M_{10} & M_{11} & \dots & M_{1, 2^k-1} \\ \dots & \dots & \dots & \dots \\ M_{2^k-1, 0} & M_{2^k-1, 1} & \dots & M_{2^k-1, 2^k-1} \end{pmatrix}$$

定理 3 $M_{(k)}$ 中任一元素 $M_{ij} (0 \leq i, j \leq 2^k - 1)$ 可表示成 k 位的布尔形式 $\bar{\pi}_i \pi_j + \pi_i \bar{\pi}_j$, 其值为 $\sum_{c=1}^k \theta \cdot \sum_{c=1}^{k-c} (1-\theta) (\alpha$ 表示其布尔形式中值为 1 的个数).

例 4 在计算频繁 2-序列 $\langle \{I_{\beta 1}, I_{\beta 2}\} \rangle$ 支持度的过程中, 有等式 $M_{(2)} \cdot \pi_{(2)} = \pi_{(2)'}$ 成立, 且 $M_{(2)}$ 中的任意元素均可验证定理 3. 以元素 M_{13} 为例, 其布尔形式表示为 $\bar{\pi}_1 \pi_3 + \pi_1 \bar{\pi}_3 = \bar{01} \wedge 11 + 01 \wedge \bar{11} = 10 \wedge 11 + 01 \wedge 00 = 10 + 00 = 10$, 其值为 $\theta(1-\theta)$.

证明: $M_{(k)}$ 中每一元素 $M_{ij} (0 \leq i, j \leq 2^k - 1)$ 表示随机集元素 $\{r_1 r_2 \dots r_k\}$, 其布尔表示形式为 k 位, 其值为 $\sum_{i=1}^k P(r_i)$, 其中 $P(r_i)$ 表示 r_i 取值为 0 或 1 的概率值. 等式(1)亦可表示成 i 个等式.

$$\sum_{j=0}^{2^k-1} M_{ij} \cdot \pi_j = \pi_i \quad (2)$$

因为 F 函数 $I_t' = \bar{I}_t r + I_t \bar{r}$, 所以:

$$\pi_j' = \bar{\pi}_j \wedge r_1 r_2 \dots r_k + \pi_j \wedge \bar{r}_1 \bar{r}_2 \dots \bar{r}_k \quad (3)$$

原始数据序列 π_j 经函数 F 和随机集元素 $\{r_1 r_2 \dots r_k\}$ 扰乱后生成 π_j' , 即伪装成式(2)中所显示的原始序列中的 π_i , 即有等式(4)成立.

$$\pi_i = \bar{\pi}_j \wedge r_1 r_2 \dots r_k + \pi_j \wedge \bar{r}_1 \bar{r}_2 \dots \bar{r}_k \quad (4)$$

根据布尔运算可得:

$$r_1 r_2 \dots r_k = \pi_i \pi_j + \pi_i \bar{\pi}_j$$

所以 M_{ij} 的 k 位布尔形式为 $\bar{\pi}_i \pi_j + \pi_i \bar{\pi}_j$, 其值为 $\sum_{i=1}^k P(r_i) =$

$$\sum_{c=1}^k \theta \cdot \sum_{c=1}^{k-c} (1-\theta).$$

定理 3 证明完毕.

因此在计算频繁 k -序列的支持度时, 可通过在伪序列库 S' 上求解出 $\pi_{(k)'}$, 然后由 $M_{(k)} \cdot \pi_{(k)} = \pi_{(k)'}$ 等式求解出 $\pi_{11\dots 11}$, 即得到频繁 k -序列 $\langle \{I_{\beta 1}, I_{\beta 2}, \dots, I_{\beta k}\} \rangle$ 在原始序列库 S 中的支持度.

4.4 算法

依据上面所描述的模型架构, 对传统 AprioriAll 算法进行改进, 设计由算法 1 和算法 2 构成的序列模式隐私保护算

法,其中算法 1 由数据拥有者执行,实现原序列数据库的数据扰乱;算法 2 由半可信第三方挖掘者执行,现在伪数据库上准确挖掘出频繁序列模式。

算法 1 序列数据扰乱算法 SDPA (Sequence Data Perturbation Algorithm)

输入:原始序列库 $S = \{s_1, s_2, \dots, s_m\}$, 扰乱函数 F , 概率 θ , 项目数据集中包含项目元素的个数 z

输出:扰乱后的伪数据库 S'

执行方:数据拥有者

1. For($i=1; i \leq m; i++$) //遍历原始序列库
2. according θ to generate $R_i = \{r_{i1}, r_{i2}, \dots, r_{iz}\}$ ($r_{it} \in \{0, 1\}, 1 \leq t \leq z$) //产生随机集
3. scan $s_i = \langle e_{i1}, e_{i2}, e_{i3}, \dots, e_{in} \rangle$ //获取原始序列
4. For($j=1; j \leq n; j++$)
5. scan $e_j = \{I_{j1}, I_{j2}, \dots\}$ //获取项集布尔形式
6. For($t=1; t \leq z; t++$)
7. $I'_t = \bar{I}_{tR_{it}} + I_t \cdot r_{it}$ //由扰乱函数生成扰乱后布尔形式
8. End for
9. get $e'_j = \{I'_{j1}, I'_{j2}, \dots\}$ //生成扰乱后项集
10. End for
11. get $s'_i = \langle e'_i, e'_2, e'_3, \dots, e'_n \rangle$ //生成扰乱后序列
12. End for

算法 2 序列模式的隐私保护算法 PPSPMA (Privacy-preserving Sequence Pattern Mining Algorithm)

输入:伪数据库 S' , 最小支持度阈值 \min_sup

输出:原始序列库 S 中的频繁序列模式 L

执行方:数据挖掘者

1. For($t=1; t \leq z; t++$) //候选 t -序列在伪序列库 S' 上的支持度
2. scan S' to compute $I_t, count'$
3. $I_t, count = (I_t, count' - \theta) / (1 - 2\theta)$
4. End for
5. $L_1 = \{I_t \mid I_t, count > \min_sup\}$ //生成 1-序列集
6. For($k=2; L_{k+1} \neq \emptyset; k++$) //伪序列库 S' 挖掘频繁序列模式
7. $C_k = \text{apriori_gen}(L_{k-1})$;
8. For each $s'_i \in S'$ do
9. $C_s = \text{subsequence}(C_k, s'_i)$
10. For each candidate $c \in C_s$ do
11. $c, count' = c, count' + 1$;
12. End for
13. End for
14. $c, count = M_{\{k\}}^{-1} \cdot c, count'$ //由扰乱函数求解出原序列库 S 的支持度
15. $L_k = \{c \mid c \in C_k \wedge c, count > \min_sup\}$
16. End for
17. Return $L = \cup L_k$

5 性能分析

对于实现隐私保护的数据挖掘方法,目前国际上尚没有一个统一的评价标准体系。本文将从数据的隐私保护质量、频繁序列模式的挖掘质量和算法的执行效率 3 方面进行理论分析和实验验证。

5.1 数据的隐私保护质量

实现隐私保护的数据挖掘即是在不提供原始数据信息的前提下准确地完成挖掘任务。在本模型中,通过扰乱集合和扰乱函数将原序列库 S 伪装成 S' 。对于序列库中每一个序列而言,若 S' 和 S 的差异性越大,则说明原始数据的隐私性被保护得越强。因此提出数据混淆性的量化指标 DC (Data Confusion) 公式:

$$DC = P_{\text{隐藏的序列项集}} + P_{\text{新产生的序列项集}} \quad (5)$$

其中, $P_{\text{隐藏的序列项集}}$ 表示 S 中出现但扰乱后在 S' 中被隐藏起来的序列项集的概率, $P_{\text{新产生的序列项集}}$ 表示 S 中未出现但扰乱后在 S' 中产生的序列项集的概率。

依据本模型中采用的扰乱函数 $F: x' = \bar{x}r + x\bar{r}$, 可以得到数据混淆的量化指标值 $DC = P(x=1, x'=0) + P(x=0, x'=1) = \theta$ 。该值说明当扰乱集合 R 中取值为 1 的元素越多即 θ 越趋近于 1 时, S' 和 S 的差异性越大。但数据的隐私保护性和挖掘结果的准确性是相互矛盾的,两序列库的差异性增加,将导致挖掘结果的准确度降低,所以应选取合适的 θ 值来保证隐私保护和准确挖掘的两者平衡。

5.2 频繁序列模式的挖掘质量

实现隐私保护的数据挖掘方法的最终目标是准确地挖掘出频繁模式。虽然 S' 提供的序列集与 S 不相同,但由于频繁序列模式的挖掘是基于所有序列的统计信息而不是一个特定的序列,如果 S' 提供的序列统计信息与 S 越近似,那么其挖掘结果就越精确。因此提出挖掘准确度的量化指标 MA (Mining Accuracy) 公式:

$$MA = \sigma^2(\hat{\pi}) = E((\hat{\pi} - E(\hat{\pi}))^2) \quad (6)$$

其中, $\hat{\pi}$ 代表算法 2 中的 $c, count$, 是由 S' 上挖掘出来的 $c, count'$ 经过扰乱函数 F 计算出来的序列支持度, $E(\hat{\pi})$ 表示在原始序列数据库 S 上真实的支持度, MA 公式的计算结果表示通过 S' 挖掘计算出来的支持度与原 S 上真实的支持度之间的偏差情况。

$$\text{对于频繁 1-序列而言, } MA = \frac{\pi_{(1)}' - \theta}{1 - 2\theta} \cdot \frac{(1 - \pi_{(1)}' - \theta)}{1 - 2\theta}$$

$\pi_{(1)}'$ 表示在 S' 上挖掘出来的频繁 1-序列。

对于频繁 k -序列而言, $MA = E((\hat{\pi} - E(\hat{\pi}))^2) = E(\hat{\pi}^2) - E(\hat{\pi})^2 = (M_{\{k\}}^{-1} \cdot \pi'_{(k)}) (1 - (M_{\{k\}}^{-1} \cdot \pi_{(k)}'))$, 其中 $\pi_{(k)}'$ 表示在 S' 上挖掘出来的频繁 k -序列。

5.3 算法的执行效率

设项目数据集 $I = \{I_1, I_2, \dots, I_z\}$ 包含 z 个项目元素, 原数据序列库 $S = \{s_1, s_2, \dots, s_m\}$ 包含 m 个序列元素。按照本文提出的模型分 4 步实现挖掘任务的过程, 第 1 步用布尔集合形式表示序列库 S 的时间复杂度为 $O(mz)$; 第 2 步通过扰乱函数和随机集产生序列库 S' 的时间复杂度为 $O(mz)$; 第 3 步求解支持度 sup' 过程中, 设频繁 1-序列的个数为 c_1 , 扫描序列库 S' 求解 $I_t, count$ 的时间复杂度为 $O(mzc_1)$; 设频繁 k -序列的个数为 c_k , 扫描序列库 S' 求解 $c, count'$ 的时间复杂度为 $O(mzc_k)$; 第 4 步由 $M_{\{k\}}^{-1} \cdot c, count'$ 重构原序列库 S 的支持度 sup 的时间复杂度为 $O(2^{k-1}c_k)$ 。

5.4 实验结果

实验使用的硬件平台是 Pentium IV 2.2GHz CPU, 1GB 内存, 操作系统是 Windows XP, 算法用 VC 实现, 采用 IBM Almaden 实验室提供的人工数据生成程序 `assocgen`^[1], 数据参数如下: 数据库大小为 503MB, 序列总数为 3100000 个, 项目总数为 10000 个, 序列中含项集的平均个数为 15。

采用 AprioriAll 算法^[1] 与本文提出的 PPSPMA 算法进行比较, 考察两者的运行时间和 PPSPMA 算法的挖掘质量。实验结果如图 2—图 4 所示。

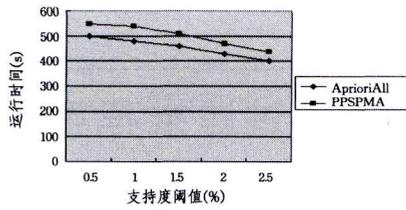


图2 支持度阈值变化时算法的运行时间

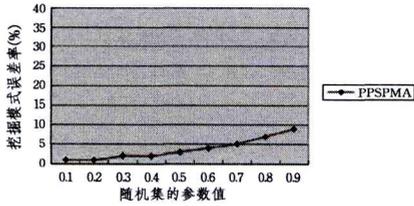


图3 参数 θ 变化时算法的挖掘误差率

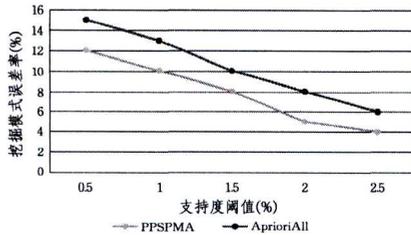


图4 支持度阈值变化时挖掘质量的比较

图2为AprioriAll算法和PPSPMA算法在设置不同的支持度阈值时运行时间的比较情况,由于PPSPMA算法是在AprioriAll算法的基础上添加数据干扰和真实支持度还原的步骤,因此运行时间比AprioriAll算法稍长。图3为当随机集 R 的参数 θ 设置发生改变时,PPSPMA算法所挖掘出的频繁序列模式的误差率 $error = \left| \frac{L-L'}{L} \right|$,其中 L 和 L' 分别表示通过AprioriAll算法和PPSPMA算法挖掘出来的频繁序列模式。随着 θ 越趋近于1,原始序列库得到更强的隐私保护,但其挖掘准确度也随之降低; θ 越趋近于0时,原始序列库的隐私保护性降低,提供给挖掘者的几乎是原始信息,其挖掘准确度接近于AprioriAll算法。图4为AprioriAll算法和PPSPMA算法在设置不同的支持度阈值时挖掘质量的比较情况,虽然PPSPMA算法运行时间比AprioriAll算法长,但是在同样的支持度阈值的情况下挖掘模式的误差率要低,表明PPSPMA算法的挖掘质量更好。因而在现实应用中,应根据可容许的误差率范围来设置扰乱参数 θ ,以兼顾隐私保护和准确挖掘。

结束语 本文针对序列模式隐私保护挖掘问题展开研究,提出了序列项集的布尔集合关系概念,设计了一种适用于序列模式挖掘的数据扰乱隐私保护方法,即基于随机集和扰乱函数对原始序列库进行数据扰乱,然后通过扰乱函数特性还原出原始序列库的真实支持度。在数据隐私保护质量、挖掘模式质量和算法执行效率3方面对PPSPMA算法进行理论分析和实验验证。

在未来的工作中,将进一步改进该挖掘算法的执行效率,并将此数据扰乱方法扩充应用到其它数据挖掘任务中。

参考文献

[1] Agrawal R, Srikant R. Mining Sequential patterns[C]// Pro-

ceeding of the 11th International Conference on Data Engineering, Los Alamitos, CA; IEEE Computer Society Press, 1995; 3-14

[2] Srikant R, Agrawal R. Mining sequential patterns: Generalizations and Performance Improvements[C]// Proceeding of the 5th International Conference on Extending Database Technology, Berlin; Springer-Verlag, 1996; 3-17

[3] Han J, Pei J, et al. FreeSpan: Frequent Pattern-projected Sequential Pattern Mining[C]// Proceeding of the 6th International Conference on Knowledge Discovery and Data Mining, New York; ACM Press, 2000; 335-359

[4] Pei J, Han J, et al. PrefixSpan: Mining Sequential Patterns Effectively by Prefix Protected Pattern Growth[C]// Proceeding of the 17th International Conference on Data Engineering, Los Alamitos, CA; IEEE Computer Society Press, 2001; 215-224

[5] Guralnik V, Garg N, Karypis G. Parallel Tree Projection Algorithm for Sequence Mining[C]// LNCS2150, 2001; 310-320

[6] Agrawal R, Srikant R. Privacy-preserving data mining [C] // Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, United States: ACM, 2000; 439-450

[7] Rizvi S J, Haritsa J R. Maintaining data privacy in association rule mining[C]// Proceedings of the 28th International Conference on Very Large Databases (VLDB), Hong Kong, China, 2002; 682-693

[8] Saygin Y, Verykios V S, Elmagarmid A K. Privacy preserving association rule mining [C] // Proc. of the 12th International Workshop on Research Issues in Data Engineering (RIDE), San Jose, USA, 2002; 151-158

[9] Vaidya J, Clifton C1 Privacy preserving association rule mining in vertically partitioned data [C] // the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002; 639-644

[10] Wu Ying-jie, Tang Qing-ming, Ni Wei-wei, et al. Private Preserving Data Publishing Based on Clustering [J]. Computer Research and Development, 2013, 50(3); 578-593 (in Chinese)

吴英杰, 唐庆明, 倪巍伟, 等. 基于聚类杂交的隐私保护轨迹数据发布算法 [J]. 计算机研究与发展, 2013, 50(3); 578-593

[11] Li Yang, Hao Zhi-feng. Private Preserving K-means Clustering Methods Research [J]. Computer Science, 2013, 3(1); 39-45 (in Chinese)

李杨, 郝志峰. 差分隐私保护 k-means 聚类方法研究 [J]. 计算机科学, 2013, 3(1); 39-45

[12] Fang Wei-wei, Yang Bing-ru, Xia Hong-ke. Private Preserving Clustering Model Based on SMC [J]. System Engineering and Electric Technology, 2012, 34(7); 567-578 (in Chinese)

方炜炜, 杨炳儒, 夏红科. 基于 SMC 的隐私保护聚类模型 [J]. 系统工程与电子技术, 2012, 34(7); 567-578

[13] Xiong Ping, Zhu Tian-qing. One Private Preserving Algorithm Based on Decision Tree [J]. Computer Application Research, 2014, 31(10); 354-360 (in Chinese)

熊平, 朱天清. 一种面向决策树构建的差分隐私保护算法 [J]. 计算机应用研究, 2014, 31(10); 354-360

[14] Zhang Cheng-xue. Private Preserving Algorithm Based on Data Victoria Distribution [J]. Shandong Technology University Paper, 2011, 30(2); 30-38 (in Chinese)

张成学. 数据垂直分布的线性规划的隐私保护算法 [J]. 山东科技大学, 2011, 30(2); 30-38