基于序列格的隐私时序模式挖掘方法

彭慧丽[1,2],金凯忠[1],付聪聪[1],付 楠[1],张啸剑[1]
(1. 河南财经政法大学计算机与信息工程学院,河南郑州 450002;2. 河南广播电视大学信息工程学院,河南郑州 450046)

摘 要: 基于差分隐私的时间序列模式挖掘方法中,序列的最大长度以及添加拉普拉斯噪声的多少直接制约着挖掘结果的可用性.针对现有时间序列模式挖掘方法全局敏感度过高、挖掘结果可用性较低的不足问题,提出了一种基于序列格的差分隐私下时间序列模式挖掘方法 PrivTSM(Differentially Private Time Series Pattern Mining).该方法首先利用最长路径的策略对原始数据库进行截断处理;在此基础上,采用表连接操作生成满足差分隐私的序列格;结合序列格结构本身的特性,合理分配隐私预算,提高输出模式的可用性.理论分析表明 PrivTSM 方法满足 ε-差分隐私,基于真实数据库上实验结果表明,PrivTSM 方法的准确率 TPR(True Postive Rate)和平均相对误差 ARE(Average Relative Error)明显优于 N-gram 和 Prefix-Hybrid 方法.

关键词: 差分隐私;时间序列;全局敏感度;数据挖掘;数据截断;序列格
中图分类号: TP309.2 **文献标识码:** A **文章编号:** 0372-2112 (2020)01-0153-11
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.01.019

Private Time Series Pattern Mining with Sequential Lattice

PENG Hui-li[1,2],JIN Kai-zhong[1],FU Cong-cong[1],FU Nan[1],ZHANG Xiao-jian[1]
(1. *School of Computer& Information Engineering,Henan University of Economics and Law,Zhengzhou,Henan* 450002,*China*;
2. *School of Information Engineering,Henan Radio&Television University,Zhengzhou,Henan* 450046,*China*)

Abstract: Many methods of differentailly private time series pattern mining have been proposed,while in those methods,the length of sequence pattern and Laplace noise directly constrain the utility of the mining results. To address the questions caused by the global query sensitivity and lower utility of the existing works,an efficient method,called PrivTSM(differentially Private Time Series Pattern Mining) is proposed,which is based on sequence lattice for mining time series pattern with differential privacy. This method relies on the longest path strategy to truncate the original database;based on the truncated database,this method uses the table join operation to construct a differentially private sequence lattice. Furthermore,this method uses the property of the sequence lattice structure itself to allocate privacy budget reasonably and boost the accuracy of the noisy counts. PrivTSM satisfies ε-differential privacy through theoretical analysis. The experimental results on real datasets show that the accuracy (TPR) and average relative error (ARE) of the PrivTSM are better than those of the N-gram and Prefix-Hybrid algorithms.

Key words: differential privacy;time series;global sensitivity;data mining;data truncate;sequential lattice

1 引言

随着智能设备及移动应用的迅猛发展,产生了大量的时间序列数据,对此类数据进行分析可以从中获得对交通规划、以及公共服务等有用的信息.然而,该类数据通常蕴含丰富的敏感信息.因此,如何在保护个人隐私的同时,尽可能提高挖掘方法的可用性是主要的技术挑战.

近年来,差分隐私[1~3]已成为一种标准的保护模型,该模型要求数据库中任何一个用户的存在都不应显著地改变任何查询的结果,从而保证了每个用户加入该数据库不会对其隐私造成危险.基于差分隐私保护模型出现了多种基于差分隐私的时间序列模式挖掘方法.现存的方法包括 STM-Full[4]、DiffPart[5]、PT-Sam-

收稿日期:2018-10-18; 修回日期:2019-5-23;责任编辑:蓝红杰
基金项目:国家自然科学基金(No. 61502146, No. 91646203, No. 91746115, No. 61772131, No. 61702161);河南省自然科学基金(No. 162300410006);河南省科技攻关项目(No. 162102310411);河南省教育厅高等学校重点科研项目(No. 16A520002);河南省高等学校青年骨干教师培养计划(No. 2017GGJS084);河南财经政法大学青年拔尖人才资助计划

ple[6]等方法. 上述方法均通过构建前缀树挖掘频繁序列,然而这些方法存在诸多问题:(1)挖掘序列模式通常具有较高的全局敏感度,上述方法仅适用于短序列数据;(2)上述方法均采用前缀树作为索引结构,而在大规模数据中挖掘频繁模式会产生大量候选序列,进而造成前缀树过于冗余;(3)随着前缀树高度的增长,每一个子分割的序列数量会急剧减小,由于拉普拉斯噪声的注入,会严重影响最终序列模式的可用性. 结合上述方法存在的问题,本文提出了一种基于序列格的时间序列模式挖掘方法 PrivTSM(Differentially Private Time Series Pattern Mining),该方法蕴含三种操作:(1)为了解决问题 1,提出了一种利用最长路径策略获取最优截断长度的方法,该策略可以有效降低全局敏感度以及序列截断带来的误差;(2)为了解决问题 2,借鉴 SPADE[7]技术引入时间戳与表连接操作,并利用隐私序列格挖掘时间序列模式;(3)为了解决问题 3,基于序列格本身特性,提出一种有效的隐私预算分配策略;(4)理论分析了 PrivTSM 满足 ε-差分隐私. 结合 4 种真实数据验证了 PrivTSM 在准确率(TPR)和平均相对误差(ARE)方面均优于 N-gram 和 Prefix-Hybrid 方法.

2 相关工作

基于差分隐私的频繁序列挖掘已存在多种方法. 文献[8]采用(α,β)-userfullness 技术与 STM-Full[4]方法相结合,提出了一种混合粒度前缀树划分方法 Prefix-Hybrid,然而该方法却忽视了序列中蕴含的时间信息. 文献[9]所提出的 N-gram 方法通过限制序列最大长度来降低序列维度对挖掘结果的影响. 然而,该方法仅适用于短序列. 文献[10]结合长序列导致的信息损失与挖掘效率低问题,提出了智能权重截取方法和支持度评估方法. PFS^{2}方法[11]利用基于采样的候选剪枝技术减少候选序列数目,而该方法在序列重构时未考虑不同序列的重要性. 文献[12]利用 k-means 方法对轨迹中位置进行聚类,并通过指数机制实现最优泛化. 上述满足差分隐私的频繁模式挖掘方法均存在各自的不足,为此本文提出一种基于隐私序列格的时间序列模式挖掘 PrivTSM 方法,该方法能够有效降低全局敏感度,以及提高挖掘结果的可用性.

3 定义与概念

3.1 差分隐私

相比于传统保护模型,差分隐私保护模型具有两个显著的特点:(1)定义了一个相当严格的攻击模型,不关心攻击者拥有多少背景知识,假设攻击者已掌握除某一条记录之外的所有记录信息,该攻击者也无法从统计结果中推测出这条记录是否存在于数据库中;(2)对隐私保护水平给出了严格的定义和定量分析评估方法.

定义 1 近邻数据库:设 $T=\{s_1,s_2,\cdots,s_n\}$ 为原始时序数据库. $T'=\{s_1,s_2,\cdots,s_{r-1},s_{r+1},\cdots,s_n\}$,$T$ 与 T' 相差一条记录,则二者互为近邻数据库.

结合 T 与 T' 给出 ε-差分隐私的形式化定义,如定义 2 所示.

定义 2 ε-差分隐私:给定一个时间序列模式挖掘方法 A,Range(A)为 A 的输出范围,若方法 A 在 T 与 T' 上任意输出结果 O($O\in$Range(A))满足下列不等式,则 A 满足 ε-差分隐私.

$$\Pr[A(T)=O]\leqslant\exp(\varepsilon)\times\Pr[A(T')=O] \tag{1}$$

其中,ε 表示隐私预算,用于衡量差分隐私保护的强度.

实现差分隐私保护需要噪声机制的介入,拉普拉斯与指数机制是实现差分隐私的主要技术. 而所需要的噪声大小与其响应查询函数 f 的全局敏感性密切相关.

定义 3 全局敏感性:设 f 为某个查询函数,且 $f:T\rightarrow R^d$,f 的全局敏感度为

$$\Delta f=\max_{T,T'}\|f(T)-f(T')\|_1 \tag{2}$$

其中,R 为映射的实数空间,d 为 f 的查询维度.

3.2 时间序列模式挖掘

时间序列是具有时间戳的数据. 在含有空间位置信息的时序数据库中,时间序列 s 可表示为 $s=\{(x_1,y_1,t_1),\cdots,(x_n,y_n,t_n)\}$,其中 $t_1<t_i<t_n$,x_i,y_i表示地理空间中的位置点信息,t_i为 x_i和 y_i位置的时间戳信息. 一条时间序列如图 1 所示,时序数据库就是这些序列的集合.

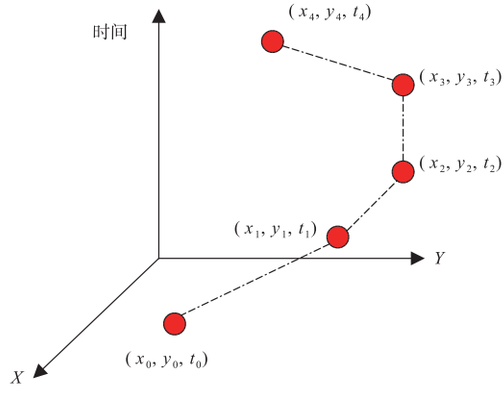


图1 时间序列

序列模式是否频繁,取决于模式的支持度与最小支持度阈值的关系,时间序列模式挖掘的任务就是在时序数据库中找出所有大于或等于最小支持度阈值的时间序列. 例如,表 1 包含 5 条时间序列,每一条序列按照时间先后排序,其中 A 表示某一敏感位置. 攻击者想

窃取“Mary 是否去过 A 位置”这种信息,通过分析获知 A 为频繁模式,其真实支持度计数为 3. 如果攻击者已经知道有 2 人去过 A 位置,则可推理出 Mary 也去过 A 位置,进而导致 Mary 的敏感位置泄露. 因此,如何在保护个人隐私的同时,尽可能提高挖掘方法的可用性是主要的技术挑战.

表 1　时序数据库

ID	序列	ID	序列
1	$A\rightarrow B\rightarrow C$	4	$A\rightarrow D\rightarrow C$
2	$B\rightarrow C$	5	$A\rightarrow B\rightarrow D\rightarrow C$
3	$B\rightarrow D\rightarrow C\rightarrow A$		

本文利用序列格结构挖掘时间序列模式,该结构是一种层次数据结构,格中第 i 层所有节点代表的序列是第 $i-1$ 层所有节点代表的序列排列组合的所有情况. 利用序列格挖掘时间序列模式的过程中,采用 Apriori[13] 的思想,通过连接和剪枝操作获得候选序列及其支持度,并通过与最小支持度阈值比较得到频繁序列.

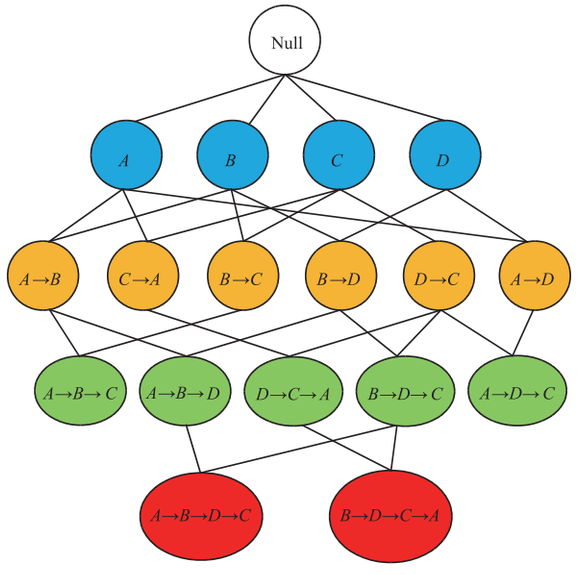


图2　序列格

但是,上述方法会产生大量候选序列,导致挖掘效率低下,因此本文借鉴 SPADE[7] 思想,采用表连接操作构建序列格. 图 2 是针对表 1 代表的时序数据库,通过表连接操作构建的序列格结构,其中最小支持度阈值为 1. 具体构建过程将在 4.3 节详细介绍.

4　基于序列格的时间序列模式挖掘方法

本节主要介绍 PrivTSM 算法的概述以及该算法的具体实现细节,其中包括序列截断策略、满足差分隐私的序列格构建方法、隐私预算分配策略以及判断 PrivTSM 算法是否满足 ε-差分隐私.

4.1　PrivTSM 算法整体描述

本小节描述 PrivTSM 算法的具体实现过程,如算法 1 所示:

算法 1　PrivTSM 算法

输入:时序数据库 T,隐私预算 ε,最小支持度阈值 min_sup
输出:满足 ε-差分隐私的频繁序列集合 F 及其噪声支持度
1. $\varepsilon=\varepsilon_1+\varepsilon_2+\varepsilon_3$
2. $\tilde{T}\leftarrow$Dynamic-Truncation(T,ε_1,min_sup)
3. $S_{\text{Lattice}}\leftarrow$Build-Noisy-Lattice($\tilde{T},\varepsilon_2$,min_sup)
4. $F\leftarrow$Generate-TSeries($S_{\text{Lattice}},\varepsilon_3$)
5. Return F

该算法主要包含三个过程:截断时序数据库过程(行 2)、构建序列格过程(行 3)、生成噪声频繁时间序列模式过程(行 4). 图 3 给出了 PrivTSM 算法的整体框架,4.2,4.3,4.4 节分别对上述三个过程做了详细介绍.

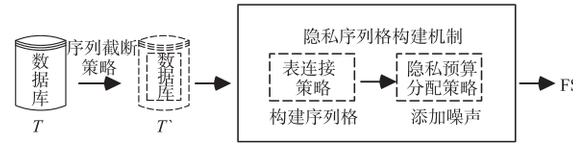


图3　PrivTSM算法整体框架

4.2　序列截断策略

4.2.1　全局敏感度分析

在时序数据上挖掘频繁隐私序列模式具有较高的全局敏感度. 然而,全局敏感度越大,添加的噪声越大,挖掘结果的可用性越低.

定理 1　给定时序数据库 T,T 中序列的最大长度为 $l_{\max}$,查询 $Q=[q_1,q_2,\cdots,q_n]$,用以查询 q_i 在数据库 T 中的支持度. q_i 的序列长度满足 $q_i\in I=[q_{\min},q_{\max}]$,$1\leqslant q_{\min}\leqslant q_{\max}$,查询 Q 的全局敏感度满足:

$$\Delta f_Q=\Delta I\times l_{\max},\Delta I=(q_{\max}-q_{\min})+1$$

定理 1 表明,全局敏感度与时序数据库的最大长度 $l_{\max}$ 成正比,即使时序数据库仅存在一个非常长的序列,挖掘出的所有频繁序列模式的支持度都需要添加非常大的噪声. 因此降低时序数据库的最大长度,将降低全局敏感度,进而提高数据的可用性.

现有方法大都是通过截断原始数据库来降低全局敏感度,这种方法事先需要定义一个最优截断长度,因此不可避免的会造成频繁信息的丢失. 在截断过程中,如果一个频繁序列模式被错误的当成非频繁序列模式,它的所有超集都会被误认为非频繁序列模式,这样会造成很大误差. 为了减少序列截断带来的误差,尽可能保留足够多的频繁信息. 本文利用最长路径的策略,提出一种变长序列截断方法,并将问题转化为求有向

无环图 DAG 中的最长路径.

4.2.2 变长序列截断

本小节介绍变长序列截断方法的主要思想及具体实现过程,如算法 2:

算法 2 Dynamic-Truncation 算法

输入:时序数据库 T,隐私预算 ε_1,最小支持度阈值 min_sup
输出:截断后时序数据库 $\tilde{T}$

1. $\varepsilon_1 = \varepsilon_{11} + \varepsilon_{12}$
2. $F_1 \leftarrow$frequent-one-sequences$(T, \varepsilon_{11}, \text{min_sup})$
3. $C_2 \leftarrow$Apriori(F_1)
4. $F_2 \leftarrow$frequent-two-sequences$(C_2, \varepsilon_{12}, \text{min_sup})$
5. $G=(V,E) \leftarrow$DAG(F_2) //利用频繁 2 序列前后位置关系构建有向无环图.
6. for each $s \in T$ do
7. for each $s[i] \in s$ do
8. $\text{Dis}[s[i]] = \max\{\text{Dis}[s[i]], \text{Suffix}[s[i]]\}$
9. if $\text{Suffix}[s[i]] > \text{Dis}[s[i-1]]$ or $\text{Suffix}[s[i]] = 0$ then
10. flag$\leftarrow i$
11. $\tilde{s} \leftarrow \text{Suffix}[s[\text{flag}]]$
12. $\tilde{T} \leftarrow \tilde{T} + \tilde{s}$
13. Return $\tilde{T}$

该算法主要思想是首先通过频繁 2 序列构建隐私有向无环图 G(行 2 ~ 5). 然后遍历时序数据库 T 的每条序列 s,同时遍历 G,找出 G 的最长路径,且该最长路径为 s 的子序列. 最后将该最长路径作为 s 的最优截断序列 $\tilde{s}$. 这里的最长路径指的是距离最长,且无环.

例 1 给定时序数据库 T 如表 1 所示,最小支持度阈值 min_sup = 2,假设频繁 1 序列及其噪声支持度为 $\{A:4, B:4, C:5, D:3\}$,频繁 2 序列及其噪声支持度为 $\{A->B:2, B->C:2, B->D:2, D->C:3\}$,则根据频繁 2 序列中项的前后位置关系生成的隐私 DAG 图如图 4 所示.

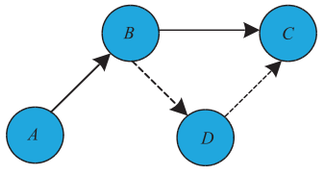


图 4 隐私DAG图

接下来需要通过隐私 DAG 图 G 来截断时序数据库 T. 通过上述分析可知,如果利用穷举法求解,时间复杂度为 $O(|T|n^2)$,因此本文利用动态规划的思想,递归的找到 s 中的最优截断序列 $\tilde{s}$,可以将时间复杂度降低为 $O(|T|n)$,递推公式如下:

$$\text{Dis}[s[i]] = \max\{\text{Dis}[s[i]], \text{Suffix}[s[i]]\} \quad (3)$$

$\text{Dis}[s[i]]$表示 G 中以序列 s 的项 $s[i]$ 开头的最长路径,且该最长路径为 s 的子序列. $\text{Suffix}[s[i]]$表示 G 中以序列 s 的项 $s[i]$结尾的最长路径,且该最长路径为 s 的子序列.

例如,给定时序数据库 T 如表 1 所示. 对于 T 中序列 $s_1 = \{B->D->C->A\}$,其子序列有 24 中情况,需从中选择一个子序列作为最优截断序列.

首先计算得到 $\text{Suffix}[s[i]]$,通过遍历 s_1 的每一项 $s_1[i]$,得到数组 Suffix = [0,1,2,0]. 根据式(3),递归的更新数组 Dis,同时记住最优截断序列的末尾位置,最终得到结果 Dis = [2,2,2,2],flag = 3,即 DAG 中以 s_1 的某项开头的最长路径且为 s_1 的子序列的路径为 $B->D->C$,如图 4 中虚线所示. 同理,给定 $s_2 = \{A->D->C\}$,则 DAG 中以 s_2 的某项开头的最长路径且为 s_2 的子序列的路径为 $D->C$.

定理 2 Dynamic-Truncation 算法满足 ε_1-差分隐私.

证明 Dynamic-Truncation 算法用到隐私预算的地方为算法 2 的行 2 与行 4,已知添加 Laplace 噪声,有下述不等式成立:

$$\ln\left(\frac{\Pr[C_1 + \text{lap}(l_{\max}/\varepsilon_{11}) > \text{min\_sup}]}{\Pr[C_1 - 1 + \text{lap}(l_{\max}/\varepsilon_{11}) > \text{min\_sup}]}\right) \leqslant \varepsilon_{11} \quad (4)$$

$$\ln\left(\frac{\Pr[C_2 + \text{lap}(l_{\max}/\varepsilon_{12}) > \text{min\_sup}]}{\Pr[C_2 - 1 + \text{lap}(l_{\max}/\varepsilon_{12}) > \text{min\_sup}]}\right) \leqslant \varepsilon_{12} \quad (5)$$

由式(4)不等式两端同时取指数可得,

$$\Pr[C_1 + \text{lap}(l_{\max}/\varepsilon_{11}) > \text{min\_sup}] \leqslant \exp(\varepsilon_{11}) \times \Pr[C_1 - 1 + \text{lap}(l_{\max}/\varepsilon_{11}) > \text{min\_sup}] \quad (6)$$

同理,由式(5)可得,

$$\Pr[C_2 + \text{lap}(l_{\max}/\varepsilon_{12}) > \text{min\_sup}] \leqslant \exp(\varepsilon_{12}) \times \Pr[C_2 - 1 + \text{lap}(l_{\max}/\varepsilon_{12}) > \text{min\_sup}] \quad (7)$$

结合式(1)对差分隐私的定义以及式(6)可知,算法 2 行 2 满足 ε_{11}-差分隐私,同理结合式(1)以及式(7)可知,算法 2 行 4 满足 ε_{12}-差分隐私,根据差分隐私的顺序性质[18],可得,Dynamic-Truncation 算法满足($\varepsilon_{11} + \varepsilon_{12}$)-差分隐私,即变长序列截断满足 ε_1-差分隐私.

4.3 构建序列格

本小节介绍构建序列格的具体实现细节,如算法 3:

算法 3 Build-Noisy-Lattice 算法

输入:时序数据库 $\tilde{T}$,隐私预算 ε_2,最小支持度阈值 min_sup
输出:满足 ε_2-差分隐私的序列格 S_{Lattice}

1. $\varepsilon_2 = \varepsilon_{21} + \varepsilon_{22}$
2. $z \leftarrow < z_1, z_2, \cdots, z_k, \cdots, z_n >$, $n = l_{\max}//z_k$ 表示候选 k 序列支持度的最大值

3. $z' \leftarrow z + \mathrm{lap}(1/\varepsilon_{21})^n$
4. max_height←max-subscript(z',min_sup) //z 中大于 min_sup 的最大下标
5. $\bar{\varepsilon} \leftarrow \varepsilon_{22}/\text{max\_height}$
6. for k from 1 to max_height do
7. $\Delta \leftarrow \min\left\{\binom{\text{max\_height}}{k}, |C_k|\right\}$
8. if $k=1$ then
9. for each $c_{1i} \subset C_1$ do //C_1为时序数据库所有不同的项
10. if $c_{1i}.\text{sup} + \mathrm{lap}(\Delta/\bar{\varepsilon}) > \text{min\_sup}$ then
11. $S_1 \leftarrow S_1 \cup c_{1i}$
12. $S_{\text{Lattice}} \leftarrow S_{\text{Lattice}} + S_1$
13. else
14. $C_k \leftarrow$table-join(S_{k-1})
15. for each $c_{ki}(c_{ki} \subset C_k)$ do
16. if $c_{ki}.\text{sup} + \mathrm{lap}(\Delta/\bar{\varepsilon}) > \text{min\_sup}$ then
17. $S_k \leftarrow S_k \cup c_{ki}$
18. $S_{\text{Lattice}} \leftarrow S_{\text{Lattice}} + S_k$ //S_k添加到 S_{Lattice}中,并根据表连接操作添加指针指向
19. Return S_{Lattice}

该算法主要包含两个过程:估计序列格的最大高度 max_height 过程(行 2 ~ 5);通过表连接操作 table-join 构建序列格过程(行 6 ~ 18). 为了满足差分隐私,算法 3 在估计序列格的最大高度时,首先计算所有候选 $k(1 \leqslant k \leqslant l_{\max})$序列的最大值集合 z,对 z 添加 $\mathrm{lap}(1/\varepsilon_{21})$ 噪声,找出 z 中大于 min_sup 的最大下标 max_subscript,该下标值 max_subscript 即为序列格最大高度 max_height. 通过表连接操作构建序列格是算法 3 的核心步骤,下面将对其着重介绍.

考虑到时序数据库的时间戳信息,本文引入 SID-TID 表,其中 SID 表示序列的 ID 号,TID 表示序列的时间戳信息. 在挖掘时间序列过程中,所有候选序列都有其对应的 SID-TID 表. 例如,给定时序数据库如表 1 所示,其 SID = 1 的部分序列的 SID-TID 表如表 2 所示.

表 2　SID-TID 表格

SID	Time(TID)	Sequences:
1	10	*B*
1	15	*A*
1	20	*C*
1	25	*B*

构建序列格过程中,当数据库规模较大时,利用 Apriori 方法的连接-剪枝操作会产生大量候选序列,严重影响挖掘效率,同时会造成序列格中大量冗余序列. 因此,本文引入时间戳信息,每个候选序列都有其对应的 SID-TID 表. 在由频繁序列 S_k产生候选序列 C_{k+1}的过程中,只有 SID 相同,TID 存在前后关系的序列才能够连接,并产生候选序列. 图 5 示出了通过表连接操作构建序列格中一条路径 $P = \{B, B->D, B->D->C\}$ 的过程. 该路径 P 如图 6 中虚线所示.

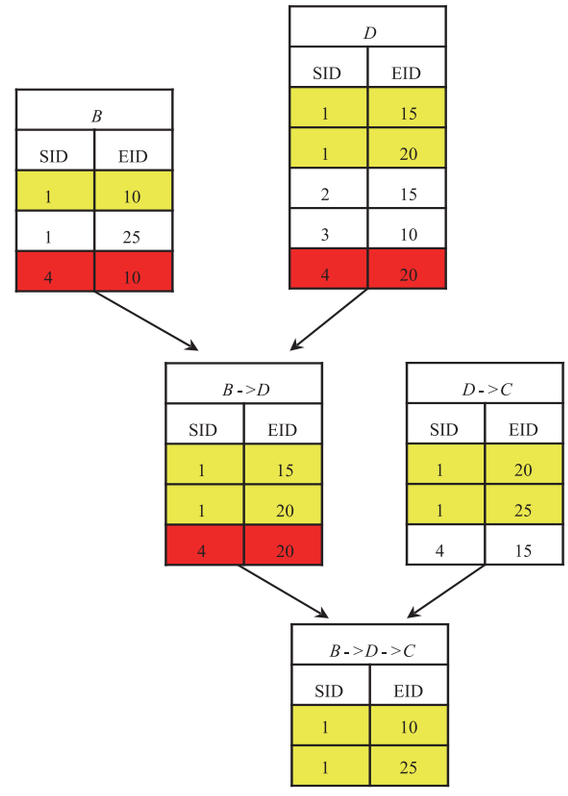


图5　表连接过程

由图 5 可看出,表中记录条数即为该序列的真实支持度. 因此,表连接操作使得在求序列真实支持度时无需重复遍历数据库,很大程度上减少了遍历时序数据库的次数;同时由于表连接操作本身蕴含的时序性,即只有存在时间前后关系的两个序列才能够进行连接操作,相比 Apriori 方法,该操作有效地减少了候选序列的数目.

本文采用全局敏感度的上界作为挖掘频繁序列模式过程中的全局敏感度,由定理 1,可以求出全局敏感度的的上界是 $\min\left\{\binom{I_{\max}}{k}, |C_k|\right\}$,如算法 3 中行 7、行 10、行 16 所示,其中 $l_{\max}$表示时序数据库中序列的最大长度,$|C_k|$为候选 k 序列集合的大小. 该算法通过层序遍历的方式构建序列格. 首先获取频繁 1 序列,并为其在序列格中创建节点. 利用当前节点通过表连接操作得出后继节点,然后迭代构建序列格 S_{Lattice}的每一层. 构建过程中某一节点是否要继续往下划分,即是否产生后继节点的条件为:(1)是否满足最小支持度阈值 min_sup;(2)层次是否超过 max_height. 最终构建的隐私序列格如图 6 所示.

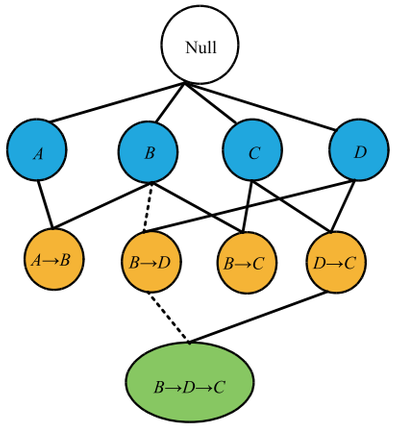


图6 隐私序列格

定理 3 构建序列格过程满足 ε_2-差分隐私.

证明 根据算法 3 描述,构建序列格过程主要包含两个步骤:估计序列格最大高度,以及表连接构建序列格. 对于时序数据库 T,由于添加(或删除)一个输入序列只影响 T 变化 1,因此,估计序列格最大高度过程在计算时添加拉普拉斯噪声的敏感度为 1,因此有下述不等式成立:

$$\ln\left(\frac{\Pr[z+\mathrm{lap}(1/\varepsilon_{21})]}{\Pr[z-1+\mathrm{lap}(1/\varepsilon_{21})]}\right)\leqslant\varepsilon_{21} \tag{8}$$

不等式(8)两端同时取指数可得,

$$\Pr[z+\mathrm{lap}(1/\varepsilon_{21})]\leqslant\exp(\varepsilon_{21})\times\Pr[z-1+\mathrm{lap}(1/\varepsilon_{21})] \tag{9}$$

结合式(1)与式(9)可知,估计序列格最大高度过程满足 ε_{21}-差分隐私. 挖掘时间序列模式的敏感度 $\Delta=\min\{C^k_{\max\_height},|C_k|\}$,构建序列格过程中判断节点 c_{ki} 加 $\mathrm{lap}(\Delta/\bar{\varepsilon})$ 噪声是否大于阈值 min_sup 时,有下述不等式成立:

$$\ln\left(\frac{\Pr[c_{ki}.\sup+\mathrm{lap}(\Delta/\bar{\varepsilon})>\min\_sup]}{\Pr[c_{ki}.\sup-1+\mathrm{lap}(\Delta/\bar{\varepsilon})>\min\_sup]}\right)\leqslant\bar{\varepsilon} \tag{10}$$

不等式(10)两端同时取指数可得,

$$\Pr[c_{ki}.\sup+\mathrm{lap}(\Delta/\bar{\varepsilon})>\min\_sup]\leqslant\exp(\bar{\varepsilon})\times\Pr[c_{ki}.\sup-1+\mathrm{lap}(\Delta/\bar{\varepsilon})>\min\_sup] \tag{11}$$

结合式(1)与式(11)可知,节点 c_{ki} 满足 $\bar{\varepsilon}$-差分隐私. 同时由算法 3 行 6 可知,该过程需要重复 max_height 次,而 $\bar{\varepsilon}=\varepsilon_{22}/\max\_height$,根据差分隐私的并行性质[18]和顺序性质[18]可知,构建前缀序列格过程满足($\varepsilon_{21}+\varepsilon_{22}$)-差分隐私,即满足 ε_2-差分隐私.

4.4 隐私预算分配及利用

在得到隐私序列格后,需要获得频繁序列模式及其噪声支持度(如算法 1 第 4 行所示). 本小节结合文献[19,20]介绍一种有效的隐私预算分配策略,通过该策略获取频繁序列模式的噪声支持度,以进一步提高挖掘结果的可用性.

序列格本身蕴含的特性,即约束条件:(1)对任意一条根节点到叶子节点的路径 P,满足 $\forall v_i\in P,|\mathrm{ns}(v_i)|\leqslant|\mathrm{ns}(v_{i+1})|$,其中 $|\mathrm{ns}(v_i)|$ 表示节点 v_i 的噪声支持度,v_i 是 v_{i+1} 的孩子节点;(2) $\forall v,|\mathrm{ns}(v)|\geq\sum_{u\in\mathrm{childs}(v)}|\mathrm{ns}(u)|$. 上述约束条件表明,序列格的层次越高,节点的真实支持度越小. 因此,本文设计的隐私预算分配策略满足的约束条件应为:序列格的层次越高,所分配到的隐私预算越多.

假设 $|p_i|$ 表示序列格的某条路径 P 从第 i 层到叶子节点的路径长度,则序列格上某条路径 P 上节点 v_i 的隐私预算满足下述等式:

$$\mathrm{pri\_budget}(v_i)=\begin{cases}\dfrac{\varepsilon_3}{2}\displaystyle\prod_{j=1}^{|p_i|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_i|}, & \text{if } v_i \text{ is not leaf}\\ \dfrac{\varepsilon_3}{2}\displaystyle\prod_{j=1}^{|p_i|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_i|}+\frac{\varepsilon_3}{2}, & \text{else}\end{cases} \tag{12}$$

定理 4 由式(12)可知,序列格上某条路径 P 上节点 v_{i-1} 分配到的隐私预算为 $\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_{i-1}|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_{i-1}|}$,节点 v_i 分配到的隐私预算为 $\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_i|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_i|}$,则有下列不等式成立:

$$\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_i|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_i|}\geq\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_{i-1}|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_{i-1}|} \tag{13}$$

证明 根据序列格结构本身可知 $|p_i|\geq|p_{i-1}|+1$,则

$$\begin{aligned}&\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_i|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_i|}\\&=\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_{i-1}|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{|p_{i-1}|-1}{|p_{i-1}|}\cdot\frac{1}{|p_i|}\\&\geqslant\frac{\varepsilon_3}{2}\prod_{i=1}^{|p_{i-1}|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{|p_i|}{|p_{i-1}|}\cdot\frac{1}{|p_i|}\\&=\frac{\varepsilon_3}{2}\prod_{j=1}^{|p_{i-1}|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p_{i-1}|}\end{aligned}$$

由定理 4 可知节点 v_i 分配到的隐私预算大于等于节点 v_{i-1} 分配到的隐私预算. 随着序列格层次的增加,序列的真实支持度急剧减小,同时由定理 4 可知,当序列格的层次越高,利用式(12)分得的隐私预算越多,这样可以减少噪声注入带来的误差,进而提高挖掘结果的可用性.

例如,给定隐私序列格如图 6 所示,路径 P 如图中虚线所示,P 上频繁序列{$B,B->D,B->D->C$}的噪声支持度分别为{10,5,1},则根据式(12),路径 P 上

隐私预算的分配为:$\{B:\frac{\varepsilon_3}{6},B->D:\frac{\varepsilon_3}{6},B->D->C:\frac{2\varepsilon_3}{3}\}$.

定理 5　Generate-Tseries 过程满足 ε_3-差分隐私.

假设$|P|$表示一条从根节点到叶子节点的路径长度,由式(12)隐私预算的分配策略可知,证明 Generate-Tseries 过程满足 ε_3-差分隐私可转化为证明下面不等式:

$$\frac{\varepsilon_3}{2}\cdot\frac{1}{|p_1|}+\frac{\varepsilon_3}{2}\sum_{i=1}^{|P|-1}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)+\frac{\varepsilon_3}{2}\leqslant\varepsilon_3 \tag{14}$$

证明

$$\begin{aligned}
&\frac{1}{|p_1|}+\sum_{i=1}^{|P|-1}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)\\
&\quad+\prod_{i=1}^{|P|-1}\left(1-\frac{1}{|p_i|}\right)\cdot\left(1-\frac{1}{|p|}\right)\\
&=\frac{1}{|p_1|}+\sum_{i=1}^{|P|-2}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)\\
&\quad+\prod_{j=1}^{|P|-1}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{|p|}\\
&\quad+\prod_{i=1}^{|P|-1}\left(1-\frac{1}{|p_i|}\right)\cdot\left(1-\frac{1}{|p|}\right)\\
&=\frac{1}{|p_1|}+\sum_{i=1}^{|P|-2}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)+\prod_{j=1}^{|P|-1}\left(1-\frac{1}{|p_j|}\right)\\
&=\frac{1}{|p_1|}+\sum_{i=1}^{|P|-2}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)+\prod_{j=1}^{|P|-2}\left(1-\frac{1}{|p_j|}\right)\\
&\quad\cdot\left(1-\frac{1}{|p|-1}\right)\\
&=\cdots.\\
&=1
\end{aligned}$$

去除非负项 $\prod_{i=1}^{|P|-1}\left(1-\frac{1}{|p_i|}\right)\cdot\left(1-\frac{1}{|p|}\right)$,可得,

$$\frac{1}{|p_1|}+\sum_{i=1}^{|P|-1}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)\leqslant 1 \tag{15}$$

式(15)不等式两端同乘以$\frac{\varepsilon_3}{2}$,可得,

$$\frac{\varepsilon_3}{2}\cdot\frac{1}{|p_1|}+\frac{\varepsilon_3}{2}\sum_{i=1}^{|P|-1}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)\leqslant\frac{\varepsilon_3}{2}$$

即,

$$\frac{\varepsilon_3}{2}\cdot\frac{1}{|p_1|}+\frac{\varepsilon_3}{2}\sum_{i=1}^{|P|-1}\left(\prod_{j=1}^{i}\left(1-\frac{1}{|p_j|}\right)\cdot\frac{1}{i+1}\right)+\frac{\varepsilon_3}{2}\leqslant\varepsilon_3$$

假设根据上述隐私预算分配策略,路径$|P|$中节点 v_i 分配到的隐私预算为 ε_i,则有下述不等式成立:

$$\ln\left(\frac{\Pr[v_i.\sup+\mathrm{lap}(\Delta/\varepsilon_i)]}{\Pr[v_i.\sup-1+\mathrm{lap}(\Delta/\varepsilon_i)]}\right)\leqslant\varepsilon_i \tag{16}$$

式(16)不等式两端同时取指数可得,

$$\Pr[v_i.\sup+\mathrm{lap}(\Delta/\varepsilon_i)]\leqslant\exp(\varepsilon_i)\times\Pr[v_i.\sup-1+\mathrm{lap}(\Delta/\varepsilon_i)] \tag{17}$$

由式(17)可知,节点 v_i 满足 ε_i-差分隐私,结合式(14)以及差分隐私的并行性质[18]和顺序性质[18]可知,隐私预算分配过程满足 ε_3-差分隐私.

获得隐私序列格后,通过深度优先遍历或者广度优先遍历,即可得到所有的频繁时间序列及其噪声支持度.

4.5　PrivTSM 算法可用性分析

本小节主要从差分隐私定义角度,证明 PrivTSM 算法如何满足 ε-差分隐私,并从时间复杂度角度分析算法的性能.

4.5.1　算法隐私性分析

定理 6　PrivTSM 算法满足 ε-差分隐私.

证明　根据算法 1 描述,PrivTSM 算法主要包含三个过程:截断时序数据库过程、构建序列格过程、生成噪声频繁时间序列模式过程. 根据定理 2、定理 3 与定理 5,以及结合差分隐私的顺序性质[18]可知 PrivTSM 满足 $(\varepsilon_1+\varepsilon_2+\varepsilon_3)$-差分隐私,即满足 ε-差分隐私.

4.5.2　算法时间复杂度分析

截断时序数据库过程中,通过隐私 DAG 图找最优截断序列的时间复杂度为 $O(|T|n)$;构建序列格过程采用层序的方式,在构建当前层序列格过程中,表连接操作(行 14)以及判断表中的频繁序列都只需线性时间 $O(n)$,构建序列格过程需重复 max_height 次,所以最坏情况下,构建序列格过程总的时间复杂度为 $O(n\times\text{max\_height})$,因此 PrivTSM 算法的时间复杂度为 $O(n\times\text{max\_height})$. 基于前缀树划分的 Prefix-Hybrid 方法的时间复杂度为 $O(h|T|n)$,基于变长模型的 N-gram 方法的时间复杂度为 $O(l_{\max}|T|n)$,由于 $\text{max\_height}\leqslant h|T|\leqslant l_{\max}|T|$,则 PrivTSM 算法的性能较好.

5　实验结果与分析

实验平台是 4 核 Intel i7-4790 CPU(4GHz),8GB 内存,Win7 系统. 所涉及代码采用 Python 实现,部分代码采用 C++ 实现. 实验采用四种真实时序数据库 Geolife、MSNBC[9]、BMS1[14]和 Kosarak[16],其中 Geolife 数据库是微软亚洲研究院从 2007 年 4 月至 2012 年 8 月,收集的包含 182 个用户、17621 条轨迹信息的数据库. 本文利用随机采样的方法选取了部分数据作为实验数据. MSNBC 数据库是 msnbc.com 网站用户点击流数据,每条记录描述的是某个用户一天之内访问的新闻页面的序列,包含 989818 条记录,数据来源于 UCI repository;Kosarak 数据库是匈牙利新闻门户网站 kosarak.org 的用户点击流数据,包含 990002 条记录;BMS1 数据库是电子商务网站 Gazelle.com 几个月内用户的点击流数据,每条记录描述的是用户按照时间顺序访问的所有

商品的序列,包含 59602 条记录. 四种时序数据库的具体特征如如表 3 所示.

表 3 实验数据库特征

数据库	实际大小	项集大小	最大长度	平均长度
Geolife	17621	50000	3879	55.7
MSNBC	989818	17	14795	4.7
Kosarak	990002	41270	2498	8.1
BMS1	59602	497	267	2.5

本节实验分为两组,一组实验用于比较分析 PrivTSM、N-gram、Prefix-Hybrid 三种算法的可用性;另一组实验用于比较分析变长序列截断方法 Dynamic-Trunc 与随机序列截断方法 Random-Trunc 的好坏.

本文采用准确率 TPR 和平均相对误差 ARE 衡量 PrivTSM、N-gram、Prefix-Hybrid 三种算法的可用性;采用准确率和召回率(Recall)衡量 Dynamic-Trunc、Random-Trunc 两种序列截断方法的好坏. 设 $F(T)$ 表示原始时序数据库 T 对应的真实频繁序列模式集合,$F(\tilde{T})$ 表示满足差分隐私的的频繁序列模式集合.

定义 4 准确率:该指标用于衡量挖掘出的时间序列模式本身的可用性,公式如下

$$\mathrm{TPR}=\frac{F(T)\cap F(\tilde{T})}{F(\tilde{T})} \tag{18}$$

根据式(18)可知 TPR 值越大,算法的可用性越高.

定义 5 召回率:该指标同样用于衡量挖掘出的时间序列模式本身的可用性,公式如下

$$\mathrm{Recall}=\frac{F(T)\cap F(\tilde{T})}{F(T)} \tag{19}$$

根据式(19)可知 Recall 值越大,算法的可用性越高.

定义 6 平均相对误差:该标准用于衡量时间序列模式噪声支持度计数的可用性,公式如下

$$\mathrm{ARE}=\frac{\sum_{F_i\in TF(T)}\dfrac{|\ \mathrm{TC}(F_i,F(T))-\mathrm{NC}(F_i,F(\tilde{T}))\ |}{\mathrm{TC}(F_i,F(T))}}{\max\{\Delta,F(\tilde{T})\}} \tag{20}$$

其中 $\mathrm{TC}(F_i,F(T))$ 和 $\mathrm{NC}(F_i,F(\tilde{T}))$ 分别表示模式 F_i 对应的真实支持度计数和噪声支持度计数,Δ 为平滑因子,其值为时序数据大小的 1%. 由式(20)可知 ARE 值越小,算法的可用性越高.

本文设置参数 ε 为 0.2,0.4,0.6,0.8,1.0. 最小支持度阈值 min_sup 为相对阈值,并根据不同数据库设置不同的 min_sup. 每个算法分别执行 100 次,并记录输出结果的平均值.

(1)基于 min_sup ,ε 的变化,对比 PrivTSM、N-gram、Prefix-Hybrid 三种算法的可用性.

在 Geolife 数据库上,图 7(a)和图 7(b)显示,固定 $\varepsilon=0.5$,最小支持度阈值 min_sup 取值分别为数据库大小的 0.02,0.025,0.03,0.035,0.04. 当 min_sup 从 0.02 变化到 0.04 时,PrivTSM 算法的准确率明显比 N-gram 和 Prefix-Hybrid 算法表现好,其原因是虽然 N-gram 和 Prefix-Hybrid 算法也限制序列长度,但是直接删除超过限定长度的项,因此会丢失大量信息. 而 PrivTSM 算法使用变长序列截断方法方法,在限制序列长度的同时,很大程度上保留了序列的频繁信息. 由图 7(c)和图 7(d)可看出,当固定 min_sup = 0.02,ε 从 0.2 变化到 1.0 时,PrivTSM 算法的准确率比 N-gram 和 Prefix-Hybrid 表现要好,维持在 0.8 左右. 而 N-gram 和 Prefix-Hybrid 算法在 0.6 左右. PrivTSM 算法的 ARE 随着 ε 的增大明显降低,是 Prefix-Hybrid 算法的将近 5 倍. 由于 MSNBC 数据库包含大量短序列记录,所以从图 7(e)和图 7(f)可看出,N-gram 和 Prefix-Hybird 表现较好,但仍然高于 PrivTSM 算法. 由于 Kosarak 数据库特征与 Geolife 数据库特征比较相似,所以 PrivTSM 在两种数据库上表现相似. 由于 Kosarak 数据库与 Geolife 数据库的平均长度分别为 8.1、55.7,N-gram 和 Prefix-Hybrid 算法在 Geolife 和 Kosarak 数据库上表现差别比较大,这也证明了 PrivTSM 算法即能够适用于短序列居多的数据库,同时在长序列居多的数据库上表现也很好. 由于 BMS1 数据库规模较小,序列平均长度非常小,因此三种算法在 BMS1 数据库上表现都非常好,相对来说 PrivTSM 算法仍然好于 N-gram 和 Prefix-Hybrid 算法.

总的来说,PrivTSM 由于采用变长序列截断方法,在保留足够频繁信息的同时,很好的提高了挖掘结果的可用性. 基于序列格本身蕴含的约束条件,采用合理的隐私预算分配策略,进一步的提高的挖掘结果的准确率,因此 PrivTSM 算法的表现才能够优于 N-gram 和 Prefix-Hybrid 算法.

(2)固定 ε,基于参数 min_sup 的变化,比较 Dynamic-Trunc、Random-Trunc 两种序列截断方法的好坏.

由图 8(a)和图 8(e)可知,在长序列居多的 Geolife 数据库上,Dynamic-Trunc 方法的召回率还是明显优于 Random-Trunc 方法,准确率稍微好于 Random-Trunc 方法. 在 MSNBC 和 BMS1 数据库上,绝大多数是短序列记录,只有极少数的长序列记录,恰恰因为这些长序列记录给挖掘过程带来了很高的全局敏感度,因此,序列截断方法能够很大程度提高挖掘结果的可用性,从图 8(f) ~ 8(h)也可看出,Dynamic-Trunc 方法明显优于 Random-Trunc 方法.

总的来说,变长序列截断方法 Dynamic-Trunc 在截断过程中利用挖掘出的隐私频繁 2 序列构建 DAG 图,通过遍历 DAG 图来获取最优的截断序列,因此该截断方法保留了足够多的频繁信息,相比随机截断方法 Random-Trunc 来说,挖掘结果的可用性更高.

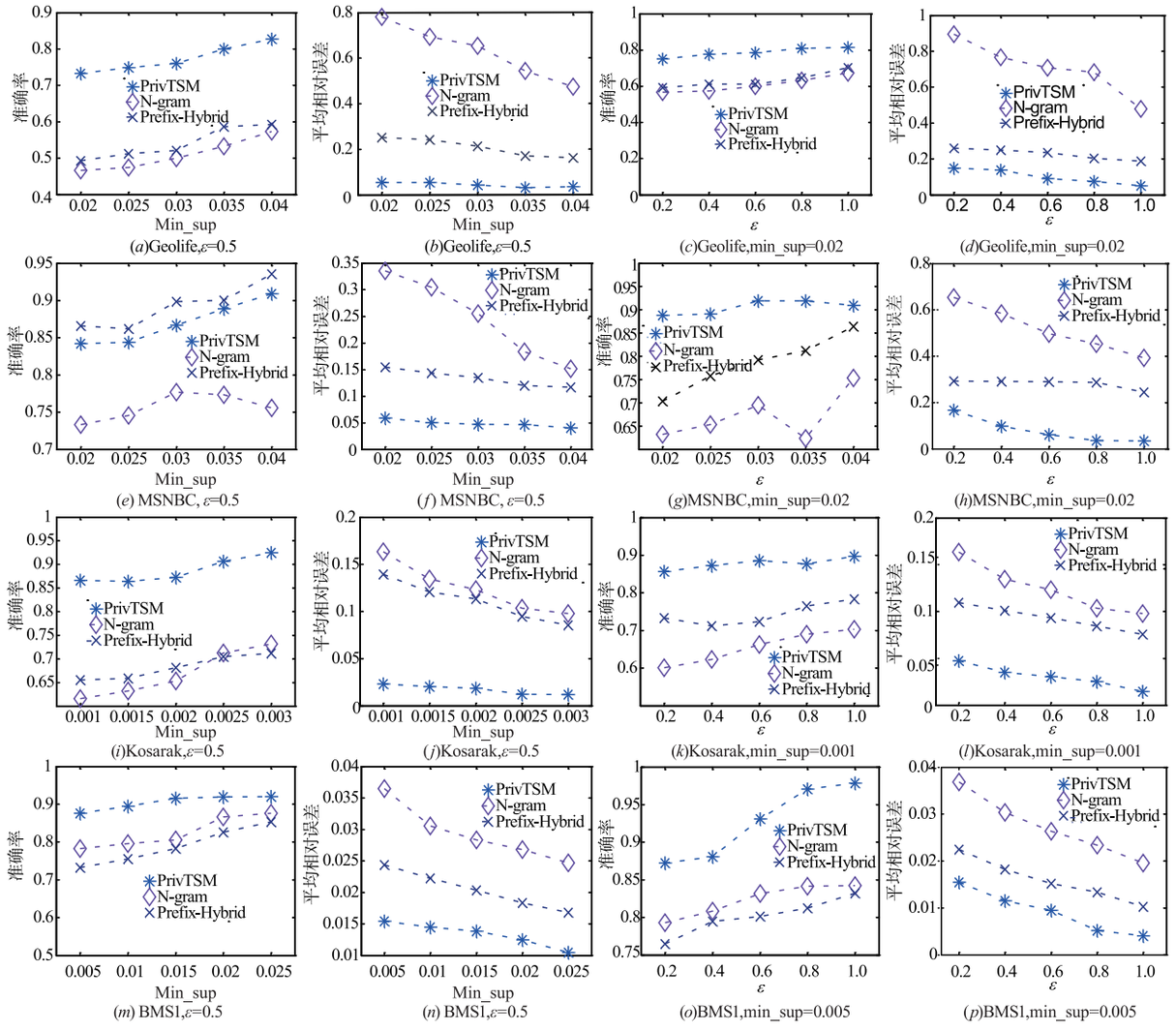


图7　时间序列模式挖掘结果

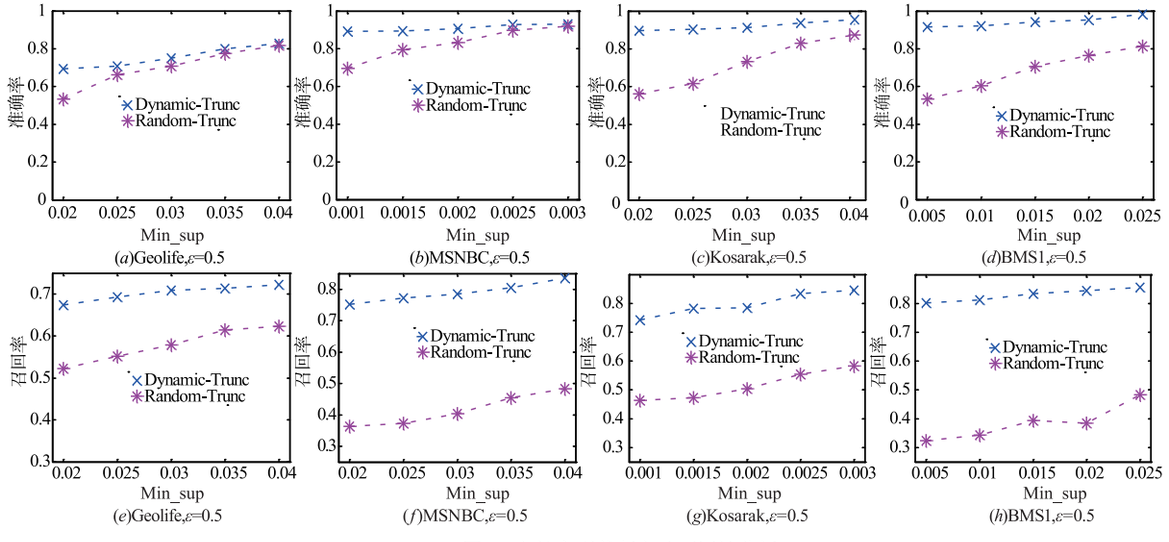


图8　变长序列截断方法可用性分析

6 结束语

针对差分隐私保护下时间序列模式挖掘存在的问题,本文提出了一种基于序列格的差分隐私下时间序列模式挖掘方法 PrivTSM. 该方法通过截断时序数据库来降低全局敏感度. 在此基础上,结合表连接操作构建序列格,减少候选序列的规模. 针对序列格本身特性,采用一种有效的隐私预算分配策略,提高了挖掘结果的可用性. 最后,通过真实数据库上的实验结果表明 PrivTSM 有比较高的数据可用性. 在未来的工作中,我们将研究如何更加合理地分配隐私预算,以进一步提高挖掘结果的可用性.

参考文献

[1] Dwork C. Differential privacy[A]. Proceedings of the 33rd Int Colloquium on Automata, Languages and Programming [C]. Berlin: Springer, 2006. 1 – 12.

[2] Dwork C. Lei J. Differential privacy and robust statistics [A]. Proceedings of the 41th Annual ACM Symp on Theory of Computing[C]. New York: ACM, 2009. 371 – 380.

[3] Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis[A]. Proceedings of the 3th Theory of Cryptography Conference[C]. Berlin: Springer, 2006. 363 – 385.

[4] Chen R, Fung B, Desai B C. Differentially private trajectory data publication[J]. arXiv, 2011, 1(9): 1112 – 2020

[5] Chen R, Mohammed N, Fung B C M, et al. Publishingset-valued data via differential privacy[A]. Proceedings of the 37th Conference of Very Large Databases[C]. New York: ACM, 2011. 1087 – 1098.

[6] Bonomi L, Xiong L. A two-phase algorithm for mining sequentialpatterns with differential privacy[A]. Proceedings of the 22nd ACM International Conference on Information and Knowledge Management[C]. New York: ACM, 2013. 269 – 278.

[7] Mohammed J Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences[J]. Machine Learning, 2001, 1(42): 31 – 60.

[8] Chen R, Fung B, Desai B. C, et al. Differentially private transit data publication: a case study on the montreal transportation system [A]. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [C]. New York: ACM, 2012. 213 – 221.

[9] Chen R, Gergely A, Claude C. Differentially private sequential data publication via variable-length n-grams[A]. Proceedings of the 19th ACM Conf on Computer and Communications Security [C]. New York: ACM, 2012. 638 – 649.

[10] Cheng X, S Su, S Xu, et al. Dp-apriori: A differentially private frequent itemset mining algorithm based on transaction splitting[J]. Computers & Security, 2015, 50(1): 74 – 90.

[11] Su S, Cheng X, et al. Differentially private frequent sequence mining via sampling-based candidate pruning [A]. Proceedings of the 31st IEEE International Conference on Data Engineering [C]. Washington, DC: IEEE Computer Society, 2015. 1035 – 1046.

[12] Hua Jingyu, Gao Yue, Zhong Sheng. Differentially private publication of general time-serial trajectory data[A]. Proc of INFCOM[C]. Piscataway, NJ: IEEE, 2015. 163 – 175.

[13] R Agrawal, R Srikant. Fast algorithms for mining association rules in large databases[J]. Proceedings of the VLDB Endowment, 1994, 23(3): 487 – 499.

[14] Zeng C, J F Naughton, J Y Cai. On differentially private frequent itemset mining [J]. Proceedings of the VLDB Endowment, 2012, 6(1): 25 – 36.

[15] 李杨,郝志峰,温雯,等. 差分隐私保护 k-means 聚类方法研究[J]. 计算机科学,2013,40(3):287 – 290.
Li Y, Hao Z, Wen W, et al. Research on differential privacy preserving *k*-means clustering[J]. Computer Science, 2013, 40(3): 287 – 290. (in Chinese)

[16] 张啸剑,王淼,孟小峰. 差分隐私保护下一种精确挖掘 top-k 频繁模式方法[J]. 计算机研究与发展,2014,51(1):104 – 114.
Zhang X, Wang M, Meng X. An accurate method for mining top-*k* frequent pattern under differential privacy[J]. Journal of Computer Research and Development, 2014, 51(1): 104 – 114. (in Chinese)

[17] 张啸剑,孟小峰. 面向数据发布和分析的差分隐私保护研究综述[J]. 计算机学报,2014,37(4):927 – 949.
Zhang X, Meng X. Differential privacy in data publication and analysis[J]. Chinese Journal of Computers, 2014, 37(4): 927 – 949. (in Chinese)

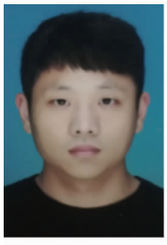
[18] McSherry F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis [A]. Proceedings of the 35th ACM SIGMOD Int Conf on Management of Data[C]. New York: ACM, 2009. 19 – 30.

[19] Xiang Cheng, Sen Su, Shengzhi Xu, Li Xiong, et al. A two-phase algorithm for differentially private frequent subgraph mining [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(8): 1411 – 1425.

[20] Chen R, Noman M. , B. C. M. Fung, et al. Publishing set-valued data via sifferential Privacy[J]. PVLDB, 2011, 4(11): 1087 – 1098.

作者简介

彭慧丽　女,1981 年生于河南周口. 主要研究方向为数据库、隐私保护.
Email:phl81@163.com



金凯忠　男,1991 年生于河南开封. 河南财经政法大学硕士. 主要研究方向为差分隐私、数据库.
E-mail:kaizhong@huel.edu.cn

付聪聪　女,1995 年生于河南商丘. 硕士研究生. 主要研究方向为差分隐私、图像处理.
E-mail:congf@huel.edu.cn

付　楠　男,1988 年生于河南开封. 硕士研究生. 主要研究方向为差分隐私、数据库.
E-mail:funan@huel.edu.cn

张啸剑(通信作者)　男,1980 年生于河南周口. 现为河南财经政法大学计算机与信息工程学院副教授,硕士研究生导师. 主要研究方向为差分隐私、数据库.
E-mail:xjzhang82@ruc.edu.cn