差分隐私下的一种频繁序列模式挖掘方法

卢国庆1,2 丁丽萍1 张啸剑3

1 (中国科学院 软件研究所 基础软件国家工程研究中心, 北京 100190)

2 (中国科学院大学, 北京 100190)

3 (河南财经政法大学计算机与信息工程学院, 郑州 450002)

(guoqing@nfs.iscas.ac.cn)

Frequent Sequential Pattern Mining Under Differential Privacy

Lu Guo-qing1,2 Ding Li-ping1 Zhang Xiao-jian3

1 (*National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*)

2 (*University of Chinese Academy of Sciences, Beijing 100190, China*)

3(*School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450002, China*)

**Abstract** Mining frequent sequential pattern is an exploratory problem in data mining. However, directly releasing discovered frequent patterns and corresponding true support are likely to reveal the individual privacy. Differential privacy has emerged as a new privacy protection technology with a rigorous defined attack model, which makes data distortion to achieve privacy protection purposes by means of adding noise. Due to the inherent sequentiality and high-dimensionality, it is challenging to apply differential privacy to sequential data. In this paper, we presents an efficient frequent sequential patterns mining algorithm called Diff-FSPM, based on differential privacy model.

**Key words** Differential Privacy; privacy preserving; Sequential Data; Frequent Patterns; Data Mining

**摘 要** 频繁序列模式挖掘是数据挖掘领域的一个基本问题, 然而模式本身以及相应计数有可能泄露用户隐私信息. 差分隐私作为一种新出现的隐私保护技术, 定义了一个相当严格的攻击模型, 通过添加噪音使数据失真达到隐私保护的目的.

由于序列数据内在序列性和高维度的特点, 给差分隐私结合序列数据带来了挑战. 对此提出了一种满足-差分隐私的频繁序列模式挖掘算法Diff-FSPM. 该方法利用拉普拉斯机制截断原始序列数据集获得其规约表示, 降低全局敏感度; 采用拉普拉斯机制产生的噪音扰动序列模式的真实支持度计数, 构建扰动前缀树; 为了增强输出模式的可用性, 采用一致性约束后置求精处理, 获得频繁序列模式结果集. 理论角度证明了该算法满足-差分隐私, 实验结果证明了算法具有较好的准确性、可用性和可扩展性.

第一篇基于差分隐私交互式数据保护框架

**关键词** 差分隐私；隐私保护；序列数据；频繁模式；数据挖掘

**中图法分类号** TP392

# 引言

频繁序列模式挖掘[1]是数据挖掘研究中的一个重要课题, 其目的是从序列数据集中发现相对时间或者其他顺序出现的高频率子序列模式, 被广泛应用在客户购买行为模式预测、交通轨迹推断、用户Web访问模式预测、自然灾害预测、DNA序列分析等方面. 然而频繁序列模式本身的内容以及计数信息都有可能泄露用户隐私信息, 或者披露用户的真实身份.

传统隐私保护下的模式挖掘算法大多基于*K*-匿名模型[2]. 然而, Zhou等人[3]和Ohm等人[4]指出, 当攻击者具备一定背景知识时, *K*-匿名模型就存在隐患, 攻击者可以利用再识别攻击(Re-identification Attack)等方法来确认用户隐私信息. 例如, 如果攻击者获得用户轨迹数据的同时, 也拿到了一份当地人口普查数据, 即使采用*K*-匿名模型来防治序列模式挖掘可能造成的隐私泄露, 攻击者依然可以利用Link攻击推理出用户的隐私信息. 此外, 传统隐私保护模型无法定量分析其隐私保护水平.

差分隐私(Differential Privacy, DP), 是Dwork在2006年提出的一种新的的隐私保护模型[5]. 该方法能够解决传统隐私保护模型的两大缺陷[6]: (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, 即使攻击者已掌握除某一条记录之外的所有记录信息(即最大背景知识假设), 该记录的隐私也无法被披露; (2) 对隐私保护水平给出了严谨的定义和量化评估方法.

差分隐私下的频繁模式挖掘已得到广泛关注. 频繁项集挖掘, 作为最简单的模式挖掘类型, 最先开展了相关研究工作, 如文献[7, 8, 9, 10]. 但由于序列模式本身内在序列性和高维度的特点, 上述方法不能有效解决差分隐私下的频繁序列模式挖掘问题. 为了实现满足差分隐私的频繁序列模式挖掘,当前已有文献[11, 12, 13]开始关注相关研究工作, 但普遍存在无法高效兼容隐私性和可用性的不足. 由于序列数据内在序列性, 序列模式呈指数规模增长, 序列冗余问题比较严重. 此外, 高维度性质导致某一长序列的存在会极易打破可用性和隐私性之间的平衡. 在国内, 差分隐私下数据挖掘(如频繁序列模式挖掘)相关研究也是一个崭新的领域.

本文的主要贡献在于:

1. 提出了一种差分隐私下的频繁序列模式挖掘方法Diff-FSPM(Differential-privacy Frequent Sequential Pattern Mining).
2. 应用数据规约技术, 获得最优序列长度用以截断原始序列数据集, 保证接近于完整表达原始序列数据集的前提下, 降低高维度的影响.
3. 应用闭频繁序列模式挖掘方法, 与频繁序列模式具有相同的信息表达, 但具有更少的模式, 有效解决序列冗余问题; 同时, 借鉴马尔科夫假设理论引入预测计数值, 保证序列模型的可用性.
4. 从隐私性、可用性、扩展性等方面进行详细的分析和实验评估, 并与已有方法进行比较.

本文内容共分为5节, 各节内容安排如下: 第1节给出相关工作, 第2节介绍定义和理论基础, 对算法的详细分析和评价在第3节中给出, 第4节展示实验结果, 第5节给出结论与展望.

# 相关工作

差分隐私下频繁模式挖掘领域的研究工作, 最早开始于频繁项集挖掘类型. 2010年Bhaskar等人[7]第一次提出了差分隐私下的频繁项集挖掘方法TF, 应用截断频率思想缩小候选项集合规模并通过拉普拉斯机制实现对记录长度不大于的频繁项集真实计数的保护. 然而, 该方法处理较大或值时性能与效率比较差; 为了弥补TF方法的不足, Li等人[8]提出PrivBasis方法, 结合-基和映射技术, 并借鉴极大团思想实现top-*k*频繁项集挖掘. 此外, 人大张啸剑博士等人[9]提出DP-topkP方法, 采用一种满足一致性约束的后置处理方法对TF方法中添加拉普拉斯噪音的计数进行求精处理, 提高频繁项集的可用性. Chen等人[10]依据长事务记录会导致较高全局敏感性的缺陷, 提出一种事务截断的贪婪方法SmartTrunc, 该方法利用阈值和动态权重频率对数据集中每条记录进行局部转换, 截断长记录来降低全局敏感性, 进而提高项集的可用性. 综上, 已有方法能够较好的实现差分隐私下的频繁项集挖掘. 但由于序列数据本身内在序列性和高维度的特性, 上述方法不能有效解决差分隐私下的频繁序列模式挖掘问题.

为了实现差分隐私下的频繁序列模式挖掘, Chen等人[11]结合自顶向下的前缀树, 第一次提出发布轨迹数据集的Prefix-Hybrid方法, 支持频繁序列挖掘. 该方法通过对原始轨迹数据集对应序列模式添加拉普拉斯噪音, 构建扰动前缀序列树, 最终重构发布满足差分隐私的轨迹数据集. 但随着前缀序列树的增长, 每一子分割的序列数量会急剧减小, 严重影响发布序列的可用性. Chen等人[12]基于变长n-gram模型提出发布序列数据集的方法N-gram, 该方法仅直接发布部分小维度序列模式, 然后采用马尔科夫假设迭代计算获得所有可得的长序列模式, 最终重构发布序列数据集, 支持频繁序列挖掘. 该方法能够较好的兼容隐私性和可用性, 但容易受到序列维度的影响, 存在一定的信息丢失. 为了解决N-gram方法所发布序列模式可用性较低的问题, Bonomi等人[13]提出一种两阶段方法PT-Sample, 分别应用前缀树划分和字符串指纹技术支持字符串的频繁前缀和频繁序列模式挖掘, 但该方法仅针对字符串类型的数据. 综上, 上述方法存在各自的不足, 如何设计同时兼顾隐私性和可用性的方法是本文的主题, 也是未来的研究方向.

# 定义和理论基础

## 序列模式挖掘

设是项的集合, 表示项的总数. 序列数据库DB, 其中任意一条记录*S*是一个非空项集, 满足, 且都有一个标识符, 称为*ID*. 每一条记录满足时间或者其他顺序出现. 例如, 一条序列记录表示某一用户的历史移动轨迹, 其中项代表某一车站; 或者一串密码, 项代表某一字符.

一条长度为的序列*S*. 一个项在某一序列中可能出现多次, 并可能连续出现. 因此, *S*是一合法序列. 一个序列数据库DB有条序列组成, 即DB*.* 如表2所示的样本序列数据库DB, .

Table 1 Sequence Database

表 1序列数据库*DB*

|  |  |
| --- | --- |
| **ID** | **序列数据** |
|  | I2→I3→I1 |
|  | I2→I3 |
|  | I3→I2 |
|  | I2→I3 →I1 |
|  | I2→I3→I1→I2→I3 |
|  | I3→I2 |
|  | I2→I3 →I1 |
|  | I3→I1→I2→I3 |

1. (包含关系[14]) 给定两个序列*s*和, 如果存在一组整数使得, 则称序列*s*被序列包含, 即是超序列, s是子序列, 表示为.
2. (*FS*[14]) 频繁序列模式集(Frequent Sequence Patterns)满足：
3. (*CS*[15]) 闭频繁序列模式集(Closed Frequent Sequence Patterns)满足：
4. (*PS*[15]) 投影数据库闭集(Projected Database Closed Set)满足：
5. (等价关系[15]) 给定两个序列*s*和,且, 那么, 满足.
   1. (反向子模式剪枝[15]) 给定两个序列*s*和,, 则只需条件成立, 即可停止搜索序列在前缀序列树中的所有后代.
   2. (反向超模式剪枝[15]) 给定两个序列*s*和,, 则只需条件成立, 则将序列*s*在前缀序列树中的所有后代移植给序列, 而不必搜索序列在前缀序列树中的任何后代.

挖掘*FS*过程中, 仅有一个剪枝条件, 即当时, 停止扩展序列*s*. 挖掘*CS*, 新添加两个有效的剪枝条件: 反向子模式剪枝和反向超模式剪枝, 能够尽可能早地结束序列扩展, 搜索空间急剧减小. 具体挖掘过程中首先产生完全的*PS*集合, 然后排除非闭合的*PS*序列生成*CS*. 对应样本序列数据库*DB*, 如果*min\_sup=*2, 对应的*FS*集合有16个序列(详见图1(a)), *LS*集合有10个序列(详见图1(b)), 而*CS*集合仅有8个序列(详见图1(c)), 满足.



(a) (c) (b)

|  |  |  |
| --- | --- | --- |
| **ID** | **序列模式** | **计数** |
|  | I2 | 9 |
|  | I3 | 10 |
|  | I2→I3 | 7 |
|  | I3→I1 | 5 |
|  | I3→I2 | 2 |
|  | I2→I3→I1 | 4 |
|  | I3→I1→I2→I3 | 2 |

图 1 频繁序列模式挖掘

Figure 1 Frequent Sequential Patterns Mining

*CS*与*FS*有着相同的信息表达, 但具有更少的模式. 闭合序列模式比频繁序列模式更紧凑. 即挖掘闭序列可以在保持信息完备性的前提下, 更加有效地解决序列冗余问题.[[1]](#footnote-1)

## 差分隐私保护

差分隐私(Differential Privacy, DP)[5-6, 16-20]是基于数据失真的隐私保护技术. 通过向查询或者分析结果中添加噪音使数据失真, 确保在某一数据集中插入或者删除某一条记录的操作不会影响任何查询的输出结果, 从而达到隐私保护的目的. 差分隐私的形式化定义如下:

1. (差分隐私[5]) 对于所有差别至多为一条记录的两个相邻数据集和, 给定隐私算法, 表示的取值范围. 若算法提供差分隐私保护, 则对于所有, 有

|  |  |
| --- | --- |
|  |  |

其中, 概率表示隐私披露风险, 隐私预算表示隐私保护水平, 越小隐私保护水平越高.

噪音机制是实现差分隐私的主要技术, 常用的噪音机制包括拉普拉斯机制(Laplace Mechanism)[2]和指数机制(Exponential Mechanism)[2]. 基于不同的噪音机制满足差分隐私的算法所添加噪音大小与全局敏感性(Global Sensitivity) [2]密切相关.

1. (全局敏感性[18]) 对于任意一个函数, 的全局敏感性定义为：

|  |  |
| --- | --- |
|  |  |

其中, 和之间至多相差一条记录, 表示函数的查询维度, 表示所映射的实数空间.

1. (拉普拉斯机制[18]) 对于任一个函数, 若算法的输出结果满足下列等式, 则满足差分隐私保护.

|  |  |
| --- | --- |
|  |  |

其中, 是相互独立的拉普拉斯变量, 对应概率密度函数为. 噪音大小与成正比, 与成反比, 即函数全局敏感性越大, 所需噪音越大. 拉普拉斯机制主要处理一些输出结果为实数型的算法.

1. (指数机制[19]) 给定一个打分函数，若算法满足下列等式，则满足-差分隐私.

|  |  |
| --- | --- |
|  |  |

其中，为打分函数的全局敏感性, 表示从输出域中所选择的输出项. 由公式(4)可知, 打分越高, 被选择输出的概率越大. 指数机制主要处理一些输出结果为非数值型的算法.

差分隐私保护技术本身蕴含着两种重要的组合性质[20]: 序列组合性(Sequential Composition)和并行组合性(Parallel Composition).

1. (序列组合性[20]) 给定数据库, 设为任一个随机算法()满足-差分隐私, 则算法在上的顺序操作满足-差分隐私.
2. (并行组合性 [20]) 给定数据库, 设为任一个随机算法()满足-差分隐私, 则算法在上的一系列不相交的操作满足*max*)-差分隐私.

# 差分隐私下频繁序列模式挖掘算法

## 问题说明

问题定义: 给定一个序列数据集, 隐私预算和最小支持度计数阈值的情况下, 本文的目标是设计Diff-FSPM算法来挖掘频繁序列模式. 该算法同时满足如下要求: (1) Diff-FSPM算法满足-差分隐私; (2) Diff-FSPM算法输出频繁序列模式对应扰动支持度计数满足一致性约束.

差分隐私下频繁序列模式挖掘, 由于序列数据本身内部序列性和高维度的特点, 存在两个主要问题需要解决: (1) 序列模式计数具有较高的全局敏感性, 需要添加非常大的噪音满足差分隐私. 定理2给出了计数全局敏感性的形式化分析; (2) 候选序列集合呈指数规模, 序列冗余问题比较严重. 导致在前缀序列树构建过程中, 存在较多的隐私预算重复分配, 浪费隐私预算进而影响数据的可用性.

1. (计数全局敏感性[13]) 已知序列数据集最大序列长度为, 设函数用以查询给定序列模式的计数, 其中任一序列模式长度满足, 则计数全局敏感性满足:

|  |  |
| --- | --- |
|  |  |

证明. 给定一条序列记录*x*和长度满足|*p*| 的序列模式*p*, *x*中出现*p*的计数最大值为. 若给定序列模式长度范围I, 则最大计数满足. 已知最大序列长度, 则计数全局敏感性

证毕.

由定理3得到计数全局敏感性正相关于最大序列长度. 因此即使只有一条很长的序列记录存在, 所有序列模式计数都需要添加更大的噪音, 进而严重影响数据可用性. 因此, 如何尽可能的降低计数全局敏感性, 是本文需要解决的问题.

随着最小支持度阈值的的减小或者最长序列长度的增长, 候选序列集合呈指数规模增长. 设候选序列集合C, 则. 如果直接处理, 计算量非常大且不能保证结果的准确性. 已有方法借鉴频繁模式增长思想, 即将候选频繁序列集压缩到一棵前缀序列模式树, 不需要产生候选. 但普遍存在序列冗余问题, 如图1(a)所示, 以序列为前缀的分支和以序列为前缀的分支满足等价关系, 即在前缀的序列树中所有后代都为冗余序列. 前缀序列树构建过程中, 对相同的候选频繁序列模式重复分配隐私预算, 不仅造成隐私预算浪费, 同时严重影响数据可用性. 因此, 如何尽可能的降低序列冗余, 节省隐私预算进而保证数据可用性, 是本文需要解决的问题.

为了解决上述两个问题, 首先, 本文应用截断数据规约(Data Reduction)技术, 获得最优序列长度截断原始序列数据集, 得到规约表示. 保证接近于完整表达原始序列数据集的前提下, 限定计数全局敏感性为最优序列长度进而提高数据的可用性; 然后, 应用闭频繁序列模式挖掘技术, 在前缀序列树构建过程中添加等价关系判定, 并添加反向子模式剪枝和反向超模式两个剪枝条件去除冗余序列, 减少隐私预算重复分配. 同时, 借鉴马尔科夫假设理论引入预测计数值, 与噪音计数比较择优选择作为前缀序列树划分所得序列对应扰动计数, 节约隐私预算同时保证较好的数据可用性.

## 算法描述

|  |  |
| --- | --- |
| 1. Diff-FSPM | |
| **输入:** 序列数据库*DB* | |
| **输入:** 隐私预算 | |
| **输入:** 最小支持度计数阈值*min\_sup* | |
| **输出:** 满足-差分隐私的频繁序列模式集 | |
|  | |
| 1. 截断***TrancatedSeqDB***(*DB*,); | //详见4.3.1 |
| 1. 扰动***BuildNoisyPreSeqTree***(); | |
|  | //详见4.3.2 |
| 1. 扰动***DisplayFreSeq***(); | //详见4.3.3 |
| 1. **return** ; | |

算法1给出Diff-FSPM算法的整体框架, 输入为序列数据集, 隐私预算和最小支持度计数阈值, 最终输出满足-差分隐私的频繁序列模式集. 算法分为三个过程: (1)过程1(行2)应用截断技术规约原始得到截断; (2)过程2(行3)构建满足差分隐私的扰动闭前缀序列树; (3)过程3(行4)排除非闭合的序列并生成满足差分隐私的频繁序列模式集.

## 算法说明

### 规约原始序列数据集

本文应用最优序列长度截断原始序列数据集的方法得到的规约表示. 中任意一条违背最优序列长度约束的序列记录, 截断并仅保留前项子序列, 即使得.

显然, 该截断方法带来了一定的信息损失. 但是, 如果序列数据集中记录长度满足幂律分布, 即短序列占多数、长序列占少数, 少量的长序列对频繁序列模式结果集影响较小, 同时有效地减小了全局敏感度. 序列数据, 如铁路交通网中的车站序列、自然语言处理应用中的单词序列等, 普遍存在幂律分布. 如图2所示, 短序列记录主导了MSNBC数据集, 如长度不大于15的记录占整个数据集的95%之多. 因此, 截断使得添加噪音减少能够广泛抵消信息损失, 进而可以提高数据可用性, 得到更准确的频繁序列模式结果集.

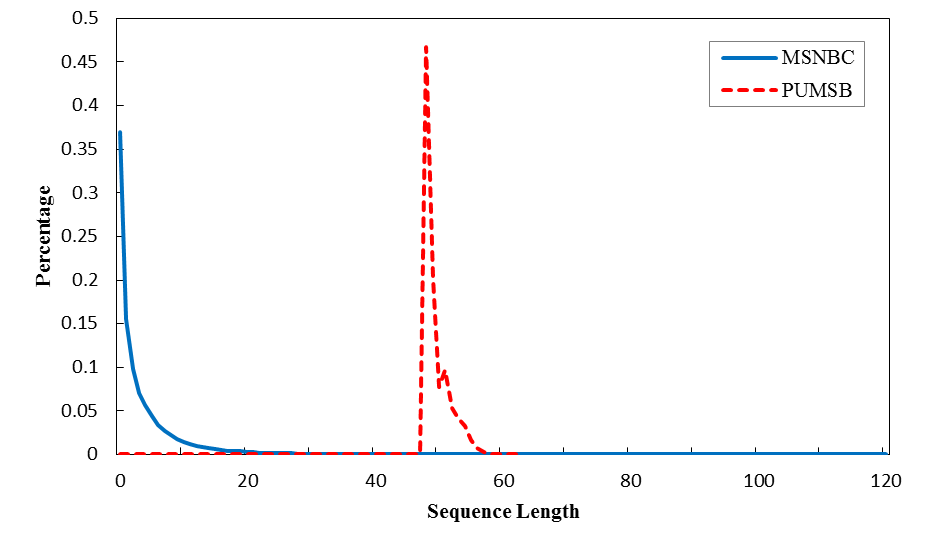


图 2 序列数据集记录长度分布

Figure 2 Distribution of seqnence length of dataset

然而, 截断DB需要保证不违反差分隐私, 即在截断DB上执行的满足-差分隐私的操作需要保证原始序列数据集满足-差分隐私. 文献[10]给出了局部变换(Local Transformation)的定义, 如定义7所示, 局部即输出仅仅依赖于输入的序列记录. 定理2证明在变换数据集(Transformed Database)上执行的满足差分隐私的任何操作都满足差分隐私.







TrancatedSeqDB过程算法描述如下:

|  |
| --- |
| 1. ***TrancatedSeqDatabase***(*DB*,) |
| **输入:** 序列数据库 |
| **输入:** 隐私预算 |
| **输出:** 规约序列数据集 |
| 1. 最优序列长度; |
| 1. ; |
| 1. **for each** 序列记录 |
| 1. 截断序列记录保留前项并添加到; |
| 1. **end for** |
| 1. **return** ,; |
|  |
| 1. **function** *GetOptSeqLength*(*DB,*) |
|  |
| 1. 满足不等式的最小整数; |
| 1. **return** ; 2. **end function** |
|  |
| 1. **function** *EstimateDistribution*(*DB*,) |
| 1. 为*DB*中序列记录长度为*i*计数值; |
| 是相互独立的拉普拉斯变量, ; |
| 1. **return** ; |
| 1. **end function** |

本文使用启发式的方法获取最优序列长度, 即不大于的序列记录计数比率和不小于0.85. 该方法要求计算每一长度序列记录计数, 存在隐私泄露风险, 应用拉普拉斯机制实现-差分隐私. 其中, 是一个配置参数, 通过数据集上反复试验获得. 在序列长度(数据表达)和全局敏感性(数据可用性)之间存在一个基本的权衡: (1) 较小的, 一方面对应较小的全局敏感度, 添加更小的噪音, 但另一方面带来了更多的原始序列信息损失; (2) 较大的, 能够较多的表达原始序列信息的同时, 却添加了更大的噪音. 本文将在第5节中讨论了不同值的影响.

借鉴中位数思想, 最优序列长度应该能够描述数据集中趋势, 且不易受数据中极端值的影响.

指数机制的关键是设计打分函数. 设表示数据集中序列长度小于的记录数目, 打分函数. 由于从数据集中添加或者删除一条序列记录, 影响至多为1, 则打分函数的全局敏感性为1, 即. 根据式(4)可知, 选择作为最优序列长度的概率为, 其中为选择最优序列长度操作所分得的隐私预算.

### 构建扰动闭FSP树

BuildNoisyPreSeqTree过程算法描述如下:

|  |
| --- |
| 1. ***BuildNoisyPreSeqTree*** |
| **输入:** 截断数据库 |
| **输入:** 最优序列长度 |
| **输入:** 最小支持度计数阈值 |
| **输入:** 隐私预算 |
| **输出:** 扰动闭前缀序列树 |
| 1. 创建树的虚拟根结点*Root*; |
|  |
|  |
| 1. **while**  **do** |
| 1. **for each** 非叶子结点 and  **do** |
| 1. 结点集合结点所有可能的后继结点 |
| 1. 为序列模式   对应的真实支持度计数 |
|  |
| 1. **if** **then** |
| 1. 为序列   模式对应的预测支持度计数 |
| 1. **end if** |
| 1. **for each** 结点  **do** |
| 1. 扰动支持度计数 |
| 1. 添加到中; |
| 1. **if**  **then** |
| 1. 标记为叶子结点; |
| 1. **end if** |
| 1. **end for** |
| 1. **end for** |
|  |
| 1. **end while** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 1. 依据一致性约束后置处理优化; |
| 1. **return** |

扰动前缀序列树的构建过程如下: 首先, 创建前缀树的虚拟根结点, 用“Root”标记(行1), 后扫描截断数据集迭代构造每一层(行4-17), 最后依据一致性约束后置处理优化. 构建过程中非叶子结点划分需要同时满足以下四个条件: (1) 树的深度小于; (2) 扰动计数不小于*min\_sup*; (3) 隐私预算未耗尽; (4) 未满足等价关系(反向子模式剪枝和反向超模式剪枝). 例1给出样本序列数据集对应前缀序列树可能的构建过程.

1. 设*min\_sup*=3, =3. 针对表2样本序列数据集可能的前缀序列树如图3所示(先不考虑隐私预算信息、结点及红色实线和虚线部分).



图 3 样本序列数据集对应的前缀序列树

Figure 3 The exploration tree of the sample database

下面将重点介绍BuildNoisyPreSeqTree过程中三个重要组成部分: (1) 采用闭频繁序列模式技术, 降低序列冗余, 进而避免前缀序列树构建过程中隐私预算的重复添加, 保证数据可用性; (2) 基于马尔科夫假设(Markov Assumption)引入预测计数值, 节约隐私预算的同时保证数据可用性; (3) 依据一致性约束实施后置处理, 保证数据可用性.

已有方法在前缀序列树的构建过程中, 非叶子结点划分孩子结点集合为, 其中表示序列结束符. 由此候选频繁序列集呈指数规模增长, 序列冗余问题较严重, 造成了较多的隐私预算重复分配情况出现. 如例1所述前缀序列树, 易得以序列为前缀的分支和以序列为前缀的分支满足等价关系. 若直接划分, 则对应两个分支重复分配隐私预算, 即存在大小为的浪费. 本文借鉴文献[15]提出的CloSpan方法添加反向子模式和反向超模式剪枝条件, 对等价分支只添加一次隐私预算. 上述分支满足反向超模式剪枝, 即只有以序列为前缀的分支分配隐私预算, 将序列在前缀序列树中的所有后代移植给序列, 而不必搜索序列在前缀序列树中的任何后代.

获取最优序列长度后, 最简单的隐私预算分配策略就是平均分配, 即前缀序列树每一层分得大小的隐私预算用以计算结点的噪音计数. 本文提出应用马尔科夫假设, 根据已划分得到的分割链来计算下一层结点的预测计数值, 并与噪音计数对比, 如果预测计数值更接近于真实支持度计数, 则此结点对应扰动计数值定为预测计数值, 无需分配隐私预算, 保证节省大小的隐私预算, 保证更高的准确性. 同时, 在前缀序列树构建结束后, 如果依然存在孩子结点对应扰动计数值不小于min\_sup, 仍可以采用预测计数值, 直至找到所有不小于min\_sup的结点或为止. 依据马尔科夫假设, 长序列模式可以通过短序列模式连接来获得.

1. (连接操作) 给定两个长度为的序列和. 如果满足, 则称序列和可连接, 表示为. 其计数值计算公式如下:



需要注意的是, 连接操作不满足交换律, 即存在序列和可连接, 但和不可连接的情况. 同时在整个计算过程中, 序列模式对应扰动计数值均是使用的噪音计数值或者预测计数值.

1. 例1中短序列和可以连接得到序列. 预测计数值为, 假设噪音计数为3.7, 真实支持度计数为3. 由于, 故选择预测计数值, 此处节省大小的隐私预算. 同理, 和可以连接得到长序列. 即使树深度为4大于, 但因为, 预测计数值为, 停止预测, 此处不消耗隐私预算.

前缀序列树本身蕴含有一致性约束: (1) 对任意一条根结点到叶子结点的路径, 满足, 其中表示结点扰动计数, 是的孩子结点; (2) . 若直接基于扰动前缀序列树进行频繁序列模式挖掘, 有可能违背一致性约束导致序列的可用性很差. 本文借鉴文献[12]提出的一种一致性约束推理策略，即通过评估扰动前缀序列树中所有结点(不包括根结点)对应的扰动计数值来满足一致性约束, 保证序列数据的可用性.

1. 例1中假设扰动支持度计数,,,,进行一致性估计得到,,,, 均满足一致性约束.

### 获取频繁序列模式集

获得扰动前缀序列树后, 剩下的任务就是获得满足-差分隐私的闭频繁序列模式集和频繁序列模式集. 该阶段分为两步来完成: (1) 排除非闭合的序列生成, 即满足-差分隐私的闭频繁序列模式集; (2) 分解获得其子集组合, 最终获得频繁序列模式集. 因此, 本文Diff-FSPM算法支持两种频繁模式挖掘, 即频繁序列模式挖掘和闭频繁序列模式挖掘, 两者具有相同的信息表达.

排除非闭合的序列, 即检测中每一条序列记录*s*, 是否存在超序列满足. 本文借鉴文献[15](CloSpan方法)提出的快速包容检查算法(Fast Subsumption Checking Algorithm)执行上述检测工作.

1. 例1中前缀序列树对应中, 执行快速包容检查算法后获得序列和与超序列满足等价关系, 故被排除.

闭序列模式的分解过程, 需要获得所有闭序列的子序列, 对于重复的序列模式, 选择对应扰动计数值最大的, 最终获得频繁序列模式集.

1. 例1中某一闭频繁序列, 其对应7个子序列为: ,其中有三个子序列存在重复且对应扰动计数值都大于当前值, 故该序列被抛弃. 对应序列模式及其扰动计数值为: .

## 隐私性分析

1. TrancatedSeqDB过程满足差分隐私.

证明. 对于任意序列模式,设,,分别表示数据集中对应的真实支持度计数、噪音支持度计数和所有噪音支持度计数集合. 给定两个相邻数据集和, 则:











因此, 该过程满足-差分隐私.

证毕.

1. BuildNoisyPreSeqTree过程满足差分隐私.

证明. 前缀序列树构建过程中, 根结点到叶子结点被称为一条分割链, 分割链上的隐私预算分配满足序列组合特性, 而分割链之间满足并行组合特性. 过程2行4-17部分定义为, 是前缀序列树的构建过程. 给定两个相邻数据集和, 需要证明不等式成立. 定义*V*表示所有可能出现在中的结点集合, 结点*v*分得的隐私预算为, 在数据集和中对应的真实计数分别表示为和.



某一条序列记录影响至多条分割链. 定义表示某一分割链中所有结点集合.等式(?)变换如下:

其中, 最后一个等式中表示影响到的包含结点*v*的分割链总数. 由于满足, 因此最终证明如下:

因此, 过程BuildNoisyPreSeqTree满足差分隐私.

证毕.

1. Diff-FSPM算法满足差分隐私.

证明. 由算法1的描述可知, 算法1的行4没有涉及到隐私问题, 仅仅是满足差分隐私的前缀序列树后置处理, 不需要隐私代价, 满足差分隐私.

由定理2可知行2满足差分隐私, 由定理3可知行3满足差分隐私, 并且顺序执行, 根据性质1序列组合特性可证明, Diff-FSPM算法满足差分隐私, 即满足差分隐私.

证毕.

# 实验结果与分析

在本节中, 将通过具体的实验来对Diff-FSPM算法的可用性和扩展性进行分析、验证和说明. 实验环境为Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 8GB内存, Windows 7 64位操作系统, 实验代码用python实现. pumsb-star 分别描述PUMS的人口普查数据(还包括retail购物篮数据, kosarakWeb点击流数据).上述序列数据集均来自于UCI Machine Learning Repository[[2]](#footnote-2), 具体描述如表2所示. 其中, 表示序列数据集中记录总数, 表示项总数, 表示序列记录最大长度, 表示平均长度.

Table 2 Experimental Dataset Characteristics

表 2 实验数据集特征描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | *|D|* | *|T|* | *max(S)* | *avg(S)* |
| MSNBC | 989, 818 | 17 | 14, 795 | 5.7 |
| PUMSB | 49,046 | 2,088 | 63 | 50 |

## 可用性分析

Diff-FSPM算法引入的误差, 主要来自于三方面: (1) 原始序列数据集截断带来的信息丢失; (2) 预测计数值带来的误差; (3) 噪音计数带来的误差. 我们称前两种为估计误差, 后一种为噪音误差. 给定隐私预算, 误差大小是由最优序列长度决定的. 越小估计误差越大, 但相反噪音计数减小, 原因在于计数全局敏感性降低进而添加的拉普拉斯噪音减小. 反之, 越大噪音误差越大, 同时估计误差减小. 接下来, 首先展示频繁序列模式挖掘和闭频繁序列模式挖掘的结果, 并与已有方法N-gram可用性比较; 然后实验展示不同对误差的影响, 说明0.85经验值的获得过程.

## 具体实验指标

本文执行Diff-FSPM和N-gram算法发布*top-k*频繁序列模式及其支持度计数. 分别采用纳真值(True Positive, TP), 纳伪值(False Positive, FP)和平均相对误差(Average Relative Error, ARE)[9]两类可用性度量指标来衡量上述两种算法的可用性.

设是原始序列数据集对应的*top-k*频繁序列模式集, 为满足差分隐私保护的*top-k*频繁序列模式集.

(1) 纳真值、纳伪值: 主要度量频繁序列模式的可用性. TP值越大, FP值越小说明频繁序列模式的可用性越高.

纳真值表示中*top-k*频繁序列出现在中的数量, 即. 纳伪值表示非频繁序列错误出现在中的数量, 即.

(2) 平均相对误差: 主要度量频繁序列模式对应计数的可用性.值越大说明频繁序列模式对应计数的可用性越高. 计算公式如下:



其中, 和分别表示序列模式对应*top-k*频繁序列模式集的真实支持度计数和扰动支持度计数. 如果不在中, 则设定. 因此, 当时, 说明和完全相等(包括序列模式和对应的支持度计数); 当时, 则完全不相等.

## 实验结果分析

### 可用性分析

本文使用PrefixSpan[14]算法挖掘原始序列数据集获得*top-k*频繁序列模式集. 实验过程中的变量包括隐私预算, *top-k*值.

表3展示不同*k*值对于两种算法纳真值、纳伪值的影响(默认设定=1.0). 随着*k*值的增大, 对应纳真率(即TP相对于*k*比率)减小. 同时, 能够观察到一致性约束在保证序列数据可用性方面的贡献.

Table 3 Changes of TP and FP value under different *k* value

表 3 *k*值变化时纳真值,纳伪值的变化情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *k* | **TP**  Diff-FSPM | **FP**  Diff-FSPM | **TP**  N-gram | **FP**  N-gram |
| 100 | 100/100 | 0/0 | 100/100 | 0/0 |
| 150 |  |  |  |  |
| 200 |  |  |  |  |
| 250 |  |  |  |  |
| 300 |  |  |  |  |

Table 4 Changes of TP and FP value under different *k* value

表 4 *k*值变化时纳真值,纳伪值的变化情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *k* | **TP**  Diff-FSPM | **FP**  Diff-FSPM | **TP**  N-gram | **FP**  N-gram |
| 100 | 100/100 | 0/0 | 100/100 | 0/0 |
| 150 |  |  |  |  |
| 200 |  |  |  |  |
| 250 |  |  |  |  |
| 300 |  |  |  |  |

表4展示不同值对于两种算法纳真值、纳伪值的影响(默认设定=200). 值越大, 对应纳真值越大、纳伪值越小, 充分证明理论分析中指出的较大隐私预算对应较小的噪音添加进而保证较高的可用性. 同时, 能够观察到一致性约束在保证序列数据可用性方面的贡献.

Table 5 Changes of TP and FP value under different value

表 5 值变化时纳真值,纳伪值的变化情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TP**  Diff-FSPM | **FP**  Diff-FSPM | **TP**  N-gram | **FP**  N-gram |
| 0.5 | 100/100 | 0/0 | 100/100 | 0/0 |
| 0.75 |  |  |  |  |
| 1.0 |  |  |  |  |
| 1.25 |  |  |  |  |
| 1.5 |  |  |  |  |

Table 6 Changes of TP and FP value under different value

表 6 值变化时纳真值,纳伪值的变化情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TP**  Diff-FSPM | **FP**  Diff-FSPM | **TP**  N-gram | **FP**  N-gram |
| 0.5 | 100/100 | 0/0 | 100/100 | 0/0 |
| 0.75 |  |  |  |  |
| 1.0 |  |  |  |  |
| 1.25 |  |  |  |  |
| 1.5 |  |  |  |  |

图4展示不同*k*值对于两种算法相对平均误差的影响(默认设定=1.0). (不同的实验数据集分别展示: 短序列为主, 长序列为主)

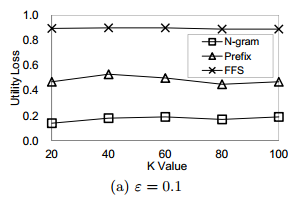


图 4 MSNBC 可用性 VS.

Figure 4 Utlity loss VS. on MSNBC

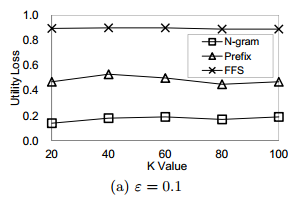


图 5 MSNBC 可用性 VS.

Figure 5 Utlity loss VS. on MSNBC

图5展示不同值对于两种算法相对平均误差的影响(默认设定=200).

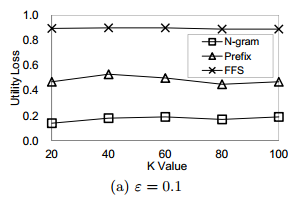


图 6 MSNBC 可用性 VS.

Figure 6 Utlity loss VS. on MSNBC

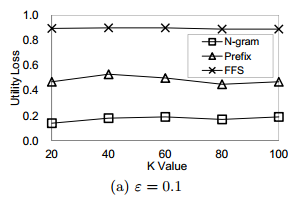


图 7 MSNBC 可用性 VS.

Figure 7 Utlity loss VS. on MSNBC

图6展示最优序列长度对Diff-FSPM算法相对平均误差的影响(默认设定=1.0, =200). 通过图6可以很明显的观察到权衡关系: 小, 估计误差是误差主要来源; 相反大, 噪音误差主导.

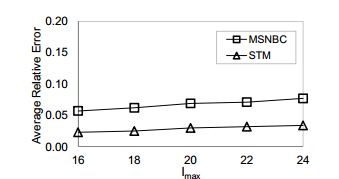
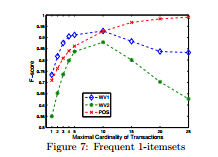
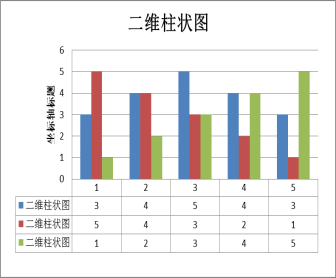


图 8可用性 VS.

Figure 8 Utlity loss VS.

图7展示经验值0.85与Diff-FSPM算法相对平均误差之间的权衡关系. 即值越大对应大, 噪音误差主导; 值越小对应小, 估计误差是误差主要来源.





# 结束语

在本文中, 我们提出了一种基于差分隐私的频繁序列模式挖掘方法Diff-FSPM, 有效地兼容了隐私性和数据可用性. 通过理论分析和实验结果说明了Diff-FSPM方法具有较好的可用性和扩展性.

在未来的工作中, 我们希望能够进一步关注频繁序列模式挖掘方面的研究, 包括频繁子图挖掘等复杂工作, 同时当前处于理论阶段, 致力于应用系统的开发, 如出租车轨迹数据, 差分隐私在推荐系统中的实际应用等.

致 谢 \*致谢内容.\* 致谢

参 考 文 献

1. Agrawal R, Srikant R. Mining sequential patterns[C] //Data Engineering, 1995. Proceedings of the Eleventh International Conference on. IEEE, 1995: 3-14.
2. Sweeney L. k-anonymity: A model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557-570.
3. Zhou S, Li F, Tao Y, Xiao X. Privacy Preservation in Database Applications: A Survey[J]. Chinese Journal of Computers, 2009, 32(5):847-861. (in Chinese)

(周水庚, 李丰, 陶宇飞, 肖晓奎. 面向数据库应用的隐私保护研究综述[J]. 计算机学报, 2009, 32(5):847-861.)

1. Ohm P. Broken promises of privacy: Responding to the surprising failure of anonymization[J]. UCLA Law Review, 2010, 57(6).
2. Dwork C. Differential Privacy[C] //Proc of the 33rd International Colloquium on Automata, Languages and Programming. Berlin: Spinger-Verlag, 2006.
3. Bhaskar R, Laxman S, Simth A, Thakruta A. Discovering Frequent Patterns in Sensitive Data. //Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD). Washington, DC, USA, 2010: 503-512.
4. Li N, Qardaji W, Su D, Cao J. PrivBasis: Frequent Itemset Mining with Differential Privacy. //Proceedings of the 38nd Conference of Very Large Databases (VLDB), Istanbul, Turkey, 2012:1340-1351.
5. Xiaojian Zhang, Miao Wang, Xiaofeng Wang. An Accurate Method for Mining top-k Frequent Pattern Under Differential Privacy[J]. Journal of Computer Research and Develoment, 2014,1(51):104-114. (in Chinese)

(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, 2014,1(51):104-114.)

1. Chen Zeng, Jeffrey F. Naughton, Jin-Yi Cai: On differentially private frequent itemset mining. PVLDB 6(1): 25-36 (2012).
2. Chen R, Fung B, Desai B C, et al. Differentially private transit data publication: A case study on the montreal transportation system[C] //Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012: 213-221.
3. Chen R, Gergely Ács, Claude Castelluccia: Differentially private sequential data publication via variable-length n-grams. CCS 2012: 638-649.
4. Bonomi L, Xiong L. A two-phase algorithm for mining sequential patterns with differential privacy[C] //Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013: 269-278.
5. Pei J, Pinto H, Chen Q, et al. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth[C] //2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE Computer Society, 2001: 0215-0215.
6. Yan X, Han J, Afshar R. CloSpan: Mining closed sequential patterns in large datasets[C] //Proceedings of SIAM International Conference on Data Mining. 2003: 166-177.
7. Dwork C. A firm foundation for private data analysis[J]. Communications of the ACM, 2011, 54(1):86-95.
8. Xiaojian Zhang, Xiaofeng Meng. Differential Privacy in Data Publication and Analysis[J]. Chinese Journal of Computers, 2014,37(4):927-949. (in Chinese)

(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.)

1. Dwork C., F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In TCC, pages 265–284, 2006.
2. F. McSherry, K. Talwar. Mechanism design via differential privacy. In FOCS, pages 94–103, 2007.
3. F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In WWW, pages 171–180, 200 In SIGMOD Conference, pages 19–30, 2009.

附录X.

1. **标识约定**

Table 7 Notational Conventions

表 7 标识约定

|  |  |
| --- | --- |
| **参数** | **描述** |
|  | 隐私预算 |
| *k* | 待挖掘频繁模式数目 |
| *,* | 全局敏感度 |
| *DB* | 样本序列数据库 |
| *FS* | 频繁序列模式集 |
| *CS* | 闭频繁序列模式集 |
| *LS* | 投影数据库闭集 |
| *s,s’* | (频繁)序列 |
| *S* | *DB*中一条完整序列记录 |
|  | 真实支持度计数值 |
|  | 噪音支持度计数值 |
|  | 预测支持度计数值 |
|  | 扰动支持度计数值 |
|  | 中*Root*到结点*v*表示的序列模式 |
| *min\_sup* | 最小支持度计数阈值 |
| *K* | 差分隐私算法 |
| *|DB|* | *DB*中记录总数 |
|  | 投影数据库 |
| ||,|| | 投影数据库项总数 |
| *sr* | 序列*s*扩展(即*s*末尾加一新项) |
|  | 序列先于出现在前缀序列树 |
| *PT,* | 前缀序列树 |
| *lmax* | DB中最大序列长度 |
| *lopt* | 最优序列长度 |
| *levelSet(i, PT)* | *PT*中第*i*层结点全集 |
|  | 结点对应项 |
| *n-seq, n-gram* | 长度为*n*的序列 |
|  | 序列数据集对应项集合 |
|  | 经验值 0.85 |

1. **定理证明**
   1. **定理2**

**定理2.**(支持度计数敏感度) 已知最大序列长度, 设查询函数用以计算特定序列模式的支持度计数, 其中任一序列模式长度满足, 则全局敏感度满足:

|  |  |
| --- | --- |
|  |  |

证明. 给定一条序列记录*x*和长度满足 的序列模式*p*, *x*中出现*p*的计数最大值为. 若给定序列模式长度范围I, 则最大计数满足. 已知最大序列长度, 对应支持度计数查询的全局敏感度为

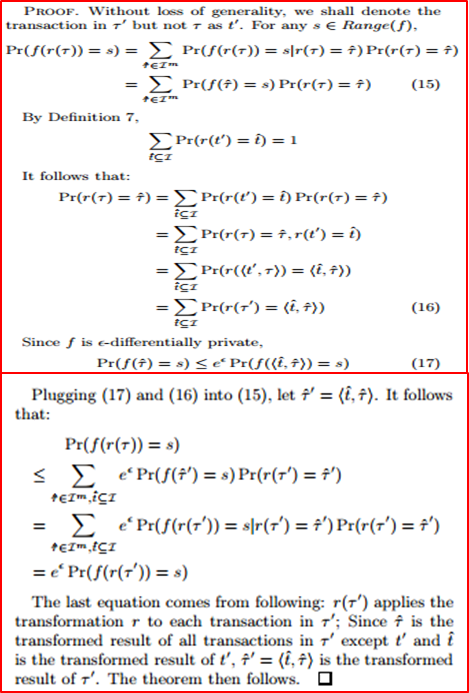
证毕.

* 1. **定理1**

1. (局部变换-差分隐私[2]) 设任一局部变换, -差分隐私算法, 满足所有相邻数据集和, 对于所有*SRange*(), 有

|  |  |
| --- | --- |
|  |  |

证明.

[[3]](#footnote-3)

证毕.

1. **实验结果注意事项**

此处不一定非要top-k, 参照Zeng C对于min\_sup和top-k两种方法实验部分的处理！

*top-k*频繁序列模式挖掘为例, 忽略长度为1的频繁序列集合.

参照N-gram和SmartTrunc方法, 可用性分析中的误差, 分为两种类型来实验分析:

1. 估计错误 approximation error

1.1 直接长度与可用性

1.2 0.85与可用性

都得说明前提: 数据集DB被短序列记录主导.

2. 噪音错误 Laplace error

此处可以分为两种形式的序列模式来展示:

(1) 频繁序列模式

True 应用prefixSpan算法获得

noise 闭频繁序列求子集, 排列组合

(2) 频繁闭序列模式

Ture 应用CloSpan算法获得

noise 闭频繁序列, 取前*k*项

\***附录内容**置于此处，字体为小5号宋体。附录内容包括：**详细的定理证明、公式推导、原始数据**等\*

**Lu Guo-Qing**, born in 1989. B.S. candidate at Institute of Software, Chinese Academy of Sciences. His main research interests include differential privacy and data mining.



**Ding Li-Ping**, born in 1965. Professor and PhD supervisor at Institute of Software, Chinese Academy of Sciences. Her main research interests include computer forensics, system security, and trusted computing, and etc.

**Zhang Xiao-Jian**, born in 1982. PhD candidate at Renmin University of China. Student member of China Computer Federation. His main research interests include differential privacy and data mining.

**Background**

Differential privacy is a relatively new privacy model stemming from the field of statistical disclosure control. Differential privacy, in general, requires that the removal or addition of a single database record does not significantly affect the outcome of any analysis based on the database. Therefore, for a record owner, any privacy breach will not be a result of participating in the database since anything that can be learned from the database with his record can also be learned from the one without his record.

In the paper, we present a method called Diff-FSPM, solving the frequent sequential patterns mining roblem, in which the goal is to find all sequential patterns whose support exceeds a theshold, and guarantee differential privacy without obliterating the utility of the algorithm.

This research is partially supported by the grants from the Natural Science Foundation of China.

\*论文背景介绍为**英文**，字体为小5号Times New Roman体\*

论文后面为400单词左右的英文背景介绍。介绍的内容包括：

本文研究的问题属于哪一个领域的什么问题。该类问题目前国际上解决到什么程度。

本文将问题解决到什么程度。

课题所属的项目。

项目的意义。

本研究群体以往在这个方向上的研究成果。

本文的成果是解决大课题中的哪一部分，如果涉及863\973以及其项目、基金、研究计划，注意这些项目的英文名称应书写正确。

1. 注: 李锡娟, 刘锋, 闰娜娜, 等. PrefixSpan算法与CloSpan算法的分析与研究[J]. 计算机技术与发展, 2008, 8(0): 1. [↑](#footnote-ref-1)
2. 注: 获取地址: <http://archive.ics.uci.edu/ml/datasets.html> [↑](#footnote-ref-2)
3. Chen Z. On Differentially Private Frequent Itemsets Mining. VLDB 2013. (THEOREM 7) [↑](#footnote-ref-3)