满足差分隐私的一种频繁序列挖掘算法

李玉伟[1],杨 庚[1,2]
(1. 南京邮电大学 计算机学院、软件学院、网络空间安全学院,江苏 南京 210023;
2. 江苏省大数据安全与智能处理重点实验室,江苏 南京 210023)

摘 要:在这个大数据时代,无论是数据量还是数据种类都在以极快的速度增长,因此数据挖掘技术在各行各业(例如移动轨迹预测、广告投递、医疗诊断等方面)中都得到了广泛的运用。频繁序列挖掘是数据挖掘领域中的一个重要方向,但是在挖掘过程中和发布序列数据时很有可能会泄露一些用户的隐私信息,产生严重的安全隐患。Dwork 等人提出的差分隐私模型可以为数据挖掘的隐私保护提供安全保证,与传统的隐私保护方法(基于 k-匿名及其扩展分组模型)相比,该模型通过添加噪音对数据进行扰动,即使攻击者拥有最大的背景知识也能达到差分隐私保护的目的。文章设计了一种渐进式序列挖掘差分隐私保护算法,该算法通过改进的稀疏向量技术实现对挖掘过程添加拉普拉斯噪音,并对候选频繁序列的真实支持度以及阈值进行扰动。算法在理论角度被证明满足差分隐私,在真实数据集上的实验结果表明该算法具有较好的可用性。

关键词:频繁模式;序列数据;差分隐私;拉普拉斯噪音;稀疏向量技术
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2022)05-0099-07
doi:10.3969/j.issn.1673-629X.2022.05.017

An Algorithm for Mining Frequent Sequence under Differential Privacy

LI Yu-wei[1], YANG Geng[1,2]
(1. School of Computer, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China)

Abstract: In this era of big data, both the amount and types of data are growing at a very fast speed, so data mining technology has been widely used in all walks of life (such as trajectory prediction, advertising delivery, medical diagnosis and so on). Frequent sequence mining is an important direction in the field of data mining, but in the process of mining and publishing sequence data, it is likely to leak some users' privacy information, resulting in serious security risks. The differential privacy model proposed by Dwork can provide security guarantee for the privacy protection of data mining. Compared with the traditional privacy protection method (based on k-anonymity and its extended grouping model), this model can achieve the purpose of differential privacy protection by adding noise to disturb the data, even if the attacker has the largest background knowledge. An improved SVT (sparse vector technology) method is used to add Laplace noise to a new progressive mining algorithm, which disturbs the real support of candidate frequent sequences and threshold. The algorithm is proved to satisfy the differential privacy in theory, and the experiment on real data sets also shows high-quality usability.

Key words: frequent patterns; sequential data; differential privacy; Laplace noise; sparse vector technology

0 引 言

频繁模式挖掘在数据处理方面具有重要意义,频繁模式挖掘最早由 Agrawal 等人[1]提出,在他们的研究中,频繁模式被定义为频繁项集,其目的是在事务数据库中挖掘出消费者的购买习惯,通过分析消费者已经购买的商品中不同物品之间的联系,观察者可以制定出更优的市场策略。频繁模式挖掘算法目前在网络信息安全、金融预测、地震监测和营销策略等领域均得到了广泛应用。频繁模式算法的范围随着研究的不断深入也已经从最初的频繁项集挖掘,扩展至挖掘更复杂的模式,比如频繁子图挖掘[2]、频繁序列挖掘[3]等等。

收稿日期:2021-04-20 **修回日期**:2021-08-24
基金项目:国家自然基金项目(61872197,61972209)
作者简介:李玉伟(1996-),男,硕士,研究方向为信息安全;杨 庚,博士,教授,研究方向为物联网安全、隐私保护、云计算安全。

频繁序列挖掘算法的目的就是在以序列形式储存的数据集中挖掘出频繁的序列事务。对于序列而言,某一序列中可以多次出现相同项,这是其与项集的差别。因此序列的长度就能突破字符种类上限的限制,序列的种类也会随序列的大小增加呈指数级增长。序列模式和序列数据,如轨迹或 DNA 序列,在许多应用中被广泛使用。例如,从主要道路收集的交通数据可用于确定出行最多的地区和预测交通拥堵。然而频繁序列模式的内容和支持度的计数都会导致用户信息的安全得不到保障[4]。为了解决该问题,Dwork 等人[5]在 2006 年提出了差分隐私模型。该模型在挖掘过程和发布数据中对支持度添加扰动噪音,为数据隐私提供了强有力的、可证明的保障。差分隐私保护模型的隐私保护手段是可量化并且严格的,其隐私保护强度与入侵者持有的信息无关。由于该模型在挖掘过程中对数据进行噪音扰动,可以达到在原数据集中改变序列记录并不会导致挖掘结果出现较大波动的目的。

目前已经有了一些具有差分隐私保护的序列挖掘算法,但这些挖掘算法大多是缺乏交互性的非渐进式的算法。在选择了数据集和确定了最小阈值之后,用户启动算法(比如 PrefixSpan 算法),在算法停止之前,中间的过程中不会得到任何回应。这样的延迟对于数据挖掘的生产力有很大的影响,因此将操作者对于挖掘信息的即时判断处理加入到整体的挖掘过程中十分有必要。Sacha 等人提出的 Prosecco 算法[6]就是一种渐进式的、满足交互性的频繁序列挖掘算法。但此算法同样存在隐私泄露的问题。该文在 Prosecco 算法的基础上,采用差分隐私保护技术对其进行安全性保障,设计了一种满足差分隐私保护的序列挖掘算法 ProSVT。主要贡献如下:

(1)针对交互式、渐进式的序列挖掘算法添加差分隐私保护机制。ProSVT 周期性地返回给用户高度近似的频繁序列结果。这种渐进式的过程体现在根据用户定义的对数据集分块(blocks)的基础下逐渐分析挖掘的过程。

(2)运用并改善了添加噪音的机制,即双层拉普拉斯噪音稀疏向量算法。稀疏向量法[7]是满足差分隐私保护的一种添加拉普拉斯噪音的方法,该文将其运用于挖掘频繁序列的算法中,并且在添加阈值噪音的位置做了改变,使添加噪音后算法挖掘结果更精确和稳定。

1 相关工作

在目前的隐私保护模型方法中,k-匿名模型[8]已经得到了广泛的研究并应用于各领域,而后续研究表明,信息入侵者持有的信息量在很大程度上会影响模型的安全性,而且 k-匿名模型的隐私保护水平并未得到有效且严格的证明。针对这些问题,Dwork 提出的差分隐私保护模型(2006)可以抵挡各类对数据信息的攻击,并且同时可以设定隐私参数来决定其隐私保护水平。Dwork 证明了在背景知识存在的情况下,绝对隐私保护是不可能的,由此产生了基于不可区分性的差分隐私概念。差分隐私要求任何计算对单个记录的更改不敏感,也就是说,针对任意一条记录,数据库包含或者不包含该条记录不影响最后计算的结果。因此,这意味着掌握某条记录的攻击者不能根据对数据库的操作得到关于该条记录的任何有价值信息。实现该模型的噪音机制一般分为两种情况,如果数据为非数值型采用指数机制,反之则采用拉普拉斯机制。

该领域的研究现状在文献[9]中已经得到了非常详尽的介绍。满足差分隐私保护的频繁模式挖掘的研究,最早从频繁项集挖掘相关算法开始。文献[4]定义了一个新的效用概念 top-k 用来量化频繁项集挖掘算法的输出精度,即返回数据集中前 k 个最频繁的项集作为输出结果,该算法将模式长度大于 l_{max} 的事务截断,然后添加 Laplace 噪音对支持度进行扰动以达到安全性,但当参数 k 和 l_{max} 较大时,算法不能保证其挖掘性能。文献[10]提出的 PrivBasis 算法利用了一种称为 θ-基的新概念,θ-基集具有频率大于 θ 的项集是某个基集的子集的性质,该算法采用的 top-k 频繁项集挖掘可以看作是通过降维的方式来处理高维数据。文献[11]中提出的 DP-topkP 算法通过后置处理噪音支持度的方式使结果满足一致性约束,同时也增强了其可用性。由于较长的数据事务会使数据集的敏感度提高,文献[12]提出了一种截断长事务的方法,并且将截断引发的错误与噪音引发的错误进行权衡,该 top-k 算法在 k 不是很小的情况下能获得较好可用性的挖掘结果。和频繁项集挖掘相比,频繁序列挖掘存在高维和序列性的特点,因此这些算法尚不能应用于挖掘频繁序列。

文献[13]提出的基于混合粒度前缀树结果算法首次实现了发布轨迹数据(频繁序列挖掘)中的差分隐私保护。该算法在构建前缀树的过程中使用 Laplace 噪音对数据进行扰动,使其发布的数据结果满足差分隐私保护,但是算法存在前缀树高度增长导致发布数据的效用大大降低的问题。文献[14]采用变长 n-gram 来提取序列数据库的基本信息,利用了前缀搜索树结果和一组基于马尔可夫假设的新技术来降低噪音量,使算法在挖掘的过程中满足差分隐私保护。该算法在短序列为主的数据集下有较好的表现,当存在较多高维数据时挖掘结果的可用性得不到保证。PT-Sample 算法[15]利用数据的统计特性来构造一个

基于模型的前缀树,用于挖掘前缀和子串模式的候选集,但是该算法并不适用于挖掘非字符类型的数据。文献[16]提出的 DPFSM 算法,设计了一种打分函数来区分不同候选序列的优先权,然后通过阈值修正策略来减少截断误差与传播错误,但尚未考虑到隐私预算的分配问题。以上算法都各自有其适用的方向并且都还存在一些问题,因此如何通过设计一种算法使频繁序列挖掘的结果可用性与差分隐私保护的安全性达到较好的平衡是现在研究的重点和难点。

2　理论基础

2.1　差分隐私

差分隐私是通过在挖掘过程中或者输出结果中添加噪音对数据进行扰动来保证数据的安全性,使得在数据集中改变一条记录(移除或者添加一条记录)之后,任何查询的输出结果都不会改变。差分隐私的定义如下:

定义 1:对于相邻数据库 D_1 和 D_2(D_1 和 D_2 之间只相差一条记录),给定算法 A ,算法 A 可能输出的结果集合为 S ,若算法 A 在数据库 D_1 中输出 S 的概率与算法 A 在数据库 D_2 中输出 S 的概率的比值小于常数值 e^{ε} ,称算法 A 满足 ε - 差分隐私保护, 即:

$$\Pr[A(D_1) \in S] \leqslant e^{\varepsilon} \times \Pr[A(D_2) \in S] \quad (1)$$

隐私预算参数 ε 可以衡量算法的隐私保护强度,ε 值越小表示安全性(隐私强度)越高,但是同时也会使数据的可用性降低。

实现差分隐私保护主要依靠对数据的支持度添加噪音的机制,主要包括 Laplace 机制和指数机制,前者针对实数型数据,后者针对字符型数据。噪音量的大小取决于隐私参数和全局敏感度,隐私参数是用户自己设定的,敏感度是数据集在算法下的属性,函数的敏感度为数据库中改变一条事务之后函数输出结果的最大改变量,其数学的定义如下:

定义 2:对于任意一个函数 f , $D \rightarrow R^d$,函数 f 的敏感度 Δf 定义为:

$$\Delta f = \max_{D_1, D_2} \| f(D_1) - f(D_2) \| \quad (2)$$

其中,f 的查询维度为 d , R 为 f 映射的实数空间。

该文的数据集为实数型数据,因此采用 Laplace 机制。为了使算法满足差分隐私保护,Laplace 机制在算法输出结果中加入服从 Laplace 分布的随机噪音。Laplace 机制的定义如下:

定理 1:对于敏感度为 Δf 的函数 f ,算法

$$A(D) = f(D) + \mathrm{Lap}(\lambda) \quad (3)$$

满足 ε - 差分隐私保护,其中 $\mathrm{Lap}(\lambda)$ 是服从 $\lambda = \Delta f/\varepsilon$ 的 Laplace 分布,Laplace 分布的概率密度函数为:

$$\Pr[x \mid \lambda] = (1/2\lambda)\, e^{-|x|/\lambda} \quad (4)$$

参数 λ 是根据隐私参数 ε 和函数敏感度 Δf 共同决定的。

由于处理的问题有时比较复杂,单个差分隐私保护算法不能解决问题,通常需要将用户在算法中指定的隐私参数进行合理分配并采用多个满足差分隐私保护的安全性算法。

定理 2(串行性质):假设 $A_1, A_2, \cdots, A_k$ 为 k 个满足差分隐私保护的算法,那么,当这些算法的组合算法 A 作用于某一数据库时,该算法满足 $\sum_{i=1}^{k} \varepsilon_i$ - 差分隐私保护。

定理 3(并行性质):假设 $A_1, A_2, \cdots, A_k$ 为 k 个满足差分隐私保护的算法,其中,每个算法 A_i 依次满足 ε_i - 差分隐私保护($1 \leqslant i \leqslant k$)。那么,当算法 $A_1, A_2, \cdots, A_k$ 分别作用于数据库 $D_1, D_2, \cdots, D_k$ (所有数据库均不相交)时,这些算法构成的组合算法 A 满足 $\max\{\varepsilon_i\}$ - 差分隐私保护。

由差分隐私保护算法的串行性质和并行性质可以推出如下结论:当存在多个满足差分隐私保护的算法 $A_1, A_2, \cdots, A_k$,对于其构成的组合算法 A 的隐私预算总和,根据需要处理的数据库彼此是否相交有不同的结果,前者为所有算法的隐私预算之和,后者为所有算法中隐私预算的最大值。

2.2　频繁序列挖掘

多条序列记录构成了频繁序列挖掘的数据集,每一条用户序列记录都可能包含该用户的涉及隐私安全的信息。在频繁序列挖掘中,对于某一条序列,其在数据集中出现的次数被定义为该序列的支持度,支持度与数据集中所有序列记录的数量的比值被定义为该序列的频率。类似地,阈值也分别被定义为绝对阈值和相对阈值,对应于支持度和频率。如果某一序列的支持度大于绝对阈值或者其频率大于相对阈值,就可以将该序列加入所要输出的频繁序列集合之中。如果未经特殊说明,文章中的阈值一般指绝对阈值。

已知 $I = \{i_1, i_2, \cdots, i_n\}$ 为序列数据集中项的集合(即字母表)。在频繁序列挖掘中,字母表中任意项的一个非空集组合就构成了一条用户序列,例如 $S = a_1 a_2 \cdots a_{|s|}$ 表示长度为 $|S|$ 的序列 $(a_1, a_2, \cdots, a_{|s|} \in I)$ 。对于一条序列 S ,如果 $|S| = k$,称 S 为一个长度为 k 的序列(即一个 k -序列)。频繁序列挖掘的目的是要发现序列数据库中的所有频繁子序列,并且计算每个频繁序列的支持度。

3　ProSVT 算法

本节详细描述所提出的用于挖掘频繁序列的 ProSVT(Prosecco-SVT)算法。

3.1 概 述

基于 Prosecco 算法,设计了一种具有隐私保护功能的序列挖掘算法,Prosecco 算法首先将数据集分成数量为 b 的子集块, b 为用户指定的参数。在挖掘过程中,当分析完第 i 个子集块后,算法输出一个满足 α_i -近似的中间结果。中间结果在不断更新,用户可以根据当前结果的价值来决定是否继续挖掘,因此整个挖掘过程中得到的数据具有交互性质。

一种挖掘频繁序列的流算法采用了一种用户相关的降低阈值($\xi < \theta$)的方法,并且在所有的子集块中都使用相同的降低后的阈值,这种策略不足以保证中间结果的有效性,因为在某些块中有的序列支持度低于 ξ ,于是就会被忽略,导致结果不准确,从而误导用户。Prosecco 为了避免这种隐患,使用了块相关的降低阈值的方法。这个过程是根据统计学中的相关定理确定了每个子集块的特定参数,结合用户先前设定的错误概率,共同决定误差参数,即阈值下降的幅度。随着挖掘过程的继续,该幅度越来越小,最终收敛于 0,阈值也收敛于标准阈值,算法输出精确挖掘结果。

在挖掘频繁序列的部分使用双噪音(改善过的 SVT)来为候选序列添加拉普拉斯噪音,即在序列支持度和阈值上同时添加噪音并控制噪音大小。此前的 SVT 法是在每个查询序列上添加不同的随机噪音,而在阈值上添加统一的一个噪音,这种做法存在弊端,如果阈值噪音过大就会导致整体比较结果,因此将统一的阈值噪音改为在每一次对查询序列操作时在阈值上添加的不同随机噪音,在比较噪音支持度和噪音阈值时,剔除未能达到噪音阈值的候选序列,反之就将其加入频繁序列集合中。

3.2 算法描述

首先来详细描述提出的 ProSVT 算法(如 Algorithm1 所示),其输入参数有数据集 D ,子集块数量 b ,阈值 θ ,错误概率 δ ,敏感度 Δ ,隐私预算 ε ,算法的输出为最终的频繁序列集合。

ProSVT 算法将数据集 D 分割成 β 个子集块 B_1 , B_2 ,…, B_β ,其中 $\beta = |D|/b$,在同一时间只处理一个子集块。根据统计学的 Vapnik - Chervonenkis (VC) dimension 及相关公式计算出的误差参数 α ,然后由该误差参数 α 得到降低后的阈值 ξ ,将参数 ξ 作为新的阈值进行频繁序列挖掘,获得第一个子集块的输出结果。接着依次对后面的子集块进行操作,不断更新输出序列集,得到最终结果 F 并输出。

Algorithm 1:ProSVT

Input:dataset D , block size b , minimum frequency threshold θ ,failure probability δ ,sensitivity Δ ,privacy budget ε

Output:a set F

1 $\beta \leftarrow |D|/b$

2 $\alpha \leftarrow 2\sqrt{\dfrac{d - \ln(\delta) + \ln(\beta - 1)}{2b}}$

3 $\xi \leftarrow \theta - \alpha/2$

4 $Q \leftarrow$pretreatment(B_1)

5 $F \leftarrow$doubleNoise-getFS(B_1 , Q , Δ , ξ , ε)

6 returnIntermeiateResult (F , α)

7 for each $i \leftarrow 2, \cdots, \beta - 1$ do

8 $\alpha \leftarrow 2\sqrt{\dfrac{d - \ln(\delta) + \ln(\beta - 1)}{2b_i}}$

9 $\xi \leftarrow \theta - \alpha/2$

10 $Q \leftarrow$pretreatment(B_i)

11 $F \leftarrow$upadteRunningSet (F ,doubleNoise-getFS(B_i , Q , Δ , ξ , ε))

12 returnIntermediateResult(F , α)

13 $F \leftarrow$upadteRunningSet(F , B_β , ξ)

14 return F

Algorithm2 是改善过后的 SVT 法,即 doubleNoise-getFS 算法。该算法的输入为子集块 B ,查询队列 Q ,敏感度 Δ ,下调阈值 ξ ,隐私参数 ε ,输出为查询结果。

doubleNoise-getFS 算法中,先将给定的隐私参数 ε 分为 ε_1 (分配给支持度的隐私预算)和 ε_2 (分配给支持度的隐私预算),然后分别计算出支持度噪音和阈值噪音,接着对于查询队列 Q 中每个查询序列的支持度和阈值添加噪音,保留噪音支持度大于噪音阈值的序列,并将其加入挖掘算法的中间结果。

Algorithm 2:doubleNoise-getFS

Input:subset B ,query queue Q ,sensitivity Δ ,lower frequency ξ ,privacy budget ε

Output:query answer a_1 , a_2 ,…,a set F

1 $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$

2 $\rho_1 = \text{Lap}(\dfrac{\Delta}{\varepsilon_1})$, $\rho_2 = \text{Lap}(\dfrac{\Delta}{\varepsilon_2})$, count = 0

3 for each query $q_i \in Q$ do

4 if $q_i(B) + \rho_1 \geqslant \xi + \rho_2$ then

5 Output $a_i = q_i(B)$ and $F.add(a_i)$, count + 1

6 else

7 Output $a_i = \perp$

3.3 隐私性证明

定理 4:ProSVT 算法满足 $\varepsilon_1 + \varepsilon_2$ -差分隐私。

证明:由于在 ProSVT 整体算法中,doubleNoise-getFS 算法中加入了支持度噪音和阈值噪音,因此需要证明加噪过程满足差分隐私保护。对于任意的输出 $a = \{\top, \perp\}^1$,只要证得:

$$\Pr[A(B) = a] = \int_{-\infty}^{\infty} \Pr[\rho_1 = z] \prod_{i \in I_\top} f_i(B, z) \prod_{i \in I_\perp} g_i(B, z)\, \mathrm{d}z \leqslant \mathrm{e}^{\varepsilon_1 + \varepsilon_2} \Pr[A(B') = a] \tag{5}$$

首先考虑第一种情况:

$\forall_i q_i(B) \geqslant q_i(B')$,在这样的条件下可得:

$\frac{f_i(B,z)}{f_i(B',z)} \leqslant 1$,不用变换积分变量来约束 f_i 项的比率,因此 $\rho_2 = \mathrm{Lap}(\frac{\Delta}{\varepsilon_2})$ 足够约束 g_i 项的比率。

$$\begin{aligned} f_i(B,z) &= \Pr[q_i(B) + \rho_2 < \xi + z] \leqslant \\ &\Pr[q_i(B') + \rho_2 < \xi + z] = \\ &f_i(B',z) \end{aligned} \tag{6}$$

$$\begin{aligned} g_i(B,z) &= \Pr[q_i(B) + \rho_2 \geqslant \xi + z] \leqslant \\ &\Pr[q_i(B') + \rho_2 + \Delta \geqslant \xi + z] \leqslant \\ &\mathrm{e}^{\varepsilon_2}\Pr[q_i(B') + \rho_2 \geqslant \xi + z] = \\ &\mathrm{e}^{\varepsilon_2} g_i(B',z) \end{aligned} \tag{7}$$

又因为 $|I_\top| \leqslant c$,所以,

$$\begin{aligned} \Pr[A(B) = a] &\leqslant \int_{-\infty}^{\infty} \Pr[\rho_1 = z] \prod_{i \in I_\top} f_i(B',z) \cdot \\ &\prod_{i \in I_\perp} \mathrm{e}^{\varepsilon_2} g_i(B',z)\,\mathrm{d}z \leqslant \\ &\mathrm{e}^{\varepsilon_2}\Pr[A(B') = a] < \\ &\mathrm{e}^{\varepsilon_1+\varepsilon_2}\Pr[A(B') = a] \end{aligned} \tag{8}$$

然后考虑第二种情况:

$\forall_i q_i(B) \leqslant q_i(B')$,记 $q_i(B) \geqslant q_i(B') - \Delta$,所以有:

$$\begin{aligned} f_i(B,z-\Delta) &= \Pr[q_i(B) + \rho_2 < \xi + z - \Delta] \leqslant \\ &\Pr[q_i(B') - \Delta + \rho_2 < \xi + z - \Delta] = \\ &f_i(B',z) \end{aligned} \tag{9}$$

考虑到约束条件 $q_i(B) \leqslant q_i(B')$,所以有:

$$\begin{aligned} g_i(B,z-\Delta) &= \Pr[q_i(B) + \rho_2 \geqslant \xi + z - \Delta] \leqslant \\ &\Pr[q_i(B') + \rho_2 \geqslant \xi + z - \Delta] \leqslant \\ &\mathrm{e}^{\varepsilon_2}\Pr[q_i(B') + \rho_2 \geqslant \xi + z] = \\ &\mathrm{e}^{\varepsilon_2} g_i(B',z) \end{aligned} \tag{10}$$

再将积分变量 z 转换为 $z - \Delta$,可得:

$$\begin{aligned} \Pr[A(B) = a] &= \int_{-\infty}^{\infty} \Pr[\rho_1 = z - \Delta] \cdot \\ &\prod_{i \in I_\top} f_i(B,z) \prod_{i \in I_\perp} g_i(B,z-\Delta)\,\mathrm{d}z \leqslant \\ &\int_{-\infty}^{\infty} \mathrm{e}^{\varepsilon_1}\Pr[\rho_1 = z] \cdot \\ &\prod_{i \in I_\top} f_i(B',z) \prod_{i \in I_\perp} g_i(B',z)\,\mathrm{d}z \leqslant \\ &\mathrm{e}^{\varepsilon_1+\varepsilon_2}\Pr[A(B') = a] \end{aligned} \tag{11}$$

因此,该算法满足 $\varepsilon_1 + \varepsilon_2$-差分隐私。

4　实验及分析

本节对提出的 ProSVT 算法进行实验仿真,并从精确率、召回率和 F-score 等方面对 ProSVT 算法进行性能分析。

4.1　实验设置

实验环境为 Intel® Core™ i7-9750H CPU@2.60 GHz,16 GB 内存,Windows10 64 位操作系统,实验用 Java 语言实现具体的 ProSVT 算法。

4.1.1　数据集

实验基于两个公开可获得的真实序列数据集(见表 1)。

表 1　真实数据集参数

数据集	序列数目	项数目	序列平均长度
MSNBC	989 818	17	5.7
BMSWebView1	59 601	497	2.5

MSNBC 数据集为 1999 年 9 月 28 日 msnbc.com 网站按照时间顺序记录下的用户访问,数据集中的每个序列对应于该用户在 24 小时内的页面浏览点击,该数据集共有 989 818 条序列记录,点击项为 17 项,对应了 17 个页面类别,序列平均长度为 5.7。BMSWebView1 数据集用于 KDD CUP 2000,它包含来自电子商务的 clickstream 数据,序列量为 59 601,共包含 497 个不同的项,序列平均长度为 2.5。

4.1.2　测试对象

测试主要集中在三个算法:(a)ProSVT:此算法是该文提出的差分隐私保护下的频繁序列挖掘算法;(b)Prefix:这是文献[13]中提出的一种构造前缀树挖掘频繁序列的差分隐私保护算法;(c)N-gram:此算法是利用可变长度的 n-gram 模型来挖掘频繁序列的差分隐私保护算法[14]。

4.1.3　指　标

在实验中测试的衡量标准为 Precision、Recall 和 F-score。其中 Precision 为挖掘的精确率,Recall 为挖掘的召回率,F-score 为两者的综合评价指标。设 Up 是由具有隐私保护的算法挖掘得到的频繁项集,Uc 是真实的频繁项集,具体定义如下:

$$\begin{cases} \mathrm{Precision} = |\mathrm{Up} \cap \mathrm{Uc}| / |\mathrm{Up}| \\ \mathrm{Recall} = |\mathrm{Up} \cap \mathrm{Uc}| / |\mathrm{Uc}| \\ \mathrm{F-score} = 2 \times \dfrac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \end{cases} \tag{12}$$

4.2　实验结果

本节首先针对最低阈值 θ 和隐私预算 ε 对 ProSVT 算法进行测试,然后将 ProSVT 与 Prefix、N-gram 进行对比,并分析实验得出的结果。由于敏感度和序列长度成正比,因此在本实验中用平均序列长度来表征敏感度。

4.2.1　不同阈值

这里测试了 ProSVT 算法在隐私预算固定时,改变阈值对其挖掘性能造成的影响。图 1 和图 2 分别是在 MSNBC 和 BMSWebView1 数据集下的测试结果。实验结果表明,随着阈值的提升,挖掘结果的精确率、召回率都有明显的增加,从而使 F-score 也得到提升。因为阈值提高后,算法挖掘的候选频繁序列数目减少,

出现误差的可能减少,由此提高了挖掘性能。用户在挖掘频繁序列时,不宜将阈值设置得过低,否则会大大降低挖掘结果的可用性。

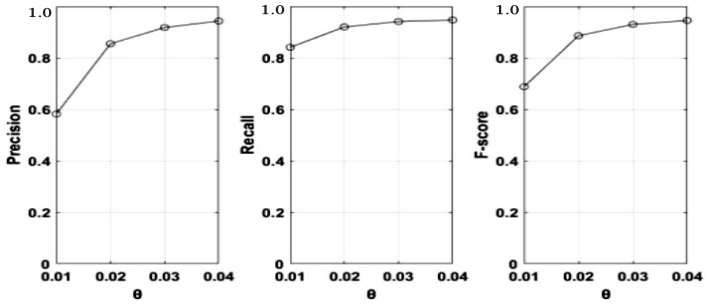

图 1 MSNBC:不同 θ 下的挖掘性能

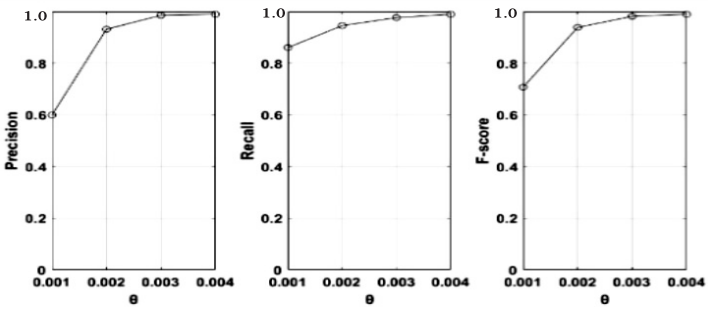

图 2 BMSWebView1:不同 θ 下的挖掘性能

4.2.2 不同隐私预算

这里测试了 ProSVT 算法在最低阈值固定时,改变隐私预算对其挖掘性能造成的影响。图 3 和图 4 分别是在 MSNBC 和 BMSWebView1 数据集下的测试结果。可以看出随着隐私预算的提升,挖掘结果的精确率、召回率都有明显的增加,从而使 F-score 也得到提升。当隐私预算增加时,所添加的拉普拉斯噪音随之降低,因此理论上会使挖掘结果更加精确。

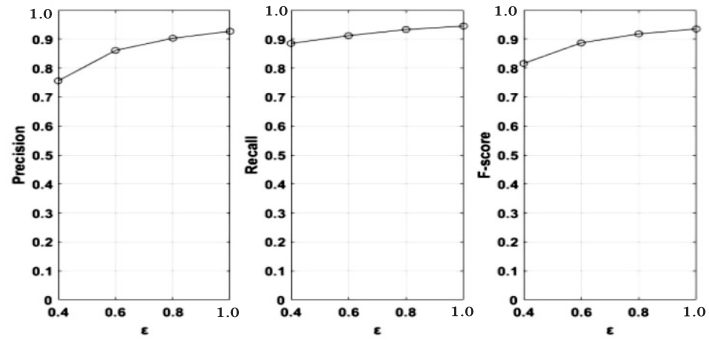

图 3 MSNBC:不同 ε 下的挖掘性能

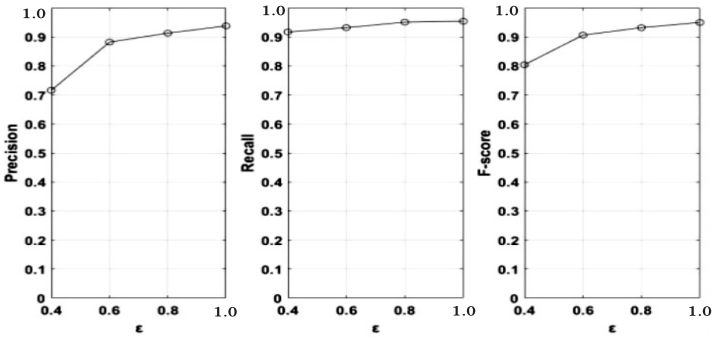

图 4 BMSWebView1:不同 ε 下的挖掘性能

4.2.3 ProSVT 与其他算法

这里测试当固定阈值,改变隐私预算时在两个数据集中 ProSVT、Prefix 以及 N-gram 算法的 F-socre 指标变化,如图 5 和图 6 所示。对于 MSNBC 数据集,ProSVT 在低隐私预算时表现较差,当隐私预算大于 0.4 后,其挖掘性能较 Prefix 和 N-gram 更优秀;对于

BMSWebView1 数据集,ProSVT 在各种大小的隐私预算情况下的挖掘性能表现都要优于 Prefix 和 N-gram。

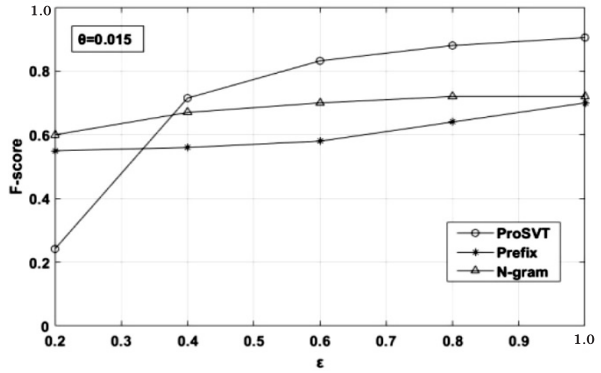

图 5　MSNBC:三种算法的挖掘性能对比

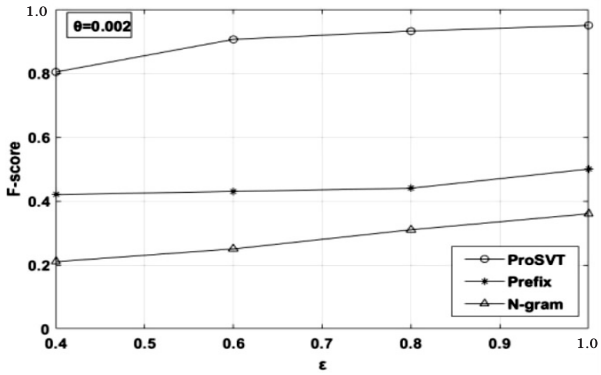

图 6　BMSWebView1:三种算法的挖掘性能对比

5　结束语

提出了一种满足差分隐私保护的频繁序列挖掘算法 ProSVT。此算法是在 Prosecco 算法的基础上加入了改善过后的双拉普拉斯噪音方法以保证挖掘期间的差分隐私机制,以其渐进式、交互性的特点给挖掘者更便捷、有效的挖掘频繁序列的体验。在真实数据集下的相关实验表明了最低阈值和隐私预算的分配对于挖掘性能的影响,在和其他算法对比的过程中,ProSVT 也有比较出色的表现。在隐私预算的分配中,对于支持度噪音和阈值噪音的隐私预算分配采用均分的方法,进一步的研究中可以针对隐私预算分配方法作深入的探讨。

参考文献:

[1] AGRAWAL R, IMIELINSKI T, SWAMI A. Mining association in large databases[C]//Proceedings of the ACM SIGMOD international conference on management of data. Washington, D. C.: ACM, 1993: 207-216.

[2] KURAMOCHI M, KARYPIS G. Frequent subgraph discovery[C]//Proceedings of IEEE international conference on data mining. San Jose, CA, USA: IEEE, 2001: 313-320.

[3] PEI J, HAN J, MORTAZAVI-ASL B. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth[C]//Proceedings of IEEE international conference on data engineering. San Jose, CA, USA: IEEE, 2001: 215-224.

[4] BHASKAR R, SMITH A. Discovering frequent patterns in sensitive data[C]//Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining. Washington, D. C.: ACM, 2010: 503-512.

[5] DWORK C. Differential privacy[C]//Proceedings of the 33rd int colloquium on automata, languages and programming. Berlin: Springer, 2006: 1-12.

[6] SERVAN-SCHREIBER S, RIONDATO M, ZGRAGGEN E. ProSecCo: progressive sequence mining with convergence guarantees[J]. Knowledge and Information Systems, 2020, 62(4): 1313-1340.

[7] LYU M, SU D, LI N. Understanding the sparse vector technique for differential privacy[J]. Proceedings of the VLDB Endowment, 2016, 10(6): 637-648.

[8] SWEENEY L. K-anonymity: a model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557-570.

[9] 卢国庆,张啸剑,丁丽萍,等. 差分隐私下的一种频繁序列模式挖掘方法[J]. 计算机研究与发展,2015,52(12):2789-2801.

[10] LI N, QARDAJI W, SU D. Privbasis: frequent itemset mining with differential privacy[J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1340-1351.

[11] 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.

[12] ZENG C, NAUGHTON J F, CAI J Y. On differentially private frequent itemset mining[J]. Proceedings of the VLDB Endowment, 2012, 6(1): 25-36.

[13] CHEN R, FUNG B C M, DESAI B C. Differentially private transit data publication: a case study on the montreal transportation system[C]//Proceedings of the ACM international conference on knowledge discovery and data mining. Beijing, China: ACM, 2012: 213-221.

[14] CHEN R, ACS G, CASTELLUCCIA C. Differentially private sequential data publication via variable-length n-grams[C]//Proceedings of the 2012 ACM conference on computer and communications security. Raleigh, NC: ACM, 2012: 638-649.

[15] BONOMI L, XIONG L. A two-phase algorithm for mining sequential patterns with differential privacy[C]//Proceedings of international ACM conference on information and knowledge management. Washington, D. C.: ACM, 2013: 269-278.

[16] ZHOU F, LIN X. Frequent sequence pattern mining with differential privacy[M/OL]//Lecture notesin computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 2018. http://dx. doi. org/10. 1007/978-3-319-95930-6_4.