高技术通讯 2013 年 第23 卷 第3 期:251 ~256

doi:10.3772/j.issn.1002-0470.2013.03.005

面向数据流的敏感规则 k-匿名保护算法①

张君维② 杨 静 张健沛 张乐君

(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

摘 要 为了使攻击者通过降低阈值来发现被隐藏规则的概率小于 $1/k$,以此实现对敏感规则的二重保护,提出了一种面向数据流的敏感规则 k-匿名保护算法。该算法首先用时间滑动窗口技术来获取数据使用者最感兴趣的最新 n 个时刻到达的数据,然后通过追加交易的方式而并非修改交易项的方式来实现对敏感规则的 k-匿名,从而避免对数据流的二次访问以及被隐藏敏感规则的泄漏。同时采用素数编码的方法,用素数集代替原始项集参与运算,以提高算法的效率和降低算法的空间复杂度。实验结果表明,此算法可以在数据流环境下高效进行敏感规则的 k-匿名,并且能够保证挖掘结果的有用性。

关键词 k-匿名,数据流,关联规则,敏感规则

0 引 言

k-匿名技术[1]已广泛应用于原始数据保护,文献[2,3]将 k-匿名技术加以改进与完善,实现了对原始数据中的敏感属性值的保护。然而,将 k-匿名技术应用于知识保护的研究还不是很多。文献[4]将 k-匿名应用于关联规则隐藏(association rule hiding, ARH)——通过对原始数据的修改产生屏蔽规则,将其加入到隐匿区域,快速实现敏感规则的 k-匿名保护。但此算法针对静态数据,不能被直接应用于数据流。数据流作为普遍存在的数据形态,已经吸引了越来越多的研究人员的关注,文献[3,5]就提出了面向数据流发布的隐私保护算法。然而,迄今为止,在数据流环境下,对挖掘到的关联规则结果实现敏感规则隐藏[4,6-8]的 k-匿名保护还鲜有研究。敏感规则隐藏通常是通过降低强敏感规则的支持度或置信度到最小阈值以下的方法来实现,并且,为了最大程度地保证隐藏后所产生的负面效应最小,要尽量最小程度地降低支持度或置信度到最小阈值以下,所以这就会使得攻击者通过降低最小支持度或置信度来发现被隐藏的敏感规则,造成隐私的泄露。为了解决这一问题,可以采用对敏感规则进行 k-匿名的方法,即攻击者通过降低相应的最小支持度与置信度阈值后,将得到至少 k 条规则,这就使得其确定敏感规则的概率下降到 $1/k$ 及以下,从而敏感规则得到了保护。为此,本文提出了一种面向数据流的敏感规则 k-匿名保护算法。因为面向的是敏感规则隐藏后的数据流,而规则隐藏常用方法是通过清洗数据项来实现的,这意味着基于数据流的特征清洗数据项后的数据流不能再进行访问,所以对敏感规则的匿名化,如果也通过交易清洗来完成,难度较大,也可能会影响规则隐藏的准确度,产生过量的负面效应。因此,本文通过扩展交易数据流的方法来做敏感规则的匿名化。扩展的交易数据被追加到滑动窗口,并将其时间戳设置为窗口中最新的时间刻度 t_i,随着滑动时间窗口[9,10]的移动,扩展的交易数据将更新得到的规则集,实现规则的 k-匿名。

1 基本概念及相关技术

1.1 滑动时间窗口

考虑到数据使用者对最新到达数据比对历史数据更感兴趣,我们结合滑动时间窗口技术来随时捕获数据流中最新时刻到达的数据,并对挖掘到的结果进行敏感规则 k-匿名保护。

我们面对数据流 DS,要实现敏感规则的 k-匿名保护,其中 DS 为已经进行了敏感关联规则隐藏的

① 国家自然科学基金(61073041,61073043),黑龙江省自然科学基金(F200901),高等学校博士学科点基金(20112304110011)和优秀学科带头人专项资金(2011RFXXG015, 2010RFXXG002)资助项目。

② 女,1983 年生,博士生;研究方向:数据挖掘,隐私保护;联系人,E-mail: zhangjunwei20@hrbeu.edu.cn

(收稿日期:2012-06-12)

数据流，即 $\forall sr \in SR$, s.t. $sr \notin f(DS, MST, MCT)$，这里，$SR$ 为敏感关联规则集，$f(data, sup, conf)$ 为关联规则挖掘算法，即从数据 $data$ 中挖掘得到 $(sup, conf)$ -强关联规则。sup 与 $conf$ 分别为支持度与置信度，MST 为最小支持度阈值，MCT 为最小置信度阈值。将最新的 n 个时刻到达的数据组成滑动时间窗口 DW，即 $DW = \{DW_1, DW_2, \cdots, DW_n\}$，其中，$DW_i$ 为第 t_i 时刻到达的数据集组成的子窗口，DW_1 较 DW_n 为时间更早到达的数据集，如图 1 所示。

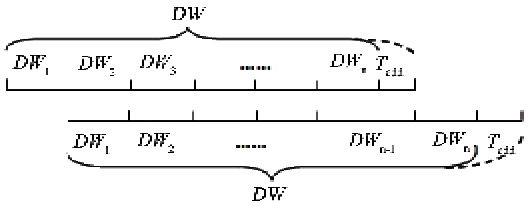

图 1 滑动时间窗口 DW

为了实现敏感规则的 k-匿名，我们采取了追加交易的方法，即通过我们提出的算法，确定追加的交易数据集 T_{add}，将其时间戳赋值为 t_n，与 DW_n 组成新的子窗口 DW_n，从而隐藏追加了新数据的痕迹。当新数据到达时，滑动时间窗口移动，DW 数据更新，再次执行敏感规则的 k-匿名算法。

1.2 敏感规则 k-匿名模型

通过图 2，我们可以很直观地了解到备选规则集 $RS\text{Ⅲ} = f(DW, MST, MCT)$，即 DW 在最小支持度阈值 MST 及最小置信度阈值 MCT 下，通过挖掘算法 $f(data, \text{sup}, conf)$ 所得到的强关联规则集。并且 $RS\text{Ⅲ} \cap SR = \varnothing$。$RS\text{Ⅱ} = f(DW, \text{sup}', conf') - RS\text{Ⅲ}$ 是目标的 k-匿名规则集，其存在的前提条件是 $RS\text{Ⅱ} \cap SR \neq \varnothing$。$RS\text{Ⅰ}$ 为用于规则 K-匿名的备选规则集，$RS\text{Ⅰ} = RS\text{Ⅰ}' + RS\text{Ⅰ}'' = f(DW, \text{sup}', conf'') + f(DW, \text{sup}'', conf') - f(DW, \text{sup}', conf')$，从备选规则集 $RS\text{Ⅰ}$ 中选取至少 $k\text{-}count(RS\text{Ⅱ})$ 条规则，通过提高支持度或置信度，调整到 $RS\text{Ⅱ}$，从而实现 $RS\text{Ⅱ}$ 中的规则 k-匿名，被选中规则定义为准入规则。

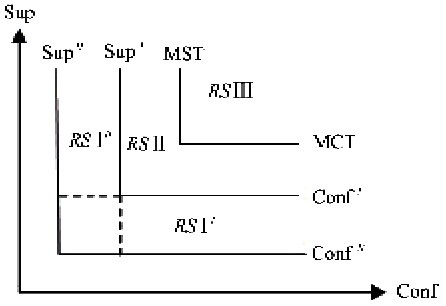

图 2 阈值空间

定义 1：数据流 DS 的任意一个滑动时间窗口 DW 有 $\exists sr \in SR$, s.t. $sr \in RS\text{Ⅱ}$ 并且 $count(RS\text{Ⅱ}) \geqslant k$ 或者 $\nexists sr \in SR$, s.t. $sr \in RS\text{Ⅱ}$，我们就说数据流 DS 满足敏感规则的 k-匿名。

下面计算 sup′, sup″, $conf'$, $conf''$ 的取值。

定理 1：给定规则挖掘的最小支持度阈值与最小置信度阈值为 MST 与 MCT，进行敏感规则隐藏后，被隐藏的敏感规则的支持度与置信度的下界 sup' 与 $conf'$ 分别为 $MST - \frac{1}{N}$ 和 $MCT - \frac{1}{MST \cdot N}$。

证明：在最小支持度与置信度阈值为 MST 与 MCT 的情况下，可以得到敏感规则 $x \rightarrow y$，它的支持度与置信度分别为 S 和 C，$N = |DW|$。在最小负面效应前提下执行隐藏敏感规则，一方面是降低敏感规则的支持度，即 $sup(x \rightarrow y) = \frac{(P(x \cup y) - 1)}{N} = S - \frac{1}{N}$, $\min(sup(x \rightarrow y)) = MST - \frac{1}{N}$，另一方面是降低敏感规则的置信度，即 $conf(x \rightarrow y) = \frac{(P(x \cup y) - 1)}{P(x)} = C - \frac{1}{P(x)}$, $\min(conf(x \rightarrow y)) = MCT - \frac{1}{MST \cdot N}$。我们这里假设的前提是通过移除项的方法进行规则隐藏。

定理 2：根据定理 1 中得到的 sup' 与 $conf'$，可得到备选规则的支持度与置信度的下界 sup'' 与 $conf''$ 分别为 $sup' + \frac{sup' - 1}{N}$ 和 $conf' - \frac{1 - conf'}{sup' \cdot N}$。

证明：首先计算 $RS\text{Ⅰ}''$ 的支持度下界 sup''，假设 $\exists rule\ x \rightarrow y \in RS\text{Ⅰ}''$，具有最小支持度 sup''，即 $sup'' = P(x \cup y)/N$，通过追加交易，将 $x \cup y$ 的支持数加 1，而使得 $x \rightarrow y$ 的支持度大于 sup'，即 $\frac{P(x \cup y) + 1}{N + n} \geqslant sup'$，推理得到 $\frac{P(x \cup y)}{N} \geqslant sup' + \frac{sup' \cdot n}{N} - \frac{1}{N}$，即最小支持度 $sup'' = sup' + \frac{sup' \cdot n}{N} - \frac{1}{N}$，我们要取得 sup'' 的最小值，即当 n 取最小值 1，也就是说，只追加一条交易且支持 $x \cup y$，所以，得到 $sup'' = sup' + \frac{sup' - 1}{N}$。

其次计算 $RS\text{Ⅰ}'$ 的置信度下界 $conf''$。假设

$\exists rule\ x \rightarrow y \in RS\text{Ⅰ}'$,具有最小置信度 $conf''$,即 $conf'' = P(x \cup y)/P(x)$,在追加的交易中,有一条交易支持 $x \cup y$,从而使得 $x \rightarrow y$ 的置信度大于 $conf'$,即 $\frac{P(x \cup y)+1}{P(x)+n} \geqslant conf'$,推理得到 $\frac{P(x \cup y)}{P(x)} \geqslant conf' + \frac{conf' \cdot n}{P(x)} - \frac{1}{P(x)}$,即最小置信度 $conf'' = conf' + \frac{conf' \cdot n}{P(x)} - \frac{1}{P(x)}$,我们要取得 $conf''$ 的最小值,就要取 n 的最小值1,且使 $P(x) = P(x \cup y)$,即包含 x 的交易同时都包含项 y,进而得到 $conf'' = conf' - \frac{1 - conf'}{sup' \cdot N}$。

1.3 算法损失度量

进行敏感规则 k-匿名保护前后,采用新的 (MST, MCT) 强关联规则不同于原始 (MST, MCT)-强关联规则的数量占原始 (MST, MCT)-强关联规则数量的比率来定义算法的损失度量,即

$$Loss = \frac{\| R_b - R_a \|}{| R_a |}$$

R_a 为规则 k-匿名前强关联规则集,R_b 为执行敏感规则 k-匿名之后的强关联规则集,$\| R_b - R_a \|$ 为敏感规则 k-匿名后较之前获得的强关联规则的差异数量,$Loss$ 值越大,表示进行规则 k-匿名保护后,数据的有用性越低,$Loss$ 值为0,表示敏感规则 k-匿名保护的同时,也最完美地保持了完整的挖掘结果。

2 敏感规则的 k-匿名算法

敏感规则的 k-匿名保护,实际上是对敏感规则的二次保护,即对于已经被隐藏的敏感规则,增加其所在阈值范围内的规则数量到 k,进而降低敏感规则的暴露概率。算法的基本思想是:首先确定需要进行敏感规则的 k-匿名保护条件,即 $RS\text{Ⅱ}$ 中存在敏感规则 SR_i,且 $RS\text{Ⅱ}$ 中规则数量小于 k;然后在 $RS\text{Ⅰ}$ 中确定可被调整到 $RS\text{Ⅱ}$ 去实现规则 k-匿名的准入规则,最后根据准入规则,在 DW_n 中追加完全支持准入规则的最少项交易。具体的算法描述见图3。

算法:敏感规则 k-匿名算法
输入:进行了敏感规则隐藏的数据流 DS,匿名约束 k,支持度与置信度阈值 MST, MCT,敏感规则集 SR
输出:满足敏感规则 k-匿名的数据流 DS'
步骤:1. 数据流 DS 中最新的 n 个连续时刻到达的数据构成时间滑动窗口 DW;
2. 按素数编码读取 DW 中的交易数据中的项;
3. 根据规则支持度与置信度,更新规则集合 $RS\text{Ⅰ}$, $RS\text{Ⅱ}$, $RS\text{Ⅲ}$;
4. 循环,直到 $RS\text{Ⅱ} \cap SR = \varnothing$,或者 $count(RS\text{Ⅱ}) \geqslant k$
For $R_i \in RS\text{Ⅰ}$
If $mul(R_i)/mul(R_j) \neq 0$ and $mul(R_i)/mul(r_{pre}) \neq 0$
$R_{sel} = R_{sel} + R_i$;
追加相应交易 t_{add} 到 DW_n,并且将追加交易的时间戳设为 t_n;
更新 $RS\text{Ⅰ}$, $RS\text{Ⅱ}$, $RS\text{Ⅲ}$;
5. $DS' = DS' + DW_n$;
6. 滑动时间窗口移动,$DW_n = DW_{n+1}$;

图3 敏感规则 k-匿名算法描述

2.1 项编码

使用素数集依次对项集进行编码,如,项集 I = {a,b,c,d,e},编码后的项集 I = {2,3,5,7,11}[3]。算法的第2步就是根据项编码读取到交易数据,由重新编码后的素数项组成的交易数据流得到相应的素数规则集合,并且基于素数只能被1和本身整除的特点,根据规则对应项集的素数积,唯一确定规则所包含的项。进而,通过项积整除法(稍后介绍)进行备选规则的筛选运算,帮助快速地选择到较优的备选规则,并且用素数集代替原始项集参与运算,能够降低运算占用的内存空间。

算法依据项编码后的交易及规则进行 k-匿名,我们用函数 $mul(Rule)$ 计算规则的项积,即组成规则的项的素数乘积,参与到算法的第4步,进行准入规则的选择。

2.2 确定准入规则集 R_{sel}

2.2.1 筛选原则

提高 $RS\text{Ⅰ}$ 中部分规则的支持度或置信度,使其加入 $RS\text{Ⅱ}$,实现敏感规则的 k-匿名。要保证敏感规则不会成为 (MST, MCT)-强关联规则,并且为了保证对挖掘结果所产生的影响最小,要使得追加的交易不会产生强关联规则被错误隐藏,以及不会使得非频繁规则成为 (MST, MCT)-强关联规则。

算法第4步使用 if 语句从 $RS\text{Ⅰ}$ 中确定准入规

则,即备选规则集 $RS\text{Ⅰ}$ 中,满足筛选条件且被调整其支持度或置信度到 $RS\text{Ⅱ}$ 的规则,其具体的筛选原则如下:

(a)避免 $RS\text{Ⅱ}$ 中的非频繁规则成为 (MST, MCT) -强关联规则,使得准入规则不完全包含 $RS\text{Ⅱ}$ 中的规则,即 $R_{sel} \cap RS\text{Ⅱ} = \varnothing$,在提高准入规则 R_{sel} 支持度的同时不会使 $RS\text{Ⅱ}$ 中非频繁规则的支持度增加而成为频繁规则,减少对规则挖掘结果准确性的影响。

(b)避免 (MST,MCT) -强关联规则被误隐藏,使得选规则不包含 R_i 的前项集合。R_i 定义为规则集,它们的支持度大于 MST 且置信度范围为 $\left[MCT, \frac{MCT \cdot S_r \cdot N}{S_r \cdot N - MCT}\right]$,通过增加规则前项的支持度,会使得规则的置信度降低到最小置信度阈值 MCT 以下,造成错误隐藏,影响挖掘结果的完整性。

R_i 规则集的建立,对于 $RS\text{Ⅲ}$ 中的频繁规则来说,如果追加的交易完全支持某一规则,会提高其支持度或置信度,保持其为频繁规则,如 $RS\text{Ⅲ}$ 中的频繁规则 $x \rightarrow y$, 其支持度为与置信度分别为 S_r 和 C_r, 在追加交易 (x, y) 后,支持度 $S_r' = \frac{P(x \cup y) + 1}{N + 1}$, 一定是大于等于 S_r 的,同理 $C_r' = \frac{P(x \cup y) + 1}{P(x) + 1}$, 也一定大于等于 C_r, 可见,不会影响到规则的频繁性。然而,如果追加的交易只是部分支持规则的前项,则会使其置信度降低,可能成为非频繁规则,影响挖掘结果。同样,对于 $RS\text{Ⅲ}$ 中的频繁规则 $x \rightarrow y$, 如果追加交易 (x), 只是部分支持规则前项,则有 $C_r' = \frac{P(x \cup y)}{P(x) + 1}$, 显然, C_r' 一定是小于 C_r 的, R_i 为会被错误隐藏的规则集,即 $Ci'_r < MCT$, 且 $C_r = \frac{P(x \cup y)}{P(x)}$, 因此得到 C_r 的取值范围为 $\left[MCT, \frac{MCT \cdot S_r \cdot N}{S_r \cdot N - MCT}\right]$。

2.2.2 项积整除法

依据素数的特点,即,规则的项积只能被分解成多个固定的素数之积,提出项积整除法:如果规则 R_1 的项积能够被规则 R_2 (或项集 IS) 的项积所整除,则表示规则 R_1 包含规则 R_2 (或项集 IS) 中的所有项。利用此方法,可以快速实现准入规则集的筛选。

算法第 4 步中的 if 语句即是使用项积整除法来实现对规则的筛选,其中 $mul(R_i)/mul(R_j) \neq 0$ 为筛选规则(a)的实现,从 $RS\text{Ⅰ}$ 中筛选掉那些可以被 $RS\text{Ⅱ}$ 中项积整除的规则; $mul(R_i)/mul(r_{pre}) \neq 0$ 则为筛选规则(b)的实现,筛选掉那些可以被 R_i 的前项集合对应项积所整除的规则。

2.3 确定追加的交易集 T_{add}

算法第 4 步循环确定满足筛选条件的准入规则,并根据每一次确定的规则 r_{sel}, 追加的一条完全且仅仅包含规则 r_{scl} 对应项集的交易,以此来增加规则 r_{sel} 的支持度。如:确定的准入规则集 $R_{sel} = \{(a \rightarrow b), (ce \rightarrow f)\}$, 则追加交易 $t_{add}(ab)$ 和 $t_{add}(cef)$ 到 T_{add}。

3 实验结果

3.1 实验数据

实验在 CPU 为 PⅣ2.0GHZ、内存为 2G、操作系统为 WINXP 的 PC 机上进行,所有的实验程序均采用 Visual C + + 实现。实验中的模拟数据由 IBM 模拟数据生成器[9] (http://www.almaden.ibm.com) 产生。实验从运行时间、内存使用以及信息损失度量等角度来分析算法的性能。

3.2 算法的运行性能

第一个实验是测试算法在每个数据分段上的运行性能。实验使用数据 T10I5D1000K,数据流被分割成 10 个大小为 100kb 的数据分段,每个数据分段被看作是某一时间戳下到达的数据子窗口, MST 与 MCT 分别设为 0.5% 和 80%。图 4(a)所示滑动窗口分别为 0.3Mb、0.4Mb、0.5Mb 时算法在数据流各个数据分段上运行所需要的内存空间。当滑动时间窗口固定时,每一次都处理相等的时间范围内的数据,则存入内存待处理的数据数量大致相同。如果数据分布较平均,那么处理的规则数量也比较相近,并且当新的数据流不断产生时有部分过期数据会被释放,所以内存的使用率不会发生非常显著的变化。而当滑动窗口增大时,读入的数据量增多,但是需要更新及维护的规则会存在重叠,因此内存的增幅不会太大。图 4(b)所示为在数据流各个数据分段上算法运行的平均时间。当滑动窗口固定时,其处理的数据量是大致相同的,针对每一个数据分段,进行数据的读取,对其影响的规则进行信息更新,确定准入规则实现 k-匿名操作,其时间复杂度不会有太大变化。但当滑动窗口增大时,在相同阈值下,要处理的规则集合会有所增加,则参与计算的规则数量增多,会一定程度上增长算法的运行时间。当然,增加的新规则是有限的,因此不会大幅度增长其算法的运行时间。

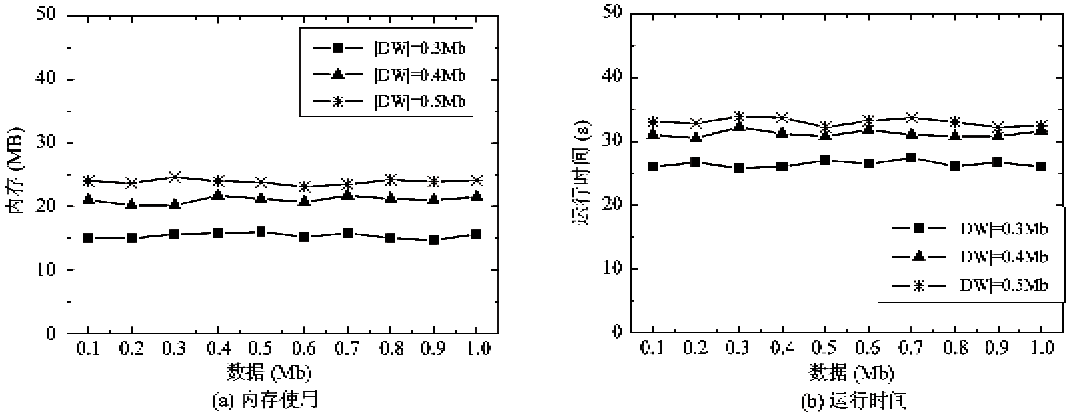

图 4　算法在每个数据分段上运行的性能

3.3　不同数据集下算法的运行性能及信息损失比较

第二个实验是进行不同的数据集下算法的运行时间以及信息损失的比较。针对不同的数据流,将滑动时间窗口设为 0.3Mb, *MST* 设为 0.5%, *MCT* 为 80%, k 分别取 10、12、14、16、18 时,对算法的执行时间以及所产生的信息损失进行比较。对整个数据流来说,k 的取值增大时,虽然会保证敏感规则的安全度更高,但是算法要确定的准入规则以及需要追加的交易数据也就会增多,则算法在数据流上的运行时间也会相应地增加。并且,追加交易的方法一方面可能提高其完全支持规则的支持度,另一方面也可能降低其不完全支持的规则的置信度。我们通过前者来实现敏感规则的 k-匿名,同时后者也会使得部分 *RS*Ⅱ 中的规则的置信度降低而脱离 *RS*Ⅱ,使得算法要消耗更多的时间完成敏感规则 k-匿名。此实验分别测试了算法在三套数据集(T10I5D1000K, T10I8D1000K, T15I8D1000K)上的运行时间以及所产生的信息损失 *Loss*。如图 5(a),当 k 取值为 10 和 12 时算法在数据集 T10I8D1000K 上的运行时间,以及 k 取值为 10 时算法在数据集 T15I8D1000K 上的运行时间均很低,这是因为每个数据段上得到的关联规则挖掘结果基本满足 k-匿名要求,算法执行确定准入规则以及追加交易的操作较少。当 k 值不断增大时,算法在各数据集上的运行时间也会不断提高,主要集中在确定准入规则上,同时,也会有越多的规则在 *RS*Ⅱ 与 *RS*Ⅰ 之间出现反复,所以会消耗更多的时间,降低算法的效率。同时,k 的取值也直接关系着 k-匿名后的数据流挖掘结果的完整性与准确性,如图 5(b)所示,对于数据集 T10I8D1000K 和 T15I8D1000K 分别在 k 取值 10、12 和 k 取值为 10 时,所产生的 *Loss* 都较低,也是因为数据分段存在满足 k-匿名的情况,调整的规则较少。随着 k 值的提高,要追加更多的交易数据,则可能造成 *Loss* 的增加。这是因为追加更多的交易数据会使得部分强规则的支持度下降到最小置信度阈值以下而被错误隐藏,并且 k 越大,追加的交易越多,错误隐藏的可能性及数量也就越高。

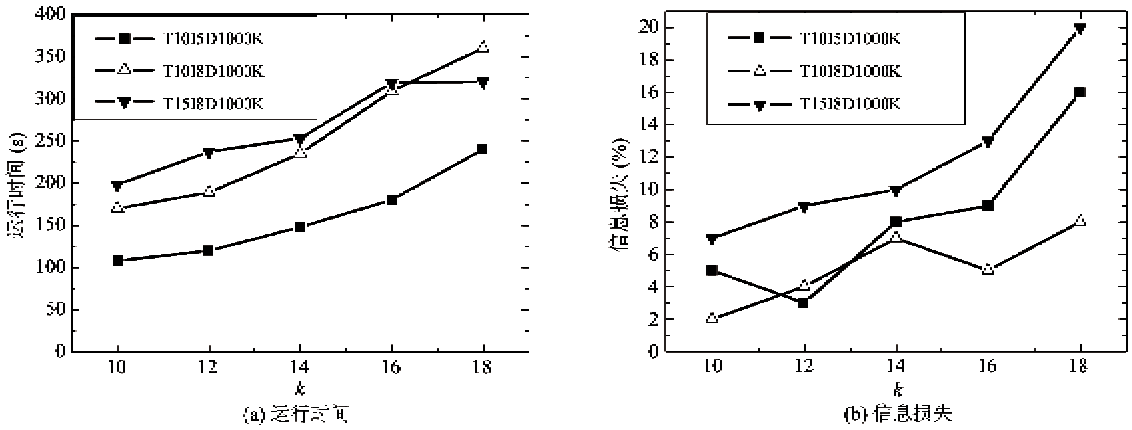

图 5　不同数据集下算法的运行时间及信息损失比较

4 结 论

为了进一步保证面向数据流的关联规则挖掘结果的安全性,我们提出了敏感关联规则 k-匿名保护算法,此算法充分考虑到了数据使用者对最新数据具有较高关注度以及对数据流无法多次访问的特点,采用了滑动时间窗口技术来获取最新 N 个时刻到达的数据,并且,通过追加交易的方式实现对敏感规则的 k-匿名,从而避免二次修改数据流对数据安全的影响。算法还应用到素数编码项集的方法,利用素数的显著特点来帮助运算,找到最适合的备选规则,在提高速度的同时,也大大降低了算法运行的空间复杂度。进行的实验验证了此算法的效率及可行性,此算法能够在保证挖掘结果有用性的前提下,高效进行敏感规则的 k-匿名。

参考文献

[1] Sweeney L. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 2002,10(5):557-570

[2] 杨高明,杨静,张健沛. 聚类的(α,k)-匿名数据发布. 电子学报,2011, 39(8):1941-1946

[3] Zhang J, Yang J, Zhang J, et al. KIDS: k-anonymization data stream base on sliding window. In: Proceedings of the 2th International Conference on Future Computer and Communication, Wuhan, China, 2010, vol.2. 311-316

[4] Zhu Z, Du W. K-anonymous association rule hiding. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, New York, USA, 2010. 305-309

[5] Zhou B, Han Y, Pei J, et al. Continuous privacy preserving publishing of data streams. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, New York, USA, 2009. 648-659

[6] Wu Y H, Chiang C M, Chen A L P. Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering*, 2007, 19(1): 29-42

[7] Verykios V S, Gkoulalas-Divanis A. A survey of association rule hiding methods for privacy. In: Privacy-Preserving Data Mining: Model and Algorithms. Chicago: Springer, 2008. 267-289

[8] Wang S L, Parikh B, Jafari A. Hiding informative association rule sets. *Expert Systems with Applications*, 2007, 33(2): 316-323

[9] 李国徽,陈辉. 挖掘数据流任意滑动时间窗口内频繁模式. 软件学报,2008,19(10): 2585-2596

[10] Li H F, Lee S Y. Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems with Applications*, 2009, 36(2): 1466-1477

A k-anonymity preservation algorithm for sensitive rules in data stream

Zhang Junwei, Yang Jing, Zhang Jianpei, Zhang Lejun

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

Abstract

To achieve the two-fold protection for sensitive rules by lowering attackers' probability of finding hidden rules to less than $1/k$ through reducing the thresholds of the minimum support threshold (MST) and the minimum confidence threshold (MCT), the paper proposes a data stream oriented k-anonymity preservation algorithm for sensitive rules. The algorithm uses the time sliding window technique to get the latest n-moment data which most interest data users, and then, completes k-anonymity of sensitive rules by adding transactions data instead of modifying items of transactions, so it can avoid the twice access to data stream and the leaking of hidden sensitive rules. At the same time, the algorithm adopts prime number coding to further improve the efficiency and reduce the space complexity. The experimental results show that the proposed algorithm can achieve k-anonymous sensitive rules hiding efficiently in data stream and keep the higher usefulness of data mining results.

Key words: k-anonymous, data stream, association rule, sensitive rule