doi:10.3969/j.issn.1000-436x.2014.11.022

基于时间密度的数据流匿名方法

谢静，张健沛，杨静，张冰
(哈尔滨工程大学 计算机科学与技术学院，黑龙江 哈尔滨 150001)

摘 要：针对数据流中的匿名问题，提出一种基于时间密度的数据流匿名算法，考虑数据流的强时态性，提出时间权重和时间密度概念，当已发布簇的个数达到上限时，删除时间密度最小的簇，以此来保证已发布簇的可重用性。此外，为了保持较高的执行效率，算法对数据采用单遍扫描，以实现数据流的高效匿名。在真实数据集上的实验结果表明，提出的方法能保持较高的效率和较好的数据效用。

关键词：隐私保护；数据流；匿名；时态性
中图分类号：TP390.2 **文献标识码**：A **文章编号**：1000-436X(2014)11-0191-08

Anonymization algorithm based on time density for data stream

XIE Jing, ZHANG Jian-pei, YANG Jing, ZHANG Bing
(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001,China)

Abstract: Aim to address the problem of anonymization on data streams, an anonymization algorithm based on time density for data stream was proposed. Time weight and time density were designed for describing the data stream's temporal, when the published clusters reach the threshold, it will delete the minimum time density cluster to ensure the availability of published clusters. Furthermore, in order to maintain the higher efficiency, the algorithm scans the data only once to satisfy the anonymization requirements for speeding up. The experimental results on the real dataset show that the algorithm is efficient and effective meanwhile the quality of the output data.

Key words: privacy preserving; data stream; anonymization; temporalitaet

1 引言

随着信息技术的发展，数据流在应用环境中屡见不鲜，如远程通信、购物篮分析、网络监控以及传感器网络等。对这些连续的数据流进行挖掘[1~3]可以帮助企业了解客户的行为而带来独特的商业机会，然而大部分的企业并不拥有内部的数据挖掘部门或技术人员，所以企业需要把挖掘任务外包给第三方的专业机构。但是，数据流中可能包含许多需要保护的个人信息，因此，在企业向第三方机构提供数据前需要对数据进行处理，以保护敏感信息。

如何在保证数据效用的同时保护个体的敏感信息不泄露是当前隐私保护领域主要的研究问题。数据中每条记录对应某个个体，包含多个属性，属性可以分为三类：1) 显式标识符属性(identifier)，是指能唯一确定出个体身份的属性，如身份证号；2) 准标识符属性(QI, quasi-identifier)，通过该类属性组合起来可以确定个体身份，如年龄、性别和邮编等；3) 敏感属性(SA, sensitive attribute)，是指包含个体隐私信息的属性，如疾病、薪资等。

目前，对于静态数据上的隐私保护问题已经有许多研究成果，但是对于数据流环境下的研究却较少。由于数据流具有无限、快速到达和变化迅速的特点[4]，因此静态数据上的匿名方法不能直接适用于数据流上。针对数据流中的隐私保护问题，提出一种基于时间密度的数据流匿名方法，首先，采用

收稿日期：2014-08-15；**修回日期**：2014-10-22
基金项目：国家自然科学基金资助项目（61073041, 61073043, 61370083, 61402126）；教育部高等学校博士学科点专项科研基金资助项目（20112304110011, 20122304110012）
Foundation Items: The National Natural Science Foundation of China (61073041, 61073043, 61370083, 61402126); The National Research Foundation for the Doctoral Program of Higher Education of China (20112304110011, 20122304110012)

k-中心点思想对元组进行聚类，对于信息损失满足要求的簇输出，然后，提出时间权重和时间密度概念度量簇的时间特性，最后，对于不满足要求的簇则为其中的元组寻找新的簇或将元组抑制输出。

2 相关工作

目前，大部分已有的隐私保护技术都是针对静态数据集的，而在现实生活中的数据大部分是数据流形式的，也正是由于这种数据会随着时间的推进而不断地发生变化，将静态数据集的隐私保护技术直接应用在数据流上不能很好地保护隐私信息，这也就激发了对面向数据流的隐私保护技术的研究。静态数据上的隐私保护技术不能直接用于数据流的原因包括：1) 该技术假设数据集中的每条记录对应于不同的个体，也就是说，每个个体在数据集中只出现一次，虽然这种假设在静态数据中是合理的，但是在动态数据中是不实际的；2) 数据流中存在时间维度，数据以某一速率到达，动态处理后在某一时间约束内输出，所以必须保证数据在最大的延迟时间内输出。

数据流的匿名技术主要是基于扰动和聚类的方法。基于扰动的方法[5]是向数据流中加入噪声以实现数据流的匿名操作。基于聚类思想的方法[6~10]是将数据流中的元组按照某些原则进行聚类，发布满足匿名要求的簇。Cao 等[6]首次提出了面向数据流聚类的隐私保护算法 CASTLE。该算法在 *k*-匿名[11]原则的基础上，通过维护候选 *k*-匿名簇和已发布的 *k*-匿名簇进行聚类过程。对于新到达的元组，为其寻找合适的簇或创建一个新簇，当有元组到达发布时延时，则将元组泛化输出。Wang 等[7]提出的 B-CASTLE 针对 CASTLE 中没有考虑数据分布的局限，对候选 *k*-匿名簇中的元组个数加以限制，避免了簇的分割过程，减少了执行时间。CASTLE 和 B-CASTLE 中都需要维护候选 *k*-匿名簇和已发布的 *k*-匿名簇。Hessam 等[8]提出的 FAANST 中为了能够提高数据流匿名的处理速度，只维护已发布的 *k*-匿名簇，并且对数据进行批量处理，当数据缓存到发布时延时才一次性处理所有缓存的数据。FADS[9]在 FAANST 的基础上限制维护的已发布 *k*-匿名簇的个数，减少了查找簇的时间消耗，降低了执行时间。Guo 等[10]将 FADS 扩展到 *l*-多样性[12]上，实现了满足 *l*-多样性原则的数据流匿名方法。

3 基本概念

数据流是以一定速率连续到达的数据项序列 $a_1,\cdots,a_i,\cdots,a_n,\cdots$，该数据项序列只可以按照下标 i 的递增顺序读取一次[13]。研究数据流中的每个数据项对应某个个体，数据项中包含个体的多个属性信息，可以形式化表述为 $\{t,id,QI_1,\cdots,QI_n,A_1,\cdots,A_m\}$，$t$ 表示元组到达的时刻，id 是个体的显式标识符属性，$QI_i(1\leqslant i\leqslant n)$ 为准标识符属性，$A_j(1\leqslant j\leqslant m)$ 为剩余属性。

3.1 基于数据流的 *k*-匿名原则

文献[6]中提出了数据流上的 *k*-匿名方法，该方法通过对数据流中元组进行处理，输出满足 *k*-匿名要求的数据流 S_{out}。

定义 1 等价类。给定数据流 S，G 为 S 中某些元组的集合，在输出数据流 S_{out} 中，如果 G 中元组在准标识符属性 $QI_i(1\leqslant i\leqslant n)$ 上具有相同的取值，则称 G 为数据流 Sr' 上的等价类。

定义 2 基于数据流的 *k*-匿名原则。令待发布数据流 $S=\{t,id,QI_1,\cdots,QI_n,A_1,\cdots,A_m\}$，$S_{\text{out}}$ 为 S 匿名后的数据流，其中 t 和 id 已经删除，称 S_{out} 满足数据流的 *k*-匿名原则，当且仅当 S_{out} 满足下列条件。

1) 对于数据流 S 中的任一元组 r，在 S_{out} 中都存在相应的元组 r'。

2) 对于数据流 S_{out} 中的任一元组，G 为 r' 所在的等价类，G 满足 $N(G)\geqslant k$。其中，$G=\{\overline{r}'\in S_{\text{out}}\mid \overline{r}'q_i=r'q_i,1\leqslant i\leqslant n\}$，$r'q_i$ 表示元组 r' 在属性 QI_i 上的取值，$N(G)$ 表示 G 中不同个体的数目。

由于数据流具有强时态性，所以对于数据流中的元组应设置输出时间约束，使输出的数据流能尽量保证其时间特性。

定义 3 时间约束。令 Z 是数据流上的 *k*-匿名模型，输入数据流 S，δ 是正整数。若对于当前时刻 t_{now}，数据流中所有到达时刻小于 $t_{\text{now}}-\delta$ 的元组已经通过模型 Z 输出，则称模型 Z 满足时间约束。

3.2 信息损失度量

泛化和抑制[14]2 种技术常用于数据的匿名。泛化是将元组具体的 QI 值替换为更加模糊的值，如将年龄 25 替换为区间[25,30]。抑制是一种特殊的泛化，是指用最模糊的值来代替元组具体的 QI 值，如将年龄替换为区间 $(-\infty,+\infty)$。数据泛化后的信息损失度量用以衡量数据的效用，也能直接反映出隐私模型的优劣。将 QI 属性的初始值与匿名后的值

之间的距离作为信息损失度量，采用一般损失度量 GLM[15](generalized loss metric)来衡量信息损失。GLM 的度量方法如下。

定义 4　元组泛化信息损失。令 $r \in S$，S_{out} 中元组 r 泛化为 $r'(\tilde{q}_1,\cdots,\tilde{q}_n)$，则元组 r 泛化为 r' 的信息损失为

$$IL(r') = \frac{1}{n}\sum_{i=1}^{n} IL(\tilde{q}_i) \tag{1}$$

$$IL(\tilde{q}_i) = \begin{cases} \dfrac{U_i - L_i}{U - L}, & \tilde{q}_i = [L_i, U_i] \\ \dfrac{|M_i| - 1}{|M| - 1}, & \tilde{q}_i = M_i \end{cases} \tag{2}$$

式(1)表示元组的信息损失，式(2)计算每维 QI 属性上的信息损失。当 QI 属性是数值型属性时，信息损失为该 QI 属性泛化区间 $[L_i,U_i]$ 与值域区间 $[L,U]$ 的比值。当 QI 属性是分类型属性时，泛化的信息损失为节点 M_i 所覆盖的叶子节点个数 $|M_i|$ 与层次树总叶子节点个数 $|M|$ 的比值，其中 M_i 是泛化值 $\tilde{q}_i$ 在层次树中所对应的节点。

由于抑制是将具体的 QI 值替换为最模糊的值，因此，由定义 4 计算可知，抑制操作带来的信息损失为 1。

定义 5　数据流平均泛化信息损失[9]。令 $r_i \in S$，则数据流 S 到 t_p 时刻为止的平均泛化信息损失为

$$AVGIL(S,t_p) = \frac{1}{t_p}\sum_{r_i \in S, r_i t \le t_p} IL(r_i)$$

其中，$r_i t$ 表示元组 r_i 到达的时刻。

4　基于时间密度的数据流匿名方法

在数据流匿名过程中将采用聚类的思想对元组进行划分，因此把元组看作聚类空间中的点，元组间的距离作为聚类过程中的距离度量标准。

定义 6　元组间的距离。令 $r_1, r_2 \in S$，$(\tilde{q}_1,\cdots,\tilde{q}_n)$ 是 r_1 和 r_2 泛化后的 QI 值，则元组 r_1 和 r_2 之间的距离为

$$D(r_1,r_2) = \frac{1}{n}\sum_{i=1}^{n} IL(\tilde{q}_i)$$

元组间的距离可以理解为 2 个元组泛化后所带来的信息损失。2 个元组泛化之后的信息损失越小，则说明元组间的 QI 属性越相似，2 个元组间的距离也就越近。

定义 7　簇的信息损失。令簇 C 中 QI 属性的泛化结果是 $(\tilde{q}_1,\cdots,\tilde{q}_n)$，则簇 C 的信息损失定义为

$$IL(C) = \frac{1}{n}\sum_{i=1}^{n} IL(\tilde{q}_i)$$

定义 8　簇的信息损失增量。令簇 C 中 QI 属性的泛化结果是 $(\tilde{q}_1,\cdots,\tilde{q}_n)$，当元组 r 加入到簇 C 中时，簇 C 的泛化结果变为 $(\tilde{q}_1',\cdots,\tilde{q}_n')$，则簇 C 加入元组 r 后的信息损失增量定义为

$$\Delta IL(C,r) = \frac{1}{n}\sum_{i=1}^{n}(IL(\tilde{q}_i') - IL(\tilde{q}_i))$$

4.1　基于聚类的数据流匿名方法分析

对于数据流中的某个簇，若该簇中元组个数不小于 k，则称该簇满足 k-匿名；若该簇中元组个数小于 k，则称该簇不满足 k-匿名。将所有满足 k-匿名的簇组成的集合记为 SC(satisfying clusters)；所有不满足 k-匿名的簇组成的集合记为 USC (unsatisfying clusters)。

CASTLE 算法是 Cao 等提出的基于聚类的数据流匿名方法。该算法中对于到达的新元组采用 2 种处理方式：加入 USC 的某个簇中或者创建一个包含该元组的新簇，并且将新簇加入 USC 集合中。CASTLE 中为了给 USC 中未满足 k-匿名要求的簇更多的机会来接收新元组，新到达的元组只能选取 USC 中的簇，不能选取 SC 中的簇，这样可能会带来较大的信息损失。

例 1　假设数据表的 QI 属性为{年龄，学历}，年龄是数值型属性，学历是分类型属性，学历的分类层次树如图 1 所示。图 2(a)中的 C_1 和 C_2 表示 2 个簇，假设 C_1 为满足 k-匿名的簇，C_2 为未满足 k-匿名的簇。由图 2(a)可以看出，C_1 的泛化区间为([27,30]，高等学校)，C_2 的泛化区间为([20,24]，中等学校)，在这里假设年龄的值域是[18,120]。当新元组 r(25，学士)到达时，若将 r 加入簇 C_1 中，C_1 中的泛化区间扩大为([25,30]，高等学校)，由定义 8 计算可得，簇 C_1 的信息损失增量为

$$\Delta IL(C_1,r) = \frac{1}{2}\times\left[\left(\frac{5}{102}+\frac{2}{4}\right)-\left(\frac{3}{102}+\frac{2}{4}\right)\right] = 0.01$$

若将 r 加入簇 C_2 中，C_2 中的泛化区间扩大为([20,25]，所有)，簇 C_2 的信息损失增量为

$$\Delta IL(C_2,r) = \frac{1}{2}\times\left[\left(\frac{5}{102}+\frac{4}{4}\right)-\left(\frac{4}{102}+\frac{1}{4}\right)\right] = 0.38$$

按照计算结果可得，新元组 r 应加入簇 C_1 中，

然而 CASTLE 算法中优先让新元组加入未满足 *k*-匿名的簇，即簇 C_2。由此可见，CASTLE 算法在为新元组选择合适簇的过程中，破坏了元组间的聚类关系，并且产生较大的信息损失。

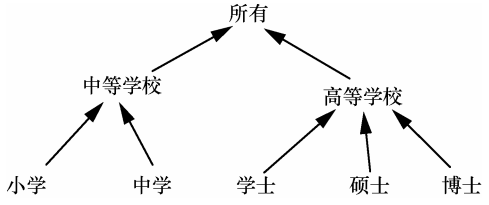


图 1　属性学历的层次树

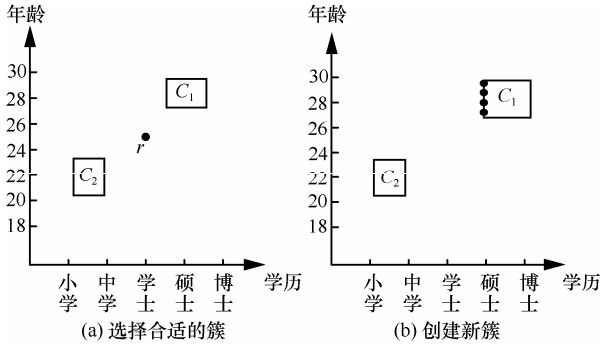


图 2　簇的选择

FADS 算法针对例 1 中 CASTLE 算法存在的不足做出了相应的改进，对于新到达的元组首先在 SC 集合中查找是否存在覆盖新元组的簇。若存在，则将新元组加入到该簇中，否则，采用 KNN 方法将元组划分成簇。虽然此操作使可以加入 SC 集合的元组无须进行聚类过程，然而，FADS 忽略了新到达元组之间的紧凑关系。

例 2　如图 2(b)所示，黑色圆点表示新到达元组，由图 2(b)可以直观看出，新到达的元组均被簇 C_1 覆盖。假设 k=4，可以发现新到达的 4 个元组可以自身组成 4-匿名的簇，该簇的泛化区间为([27,30], 学士)，泛化信息损失是 0.03，而将 4 个新元组加入簇 C_1 的泛化信息损失是 0.53。显而易见，按照 FADS 算法将新元组加入到已发布的簇 C_1 中带来的信息损失更大(信息损失为 0.53)，应为新到达的元组创建一个新簇，这样带来的信息损失更小(信息损失为 0.03)。

通过例 1 和例 2 的讨论可以发现，在聚类过程中如果强制设定元组加入 SC 或 USC 集合中都将破坏元组之间聚类的关系，并且带来较大的信息损失。采用基于时间密度的数据流匿名方法，根据簇的信息损失来控制数据流的匿名过程。此外，针对数据流的强时态性，在删除簇的过程中考虑簇的时间特性。

4.2　算法的框架及实现

定义 9　元组的时间权重。令 $r\in S$，其到达时刻为 t_0，则元组 r 在时刻 t 的时间权重为 $\omega(r,t)=2^{-(t-t_0)}$，$t>t_0$。

由定义 9 可知，元组到达时刻越早，其时间权重越小，反之，时间权重越大。新元组到达的初始时间权重为 1，然后随着时间的推移呈指数衰减。之所以采用指数衰减是为了能够更好地体现出新到达元组的重要性，此外，在许多应用中，衰减的过程与自然降温的过程近似，而自然降温的过程就类似于指数衰减。

定义 10　簇的时间密度。设 C 为数据流中的簇，簇 C 中所有元组在时刻 t 的时间权重之和的均值称为簇 C 的时间密度，记为 $D(C,t)=\dfrac{\sum_{r\in C}\omega(r,t)}{|C|}$，其中，$|C|$表示簇 C 中元组的个数。

显然，若一个簇中不断有新元组到达，则该簇的时间密度将逐步增大，否则，将逐步减小。CASTLE 算法中当 SC 中簇的个数大于上限时，则删除最早加入的簇以保持簇的个数不超过设定的上限值。然而，簇的时间特性是随着时间不断变化的，最早加入的簇中也可能会加入新到达的元组，因此，需要考虑簇的时间特性，也就是时间密度。簇的时间密度可以反映簇中是否存在较新的元组。簇的时间密度越小，说明该簇中较长时间内没有元组到达，对于这类簇可以称之为离群点或者噪音，即在数据流中这类数据是非常少见的。新元组加入该类簇中的概率很小，它们对于整个数据流上的聚类过程贡献不大，因此，将此类簇删除可以为之后接收新簇提供空间。

命题 1　给定簇 C，其平均信息损失为 $IL(C)$，将元组 r 加入簇 C 后的信息损失为 $IL'(C)$，若 $IL'(C)-IL(C)\leqslant\dfrac{1-IL(C)}{|C|+1}$，则将元组 r 加入簇 C 中所产生总的信息损失比将元组 r 抑制所产生的信息损失小。

证明　由已知条件可得

$$IL'(C)-IL(C)\leqslant\frac{1-IL(C)}{|C|+1}$$

$$|C|\,IL'(C)+IL'(C)-|C|\,IL(C)\leqslant 1$$

$$\underbrace{|C|(IL'(C)-IL(C))}_{(3)}+\underbrace{IL'(C)}_{(4)}\leqslant 1 \quad (5)$$

式(3)表示将元组 r 加入簇 C 之后，簇 C 中的$|C|$个元组因泛化范围的变化所增加的信息损失，式(4)表示将元组 r 加入簇 C 之后，元组 r 泛化所带来的信息损失。不等式右边的 1 可以看作是将元组 r 抑制所带来的信息损失。由式(5)可知，将元组 r 加入簇 C 之后，所带来的信息损失之和不大于将元组 r 抑制所带来的信息损失，也就是说，将元组 r 加入簇 C 中所产生的信息损失比将元组 r 抑制所产生的信息损失小，证毕。

针对 4.1 节的分析可以总结出数据流上的匿名算法需要满足的原则：

1) 为了尽可能地减小信息损失，对于新到达的元组，要选择合适的簇加入其中；

2) 针对数据流的时态性，算法应考虑元组和簇的时间特性；

3) 针对数据流快速到达的特点，算法的时间复杂度应为 $O(S)$。

对于原则 1)，利用簇的信息损失作为度量标准为元组选择合适的簇，以尽可能减少信息损失。对于原则 2)，给出元组的时间权重和簇的时间密度概念来反映时态性。对于原则 3)，为保证算法的时间复杂在线性时间内，应减少扫描整个数据流 S 的次数。

根据上述原则，设计一种基于时间密度的数据流匿名算法 TDAADS (time density based anonymization algorithm for data stream)，算法的基本框架如下：

1) 将新到达的元组保存至缓冲区 H 中；

2) 当缓冲区中存在元组达到发布的时间约束时，将 H 中的元组划分为大小为 k 的簇，将信息损失小于 SC 中平均信息损失的簇加入 SC 中，并将簇泛化输出，若 SC 中簇的个数到达上限(采用文献[9]中设置的上限 $c_0\delta/K$ 个)，则删除时间密度最低的簇；

3) 对于 H 中的剩余元组 r，从 SC 中查找覆盖元组 r 并且信息损失最小的簇，并以该簇的泛化形式发布元组；

4) 对于 H 中无法覆盖的元组，在 SC 中查找是否存在满足命题 1 的簇，若存在，将该元组加入该簇中，否则，抑制该元组；

5) 重复步骤 2)~4)直至没有新元组到达。

步骤 3)中是对 SC 中簇的重用，SC 中保存的是已发布簇，通过对这些簇的保存，新元组从中找出覆盖自身的簇。可见，在重用过程中，簇的时间密度越低则其可重用性越低，删除该类簇可提高 SC 簇中的重用概率。

算法 1　TDAADS(S,k, δ)

输入：数据流 S，阈值 k, δ；

输出：满足 k-匿名的泛化簇。

1)　$SC=\varnothing$；

2) while　$S\neq\varnothing$　do

3)　　令 r 为 S 中下一个元组，将 r 加入缓冲区 H 中；

4)　　if H 中存在元组到达发布时延　then

5)　　*choose_cluster*();

6)　　if　$H\neq\varnothing$　then

7)　　*Deal_tuple*();

8)　　end if

9)　　end if

10) end while

11) if　$H\neq\varnothing$　then

12)　　*Deal_tuple*();

13) end if

算法 2　*choose_cluster*()

1) 采用 PAM 算法将缓冲区 H 中的元组划分为 $\frac{\delta}{k}$ 个簇，并将簇加入集合 KC 中；

2) 对于 KC 中未满足 k-匿名的簇，将相邻的簇进行合并，使簇满足 k-匿名的要求；

3) 从 H 中将 KC 里包含的元组删除；

4) for KC 中的每个簇 C_i

5)　　if　$IL(C_i)\leqslant IL_{avg}(SC)$　then

　　// $IL_{avg}(SC)$ 表示 SC 中簇的信息损失均值

6)　　while　$|SC|\geqslant\frac{c_0\delta}{k}$　do

7)　　删除 SC 中时间密度最小的簇；

8)　　end while

9)　　将 C_i 泛化输出，并且加入 SC 中；

10)　　else

11)　　将 C_i 中的元组加入 H 中；

12)　　end if

13) end for

算法 3　*Deal_tuple*()

1) for each　$r_i\in H$

2)　　在 SC 中查找覆盖 r_i 并且信息损失最小的簇放入集合 $SC_{\min}$ 中；

3)　　if　$SC_{\min}\neq\varnothing$　then

4) 在 $SC_{\min}$ 中选取时间密度最大的簇 $C_{\max}$，以簇 $C_{\max}$ 的泛化形式输出元组 r_i；

5) else

6) for each $C_j \in SC$

7) if $IL'(C_j) - IL(C_j) \leqslant \dfrac{1 - IL(C_j)}{|C_j|+1}$ then

// $IL(C_j)$ 表示簇 C_j 的信息损失，$IL'(C_j)$ 表示加入元组 r_i 后簇 C_j 的信息损失

8) 将 r_i 以簇 C_j 的泛化形式输出；

9) else

10) 将 r_i 抑制输出；

11) end if

12) end for

13) end if

14) $H = H - \{r_i\}$；

15) end for

通过时间复杂度分析可知，*choose_cluster*()的时间复杂度为 $O(\delta^2 |QI|)$，*Deal_tuple*()的时间复杂度 $O\left(\dfrac{\delta^2 |QI|}{k}\right)$。TDAADS 算法的总时间复杂度为 $O\left(\dfrac{\delta |QI| |S|}{k}\right)$，由于 $|S| \gg \delta, |QI|, k$，因此，算法总的时间复杂度可以简化为 $O(|S|)$。

5 实验结果与分析

5.1 数据集

通过实验分析 TDAADS 的性能，并将其与文献[6]提出的 CASTLE 和文献[9]提出的 FADS 进行比较。实验所采用的数据集为隐私保护研究中广泛使用的 UCI 数据库中的 Adult 数据集，该数据集总共包含 45 222 条记录，删除包含缺失值的记录之后剩余 30 162 条记录，选取数据集中的 10 个属性进行实验，分别为{age, final-weight, education-number, capital-gain, capital-loss, hours-per-week, education, marital-status, occupation, native-country}，其中前 6 个为数值型属性，后 4 个为分类型属性。

5.2 信息损失分析

图 3~图 6 给出了 QI 维数、k 值、δ 值和数据集大小对 TDAADS、CASTLE 和 FADS 平均信息损失的影响。由图 3 可知，随着 QI 维数的增加，3 种算法的平均信息损失都将增大，这是由于 QI 维数增加使在泛化过程中需要泛化的属性维数增加，因此泛化的信息损失将增加。此外，在同等条件下，CASTLE 的平均信息损失比 FADS 和 TDAADS 的大，这是由于 CASTLE 对于新到达的元组优先选取 USC 中的簇，而 FADS 和 TDAADS 算法更好地利用了 SC 中的簇。因此，CASTLE 的平均信息损失较大。因为 FADS 中采用的 *K*-近邻思想对初始元组的选取非常敏感，而且，FADS 对于无法加入到簇中的元组直接抑制(抑制的信息损失为 1)，将会带来较大的信息损失。因此，FADS 的信息损失比 TDAADS 的大。

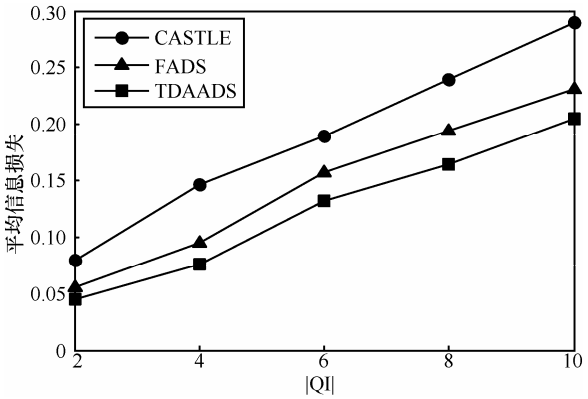


图 3 QI 维数对平均信息损失的影响

由图 4 可知，随着 k 值的增加，3 种算法的平均信息损失都将增大，这是由于 k 值增加，使已发布簇中元组个数增加，在泛化过程中的信息损失也必然增加。由图 5 可知，随着 δ 值的增加，3 种算法的平均信息损失都将减小，这是由于 δ 增大，时间约束变得宽松，使元组有更大的机会加入已发布的簇中或生成新簇，所以带来的信息损失将减小。由图 6 可知，随着数据量的增加，3 种算法的平均信息损失都将减小，这是因为数据量的增大，使已发布的簇逐渐增多，新到达的元组加入已发布簇中概率更大，从而减小了信息损失。

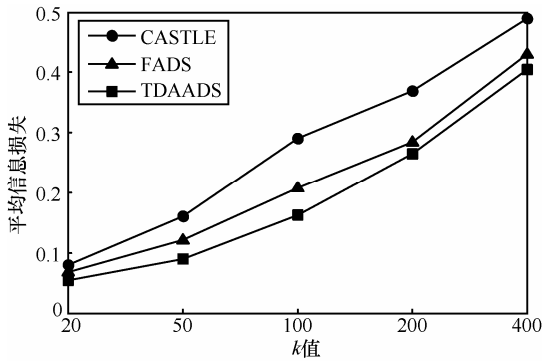


图 4 k 值对平均信息损失的影响

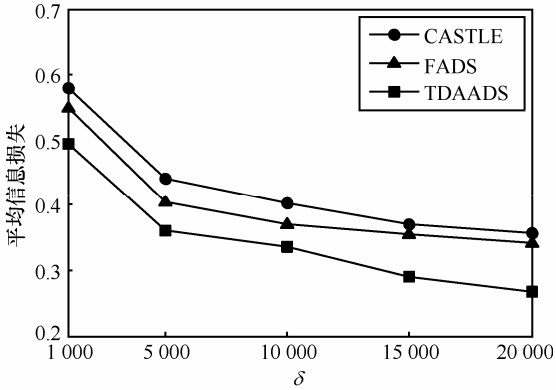


图 5　δ 对平均信息损失的影响

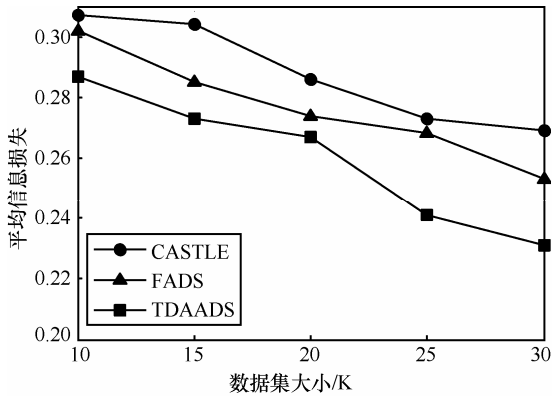


图 6　数据集大小对平均信息损失的影响

5.3　执行时间分析

图 7~图 10 给出了 QI 维数、k 值、δ 值和数据集大小对 3 种算法执行时间的影响。由图 7 可知，当 QI 维数增加时，3 种算法的执行时间将增加。这是因为 QI 维数增加，泛化过程中要处理的属性也将增多，执行时间必然增加。此外，由图 7 可以看出，随着 QI 维数的增加，CASTLE 执行时间的增长速度比 TDAADS 和 FADS 的要快，这主要是由于 CASTLE 没有限制已发布 k-匿名簇的数量，随着已发布 k-匿名簇数量的不断增加，当 QI 维数增大时，执行时间也会显著增加。而 TDAADS 和 FADS 中限制了已发布 k-匿名簇的数量，即使 QI 维数增大，也不会增加过多的执行时间。由于 TDAADS 为了减少信息损失，对于无法加入到已有簇中的剩余元组，不是采用 FADS 中直接将这些元组抑制的处理方式，而是通过判断这些元组加入到某个簇中是否会带来比抑制更小的信息损失，以此为这些剩余元组选取合适的簇。此种处理方式比 FADS 中的抑制操作需要更多的执行时间，因此 TDAADS 的执行时间更长。

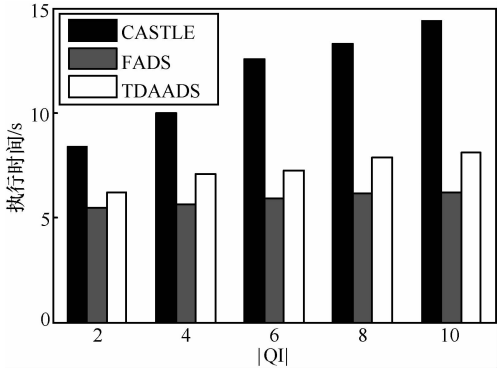


图 7　QI 维数对执行时间的影响

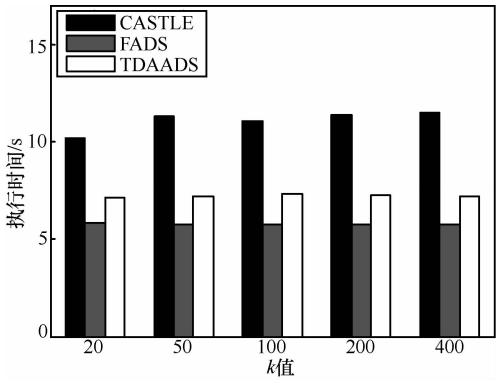


图 8　k 值对执行时间的影响

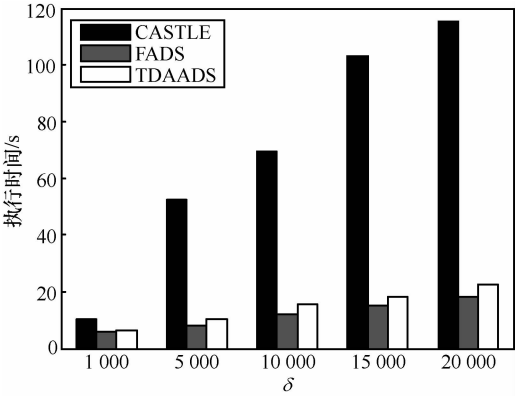


图 9　δ 对执行时间的影响

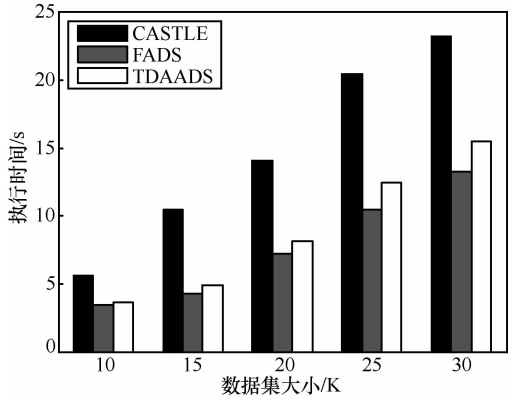


图 10　数据集大小对执行时间的影响

由图 8 可知，当 QI 维数增加时，3 种算法的执行时间产生了一些细微的波动，这是由于$|S| \gg k$，k 值的变化对总执行时间的影响很细微。与图 7 类似，CASTLE 的执行时间最长，FADS 的执行时间最短。由图 9 可知，当 δ 值增加时，3 种算法的执行时间将增大。这是由于随着 δ 值的增加，已发布的 k-匿名簇也将增多，元组进行查找时的执行时间也将增大。此外，由图 9 可以看出，CASTLE 的执行时间增长非常迅速，由文献[6]的算法时间复杂度分析可知，CASTLE 的时间复杂度与 δ^2 成正比，因此随着 δ 值的增加，执行时间增长速度较快。而 TDAADS 和 FADS 的时间复杂度与 δ 值成正比，呈线性增长，因此增长速度较慢。由图 10 可知，当数据量增加时，3 种算法的执行时间将增加，这是由于随着数据量的增加，算法需要处理的元组数增多，因此执行时间都将增大。

综上所述，在同等条件下，虽然 TDAADS 执行时间比 FADS 的稍长，但是 TDAADS 能达到较小的信息损失。因此，所提出 TDAADS 在牺牲少量时间的前提下，减小了信息损失，保证了数据的效用。

6　结束语

针对数据流中的隐私保护问题进行研究，首先，介绍了数据流匿名的基本概念和信息损失的度量；然后，提出时间权重和时间密度概念，设计一种基于时间密度的数据流匿名方法，并给出了算法实现及时间复杂度分析；最后，通过实验结果表明，提出的方法在保证较高执行效率的同时保持较好的数据效用。

参考文献：

[1] DOMINGOS P, HULTEN G. Mining high-speed data streams[A]. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York, 2000.71-80.

[2] ZHANG P, ZHU X, SHI Y. Categorizing and mining concept drifting data streams[A]. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York, 2008.812-820.

[3] LUO C, THAKKAR H, WANG H, *et al*. A native extension of SQL for mining data streams[A]. Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data[C]. New York, 2005.873-875.

[4] AGGARWAL C C, HAN J, WANG J, *et al*. A framework for clustering evolving data streams[A]. Proceedings of the 29th International Conference on Very Large Data Bases-Volume 29[C]. 2003.81-92.

[5] LI F F, SUN JM, PAPADIMITRIOU S, *et al*. Hiding in the crowd: privacy preservation on evolving streams through correlation tracking[A]. Proc of the ICDE 2007[C]. 2007. 686-695.

[6] CAO J, CARMINATI B, FERRARI E, *et al*. Castle: continuously anonymizing data streams[J]. Dependable and Secure Computing, IEEE Transactions on, 2011, 8(3): 337-352.

[7] WANG P, LU J J, ZHAO L, *et al*. B-CASTLE: an efficient publishing algorithm for k-anonymizing data streams[A]. Proc of 2010 the 2nd WRI Global Congress on Intelligent Systems[C]. 2010. 132-136.

[8] ZAKERZADEH H, OSBORN S L. FAANST: fast anonymizing algorithm for numerical streaming data[A]. Proc of the 5th Int'l Workshop on Data Privacy Management and 3rd Int'l Conf on Autonomous Spontaneous Security[C]. Springer-Verlag, 2011. 36-50.

[9] 郭昆, 张岐山. 基于聚类的快速数据流匿名方法[J].软件学报, 2013, 24(8):1852-1867.
GUO K, ZHANG Q S. Fast clustering-based anonymization algorithm for data streams[J]. Journal of Software, 2013, 24(8):1852-1867.

[10] GUO K, ZHANG Q S. Fast clustering-based anonymization approaches with time constraints for data streams[J]. Knowledge-Based Systems, 2013, 46: 95-108.

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

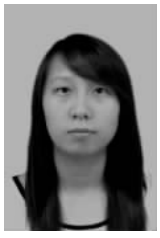
[12] MACHANAVAJJHALA A, KIFER D, GEHRKE J, *et al*. L-diversity: privacy beyond *k*-anonymity[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2007, 1(1):1-52.

[13] RAGHAVAN M R H P. Computing on data streams[A].External Memory Algorithms: DIMACS Workshop External Memory and Visualization[C]. 1999.107.

[14] XU Y, MA T, TANG M, *et al*. A survey of privacy preserving data publishing using generalization and suppression[J]. Appl Math, 2014, 8(3): 1103-1116.

[15] YENGAR V S. Transforming data to satisfy privacy constraints[A]. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. 2002.279-288.

作者简介：



谢静（1986-），女，湖北随州人，哈尔滨工程大学博士生，主要研究方向为数据挖掘、隐私保护。

张健沛（1956-），男，黑龙江哈尔滨人，哈尔滨工程大学教授、博士生导师，主要研究方向为数据挖掘、隐私保护、社会网络等。

杨静（1962-），女，黑龙江哈尔滨人，哈尔滨工程大学教授、博士生导师，主要研究方向为数据挖掘、隐私保护、机器学习等。

张冰（1986-），女，黑龙江哈尔滨人，哈尔滨工程大学博士生，主要研究方向为数据挖掘、隐私保护。