一种基于逆聚类的个性化隐私匿名方法

王 波[1,2],杨 静[1]
(1.哈尔滨工程大学计算机科学与技术学院,黑龙江哈尔滨 150001; 2.哈尔滨理工大学自动化学院,黑龙江哈尔滨 150080)

摘 要: 针对不同个体对隐私保护的不同需求,提出了一种面向个体的个性化扩展 *l*-多样性隐私匿名模型.该模型在传统 *l*-多样性的基础上,定义了扩展的 *l*-多样性原则,并通过设置敏感属性的保护属性来实现个体与敏感值之间关联关系的个性化保护需求.同时,还提出了一种个性化扩展 *l*-多样性逆聚类(PELI-clustering)算法来实现该隐私匿名模型.实验表明:该算法不仅能产生与传统基于聚类的 *l*-多样性算法近似的信息损失量以及更小的时间代价,同时也满足了个性化服务的需求,获得更有效的隐私保护.

关键词: 隐私匿名; 个性化; 逆聚类; *l*-多样性; 保护属性
中图分类号: TP309.2 **文献标识码:** A **文章编号:** 0372-2112 (2012) 05-0883-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.05.004

A Personalized Privacy Anonymous Method Based on Inverse Clustering

WANG Bo[1,2], YANG Jing[1]
(1. *College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang* 150001, *China*;
2. *School of Automation, Harbin University of Science and Technology, Harbin, Heilongjiang* 150080, *China*)

Abstract: For achieving the different privacy preservation requirements of each individual, this paper presents a personalized extension *l*-diversity privacy anonymous model orienting individuals. This model proposes an extension *l*-diversity principle based on the traditional *l*-diversity, and realizes the requirement of personalized protection of relationship between individual and sensitive value by setting up guarding attributes on sensitive attributes. In the meantime, this paper also proposes a personalized extension *l*-diversity inverse clustering algorithm (PELI-clustering) to implement the privacy anonymous model presented in this paper. The experiments show that the proposed algorithm in this paper not only meets the requirements of personalized service, but also produces similar information loss to the traditional clustering-based *l*-diversity algorithm with less time cost, which achieves more effective privacy preservation.

Key words: privacy anonymity; personalized; inverse clustering; *l*-diversity; guarding attribute

1 引言

数据发布过程中的隐私泄露问题一直是数据共享与信息安全领域的研究热点之一.对于一个纯粹的属性值并不构成敏感信息,只有当属性值与某一个体形成关联时才能使这种关联关系作为敏感信息而应得到保护,所以从攻击者的角度来看,要获取的对象是数据拥有者与敏感信息的关联关系.因此,保护或者破坏这种关联关系便成了数据发布过程所要研究的主要工作.

通常,待发布的数据集以二维表的形式存储,包含4类属性:(1)身份标识符属性(*ID*);(2)准标识符属性(*QI*);(3)敏感属性(*S*);(4)其他属性.最原始的隐私方法只是直接删除身份标识符后直接发布数据表,但这样做并不能有效的防止隐私的泄露,一种最直接的现象就是攻击者通过外部公开的数据信息,结合准标识符属性来推断出个体与敏感属性间的关联关系,这就是所谓的链接攻击[1],以 *k*-匿名[2]和 *l*-多样性[3]为主流模型的匿名技术是防止攻击者这种推理现象的有效方法之一,*k*-匿名的思想解决了链接攻击而产生的隐私泄露风险,而 *l*-多样性有效地防止了 *k*-匿名中隐含着同质攻击的缺陷,但面对复杂的背景知识攻击时,攻击者仍然可能以很高的概率推断出个体与敏感信息之间的关联关系.因此,研究者在 *k*-匿名和 *l*-多样性的基础上,又提出一些不同类型的匿名模型,例如基于聚类的匿名技术[4~7],

收稿日期:2011-06-15;修回日期:2011-10-12
基金项目:国家自然科学基金(No.61073041,No.61073043,No.61172167);黑龙江省自然科学基金(No.F200901);哈尔滨市科技创新人才研究优秀学科带头人专项基金(No.2011RFXXG015,No.2010RFXXG002)

其基本思想是:首先将待发布的数据集划分为若干簇,其中簇内记录相关,簇间记录不相关,然后将每个簇内记录的准标识符泛化为相同的属性值,生成等价类,从而实现数据集的匿名化.

然而,以上的数据匿名方法都没有考虑隐私信息个性化的需求问题,而在现实的生活中,不同个体对于同一隐私信息的敏感程度会有很大的差异性,导致不同的隐私保护需求.因此,隐私保护的个性化服务是非常有必要的.针对这一问题,Xiao 等[8]首次对个性化匿名模型进行了系统的描述,并提出了相应的解决方法.此后,研究者在这一方面作了大量的工作,面对特定环境下的个性化服务需求,提出了相应的解决方法[9~12].针对隐私保护中的个性化服务需求问题,本文在传统 l-多样性的基础上,定义了扩展的 l-多样性原则,并提出了一种面向个体的个性化扩展 l-多样性隐私匿名模型,该模型通过为每个个体指定不同的敏感属性泛化约束来实现隐私个性化的需求.同时,本文结合逆聚类的分类原则,给出了个性化扩展 l-多样性的逆聚类算法,在满足了匿名等价类扩展 l-多样性要求的同时,实现了敏感属性的个性化需求.

2 相关工作

k-匿名模型思想的实质是破坏个体与元组之间的关联关系,从而防止链接攻击带来的隐私泄露风险,但是它没有破坏个体与敏感值之间的关联关系,所以模型中隐含着同质攻击和背景知识攻击的隐私泄露缺陷,Machanavajjhala 等人在文献[3]中证明了攻击者可能以 100% 的概率获得某些个体的隐私信息,于是提出了 l-多样性模型,要求发布的等价类中不同敏感值的个数大于等于 $l(l\geqslant 2)$,等价类中敏感值的足够多样化部分地解决了 k-匿名的缺陷.

目前,在 k-匿名模型的基础上,提出了许多利用聚类来实现匿名化的算法,但这些算法大多没有考虑敏感属性值的多样性,文献[7]提出了基于聚类的敏感属性 l-多样性匿名化算法 LCA-FC 和 LCA-RC,要求在生成的每个等价聚类中至少有 l 个互异的敏感属性值,并且控制每个聚类的大小介于 l 和 $2l$ 之间,从而使聚类达到最优划分,提高了数据的安全性.

针对 k-匿名没有考虑隐私保护个性化需求的缺陷,文献[8]提出了一种个性化匿名模型,在模型中通过为不同的个体指定不同的敏感属性泛化约束来实现个性化匿名,由于模型是基于 k-匿名的,并没有考虑等价类中敏感属性值的多样性问题,故而仍然存在着同质攻击的风险.本文从等价组中敏感属性值要求 l-多样性的需求出发,在传统聚类的基础上,提出一种逆聚类的方法实现敏感属性值的多样性,同时利用对敏感属性值的泛化来满足隐私保护个性化的需求.

3 个性化隐私匿名模型

3.1 l-多样性模型

l-多样性模型[3]是 k-匿名的一种改进模型,其实质是保证发布数据表中每个等价类的敏感属性值足够多样化来解决 k-匿名所隐含的缺陷.

定义 1 ***QI*-等价类.** 给定数据表 T,T^* 为 T 泛化后的匿名数据表,T^* 中具有相同准标识符属性 QI 的记录的集合称为 T^* 的 QI-等价类.

定义 2 ***l*-多样性.** 给定数据表 T,T^* 为 T 泛化后的匿名数据表,令 SG 为 T^* 的一个 QI-等价类,如果 SG 中不同敏感属性值的个数大于等于 $l(l\geqslant 2)$,则称等价类 SG 是满足 l-多样性的;如果 T^* 的每个 QI-等价类都满足 l-多样性,则称表 T^* 是满足 l-多样性的.

如表 2 是表 1 原始医疗信息表满足 2-多样性的匿名发布表.表中准标识符属性为[Age]、[Sex]和[Zip code],属性列[Disease]为敏感属性,[Name]为个体标识符属性.

表 1 待发布的个人医疗原始信息记录表

Name	Age	Sex	Zip code	Disease
Fred	24	M	13000	HIV
Andy	25	M	13001	HIV
James	27	M	17000	Flu
Allen	29	M	17001	Cancer
Jacky	30	M	17002	Obesity
Emily	42	F	56000	Pneumonia
Judy	45	F	56010	Indigestion

表 2 2-多样性匿名表

No.	Age	Sex	Zip code	Disease
1	[20 25]	M	1300*	HIV
2	[20 25]	M	1300*	HIV
3	[26 30]	M	1700*	Flu
4	[26 30]	M	1700*	Cancer
5	[26 30]	M	1700*	Obesity
6	[40 45]	F	560**	Pneumonia
7	[40 45]	F	560**	Indigestion

3.2 个性化扩展 l-多样性模型

在本小节中,我们利用敏感属性的泛化约束,对传统 l-多样性原则进行扩展,并由不同个体指定不同的敏感属性泛化约束[8]来实现隐私保护的个性化匿名,最后给出一种满足扩展 l-多样性的个性化隐私匿名模型.

定义 3 继承分类树. 设属性 A 的有限值域为 $Dom(A)$,则属性 A 的继承分类树定义为四元组 $ITax(A)=\{r_A,L_A,IN_A,R_A\}$,其中,

(1) r_A 表示分类树的根节点;

(2) L_A 表示分类树叶子节点的集合,即 $L_A = Dom(A)$;

(3) IN_A 表示分类树中间节点的集合,集合中的元素代表了属性 A 中各取值继承分类的不同程度;

(4) R_A 表示分类树中各节点间继承关系.

在这里,继承分类树中结点的层次 level 从根结点开始依次增加,根结点为第一层,即 level = 1.图 1 给出了表 1 中敏感属性[Disease]的继承分类树.其中,叶子节点包含了属性列[Disease]中全部的取值,根节点代表所有疾病,从根节点到叶子节点的中间节点代表疾病的若干分类.

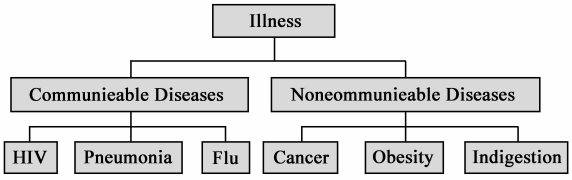


图1　敏感属性[Disease]的继承分类树

定义 4　继承分类子树.对于继承分类树 $ITax(A)$ 中的任意节点 N,称以该节点为根节点,包含该节点所有子孙节点的子树为 $ITax(A)$ 的继承分类子树,记为 $SUB\_ITax(N)$.根据树的性质,继承分类子树也可表示为一个四元组 $SUB\_ITax(N) = \{N, L_N, IN_N, R_N\}$.

结合敏感属性的继承分类树,可以对 l-多样性原则进行扩展.

定义 5　扩展 l-多样性.给定数据表 T 的 QI-等价类 SG,如果 SG 满足 $\sum_{t \in SG} |SUB\_ITax(t.S)| \geqslant l$,其中 $|SUB\_ITax(t.S)|$ 表示以 $t.S$ 为根节点的继承分类子树中包含叶子节点的个数,则称 SG 是满足扩展 l-多样性的;如果 T 的每个 QI-等价类都满足扩展 l-多样性,则称 T 是满足扩展 l-多样性的.

特别地,当 $t.S$ 为叶子节点时,$\sum_{t \in SG} |SUB\_ITax(t.S)|$ 的值就是等价类中包含不同敏感属性值的个数,也就是说,l-多样性是扩展 l-多样性的一个特例.

定义 6　保护属性.给定元组 $t \in T$,其敏感属性值为 $t.S$,则元组 t 的保护属性 $t.GA$ 定义为继承分类树 $ITax(AS)$ 从根节点 $r.AS$ 到 $t.S$ 所在叶子节点路径上任何一个节点的值.

表 3 是对表 1 中各敏感属性值根据图 1 的继承分类树,加入相应保护属性的个人医疗原始信息记录表.属性列[Guarding attribute]中各值即为属性列[Disease]中各敏感值的保护属性.

数据拥有者 o_t 通过指定保护属性 $t.GA$ 来表达对敏感属性值的个性化需求,保护属性 $t.GA$ 意味着 o_t 不希望数据攻击者得到以 $t.GA$ 为根节点的继承分类子树 $SUB\_ITax(t.GA)$ 中的任何叶子结点,即当攻击者推断出 $t.s$ 所在的继承分类子树 $SUB\_ITax(t.GA)$ 中任何叶子节点与数据拥有者的关联关系 $< o_t, A_i^S >$ 时,其中 A_i^S 为 $SUB\_ITax(t.GA)$ 的叶子节点,则数据拥有者就认为他/她的隐私受到了侵犯.显然地,指定的保护属性 $t.GA$ 在继承分类树中的层次越小,表明数据拥有者 o_t 的个性化隐私保护需求度越高,可以用隐私侵犯率作为发布数据表隐私泄露的一个评价指标,具体的定义描述如下:

定义 7　隐私侵犯率[8].给定元组 $t \in T$,其隐私侵犯率 $P_{breach}(t)$ 定义为攻击者从发布数据表 T^* 中推断出存在于原始数据表 T 中的关联关系 $< o_t, A_i^S >$ 的概率,其中 A_i^S 为 $SUB\_ITax(t.GA)$ 的叶子节点.

结合以上的概念与定义,下面给出完整的个性化扩展 l-多样性模型.

定义 8　个性化扩展 l-多样性.给定数据表 T,T^* 为 T 泛化后的匿名数据表,如果 T^* 满足扩展 l-多样性,并且 T^* 中各元组的隐私侵犯率 $P_{breach}(t) \leqslant \delta_{breach}$,其中 δ_{breach} 为系统预设的参数,则称 T^* 满足个性化扩展 l-多样性.

表 3　指定保护节点的个人医疗原始信息记录表

Name	Age	Sex	Zip code	Disease	Guarding attribute
Fred	24	M	13000	HIV	Illness
Andy	25	M	13001	HIV	Illness
James	27	M	17000	Flu	Flu
Allen	29	M	17001	Cancer	Illness
Jacky	30	M	17002	Obesity	Obesity
Emily	42	F	56000	Pneumonia	Communicable
Judy	45	F	56010	Indigestion	Noncommunicable

4　个性化扩展 l-多样性逆聚类算法

4.1　信息损失量

信息损失量指标在某种程度上较全面地反映了匿名前后数据表的可用性,另外,本文算法中需要对敏感属性进行泛化,而这一过程同样包含着一定的信息损失,这种损失也间接反映了匿名后敏感属性的保护程度.

首先,我们考虑泛化前后单个属性值的信息损失量,定义如下:

定义 9　属性值的信息损失量.给定属性 A,其值域为 $Dom(A)$,则属性值 $v \in Dom(A)$ 泛化为 v^* 的信息损失量 IL_{value} 定义为:

$$IL_{value}(v) = \frac{|v^*|}{|Dom(A)|} \tag{1}$$

其中,$|v^*|$ 和 $|Dom(A)|$ 分别表示 v^* 和属性 A 的可能取值数,对于连续型属性,表示为区间的长度,对于离散型属性,表示值域的基数.

一个元组的信息损失量定义为元组中各个属性信息损失量的叠加.

定义 10 元组的信息损失量. 给定元组 t, $t.A_i(1 \leqslant i \leqslant d)$表示 t 在属性 A_i 上的取值,则 t 泛化为 t^* 的信息损失量 IL_{tuple}定义为:

$$IL_{tuple}(t) = \sum_{i=1}^{d} w_i \cdot IL_{value}(t.A_i) \quad (2)$$

其中, w_i 表示 $t.A_i$ 泛化的精度损失权重.根据元组泛化前后的信息损失量,我们可以定义元组间的距离 $Distance(t_i, t_j)$.

定义 11 元组间的距离. 给定元组 t_i 和 t_j,两者之间的距离 $Distance(t_i, t_j)$定义为:

$$Distance(t_i, t_j) = \sum_{k=1}^{d} \left| |t_i.A_k| - |t_j.A_k| \right| \quad (3)$$

其中, $\left| |t_i.A_k| - |t_j.A_k| \right|$表示元组 t_i 和 t_j 在属性 A_k 上的距离,对于连续型属性,采用两者的区间长度之差,对于离散型属性,采用两者值域的基数之差.

一个表的信息损失量由表中全部元组的信息损失量相加得到.

定义 12 表的信息损失量. 给定数据表 T,泛化为 T^* 的信息损失量 $IL_{table}(T)$定义为:

$$IL_{table}(T) = \sum_{t \in T} IL_{tuple}(t) \quad (4)$$

4.2 敏感属性泛化算法

通过对敏感属性的泛化来使匿名组满足个性化服务的需求,敏感属性泛化的算法思想是:通过比较输入等价类中每个元组敏感属性的保护属性的隐私侵犯率是否小于等于系统给定的最大隐私侵犯阈值,如果大于该阈值,则对此元组的敏感属性值通过敏感属性继承分类树进行调整,反复这一过程,直到所有的元组都满足条件或不能再进行调整为止.算法 1 给出了具体过程.

算法的第 1 步进行初始化工作,假设等价组的元组数为 N_{tuple},集合的复制过程可在 $O(N_{tuple})$内完成;第 2 步是将保护属性不属于等价组内任何其他元组继承分类子树的节点的元组归入一个临时的集合,假设敏感属性继承分类树中的叶子结点为 N_{node},则第 2 步所需的时间代价为 $O(N_{tuple} \times N_{node})$;算法的第 3 步循环执行将不满足个性化需求的元组进行敏感值泛化操作,由于只是对元组的隐私侵犯率进行判断,所以最坏情况下所需的时间复杂度为 $O(N_{tuple})$;对于第 4 步,对整个等价类进行一次调整,时间复杂度也是 $O(N_{tuple})$.综上所述,整个算法的时间复杂度为: $O(N_{tuple}) + O(N_{tuple} \times N_{node}) + O(N_{tuple}) + O(N_{tuple}) = O(N_{tuple} \times (N_{node} + 3)) \approx O(N_{tuple} \times N_{node})$.

算法 1 敏感属性泛化算法 S-Generalization(SG)

输入:等价类 SG,敏感属性继承分类树,个性化约束参数 δ_{breach}

输出:满足个性化的等价类 SG^*

步骤:

1. 初始化:
1.1 $SG* = SG$;
1.2 $temp\_SG = \varnothing$;
2. For(每个 SG 中的元组 t)
2.1 For(每个 SG^* 中不同于 t 的其他元组 t')
2.2.1 如果 $t.GA \notin SUB\_ITax(t'.GA)$
2.2.2 则 $temp\_SG = temp\_SG \cup \{t\}$;
3. For(每个 $temp\_SG$ 中的元组 t_k)
3.1 如果 $P_{breach}(t_k) > \delta_{breach}$,则
3.1.1 如果 $t_k^*.A^S$ 是继承分类树的根节点,则返回步骤 3,检查下一个元组;
3.1.2 $t_k^*.A^S$ 泛化为 $t_k^*.A^S$ 的父节点;
4. For(每个 $SG*$ 中不同于 t^* 的其他元组 t')
4.1 如果 $t'^*.A^S$ 在 $t^*.A^S$ 的继承分类子树 $SUB_ITax(t^*.A^S)$中
4.2 则 $t'^*.A^S = t^*.A^S$;
5. Return SG^*.

4.3 逆聚类与 l-多样性

传统的聚类是将相似的对象集中在一起,在 l-多样性原则中要求在发布的等价类中至少有 $l(l \geqslant 2)$个相异的敏感属性值,基于此,本文提出了一种逆聚类的思想来满足等价类的 l-多样性.首先给出以敏感属性为判定对象的元组间相异度距离计算准则.

定义 13 相异度距离. 给定敏感属性 S 的继承分类树 $ITax(S)$,记元组 t_i 的敏感属性值 $t_i.S$ 对应 $ITax(S)$中结点 $N(t_i.S)$,则元组 t_i 和 t_j 关于敏感属性 S 的相异度距离 $\Delta(t_i, t_j)$定义为为结点 $N(t_i.S)$到结点 $N(t_j.S)$的最短距离,表示为:

$$\Delta(t_i, t_j) = \min DS(N(t_i.S), N(t_jS)) \quad (5)$$

其中, $DS(N(t_i.S), N(t_jS))$表示从结点 $N(t_i.S)$到结点 $N(t_j.S)$需要遍历的边(树枝)数.

逆聚类方法的基本思想是根据元组间的相异度距离,将性质相异的对象形成若干类,最终形成一个逆聚类集.显然地,元组 t 被分配到类簇 C 中的原则是 t 与类簇 C 中质心 t_{cm}的相异度距离最远.当考虑每个逆聚类中至少包含 l 个不同敏感值的元组时,可以将 l-多样性问题看作是逆聚类问题,对于这一特殊的聚类,我们定义为 l-多样逆聚类问题.

定义 14 l-多样逆聚类问题. 令 SG 为一个包含 m 个元组的集合, l 为多样性参数,则 l-多样逆聚类问题要求得到逆聚类集 $ICSG = \{g_1, g_2, \cdots, g_n\}$满足以下条件:

(1) $\forall\, i \neq j, g_i \neq g_j$,其中 $1 \leqslant i, j \leqslant n$;

(2)$\bigcup_{k=1}^{n} g_k = SG$;

(3)$\forall g_i \in ICSG, |g_i.S| \geqslant l$,其中$|g_i.S|$表示类$g_i$敏感属性值的基数;

(4)$\Delta(g_h) = \max \sum_{p(g_h,i)\in SG} \Delta(p(g_h,gc_h),p(g_h,i))$,其中$gc_h$为类$g_h$的质心,$\Delta(g_h)$表示类$g_h$中所有点到质心$gc_h$的相异度距离之和,$\Delta(p(g_h,gc_h),p(g_h,i))$表示点$p(g_h,i)$到质心$p(g_h,gc_h)$的相异度距离.

4.4 基于逆聚类的个性化隐私匿名方法

个性化扩展 l-多样性逆聚类算法(PELI-clustering)以逆聚类的计算准则实现等价类的个性化扩展 l-多样性,其基本思想是:首先任意选取数据集中的一个元组作为聚类质心,根据该元组的敏感属性值形成相对于该质心的逆聚类候选集,通过候选集中的元组来形成满足扩展 l-多样性的逆聚类等价类,重新计算聚类质心,并选取该聚类质心最远的元组为下一个聚类的质心,重复该过程直至全部的元组归入相应的逆聚类等价类或不满足聚类的条件为止,如果经过上述逆聚类过程还有剩余的元组未被聚类,则将这些元组归入质心离自身距离最近的等价类中,然后对每个逆聚类等价类进行个性化服务的检验和准标识符属性的泛化工作,最后形成满足个性化扩展 l-多样性的匿名表.具体描述见算法 2.

算法 2 第 1 步是初始化工作,步骤 1.1 中判断算法的可执行性,关键是参数 l 的设置,要求不大于表中敏感属性的基数,否则无法满足 l-多样性原则,可在 $O(1)$ 的时间内完成;第 2 步随机选取表中的一个元组作为初始聚类的质心;在算法第 3 步生成一个逆聚类划分空间,步骤 3.1 至步骤 3.4 是对生成当前簇的初始化工作,有一个对表中元组进行比较的过程,假设表中的元组数为 N_{table},则故时间复杂度为 $O(N_{table})$,步骤 3.5 循环选取满足逆聚类的元组加入当前簇中,每一次循环对候选集进行一次比较,最坏情况下需要 $O(N_{table})$的时间,循环的次数控制在 l 次,故步骤 3.5 所需的时间代价为 $O(l \times N_{table})$,步骤 3.6 至步骤 3.8 是重新选择下一个簇质心的过程,时间代价为 $O(N_{table} - N_{SG})$,故而算法第 3 步总的时间代价为 $O(N_{table}) + O(N_{table}) + O(l \times N_{table}) + O(N_{table} - N_{SG}) = O(N_{table} \times (l+3) - N_{SG}) \approx O(l \times N_{table})$;算法第 4 步是处理剩余元组的过程,将剩余的元组归入离自身最近的簇中,所需的时间在最坏情况下为 $O(N_{table}/l)$;算法第 5 步是等价类的泛化过程,步骤 5.1 是对簇中的准标识符属性进行泛化,在本算法采用文献[13]中的最优泛化策略,设准标识符属性个数为 N_{qi},则最坏情况下 QI 泛化的时间代价为 $O(N_{qi} \times Max\_N_{SG})$,其中 Max_N_{SG}表示等价聚类中基数最大值,步骤 5.2 是敏感属性的泛化时间复杂度为 $O(Max\_N_{SG} \times N_S)$,故泛化总的时间代价为 $N_{table}/l \times (O(N_{qi} \times Max\_N_{SG}) + O(Max\_N_{SG} \times N_S)) = O(N_{table} \times Max\_N_{SG} \times (N_S + N_{qi})/l)$.

综上所述,算法总的时间复杂度为:$O(1) + O(l \times N_{table}) + O(N_{table}/l) + O(N_{table} \times Max\_N_{SG} \times (N_S + N_{qi})/l) = O(l \times N_{table} + N_{table} \times Max\_N_{SG}) = O(N_{table}^2)$.

算法 2　个性化扩展 l-多样性逆聚类算法(PELI-clustering)

输入:带保护属性的待发布数据表 T,多样性约束参数 l
输出:满足个性化扩展 l-多样性的匿名表 PT
步骤:

1. 初始化:
1.1 计算数据表 T 中敏感属性 S 的基数 N_S,如果 $N_S < l$,则返回重新设置 l 值;
1.2 逆聚类候选集 $IC\_T = \varnothing$;
1.3 匿名表 $PT = \varnothing$;
2. 从 T 中随机选取一个元组 t;
3. While ($N_S \geqslant l$) do
3.1 令聚类 SG 质心 $t_{cm} = t$;
3.2 计算类的基数 $N_{SG} = |SG|$;
3.3 $T = T - \{t\}$;
3.4 将 T 中敏感值与 t_{cm}不同的元组加入逆聚类候选集 IC_T;
3.5 While ($N_{SG} < l$) do
3.5.1 计算 IC_T 中敏感值与 SG 中各元组不同的元组距离,取距离最小的元组 min_tuple;
3.5.2 $SG = SG \cup \{min\_tuple\}$;
3.5.3 $IC\_T = IC\_T - \{min\_tuple\}$;
3.5.4 $T = T - \{min\_tuple\}$;
3.5.5 重新计算等价聚类 SG 的质心 t_{cm};
3.6 $PT = PT \cup SG$;
3.7 重新计算 T 中敏感属性 S 的基数 N_S;
3.8 t = 距离 t_{cm}最远的元组;
4. While ($T \neq \varnothing$) do
4.1 随机选取元组 t,计算 t 与 PT 中各等价聚类质心的距离;
4.2 将 t 加入距离最小的等价聚类中;
4.3 $T = T - \{t\}$;
5. For (每个 PT 中的等价聚类 SG)
5.1 QI-generalization(SG);/* 对簇中的准标识符属性进行泛化 */
5.2 S-generalization(SG);
6. Return PT.

5 实验及结果分析

5.1 实验数据及参数

实验数据集采用 UCI Machine Learning Repository 中的标准 Adult 数据集*,该数据集由美国人口普查数据

* http://archive.ics.uci.edu/ml/datasets/Adult

构成,合并训练集和测试集,并删除存在缺省值的记录,包含 45 222 条记录,15 个属性,为了便于研究,本文选取其中的 6 个属性作为研究对象,同时假设其中的 5 个属性为准标识符属性,1 个为敏感属性,各个属性的具体描述如表 4 所示.

表 4 数据集属性描述

No.	Attribute	Type	Distinct Values
1	Age	Numeric	74
2	Work class	Categorical	8
3	Education	Numeric	16
4	Marital-status	Categorical	7
5	Sex	Categorical	2
6	Occupation	Sensitive	14

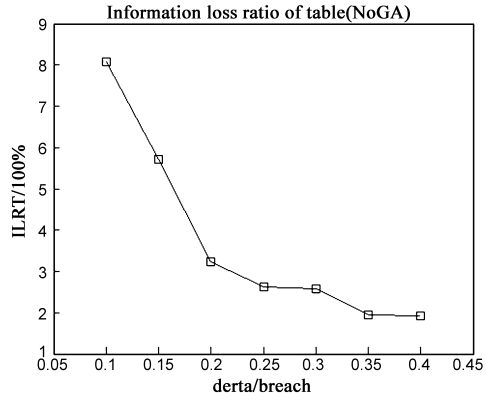
取[Occupation]为敏感属性,该属性列中有 14 个不同的属性值,将属性值按字母序设置编号,使该属性的值域变为[0,15).下面介绍构建继承分类树的过程:首先将值域分成 15 个区间,区间长度为 1,形成叶子节点;然后以长度 5 为区间长度,形成分类树的第 2 层节点,包含 3 个节点,分别是[0,5),[5,10)和[10,15);分类树的第 3 层为根节点[0,15).同时,考虑个性化服务的需求,对原始数据表中的敏感属性[Occupation]加入保护属性列[Guarding Attribute],随机选取 20%记录的 GA 值为原始敏感值的父节点,10%记录的 GA 值为空,剩余的 70%记录为原始值,并称包含 GA 的数据表为 HasGA,原始数据表为 NoGA.

实验从信息损失度与执行时间两个方面进行分析,将本文所提的算法与文献[3]中 l-多样性的实现方法以及改进文献[5]中所提的 k-member 聚类算法使之满足 l-多样性原则的 l-member 聚类算法进行比较,同时在两个算法中植入敏感属性泛化算法,使之满足个性化需求.实验的硬件环境为 Intel Core(TM)2 Duo CPU E4600 @ 2.40GHz,1.99GB 的内存;操作系统为 Microsoft Windows XP,算法均在 VC++6.0 与 Matlab 7.0 混编环境下实现.

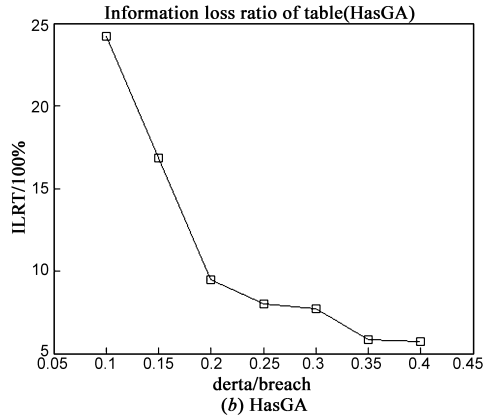
5.2 信息损失度分析

一个表的信息损失量由式(4)给出,其信息损失度 $ILR_{table}(T)=IL_{table}(T)/|T|$.下面从两组不同的实验来验证算法的性能,并考察发布数据的精度.

第一组实验比较 NoGA 和 HasGA 下的信息损失度,图 2(a)和图 2(b)分别给出了在不同系统预设的隐私侵犯率阈值 δ_{breach}下有无保护属性的信息损失度,其中多样性参数 l 取 4,从图 2 中可以看到,不管有无保护属性,发布数据的信息损失度都会随着系统阈值 δ_{breach} 的增大而减少,而在同等条件下,个性化隐私保护需求下的数据表比没有保护属性的原始数据表的信息损失要大,这是由于个性化需求下的隐私保护需要进行敏感属性的泛化,从而产生了更多的信息损失.



(a) NoGA



(b) HasGA

图2 不同系统阈值δ_{breach}下的信息损失度

第二组实验比较 PELI-clustering 算法与 l-多样性和改进的 l-member 聚类算法的性能.图 3 给出了在不同 l 值下三种算法信息损失度比较,其中准标识符属性个数|QI|为 5,数据表取 HasGA,元组数为 45 222.由图 3 可知:三种算法的信息损失度都会随着 l 值的增大而增加,因为 l 值越大,生成聚类所包含的元组个数增多,对元组的泛化程度会更高,从而引起更多的信息损失.图 4 为 l 取 6,数据表取 HasGA,元组数为 45 222,在不同准标识符属性个数|QI|下三种算法的信息损失度比较.由图 4 可知:三种算法的信息损失度都会随着|QI|值

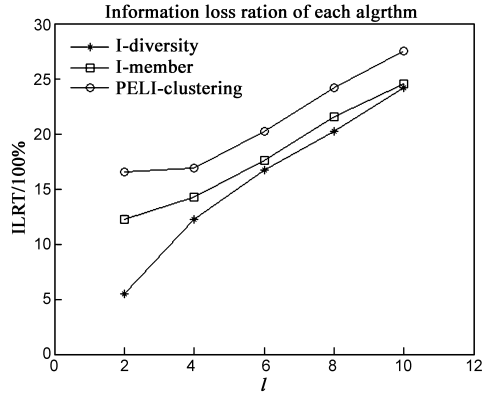


图3 不同l值下信息损失度的比较

的增大而增加,显然 $|QI|$ 值增大使得需要泛化的属性增多,信息损失度必然增加.

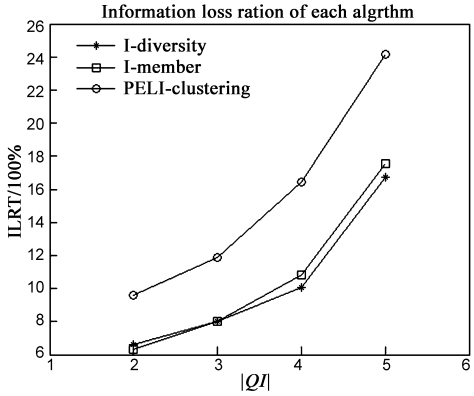


图4　不同$|QI|$值下信息损失度的比较

5.3　执行时间分析

图 5 给出了在不同 l 值下三种算法执行时间的比较,其中准标识符属性个数 $|QI|$ 为 5,数据表取 HasGA,元组数为 45 222. 由图 5 知,三种算法的执行时间都会随着 l 值的增大而增加,因为 l 值越大,聚类的次数越多,从而时间花销就会越多. 图 6 为 l 取 6,数据表取 HasGA,元组数为 45 222,在不同准标识符属性个数 $|QI|$ 下三种算法执行时间比较. 由图 6 知,随着 $|QI|$ 值的增加,三种算法的执行时间会随之增加,因为当准标识符属性个数增多时,每次聚类所涉及的属性个数增多,必然导致计算量的增大,从而引起更大的时间开销.

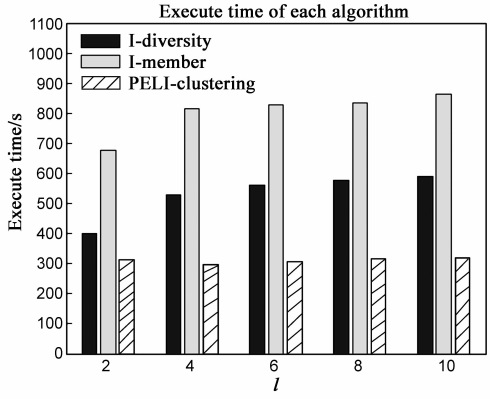


图5　不同l值下的执行时间比较

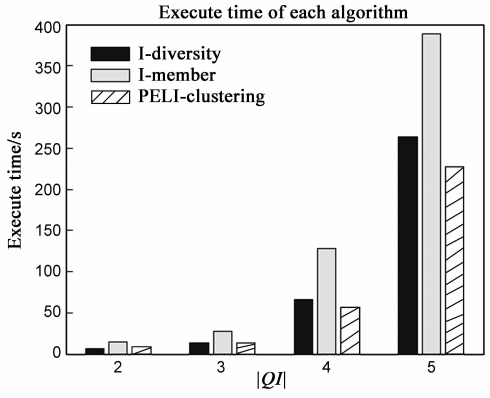


图6　不同$|QI|$值下的执行时间比较

另外,从图 5 和图 6 可以看出,在同等条件下,三种算法中 PELI-clustering 算法的执行时间较少,l-member 算法的执行时间最多,这是由于 l-member 算法在聚类的过程中,需要计算整个聚类和候选集中每个元组间的信息损失,而 PELI-clustering 算法只需要计算簇质心和候选集中每个元组间的信息损失.

6　结论

本文针对数据发布中敏感信息个性化服务的需求,提出了面向个体的个性化扩展 l-多样性隐私匿名模型. 该模型通过设置敏感属性的保护属性实现个性化服务的需求,同时,在传统 l-多样性原则的基础上,定义了扩展的 l-多样性原则,有效地满足了个性化 l-多样性的隐私保护. 另外,本文还提出了个性化扩展 l-多样性逆聚类(PELI-clustering)算法,通过对候选集进行逆向聚类,有效地实现了个性化扩展 l-多样性隐私匿名模型. 实验结果表明,PELI-clustering 算法不仅与传统基于聚类的 l-多样性算法有近似的信息损失度,而且具有更小的时间代价,更有效地实现隐私保护.

下一步工作:在本文的个性化模型中,只是考虑了敏感属性是单一的情况,而现实的很多数据集都可能包含多个敏感属性,故下一步将研究面向多敏感属性的个性化隐私保护问题.

参考文献

[1] Sweeney L. Computational disclosure control: A primer on data privacy protection[D]. Massachusetts Institute of Technology, 2001. 67 – 82.

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

[3] Machanavajjhala A, Kifer D, Gehrke J, et al. l-diversity: Privacy beyond k-anonymity[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 1 – 52.

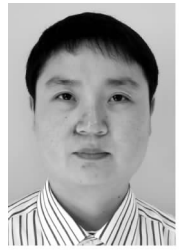
[4] Li J Y, Wong R C W, et al. Achieving k-anonymity by clustering in attribute hierarchical structures[A]. Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery[C]. Krakow, Poland, Springer Press, 2006. 405 – 416.

[5] 王智慧,许俭,汪卫,等. 一种基于聚类的数据匿名方法[J]. 软件学报,2010,21(4):680 – 693.
Wang Zhihui, Xu Jian, Wang Wei, et al. Clustering-based approach for data anonymization[J]. Journal of Software, 2010, 21(4): 680 – 693. (in Chinese)

[6] Aggarwal G, Panigrahy R, et al. Achieving anonymity via clus-

tering[J]. ACM Trans Algorithms, 2010, 6(3): 1 – 19.
[7] 滕金芳,钟诚.基于聚类的敏感属性 *l*-多样性匿名化算法[J].计算机工程与设计.2010,31(20):4378 – 4381.
Teng Jinfang, Zhong Cheng. Clustering based sensitive attribute *l*-diversity anonymization algorithms[J]. Computer Engineering and Design, 2010, 31(20): 4378 – 4381. (in Chinese)
[8] Xiao X K, Tao Y F. Personalized privacy preservation[A]. Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data[C]. New York, NY, USA: ACM Press, 2006. 229 – 240.
[9] Ye X J, Zhang Y W, et al. A personalized (a, k)-anonymity model[A]. Proceedings of the 9th International Conference on Web-Age Information Management (WAIM'08)[C]. Zhangjiajie, China: IEEE Press, 2008. 341 – 348.
[10] Shen Y G, Liu Y H, et al. Personalized granular k-anonymity [A]. Proceedings of International Conference on Information Engineering and Computer Science (ICIECS'09)[C]. Wuhan, China: IEEE Press, 2009. 1 – 4.
[11] 韩建民,于娟,虞慧群,贾▮.面向敏感值的个性化隐私保护[J].电子学报,2010,38(7):1723 – 1728.
Han Jianmin, Yu Juan, Yu Huiqun, et al. Individuation privacy preservation oriented to sensitive values[J]. Acta Electronica Sinica, 2010, 38(7): 1723 – 1728. (in Chinese)
[12] Wang P S. Personalized anonymity algorithm using clustering techniques[J]. Journal of Computational Information Systems, 2011, 7(3): 924 – 931.
[13] Bayardo R J, Agrawal R. Data privacy through optimal *k*-anonymization[A]. In Proceedings of the 21st IEEE International Conference on Data Engineering[C]. Tokyo, Japan: IEEE Press, 2005. 217 – 228.

作者简介



王 波 男,1982 年生于浙江东阳.哈尔滨工程大学计算机科学与技术学院博士研究生.研究方向为数据挖掘、隐私保护、机器学习.
E-mail: hust_wb@126.com

杨 静 女,1962 年生于黑龙江哈尔滨.哈尔滨工程大学计算机科学与技术学院博士生导师,教授.研究方向为数据挖掘、隐私保护、机器学习.