面向多敏感属性医疗数据发布的隐私保护技术

金 华 刘善成 鞠时光
(江苏大学计算机科学与通信工程学院 镇江 212013)

摘 要 针对目前多敏感属性医疗数据发布问题,在分析多维桶分组技术的基础上,继承了有损连接对隐私数据进行保护的思想,提出了一种基于相同敏感属性集的 L-覆盖性聚类分组方法。首先计算每条记录的相同敏感属性集,然后按照聚类的思想将满足 L-覆盖性的记录进行分组。同时给出了 L-覆盖性聚类分组的实现算法(LCCG)。实际数据集上的大量实验结果表明,该方法可以有效防止隐私泄露,同时增强数据的可用性。

关键词 数据发布,多敏感属性,相同敏感属性集,有损连接,L-覆盖性,聚类
中图法分类号 TP309 **文献标识码** A

Privacy Preserving Technology for Multiple Sensitive Attributes in Medical Data Publishing

JIN Hua LIU Shan-cheng JU Shi-guang
(School of Computer Science & Telecommunications Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract In view of the privacy leak problem of secure data publishing when sensitive data contains multi-attributes, on the basis of analysing the multi-dimension bucket approach, this paper proposed an L-coverage clustering grouping approach based on the same sensitive attribute set and the idea of lossy join. Firstly it calculated the same sensitive attribute set of each record, and then we grouped each record which satisfies the constraints of L-coverage following the idea of clustering. Also we designed a LCCG algorithm to implement the approach. Experimental results on the real world datasets show that the new model is able to reduce privacy disclosure apparently and enforce security of data publishing.

Keywords Data publishing, Multi-sensitive attributes, Same sensitive attribute set, Lossy join, L-coverage, Clustering

1 引言

随着网络信息技术的快速发展,有关个人信息的种类和数量都在呈指数增长。而基于信息共享、科学研究等方面的需要,数据收集者(组织或个人)须将收集到的数据进行发布。由于发布的信息会涉及到很多个人隐私信息,如果直接将收集到的原始数据进行发布,会造成大量的个人敏感信息的泄露。为了保证个人敏感信息的安全,发布数据的同时应进行隐私保护。因此面向数据发布的隐私保护已经成为众多学者关注的问题[1-3]。目前的相关研究主要为集中式数据发布[4]。即数据表中的属性通常分为3类:显式标识符属性,指能清楚标识用户身份的属性,如表1中的Name属性;准标识符属性QI(Quasi Identifier),即同时存在于隐私表与外表中,可以利用链接来标识个体信息的一组属性,称为准标识符属性,如属性组{Age、Country、Zipcode};敏感属性SA(Sensitive Attribute),该类属性包含了个体的隐私信息,如Disease。数据发布者发布数据时,一方面要使得发布的匿名数据不泄露数据中个体的隐私信息,另一方面需要保证发布的匿名数据具有高可用性,即仍然能够根据发布的匿名数据进行较准确的数据分析。

表1 多敏感属性数据集

ID	Name	Age	Country	Zipcode	Physician	Disease
1	Tom	27	USA	14248	Gregory	HIV
2	Bob	28	Canada	14207	Gregory	Cancer
3	Carl	26	USA	14246	Jennifer	Flu
4	Anne	25	Canada	14249	Jesse	Flu
5	John	41	China	13053	Cuddy	Gastritis
6	James	48	Japan	13074	Gregory	Pneumonia
7	Lily	45	India	13064	Wilson	Flu
8	Emily	42	India	14242	Cuddy	Gastritis
9	Tim	33	USA	14204	Gregory	Cancer
10	Jane	37	Canada	14205	Wilson	Flu
11	Ella	36	Canada	14204	Wilson	Pneumonia
12	Mike	35	USA	14248	Cuddy	Gastritis

现有的匿名数据发布技术大部分主要针对的是单一敏感属性的数据。而在很多现实应用中,发布的数据中往往含有多个敏感属性,而且这些敏感属性之间可能会存在一定的关联,例如表2显示了表1中Physician和Disease属性之间的关联。攻击者可以借助于这一关联增大推测目标个体隐私信息的概率。而传统的单敏感属性隐私保护技术并不能做到预防这类借助于敏感属性之间关联的隐私攻击。

到稿日期:2010-12-30 返修日期:2011-03-22 本文受江苏省自然科学基金项目(BK2010192),教育部博士点基金项目(20093227110005)资助。

金 华(1977—),男,博士生,讲师,主要研究方向为数据库安全、隐私保护,E-mail:jinhua@ujs.edu.cn;刘善成(1988—),男,硕士生,主要研究方向为隐私保护;鞠时光(1955—),男,博士,教授,主要研究方向为数据库理论和信息安全。

表 2　多敏感属性之间的关联关系

Physician	Disease
Gregory	HIV,Cancer,Pneumonia
Jennifer	Flu
Jesse	Flu
Cuddy	Gastritis
Wilson	Flu,Pneumonia

杨晓春等人首次对多敏感属性数据发布问题进行了详细研究，继承了基于有损连接对隐私数据进行保护的思想，提出了针对多敏感属性隐私数据发布的多维桶分组技术[5]。本文在前人工作的基础上对多敏感属性，特别是相关多敏感属性隐私数据发布问题进行了进一步研究，提出了一种基于相同敏感属性集的 L-覆盖性聚类分组方法。其核心思想是通过聚类的方法将满足 L-覆盖性的记录进行分组。在保护数据记录的隐私信息的同时尽可能多地保留发布数据信息的可用性。本文的主要贡献如下：

(1) 提出了基于相同敏感属性集的 L-覆盖性聚类分组隐私数据发布方法。该方法适用于含有多个敏感属性的关系型数据，在保证数据发布安全性的同时，能保留尽可能多的可用信息。

(2) 给出了基于相同敏感属性集的 L-覆盖性聚类分组实现算法。

(3)采用实际数据集进行大量实验，对所提出的方法进行了验证与分析，与多维桶分组技术的试验结果进行了对比。实验结果表明了本文方法的有效性。

2　相关工作

k-匿名模型是一种简单有效的隐私保护模型，通常采用泛化和隐匿技术[6,7]来实现该模型，但该模型不能抵制同质性攻击和背景知识攻击[8]。2006 年，Machanavajjhala 等提出了 L-多样性算法[9]，它保证每个等价类中敏感信息足够多样，以抵制同质推理攻击和背景知识攻击，但该模型不能抵制偏斜攻击和相似性攻击。同年，T. M. Truta 提出 p-Sensitive k-anonymity 模型[10]，它简化了 L-多样性实现中的参数设置，但该模型没有控制等价类中敏感值出现的频率，当 k 比 p 大得多时，无法抵制概率攻击。此后，Wong 等人提出了 (α,k)-匿名模型[11]，要求每个等价类的敏感值的频率不大于 α。但是 (α,k)-匿名模型为所有的敏感值设置统一的频率约束，适应性比较差，因为数据表中不同的敏感值可能需要不同的频率约束。Li Ninghui 等人提出了 t-closeness 框架[12]，该方法要求每个等价类的敏感值的分布要接近于其在原始数据表中的分布。但它的缺点也很显著，即使满足与原数据分布相似也并不一定能保证发布数据的安全性，而且它将很大程度地破坏数据的实用性，一个完美的 t-closeness 将完全割裂敏感属性和准标识符之间的关联。

以上提出的敏感数据发布方法都主要针对单一敏感属性数据的情况。对于具有多个敏感属性，特别是多个相关敏感属性的数据集，直接应用以上方法并不能保证隐私数据的安全。例如表 4 是(对表 3 中数据记录有损连接匿名的结果)满足单敏感属性(将 Physician 和 Disease 看作是一个整体组合) L-多样性($L=4$)的，但在实际中若攻击者获取目标个体就诊的医生是 Jennifer 或 Jesse，则攻击者可以轻易地获取目标个体敏感属性 Disease 的值为 Flu，即攻击者可以以 $1/2(>1/L,L=4)$的概率推测出目标个体的隐私信息。

表 3　元数据记录

ID	Age	Country	Zipcode	Physician	Disease
1	27	USA	14248	Gregory	HIV
2	28	Canada	14207	Gregory	Cancer
3	26	USA	14246	Jennifer	Flu
4	25	Canada	14249	Jesse	Flu

表 4　满足单敏感属性 L-多样性($L=4$)的有损连接匿名数据表

QIT

Age	Country	Zipcode	Group ID
27	USA	14248	1
28	Canada	14207	
26	USA	14246	
25	Canada	14249	

ST

Group ID	Physician	Health Condition
1	Gregory	HIV
	Gregory	Cancer
	Jennifer	Flu
	Jesse	Flu

针对多敏感属性数据发布问题，文献[5]假设安全数据发布需满足 L-多样性约束，并基于此提出了基于有损连接技术的支持多敏感属性的隐私数据发布多维桶分组技术。该技术可以很好地实现对多种敏感属性数据发布的隐私保护，但该方法的分组效率较低，往往会由于每次分组选取桶的顺序问题造成大量不必要的数据记录遭到隐匿，从而大大降低了数据的可用性。也正因此我们进一步讨论研究了多敏感属性医疗数据发布中对敏感值分布约束的问题，提出了基于有损连接技术和相同敏感属性集的 L-覆盖性聚类分组方法。

3　L-覆盖性聚类分组模型

3.1 基本概念

设 T 为多敏感属性数据集，含有 d 个准标识属性 $A_{Q_1},\cdots,A_{Q_d}$ 和 m 个敏感属性 $A_{S_1},\cdots,A_{S_m}$，符号 t 表示 T 中的一条记录，而记录 t 对于不同敏感属性列上的敏感属性值为 $t.S_1,\cdots,t.S_m$。并且我们假设准标识符属性和敏感属性之间没有交集，即假设这些记录的敏感属性不会出现在其它的已对外发布的数据集中。

定义 1(分组)　将 T 分为若干子集，T 中每条记录都属于某一子集，且子集之间互不相交。我们将这些子集称为分组，记为 $G_1,\cdots,G_p$。

定义 2(有损连接)　将 T 划分为两部分：一个准标识符表(QIT)和一个敏感属性表(ST)。QIT 用于保留所有的准标识符属性 $A_{Q_1},\cdots,A_{Q_d}$ 和每条记录所在分组的 ID 记为 GID。而 ST 用于保留分组 ID 和敏感属性值 $A_{S_j}(1\leqslant j\leqslant m)$。

定义 3(单敏感属性 L-多样性)　对于一组单敏感属性记录 G，设 v 为 G 中记录的敏感属性取值中最大频繁取值，$c(v)$ 为其出现的次数，如果 $\frac{c(v)}{|G|}\leqslant\frac{1}{L}$($|G|$ 为 G 中的记录总数)，则 G 就满足单敏感属性 L-多样性。

定义 4(移除)　对于一个分组 G，若要将 G 中任意一条记录 t 的某一敏感值 $t.S_i(1\leqslant i\leqslant m)$ 从 G 中删除，则需要将 G 中所有含有该敏感值 $t.S_i$ 的记录都删除。

定义 5(多敏感属性 L-覆盖性)　对于一个分组 G，若至少要移除 L 个不同敏感属性值才能将分组 G 中的所有记录

都删除，则 G 就满足多敏感属性 L-覆盖性。

例如对表 4 中所列匿名数据表，只需移除敏感属性“Gregory”和“HIV”就可以将分组中的所有记录移除，故该匿名分组只满足多敏感属性 2-覆盖性。

定理 1 数据集 T 中满足 L-覆盖性的分组 GT 对于发布的数据是安全的。

证明：由定义 5 可知，对于满足 L-覆盖性性质的分组 $G(t_1,t_2,\cdots,t_n)(n\geqslant L)$，其中任意两条记录 $t_i,t_j(1\leqslant i\leqslant n,1\leqslant j\leqslant n,i\neq j)$，只要满足 $t_i.S_v=t_j.S_v(1\leqslant v\leqslant m,m$ 为敏感属性列数）就可以将它们划分到同一集合 $SA_j(1\leqslant j\leqslant c,c\geqslant L,c$ 为集合数）中，即集合 SA_j 中任两条记录必在某个敏感属性列上取值相同，且对于任意两个集合 $SA_i,SA_j(1\leqslant i\leqslant c,1\leqslant j\leqslant c,i\neq j)$ 在任一敏感属性列 $S_v(1\leqslant v\leqslant m)$ 上都满足 $SA_i\cap SA_j=\varnothing$。分组 G 内所有记录的准标识符属性和多敏感属性分别发布为 QIT 表和 ST 表，通过连接操作，攻击者对于分组 G 中记录 t 在任一敏感属性列 $S_v(1\leqslant v\leqslant m)$ 中无法以大于 $1/c(c\geqslant L)$ 的概率推断出包含其真实敏感属性值的集合 SA，即无法以高于 $1/L$ 的概率推断出记录 t 的相关敏感属性值信息。因此，对这个分组来说，发布的数据是安全的。又因为数据集 T 中所有分组都满足 L-覆盖性，因此按照这个分组原则发布的 QIT 和 ST 表是安全的，也就是说数据集 T 中满足 L-覆盖性的分组 GT 是安全的。定理得证。

定义 6（多敏感属性 L-多样性） 根据对分组内敏感值分布的约束不同可分为两种：1）强多敏感属性 L-多样性，即 T 中每个分组 $G_i(1\leqslant i\leqslant p)$ 对每一个敏感属性列 $G_i.S_j(1\leqslant j\leqslant m)$ 中的敏感值分布都满足单敏感属性 L-多样性；2）弱多敏感属性 L-多样性，即 T 中的每个分组 G_i 都满足多敏感属性 L-覆盖性，即至少需要移除 L 个不同敏感属性值才能将该分组内所有记录全部移除。

定理 2 满足强多敏感属性 L-多样性的分组必然满足弱多敏感属性 L-多样性。

证明：设分组 $G(t_1,t_2,\cdots,t_n)(n\geqslant L)$ 满足强多敏感属性 L-多样性，则对于分组 G 中任意两条记录 $t_i,t_j(1\leqslant i\leqslant n,1\leqslant j\leqslant n,i\neq j)$ 都满足 $t_i.S_v\neq t_j.S_v(1\leqslant v\leqslant m,m$ 为敏感属性列数），且每移除一个敏感属性值至多移除一条记录，故至少需要进行 L 次移除才能将分组 G 内的所有记录都移除，由定义 5 可知分组 G 是满足多敏感属性 L-覆盖性的，又根据定义 6 可知分组 G 是满足弱多敏感属性 L-多样性的。定理得证。

定理 3 若分组 G 中存在一个子集 C 是满足多敏感属性 L-覆盖性的，则 G 也满足多敏感属性 L-覆盖性。

证明：若分组 $G(t_1,t_2,\cdots,t_n)(n\geqslant L)$ 中存在一个子集 $C(t_1,t_2,\cdots,t_k)(k\geqslant L)$ 满足多敏感属性 L-覆盖性，即对于子集 C 至少需要执行 L 次敏感属性值的移除操作才能将 C 中的记录全部移除。又 $C\subseteq G$，故对于分组 G 而言将至少也需要 L 次的敏感属性值的移除操作才能移除组内全部记录。根据定义 5 可知分组 G 必然也是满足多敏感属性 L-覆盖性的。定理得证。

3.2 L-覆盖性聚类分组模型

本文提出的基于相同敏感属性集的 L-覆盖性聚类分组方法采用有损连接的方法将满足 L-覆盖性的记录进行分组。同时对剩余的记录采用两种处理方式：一种是需要满足强多敏感属性 L-多样性才能将其添加到分组中，另一种是只需要保证插入记录后仍满足弱多敏感属性 L-多样性即可。对于前一种处理方法，采用附加信息损失度来衡量分组大小超过 L 时造成的附加有损连接信息损失。第二种处理方式除了会造成额外的附加信息损失，也会由于添加新记录后仅满足弱多敏感属性 L-多样性可能造成概率泄漏的情况，从而增大了该敏感属性值出现的概率。为此引入了平均概率泄漏度来衡量只满足弱多敏感属性 L-多样性的分组可能造成的敏感值出现的概率泄漏。

定义 7（附加信息损失度） 对于数据集 T 上满足多敏感属性 L-多样性的分组 $GT\{G_1,\cdots,G_m\},|G_i|\geqslant l(1\leqslant l\leqslant m)$，附加信息损失度（additional information loss）为 $\sum_{1\leqslant i\leqslant m}(|G_i|-l|)/ml$。

定义 8（平均概率泄漏度） 对于数据集 T 中仅满足弱多敏感属性 L-多样性的分组 $GT\{G_1,\cdots,G_m\},|G_i|\geqslant L(1\leqslant i\leqslant m)$。对分组 G_i 设该分组大小为 g_i，敏感属性值种类为 n，每个敏感属性值 $v_i(1\leqslant i\leqslant n)$ 出现的频率为 $c(v_i)$，则分组 G_i 的概率泄漏度为 $Leak(G_i)=\sum_{1\leqslant i\leqslant n}(\frac{c(v_i)}{g_i}-\frac{1}{L})$，数据集 T 的平均概率泄漏度为 $Leak(T)=\frac{\sum_{1\leqslant i\leqslant m}Leak(G_i)}{|T|}$，其中 $|T|$ 表示数据集的记录总数。

根据定理 2，可以将剩余记录全部添加到分组中，即不存在隐匿记录的情况。为了降低插入剩余记录而造成的附加信息损失度和平均概率泄漏度，我们将剩余的记录均匀地插入到分组中，即将剩余记录插入到分组大小较小的分组中。

定义 9（相同敏感属性集 SID） 对于数据表 T 中，含有敏感属性值 v 的所有记录组成的集合记为 $SID(v)$。

定义 10（记录相同敏感属性集 TSID） 数据表 T 中任一记录 t 的 TSID 为该记录 t 的所有敏感属性值 $t.S_1,\cdots,t.S_m$ 对应的相同敏感属性集 $SID(t.S_i)$ 的总和，即 $t.TSID=\sum_{i=1}^{m}SID(t.S_i)$。

定义 11（分组相同敏感属性集 GSID） 对任一分组 G_i，设该分组的大小即分组包含记录的条数为 g_i，则 G_i 的分组相同敏感属性集表示该分组内所有记录的记录相同敏感属性集 TSID 的总和，记为 $G_i.GSID=\sum_{j=1}^{g_i}t_j.SID$。

例如表 3 中记录 t_1 的敏感属性值“Gregory”和“HIV”的相同敏感属性集分别为 $SID(\text{"}Gregory\text{"})=\{t_1,t_2\}$ 和 $SID(\text{"}HIV\text{"})=\{t_1\}$，而记录 t_1 的记录相同敏感属性集为 $t_1.TSID=\{t_1,t_2\}$。对表 4 分组 G_i 相对于表 1 原始数据集的分组相同敏感属性集为 $G_1.GSID=\{t_1,t_2,t_3,t_4,t_6,t_7,t_9,t_{10}\}$。

3.3 L-覆盖性聚类分组算法

算法的主要思想是采用聚类的思想首先在数据集中顺序选取 L 个满足 L-覆盖性的记录作为一组，依次循环直到无法找到 L 个满足 L-覆盖性的记录构成分组。然后对剩余记录分两步处理：第一步处理剩余记录中可以添加到其它分组而且仍满足强多敏感属性 L-多样性的记录；第二步是处理第一步余下的记录，将它们均匀地添加到分组较小的分组中，以降低平均概率泄漏度。具体算法如图 1 所示。

算法：L-覆盖性聚类分组算法
输入：数据表 T，多样性参数 L
输出：准标识符属性表 QIT，敏感属性表 ST
步骤：
1. 数据表 T 的每个记录构成一个分组；
2. 循环，直到不存在大小小于 L 的分组；

(1)顺序选取大小小于 L 的分组 G_1；

(2)寻找不在分组 G_1 的分组相同敏感属性集 $G_1.GSID$ 中的且编号 ID 最小的分组 G_2；

若找到 G_2，则合并分组 G_1 和 G_2，生成新分组 $G_{1,2}$，同时更新 $G_{1,2}.GSID=G_1.GSID+G_2.GSID$，移除分组 G_1 和 G_2；

3. for each 剩余的分组 G_r

若存在满足 L-多样性的分组 G，与分组 G_r 合并后仍满足强多敏感属性 L-多样性约束，则将 G 和 G_r 合并为 G'，同时更新 $G'.GSID=G.GSID+G_r.GSID$，移除分组 G 和 G_r；

4. end for

5. for each 剩余的分组 G_r'

寻找分组大小最小的且满足 L-多样性的分组 G_S，将 G_S 和 G_r' 合并成新的分组 G_S'，同时更新 $G_S'.GSID=G_S.GSID+G_r'.GSID$，移除分组 G_S 和 G_r'；

6. end for

7. 将所有分组以 QIT、ST 形式输出；

图 1 L-覆盖性聚类分组算法

以表 1 的数据集为例，按照 $L=3$ 进行分组，首先将数据集 T 中的每条记录划分为一个分组并计算它们各自的 GSID。例如对于记录 t_1，我们有 $t_1.SID("Gregory")=\{t_1, t_2, t_6, t_9\}$，$t_1.SID("HIV")=\{t_1\}$，$t_1.TSID=\{t_1, t_2, t_6, t_9\}$，$G_1.GSID=\{t_1, t_2, t_6, t_9\}$，同理得到 $G_2.GSID=\{t_1, t_2, t_6, t_9\}$，$G_3.GSID=\{t_3, t_4, t_{10}\}$，$G_4.GSID=\{t_3, t_4, t_{10}\}$，$G_5.GSID=\{t_5, t_8, t_{12}\}$，$G_6.GSID=\{t_1, t_2, t_6, t_9, t_{11}\}$，$G_7.GSID=\{t_3, t_4, t_7, t_{10}, t_{11}\}$，$G_8.GSID=\{t_5, t_8, t_{12}\}$，$G_9.GSID=\{t_1, t_2, t_6, t_9\}$，$G_{10}.GSID=\{t_3, t_4, t_7, t_{10}, t_{11}\}$，$G_{11}.GSID=\{t_6, t_7, t_{10}, t_{11}\}$，$G_{12}.GSID=\{t_5, t_8, t_{12}\}$。本算法首先选取了 G_1，然后选择不在 $G_1.GSID$ 中含有的记录的分组 G_3，同时更新 $G_{1,3}.GSID=\{t_1, t_2, t_3, t_4, t_6, t_9, t_{10}\}$，选取不在 $G_{1,3}.GSID$ 中含有的记录的分组 G_5。可以得到分组 $G_{1,3,5}.GSID=\{t_1, t_3, t_5\}$。算法循环进行，继续分组得到分组 $G_{2,4,8}.GSID=\{t_2, t_4, t_8\}$ 和 $G_{6,10,12}.GSID=\{t_6, t_{10}, t_{12}\}$。剩余记录 $\{t_7, t_9, t_{11}\}$，可以将 t_{11} 添加到 $G_{1,3,5}$ 且满足强多敏感属性 L-多样性。然后将 t_7 和 t_9 分别添加到分组 $G_{2,4,8}$ 和 $G_{6,10,12}$ 中。故最后的分组结果是 $\{t_1, t_3, t_5, t_{11}\}$、$\{t_2, t_4, t_7, t_8\}$ 和 $\{t_6, t_9, t_{10}, t_{12}\}$。附加信息损失度为 1/9+1/9+1/9=1/3，平均概率泄漏度为 1/8+1/8=1/4。发布结果如表 5 所列。

表 5 L-覆盖性($L=3$)聚类分组算法发布的数据结果

QIT

QIs	Group ID
…	G_1
…	G_2
…	G_1
…	G_2
…	G_1
…	G_3
…	G_2
…	G_2
…	G_3
…	G_3
…	G_1
…	G_3

ST

Group ID	Sensitive Attribute
G1	⟨Gregory, HIV⟩
	⟨Jennifer, Flu⟩
	⟨Cuddy, Gastritis⟩
	⟨Wilson, Pneumonia⟩
G2	⟨Gregory, Cancer⟩
	⟨Jesse, Flu⟩
	⟨Wilson, Flu⟩
	⟨Cuddy, Gastritis⟩
G3	⟨Gregory, Pneumonia⟩
	⟨Gregory, Cancer⟩
	⟨Wilson, Flu⟩
	⟨Cuddy, Gastritis⟩

4 实验结果及分析

采用 UCI machine learning repository 的人口统计实际数据集进行实验测试，给出了算法的详细结果和对比分析。实际数据集来自 http://kdd.ics.uci.edu，对原数据集过滤掉不完整的记录，并进行数据格式转换后，随机提取 10k(1k=1000)记录，并选择 5 个属性作为敏感属性，如表 6 所列。实验的硬件环境为 Intel Pentium D 3.00GHz CPU，1024MB 内存；操作系统平台为 Microsoft Windows XP Professional；编程环境为 Microsoft Visual Studio 2008 编译器。

表 6 实验数据集信息

敏感属性	Education	Education-num	Marital-status	Occupation	Hours-per-week
基数	16	16	7	14	86

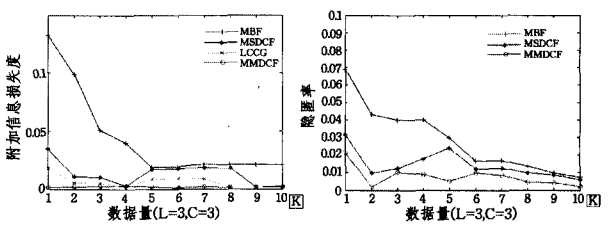
算法发布数据的安全性由 L-覆盖性保证，因此发布的数据一定是安全的。通过附加信息损失度和附加概率泄露度来衡量 L-覆盖性聚类分组算法发布数据的信息损失度和平均概率泄漏度，从而评价算法的性能。附加信息损失度越小，说明发布的数据中由于有损连接造成的信息损失越小；附加概率泄漏越小，则说明剩余记录分布越均匀，额外的概率泄漏风险越小。

实验主要从以下几个方面对算法的各个性能指标进行比较分析：(1)变化的数据集大小(数据量取 1k～10k)；(2)变化的多样性参数 L(取值为 2～9)；(3)变化的敏感属性个数 d(取值为 2～5)。表 7 给出了实验采用的不同敏感属性集描述。

表 7 实验中采用的复合敏感属性

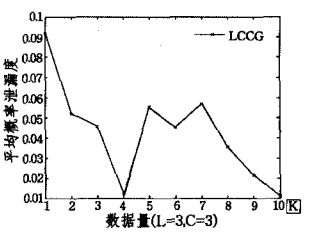
敏感属性个数	复合敏感属性
2	⟨Occupation, Education⟩
3	⟨Occupation, Education, Hours-per-week⟩
4	⟨Occupation, Education, Hours-per-week, Marital-status⟩
5	⟨Occupation, Education, Hours-per-week, Marital-status, Education-num⟩

4.1 L-覆盖性聚类分组算法性能分析



(a) 算法信息损失度的变化

(b) 算法的隐匿率变化



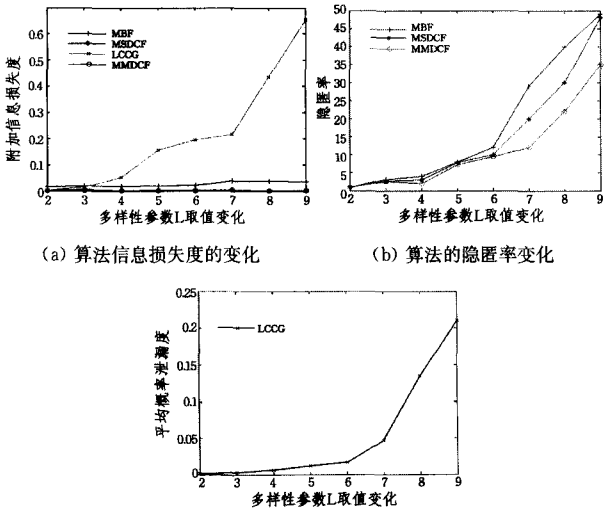
(c) 算法 LCCG 的平均概率泄漏度的变化

图 2 附加信息损失度、隐匿率和平均概率泄漏度随数据量的变化

通过大量实验测试多敏感属性隐私数据发布算法 LCCG 的附加信息损失度和平均概率泄漏度，并与文献[5]中的多敏感属性隐私数据发布算法 MBF、MSDCF 和 MDDCF 的附加信息损失度进行对比，图 2 给出了 L 取值为 3、敏感属性列数 $C=3$ 的实验结果。由图 2(a)可以看出，LCCG、MBF、MSDCF 和 MDDCF 算法的附加信息损失度都不超过 0.15，其中

MBF 算法的附加信息损失度最大，LCCG 和 MSDCF 的附加信息损失度相当。这是因为 MBF 算法形成的分组数量最少，分组中额外添加的记录也相对较多。由于 LCCG 产生的满足多敏感属性 L-多样性的分组要比 MDDCF 多，因此 LCCG 的附加信息损失度略高于 MDDCF。但 LCCG 是不以隐匿任何记录为前提的，所以 LCCG 的隐匿率为 0。而图 2(b)中基于多维桶的算法都存在一定的隐匿率。此外，从图 2(c)中可以看出，对于不同的数据集，LCCG 的平均概率泄漏度均不超过 0.1，而且随着数据集的增大，总体趋势在逐渐减小，这也是随着分组成功率的增加，剩余记录逐渐减小引起的。因此，由实验结果可以看出，LCCG 算法能够保证发布高质量的数据，且对于发布数据的质量来说，LCCG 由于不存在记录隐匿的情况，相比于其它 3 种算法可以保留更多的信息可用性。

针对多样性参数 L 的取值变化，测试算法附加信息损失度和平均概率泄漏度的影响。实验选择数据量为 5k，多敏感属性列数 $C=3$ 的数据集进行测试，结果如图 3(a)所示。LCCG 和 MBF 算法附加信息损失度都随着 L 值增大而增大。当 L 取值不超过 4 时，算法 LCCG 和 MBF 的附加信息损失度都小于 0.1。但是，当 L 取值大于 7 时，算法 LCCG 附加信息损失度迅速增加。这是由于实验数据集中 Occupation 属性的取值个数仅为 14，L 的取值越接近于这个值，满足强多敏感属性 L-多样性的分组越来越少，剩余记录越来越多，而且可以添加到分组中而不改变强多敏感属性 L-多样性的记录也很少。为了不隐匿相应记录，在这一敏感属性列上保证分组的 L 多样性越困难，这样就使得整体的分组效果显著降低。图 3(b)给出了多维桶方法随参数 L 变化时的隐匿率变化情况，而 LCCG 的隐匿率始终为 0。图 3(c)所示，LCCG 的平均概率泄漏度也随着 L 取值不断增大，但最大不超过 0.25。因此可以看出，在 L 取值不是很理想的情况下，LCCG 算法虽然可以将所有记录全部处理添加到分组中，但会造成较大的附加信息损失度，正如 MBF、MSDCF 和 MMDCF 算法在 L 取值越大时需要隐匿记录的比例越大。所以在对多样性参数 L 取值的要求上，LCCG 和其它 3 种算法的效果类似。

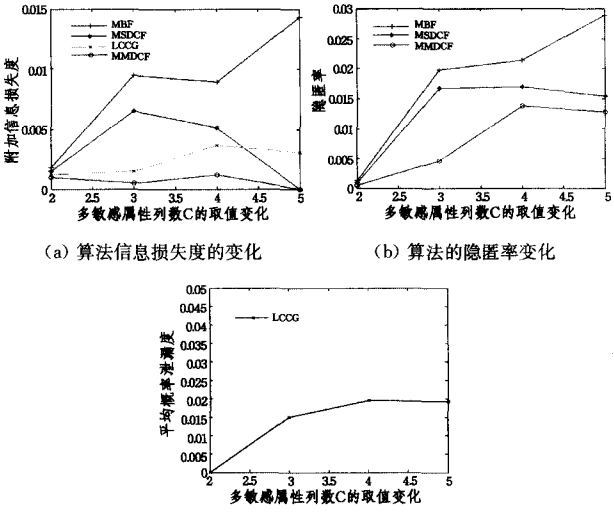


(a) 算法信息损失度的变化　(b) 算法的隐匿率变化

(c)算法 LCCG 的平均概率泄漏度的变化

图 3　附加信息损失度、隐匿率和平均概率泄漏度随多样性参数 L 的变化

图 4 给出了 7k 大小的数据集、多样性参数 L 为 4 的实验结果。从图 4(a)中可以看出，对于不同敏感属性个数的数据集，算法 LCCG 和 MDDCF 的附加信息损失度都小于 0.005，分组结果接近最优。随着多敏感属性列数的增加，算法 LCCG 的附加信息损失度有所增加，但始终在一个较小的范围内。图 4(b)显示了 MBF、MSDCF 和 MDDCF 算法隐匿率随着 C 值的变大都有增大的趋势。同时图 4(c)所示，LCCG 的平均概率泄漏度也随之缓慢增加，这是由于敏感属性的个数越多，得到在每一敏感属性列上都满足 L-多样性的分组越困难，剩余记录也随之增加，而将这些记录添加到其它分组中时又造成了每个分组中记录概率泄漏度增加。相比于 MBF、MSDCF 和 MDDCF 算法记录隐匿率的增加，LCCG 算法只是平均概率泄漏度略有增大，因而可以保留的记录可用性要更好。

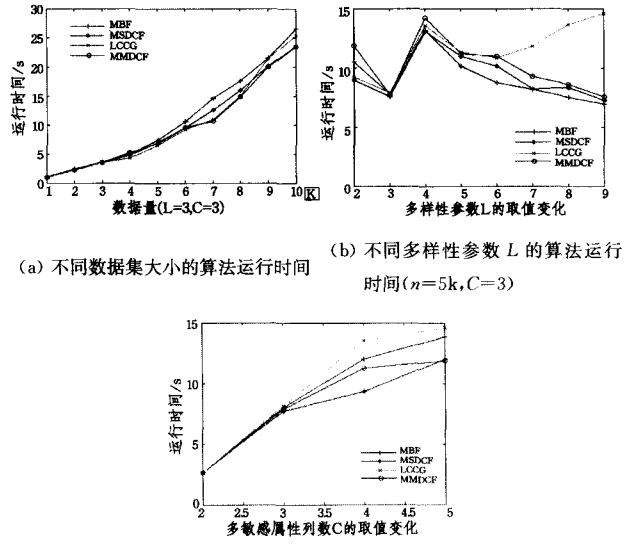


(a) 算法信息损失度的变化　(b) 算法的隐匿率变化

(c)算法 LCCG 的平均概率泄漏度的变化

图 4　附加信息损失度、隐匿率和平均概率泄漏度随敏感属性列数 C 的变化

4.2　多敏感属性隐私数据发布算法执行时间分析



(a) 不同数据集大小的算法运行时间

(b) 不同多样性参数 L 的算法运行时间($n=5k, C=3$)

(c)不同列数的算法运行时间($n=5k, L=3$)

图 5　算法执行时间随数据量、L、C 的变化

图 5 给出算法 LCCG、MBF、MSDCF 和 MDDCF 随数据集大小、参数 L、敏感属性列数 C 变化的执行时间情况。从图

5(a)可以看出，算法执行时间随着数据量的增大呈近似线性地增长，总体上4种算法的执行时间相似，LCCG和MDDCF算法的执行效率略好。由图5(b)可知，算法的执行时间都随着多敏感属性列数而变化，但幅度并不是很大。主要是由于L取值的变化并不影响MBF、MSDCF和MDDCF算法中桶的结构，因而影响并不是很大。但LCCG算法受L取值的重要影响是L取值越大，导致分组的成功率下降，剩余记录增多，进而影响了两次处理剩余记录的时间，导致后来执行时间的增大。图5(c)反映了随着多敏感属性列数C的增加，4种算法的时间也都随之增大。原因在于C值的增大使得多维桶的个数也随之增多，算法MBF、MSDCF和MDDCF对每个桶计算选择度的执行时间就增大。而算法LCCG执行时间的增加主要是因为随着C值的增大，一次分组成功率下降，剩余记录增加，进而导致了处理剩余记录的时间增加。

结束语 隐私信息的安全性是数据发布与共享环境中面临的重要问题。由于现有的隐私数据发布技术通常只针对具有单一敏感属性的数据或是没有考虑敏感属性的敏感度问题，而且在匿名化过程中将所有敏感属性值都同等对待，因此对于现实中大量存在的多敏感属性数据却无法保证其中隐私信息的安全。针对这一问题，本文提出了一种基于有损连接和相同敏感属性集的L-覆盖性聚类分组方法，并给出了方法的具体实现算法LCCG。在实际数据集上进行了大量的实验，结果表明在保证多敏感属性数据中隐私信息安全性及算法LCCG在不隐匿任何记录的前提下，其附加信息损失度和基于多维桶的方法相当。虽存在一定的平均概率泄漏度，但其值较小，且任一分组中的记录都满足L-覆盖性，可以保证发布数据的安全性，具有较高的数据发布质量。

参 考 文 献

[1] Zhao Y, Du M, Le L, et al. A Survey on Privacy Preserving Approaches in Data Publishing[C]// International IEEE Workshop on Database Technology and Applications. 2009: 128-131

[2] 周水庚，李丰，陶宇飞，等. 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)：847-860

[3] Fung B C M, Wang Ke, Chen Rui, et al. Privacy-preserving data publishing: a survey on recent developments[J]. ACM Computing Surveys, 2010, 42(4): 1-55

[4] 魏琼. 数据发布中的隐私保护方法的研究[D]. 武汉：华中科技大学，2008

[5] 杨晓春，王雅哲，王斌，等. 数据发布中面向多敏感属性的隐私保护方法[J]. 计算机学报，2008，31(4)：574-587

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

[7] Meyerson A, Williams R. On the complexity of optimal k-anonymity[C]// Proceedings of the 23rd ACM SIGACT-SIG-MOD-SIGART Symposium on Principles of Database Systems. Paris, France, 2004: 223-228

[8] Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression[J]. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002, 10(5): 571-588

[9] Machanavajjhala A, Gehrke J, Kifer D, et al. l-diversity: Privacy beyond k-anonymity[C]// Proceedings of the 22nd International Conference on Data Engineering (ICDE). 2006: 24-36

[10] Truta T M, Vinay B. Privacy protection: p-sensitive k-anonymity property[C]// 2nd International Workshop on Privacy Data Management. 2006: 94-99

[11] Wong R C-W, Li Jiu-yong, Fu A W-C, et al. (α, k)-anonymous data publishing[J]. Journal of Intelligent Information Systems, 2009, 33(2): 209-234

[12] Li N, Li T, Venkatasubramanian S. T-closeness: privacy beyond k-anonymity and l-diversity[C]// Proceedings of the IEEE ICDE. 2007: 44-56

(上接第150页)

[6] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness [M]. New York: W. H. Freeman & Co, 1979

[7] Chvatal V. A Greedy Heuristic for the Set-Covering Problem [J]. Mathematics of Operations Research, 1979, 4(3): 233-235

[8] Harrold M J, Gupta R, Soffa M L. A methodology for controlling the size of a test suite[J]. ACM Trans. Softw. Eng. Methodol, 1993, 2(3): 270-285

[9] Chen T Y, Lau M F. A new heuristic for test suite reduction[J]. Information and Software Technology, 1998, 40(5/6): 347-354

[10] Agrawal H. Dominators, super blocks, and program coverage [C]// Proceedings of the 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming anguages. Portland, Oregon, United States: ACM, 1994: 25-34

[11] Agrawal H. Efficient coverage testing using global dominator graphs[C]// Proceedings of the 1999 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering. Toulouse, France: ACM, 1999: 11-20

[12] Marré M, Bertolino A. Using Spanning Sets for Coverage Testing[J]. IEEE Trans. Softw. Eng, 2003, 29(11): 974-984

[13] Tallam S, Gupta N. A concept analysis inspired greedy algorithm for test suite minimization[C]// Proceedings of the 6th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering. Lisbon, Portugal: ACM, 2005: 35-42