面向医疗数据发布的动态更新隐私保护算法

陈虹云 王杰华 胡兆鹏 贾 露 喻纪文
(南通大学计算机科学与技术学院 江苏 南通 226019)

摘 要 随着信息技术的发展,医疗数据发布中的隐私保护技术一直是数据隐私研究的热点,医疗数据发布的同步更新是其中一个重要问题。为解决医疗数据匿名发布的同步问题,提出了一种建立在(α,k)-匿名数据基础上的支持数据动态更新的算法——(α,k)-UPDATE。该算法通过对语义贴近度的计算,在(α,k)-匿名数据集中选择最贴近的等价类,再进行相应的更新操作。更新后的匿名数据集满足(α,k)-匿名约束,可有效地保护患者的隐私信息。实验结果表明,该算法能在实际环境中稳定、有效地运行,在满足医疗数据实时一致性的同时,具有运算时间短、信息损失度小的优点。

关键词 隐私保护,数据发布,语义贴近度,(α,k)-匿名,动态更新

中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.01.032

Privacy Preserving Algorithm Based on Dynamic Update in Medical Data Publishing

CHEN Hong-yun WANG Jie-hua HU Zhao-peng JIA Lu YU Ji-wen
(College of Computer Science and Technology, Nantong University, Nantong, Jiangsu 226019, China)

Abstract With the development of information technology, the privacy protection technology in medical data publishing has always been a hotspot in data privacy research. One of the important issues is the synchronous update of medical data publishing. To solve the synchronization problem of medical data's anonymous publication, an algorithm based on (α, k) anonymous dataset to support dynamic update of data was proposed, i.e., (α,k)-UPDATE. By calculating the semantic closeness, the algorithm is able to select the most similar equivalent class in (α,k)-anonymous dataset. Then the corresponding update operation is processed. The final dynamically updated dataset can satisfy (α,k)-anonymous and protect the patient's privacy information effectively. The experimental results show that the algorithm can run stably and effectively in real environment, satisfies the real-time consistency of medical dataset and has the advantage of shorter operating time and less information loss.

Keywords Privacy preserving, Data publishing, Semantic closeness, (α,k)-anonymous, Dynamic update

1 引言

随着信息化技术的飞速发展,医院中的大量纸质化信息已逐步被电子化信息存储所替代[1],大量的数据被动态地存储在网络上。为了满足医疗信息共享和医学研究的需求,数据采集者需将已采集的数据信息进行整理并发布。但采集到的信息往往涉及到患者的个人隐私问题,如果不加以处理而直接进行发布,将会造成大量隐私信息的泄漏[2]。只有在数据发布前对其进行隐私保护处理[3-4],才能切实保证患者隐私信息的安全。在现实生活中,医疗机构的信息是不断更新的,仅进行一次匿名发布不能满足医学研究中动态更新集与匿名数据集同步统一的需求。目前,隐私匿名保护的研究主要集中于对静态数据的研究,而对动态数据的研究还处于起步阶段。

在静态数据隐私保护方面,已经有很多研究成果。k-anonymity匿名模型[5]是最早被提出的一种针对关系型数据的隐私保护模型。随后,Terrovitis等[6]提出了(k,m)-anonymity隐私模型,该模型通过泛化层次树来对数据进行泛化处理以实现隐私保护。Wong等[7]提出了(α,k)-匿名模型,其为每个等价类的敏感值设置了统一的频率约束,要求每个等价类的任意一个敏感属性值出现的频率不大于α,满足了数据的多样性要求。但上述匿名模型都只适用于静态数据集的匿名发布。

在动态数据隐私保护方面,Byun等[8]最先对持续增长数据集的发布做了相应的研究,对少量的数据先不进行发布,当

到稿日期:2017-12-19 返修日期:2018-03-07 本文受国家自然科学基金项目(61170171),江苏高校优势学科建设工程资助项目,江苏省"六大人才高峰"项目(2010-WLW-006),南通市应用基础研究计划项目(GY12016015,MS12016048)资助。

陈虹云(1993-),女,硕士生,主要研究方向为信息安全;**王杰华**(1965-),男,硕士,教授,主要研究方向为信息安全、数字水印,E-mail:wang.jh@ntu.edu.cn(通信作者);**胡兆鹏**(1994-),男,硕士生,主要研究方向为网络安全、机器学习;**贾 露**(1993-),女,硕士生,主要研究方向为数据挖掘;**喻纪文**(1992-),男,硕士生,主要研究方向为数据挖掘、机器学习。

满足一定数量之后再进行发布，但是该方法存在延迟时间不确定、数据更新不及时的问题。石秀金等[9]提出了一种基于分类树的差分隐私保护下的动态集值型数据发布的算法。武毅等[10]结合局部重编码泛化和隐匿技术，提出了一种面向动态集值属性数据重发布的隐私保护模型，但其在重发布中引入了相对较多的伪记录，数据失真度较高。鉴于此，本文引入语义贴近度的思想，在(α,k)-匿名模型的基础上提出了一种针对医疗数据发布的支持数据动态更新的算法。通过实验验证，该算法既可以保持数据的多样性，又可以有效地解决数据匿名发布不及时和发布的数据质量不高的问题。同时，该算法应用于医疗数据的匿名发布时，能保持医疗数据发布信息的同步性，从而提高医学研究的有效性。

2　相关概念

2.1　数据匿名

隐私保护通常是在数据发布或共享之前，采用数据抑制、数据泛化和数据隐匿等技术[11-16]对数据集中的相关属性进行处理，使个人的标志信息与敏感数据失去关联，从而达到保护隐私的目的。

定义 1(表的属性)　数据表的属性可以分为 3 类：标识符属性、准标识符属性和敏感属性。

1)标识符属性(Identifier)：能够唯一标识个体身份的属性，如姓名、身份证号等。

2)准标识符属性(Quasi-Identifier Attribute)：该类属性通过与外部信息相结合，可以唯一识别个体的属性。

3)敏感属性(Sensitive Attribute)：包含个体隐私信息的属性，如收入、疾病等。

假设表 1 为某医院的原始病历记录，其中标识符属性有$\{Name\}$，准标识符属性有$\{Age,Zipcode\}$，敏感属性有$\{Disease\}$。

表 1　医院病历记录

Table 1　Hospital medical records

ID	*Name*	*Age*	*Zipcode*	*Disease*
1	John	28	100100	Flu
2	Amy	32	100230	Gastritis
3	Aaron	56	100300	Flu
4	Bob	42	100220	HIV
5	Eric	27	100119	HIV
6	Helen	37	100200	Flu
7	Jim	55	100310	HIV
8	Lily	48	100220	Gastritis
9	Kitty	43	100220	Flu
10	Tom	54	100320	Gastritis

定义 2(泛化)　泛化是指对原始数据集中的准标识符属性进行修改，使用不确定的范围来代替原始数据集中具体的数据。

如图 1 所示，在 *Zipcode* 属性上的泛化操作为：第一条元组的 *Zipcode* 属性“100100”经泛化之后变为“1001＊＊”，泛化层次为 1；再次泛化后变为“100＊＊＊”，泛化层次为 2。第二条元组的 *Zipcode* 属性“100230”经泛化之后变为“1002＊＊”，泛化层次为 1；再次泛化后变为“100＊＊＊”，泛化层次为 2。

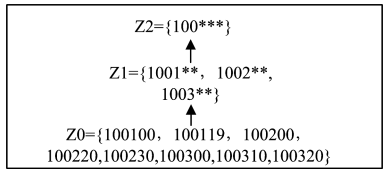

图 1　属性泛化

Fig. 1　Attribute generalization

设原数据集为 T，泛化后的数据集为 T'，原数据表中的元组为 t，泛化后的元组为 t'。对于任意元组都存在 $t\in T$，$t'\in T'$，满足 $t[A_i]\rightarrow t'[A_i]$ $(1\leqslant i\leqslant n)$，同时 $t[S]=t'[S]$。$t[A_i]$指元组 t 中的准标识符属性，$t[S]$指元组 t 中的敏感属性，$t[A_i]\rightarrow t'[A_i]$指对 $t[A_i]$的泛化操作，泛化后敏感属性值不变。

如表 1 所列，元组 1{John，28，100100，Flu}进行泛化操作之后变为{[20，29]，1001＊＊，Flu}。其中，“John”为标识符属性，直接进行隐匿操作；“Flu”为敏感属性，保持不变。

定义 3(等价类)　将数据表 T 划分为若干个数据集，$E=\{e_1,e_2,\cdots,e_n\}$，$|e_i|\geqslant k(i=1,2,\cdots,n)$，$|e_i|$代表数据集 e_i 的大小，数据表 T 划分的若干个数据集均满足条件：数据集 e_i 中每条记录在准标识符属性数据集上具有相同的值。

如表 1 所列，元组 1 与元组 5 在泛化操作之后，准标识符属性数据集均为{[20，29]，1001＊＊}，则在泛化后的匿名数据表中的元组 1 与元组 5 属于同一等价类。

定义 4(k-匿名)　给定匿名数据表 T 和对应的属性集合 $T=\{A_1,A_2,\cdots,A_n,S\}$，满足 k-匿名的数据集 T 中任意一条记录与该数据集中至少 $k-1(k\geqslant 2)$ 条记录在准标识符属性数据集 $A=\{A_1,A_2,\cdots,A_n\}$上具有相同的值。如表 2 所列，$k=2$，$k-1=1$，即每个等价类中至少都有 1 条记录。

表 2　(0.5，2)-匿名数据表

Table 2　Table of (0.5，2)-anonymity data

GID	*ID*	*Age*	*Zipcode*	*Disease*
1	1	[20，29]	1001＊＊	HIV
	2	[20，29]	1001＊＊	Flu
2	3	[30，49]	1002＊＊	Gastritis
	4	[30，49]	1002＊＊	HIV
	5	[30，49]	1002＊＊	Gastritis
	6	[30，49]	1002＊＊	Flu
	7	[30，49]	1002＊＊	Flu
3	8	[50，59]	1003＊＊	HIV
	9	[50，59]	1003＊＊	Flu
	10	[50，59]	1003＊＊	Gastritis

定义 5((α,k)-匿名)　给定匿名数据表 T 和对应的属性集合 $T=\{A_1,A_2,\cdots,A_n,S\}$、准标识符属性集 $A=\{A_1,A_2,\cdots,A_n\}$、敏感属性 S、预先给定的阈值 α，设$|e,ds|$是某一等价类 e 中所有含敏感属性 ds 的元组个数，$|e|$是该等价类中元组的总个数，如果匿名数据表 T 满足关系$|(e,ds)|\leqslant|e|\times\alpha$，那么该数据表 T 满足(α,k)-匿名。

如表 2 所列，在等价类 2 中，流感 Flu 中 $\alpha=0.5$，$k=2$；$|e|=5$，$|e|\times\alpha=2.5$，$|e,ds|=2\leqslant 2.5$，满足(0.5，2)-匿名约束。

2.2　语义贴近度

定义 6(元组与等价类的语义贴近度)　设原始数据集为

T,更新后的数据集为 T',t 为原数据集中的元组,t' 为 t 泛化后对应的元组,将元组 t 与等价类 e_j 的语义贴近度记为 $SED(t,e_j)$。

$$SED(t,e_j)=\sum_{i=1}^{n}SED(A_i) \tag{1}$$

$$SED(A_i)=\begin{cases}1, & a\in[b,c]\\ \dfrac{(c-b)/2}{|a-(b+c)/2|}, & a\notin[b,c]\end{cases} \tag{2}$$

其中,n 代表等价类 e_j 中数值型属性的个数,$t[A_i]=a$,e_j 中 A_i 的值在区间$[b,c]$内。$SED(t,e_j)$用于计算数值型属性元组与等价类的语义贴近度。如果 $SED(t,e_j)=n$,那么 $SED(A_i)=1$,$t[A_i]\in t'[A_i](1\leqslant i\leqslant n)$,其中 t' 为 e_j 中的任意元组。

例 1 计算表 1 中元组 $t=\{28,100100,\text{Flu}\}$与表 2 中等价类 1 的语义贴近度:$Age=28$ 在$[20,29]$区间上,所以 $SED(Age)=1$,$Zipcode=100100$ 也在区间$[100100,100199]$上,所以 $SED(Zipcode)=1$,则元组 t 与表 2 中等价类 1 的语义贴近度为 $SED(t,GID_1)=1+1=2$。

例 2 计算表 1 中元组 $t=\{28,100100,\text{Flu}\}$与表 2 中等价类 2 的语义贴近度:$Age=28$,不在$[30,49]$区间上,则 $SED(Age)=[(49-30)/2]/[28-(30+49)/2]=0.82$,$Zipcode=100100$ 不在区间$[100200,100299]$上,则 $SED(Zipcode)=[(100299-100200)/2]/[100100-(100200+100299)/2]=0.331$,则元组 t 与等价类 2 的语义贴近度为 $SED(t,GID_2)=0.826+0.331=1.157$,所以元组 t 与等价类 1 的语义贴近度比等价类 2 的语义贴近度更大,因此最适合的等价类为等价类 1。

定义 7(等价类与等价类的语义贴近度) 设(α,k)-匿名数据集为 T',将等价类与等价类的语义贴近度记为 $SED(e_l,e_j)$:

$$SED(e_l,e_j)=\sum_{i=1}^{n}SED(A_i) \tag{3}$$

$$SED(A_i)=\frac{1}{|(b_1+c_1)/2-(b_2+c_2)/2|} \tag{4}$$

其中,等价类 GID_l 的 A_i 在区间$[b_1,c_1]$上,等价类 GID_j 的 A_i 在区间$[b_2,c_2]$上。

例 3 如表 2 所列,计算等价类 1 和等价类 3 的语义贴近度:$SED(Age)=1/[(20+29)/2-(50+59)/2]=0.033$,$SED(Zipcode)=1/[(100100+100199)/2-(100300+100399)/2]=0.005$,$SED(e_1,e_3)=0.033+0.005=0.038$。

2.3 信息损失

定义 8(信息损失) 设数据集 $T=\{A_1,A_2,\cdots,A_n,S\}$,准标识符属性集 $A=\{A_1,A_2,\cdots,A_n\}$,元组 t 在属性 A_i 的值域为$[b,c]$,元组 t 在泛化后的值域变为$[b_i,c_i](1\leqslant i\leqslant n)$,元组 t 的信息损失记为 $InforLoss(t)$。

$$InforLoss(A_i)=(c_i-b_i)/|A_{\max}-A_{\min}| \tag{5}$$

$$InforLoss(t)=\sum_{1}^{n}InforLoss(A_i) \tag{6}$$

信息损失为元组在该 A_i 属性上泛化后$[b_i,c_i]$的长度与 A_i 属性的值域$[b,c]$的比值,元组 t 的信息损失为准标识符属性集 A 上所有准标识符属性的信息损失总和。

3 基于动态更新的(α,k)-UPDATE 算法

(α,k)-匿名模型为每个等价类的敏感值设置了统一的频率约束,满足了数据多样性的要求。但是在现实生活中,医疗机构的信息是不断更新的,(α,k)-匿名模型不能满足医学研究中动态更新集与匿名数据集同步统一的需求。针对这一问题,本文引入语义贴近度的思想,在(α,k)-匿名模型的基础上提出了一种针对医疗数据发布的支持数据动态更新的算法,称为(α,k)-UPDATE 算法。

(α,k)-UPDATE 算法是在医疗数据发布中对已经发布的(α,k)-匿名数据集进行更新操作的算法,它能满足匿名数据集发布的及时性和同步性的要求。(α,k)-UPDATE 算法包含 3 个模块:Insert 算法、Delete 算法和 Modify 算法。图 2 为(α,k)-UPDATE 算法的框图,该算法首先计算语义贴近度,然后找到最合适的等价类,再进行相应的更新操作,最后检验是否满足(α,k)-匿名约束。

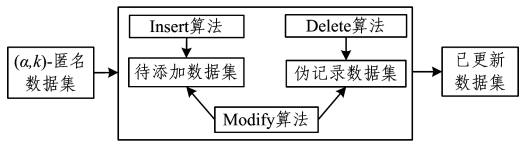

图 2 (α,k)-UPDATE 算法的框图

Fig. 2 Model of (α,k)-UPDATE algorithm

3.1 Insert 算法

Insert 算法向(α,k)-匿名数据集 T^* 中添加数据集 T。首先确定阈值 α,再计算元组 t 与等价类的语义贴近度,选择语义贴近度最大的等价类;然后判断最大语义贴近度 $sedmax$ 是否与准标识符属性个数 q 相同,再进行相应的添加操作;最后判断匿名数据集是否满足(α,k)-匿名约束。若满足约束,则添加成功;若不满足约束,则需要添加伪记录来满足约束。

算法 1 Insert 算法

输入:(α,k)-匿名数据集 T′,其中 T′有 n 个等价类;待添加数据集 T,其中有 m 个元组

输出:更新后的(α,k)-匿名数据集 TT*

```
1. For each t∈T{
2.   For i=1 to m
3.     { For j=1 to n;
4.       sed=SED(t,ej);
5.     If (sed>sedmax) Then
6.       { sedmax=sed;e=ej;}
7.     If (sedmax=q) Then
8.       { tt′[A]=t′[A];tt′[S]=t[S];TT′=T′∪{tt′};}//直接添
         加元组
9.     Else { tt′[A]=等价类与 t 相对应的属性共同泛化值
10.          tt′[S]=t[S];TT′=T′∪{tt′};}
11.      If (不满足(α,k)-匿名约束) Then
12.      {添加伪记录;}}}
13.输出 TT*
14. END
```

步骤 3—步骤 6,计算元组与等价类的语义贴近度并找出语义贴近度最大的等价类;步骤 7—步骤 8,若添加的元组 t 与等价类的最大语义贴近度等于 q(q 为准标识符属性个数),

则可以直接添加泛化元组；步骤 9－步骤 10，若最大语义贴近度不等于 q，则添加元组的准标识符属性等于等价类与 t 相对应的属性的共同泛化值，敏感属性值不变；步骤 11－步骤 12，进行(α,k)-匿名约束判断，之后进行调整。

例 4　将元组 t＝{Sandy，36，100229，HIV}添加到表 2 的匿名数据表后再进行数据发布。首先计算出元组 t 与等价类 1 的语义贴近度为 1.014，元组 t 与等价类 2 的语义贴近度为 2，元组 t 与等价类 3 的语义贴近度为 0.654，则可以比较最适合的等价类是等价类 2；再将元组 t 泛化为{[30，49]，100＊＊，HIV}，并加入等价类 2 中；最后判断更新后的等价类 2，$|e|\times\alpha=3$，$|e,ds|=2\leqslant 3$，满足(0.5，2)-匿名约束，成功完成添加操作。

3.2　Delete 算法

从(α,k)-匿名数据集 T^* 中删除数据。先确定阈值 α，再根据删除条件 φ 以及元组和泛化元组的映射关系确定所需删除元组在(α,k)-匿名数据集 T^* 中所在的等价类；然后进行删除操作，然后判断每个等价类的元组数是否小于 k，若等价类元组数小于 k，则选择语义贴近度最大的等价类进行合并；最后判断是否满足(α,k)-匿名约束，再进行调整。

算法 2　Delete 算法

输入：原始数据集 T，(α，k)-匿名数据集 T′，删除条件 φ
输出：更新后的(α，k)-匿名数据集 TT′
1. 变量初始化；
2. If（删除条件 φ 只有敏感属性 S）Then
3.　｛TT′＝T′－直接删除满足条件的元组；｝
4. Else｛T－原始数据集 T 中满足删除条件 φ 的所有元组；｝
5. For each e_i，$e_j\in T'$
6.　｛If（等价类中的元组个数＜ k）Then
7.　｛找出与 e_i 语义贴近度最大的等价类 e_j；
8.　将等价类 e_i 和等价类 e_j 进行泛化合并，形成新的等价类；｝
9.　｝
10. If（不满足(α，k)-约束条件）Then
11. ｛添加伪记录；｝
12. 输出 TT＊
13. END

步骤 2－步骤 3，若删除条件 φ 只包含敏感属性 S，则可以直接进行删除操作。步骤 4，若删除条件 φ 包含准标识符属性，则需要删除满足删除条件 φ 的所有元组。步骤 5－步骤 9，若删除数据之后等价类元组的个数少于 k，则选择与该等价类语义贴近度最大的等价类进行泛化合并。步骤 10－步骤 11，进行(α,k)-匿名约束判断，最后进行调整。

例 5　删除年龄小于 25 且患有 Flu 的病患信息。先从原始数据表 1 中找出需要删除的元组 t＝{John，28，100100，Flu}，泛化后 t'＝{[20，29]，1001＊＊，Flu}。再从表 2 中找出并删除元组 2，等价类 1 中只剩下一条记录 t_1＝{[20，29]，1001＊＊，HIV}，记录个数少于 2。之后寻找最匹配的等价类进行合并，通过计算得出 $SED(e_1,e_2)=0.077$，$SED(e_1,e_3)=0.038$，则等价类 1 需要与等价类 2 进行共同泛化后合并，再次判断得出其满足(0.5，2)-匿名约束，则删除操作成功。

3.3　Modify 算法

对匿名数据集的修改操作包含以下情况：1）当修改条件中仅包含敏感属性时，则根据语义贴近度找出相应的等价类，直接进行修改；2）当修改条件中包含准标识符属性时，则可以将修改操作拆分为删除操作和插入操作，可以先进行 Delete，再进行 Insert。下面是 Modify 算法的实现过程：

输入：原始数据集 T，(α，k)-匿名数据集 T′，修改条件 φ
输出：更新后的(α，k)-匿名数据集 TT′
1. If（φ 只包含敏感属性 S）{
2.　Then 根据 φ 修改 T′中的元组，TT′＝T′；
3. If（满足(α，k)-约束）修改成功；
4. Else　｛添加伪记录；｝
5. ｝
6. Else
7. ｛从 T 中找出满足 φ 的元组；
8. 执行 Delete 算法，从 T′中删除原泛化元组；
9. 修改对应的元组；
10. 执行 Insert 算法，将其插入到 T′中；
11. TT′＝T′｝
12. 输出 TT＊
13. END

例 6　患者 John 病情恶化，病情由流感“Flu”转化为了肺癌“Lung-cancer”。为保证医疗数据的同步性，需及时对其病情记录进行修改。由于只需要更改敏感属性，因此可以直接找出该元组并对匿名数据表进行修改，将属性“Flu”更改为“Lung-cancer”，对应的匿名数据表中的元组变为{[20，29]，1001＊＊，Lung-cancer}。最后计算出更新后的等价类 1 满足(0.5，2)匿名约束，则修改成功。

例 7　如表 2 所列，患者 John 邮编发生改变，由“100100”变为“100113”，修改前后所对应的泛化元组不同，所在的等价类也不同。因此需要先找出 John 所在元组为{John，28，100100，Flu}，执行 Delete 算法，从 T' 中删除原泛化元组；再将其修改为[John，28，100113，Flu]，之后执行 Insert 算法来将修改后泛化的元组插入匿名数据集中。由于 Insert 算法中已判断出等价类 1 满足(0.5，2)匿名约束，则修改成功。

4　实验分析

4.1　实验数据及环境

实验采用 UCI 的人口统计实际数据集中的 Adult 数据集来模拟实际生活中不断更新的医疗数据集。选取 9 个属性进行实验，其中包括 3 个数值型属性和 5 个分类型属性（用对应的数值代替），将 Occupation 作为敏感属性。目前常用的算法是 SECURE 算法[8]和 DSR 算法[10]，由于 SECURE 算法只针对数据增量，因此本文是在数据增量上进行实验对比的，本文数据删除和修改的部分仅提供设计和理论分析。分析数据采用 20 次实验的平均值，以避免实验数据的偶然性。

4.2　算法的具体实现

本文的(α,k)-UPDATE 算法共包含 3 个模块：Insert 算法、Delete 算法和 Modify 算法。其中，最关键的模块是 Insert 算法和 Delete 算法，而 Modify 算法中的修改操作可以拆分为删除操作和插入操作，限于篇幅，本文不进行详细阐述。

1）Insert 算法实现的关键步骤如下所示。

```
public static int Insert(float alpha,int k,T2 t2,T1 t1)
```

```
{//待添加数据集 T1,更新前的匿名数据集 T2
  float sedMax=0.0f;
  int sedMax_gid=0;
  List<int> list_T2=List_Group(t2);
  for (int i=0; i < t1.dataset.Count;++ i)
  {
    T1_eachData t1_i=t1.dataset[i];
    foreach (int each_gid in list_T2)
    {//找出与第 i 条记录语义贴近度最大的等价类
      float sed=SED(t1_i,t2,each_gid);//计算语义贴近度
      if (sed > sedMax)
      {
        sedMax=sed;
        sedMax_gid=each_gid;
      }
    }
    T2_eachData t2_new=new T2_eachData();
    if (sedMax==(float)q)
    {
      t2_new=add_direct(t1_i,t2,sedMax_gid);
      t2.dataset.Add(t2_new);
    }
    else
    {
      t2_new=generalize(t1_i,t2,sedMax_gid);
      t2.dataset.Add(t2_new);
    }
  }
  if (judge(p,alpha,k,t2)==true)//判断是否满足约束条件
  {
    return 0;//添加成功
  }
  else
  { //添加伪记录
    add_disguise();
    return 0;
  }
}
```

2)Delete 算法实现的关键步骤如下所示。

```
public static int Delete(string condition,T0 t0,T2 t2)
{//原始数据集 T0,更新前的匿名数据集 T2
  Boolean is_delete=del_condition(condition);
  if (is_delete)//判断删除条件中是否只含敏感属性
  {//直接从 t2 中删除只含敏感属性的元组
    delete_direct(condition,t2);
  }
  else
  {//从 t0 中找到需删除的元组进行泛化,然后从 t2 中删除
    delete_indirect(condition,t0,t2);
  }
  List<int> t2_gidList=List_Group(t2);
  float sedmax=0.0f;
  int sedmax_gid=0;
  foreach (int gid in t2_gidList)
  {
    int gid_count=count_byGroupId(t2,gid);
    if (gid_count < k)//判断等价类元组数是否小于 k
    {
      for (int i=0; i < t2_gidList.Count; ++i)
      {//找出语义贴近度最大的等价类
        sedmax=0.0f;
        sedmax_gid=1;
        if (SED_GID(t2,gid,i) > sedmax)
        {
          sedmax=SED_GID(t2,gid,t2_gidList[i]);
          sedmax_gid=t2_gidList[i];
        }
      }
      combine(t2,gid,sedmax_gid);//选择语义贴近度最大的等价类合并
    }
  }
  if (judge(p,alpha,k,t2)==true)//判断是否满足约束条件
  {
    return 0;//删除成功
  }
  else
  { //添加伪记录
    add_disguise();
    return 0;
  }
}
```

4.3 运行时间分析

本文方法的第一部分 Insert 算法通过计算待添加元组与等价类的语义贴近度,可以快速选取最适合的等价类进行插入操作,避免了将待添加元组与每个等价类进行匹配的繁琐操作,节约了大量的运行时间。第二部分 Delete 算法在删除符合条件的元组之后将小于 k 的等价类通过元组与元组间的语义贴近度进行匹配,能够快速进行合并更新,节约了大量的时间。第三部分 Modify 算法,当修改条件中仅包含敏感属性时,可以直接修改敏感属性;当修改条件中包含准标识符属性时,类似地,可以通过计算语义贴近度来快速完成数据的匿名重发布。

本文算法选取 5 个准标识符属性和 1 个敏感属性,在 $\alpha=0.3$ 的情况下来进行实验。从图 3 可以看出,随着待更新数据量的增加,运行时间也在不断增加。相同的实验条件下,DSR 算法为了维持敏感元素在更新过程中的多样性和连续性,需要更大的时间开销。(α,k)-UPDATE 算法和 SECURE 算法均采用增量插入方法,使增量数据集远小于原始数据集,消耗时间短。SECURE 算法需对部分元组进行延迟和隐匿操作,而(α,k)-UPDATE 算法每次仅需针对待添加元组进行处理,对原始数据的修改很少,因此(α,k)-UPDATE 算法的耗时最少。

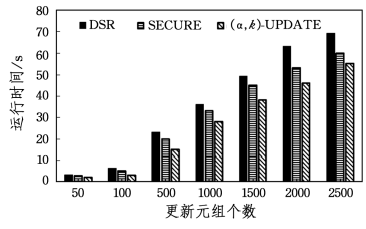

图 3 运行时间对比

Fig. 3 Comparison of execution time

4.4 信息损失的分析

如图 4 所示，在参数准标识符属性的个数不相同的情况下，数据的信息损失不断变化。随着等价类中准标识符个数的增加，只有泛化到更高层次，才能使所有元组都具有相同的准标识符属性。因为泛化层次越高，信息损失就越大，所以本文使用标识符属性的个数作为隐私泄漏的评价数据。随着准标识符属性个数的增加，3 种算法的信息损失都增加，但增长幅度各不相同。相同实验条件下，SECURE 算法的信息损失最大，DSR 算法次之，本文的(α,k)-UPDATE 算法的信息损失最少。SECURE 算法需对部分元组进行延迟隐匿操作，信息损失量偏多。DSR 算法只需进行局部重编码，数据质量较高。(α,k)-UPDATE 算法在进行更新操作时，只需选取语义贴近度最大的元组进行更新操作，所以信息损失度最小，数据质量最高。

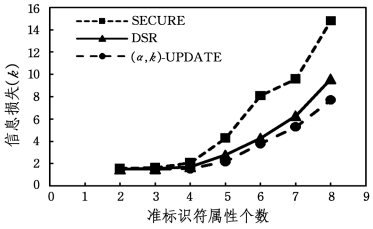

图 4　信息损失对比图

Fig. 4　Comparison of information loss

结束语　本文引入语义贴近度的思想，在(α,k)静态数据匿名发布模型的基础上进行修改，提出并实现了一种针对医疗数据发布的支持数据动态发布的算法，解决了医疗匿名数据重发布的问题。该算法通过计算元组与等价类间的语义贴近度和等价类与等价类之间的语义贴近度，在匿名数据集中快速定位，选择最适合的等价类进行更新操作，使得更新后的匿名数据集满足(α,k)-匿名约束，既减少了运行时间，又满足了数据的多样性。

实验结果表明，该算法具有运行时间较短、信息损失小、数据质量高的优点，在满足医疗数据动态发布高效性的同时，保证了医疗数据发布的质量，有效降低了实际运算开销。在接下来的工作中，将进一步对多敏感属性数据进行研究和分析，进而完善本文所提算法，使其能更好地应用于医学研究。

参 考 文 献

[1] GARTRELL K, TRINKOFF A M, STORR C L, et al. Electronic Personal Health Record Use Among Nurses in the Nursing Informatics Community [J]. Computers Informatics Nursing, 2015, 33(7): 306-314.

[2] WU D. Research on Patient Privacy Protection for Medical Data in Cloud Computing[J]. Journal of Networks, 2013, 8(11): 2678-2684.

[3] ZHANG X J, MENG X F. Differential Privacy in Data Publication and Analysis[J]. Journal of Computers, 2014, 37(4): 101-122. (in Chinese)
张啸剑，孟晓峰. 面向数据发布和分析的差分隐私保护[J]. 计算机学报，2014，37(4)：101-122.

[4] ZHOU S G, LI F, TAO Y F, et al. Privacy Preservation in Database Application: A Survey[J]. Journal of Computers, 2009, 32(5): 847-861. (in Chinese)
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)：847-861.

[5] SWEENY L. k-anonymity: A model for Protecting Privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557-570.

[6] TERROVITIS M, MAMOULIS N, KALNIS P. Privacy-preserving anonymization of set-valued data[J]. Proceedings of the Vldb Endowment, 2008, 1(1): 115-125.

[7] WONG C W, LI J, FU W C, et al. (α,k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing[C]// Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006: 754-759.

[8] BYUN J W, SOHN Y, BERTINO E, et al. Secure Anonymization for Incremental Datasets[M]. Secure Data Management. Berlin: Springer, 2006: 48-63.

[9] SHI X J, HU Y L. Privacy Preserving Based on Taxonomy Tree for Dynamic Set-valued Data Publishing[J]. Computer Science, 2017, 44(5): 120-124. (in Chinese)
石秀金，胡艳玲. 基于分类树的动态集值型数据发布的隐私保护[J]. 计算机科学，2017，44(5)：120-124.

[10] WU Y, WANG D, JIANG Z L. Privacy Preserving in Re-Publication of Dynamic Set-Valued Data Based on Transactional K-Anonymity[J]. Journal of Computer Research and Development, 2013, 50(S1): 248-256. (in Chinese)
武毅，王丹，蒋宗礼. 基于事务型 K-Anonymity 的动态集值属性数据重发布隐私保护方法[J]. 计算机研究与发展，2013，50(S1)：248-256.

[11] WANG Z H, XU J, WANG W, et al. Clustering-Based Approach for Data Anonymization[J]. Journal of Software, 2010, 21(4): 680-693. (in Chinese)
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.

[12] ABAWAJY J H, NINGGAL M I H, HERAWAN T. Privacy Preserving Social Network Data Publication[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3): 1974-1997.

[13] CHEN B C, KIFER D, LEFEVRE K, et al. Privacy-Preserving Data Publishing[J]. Acm Computing Surveys, 2009, 2(1-2): 1-167.

[14] XIAO X K, TAO Y F. Personalized privacy Preservation[C]// Proceedings of the 2006 AcmSigmod International Conference on Management of Data. Chicago, Illinois, USA: ACMPress, 2006: 229-240.

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

[16] TAKENOUCHI T, KAWAMURA T, OHSUGA A. Distributed Anonymization Method with Hiding the Presence of Individuals[J]. IEICE Transactions on Information & Systems, 2013, 96(3): 596-610.