

吉林大学申请博士学位论文自我评价表

学 科 专 业	计算机应用技术
研 究 方 向	隐私保护数据发布技术
作者自述申请博士学位论文的主要创新（学位申请人填写，限填 5 项）：	
<p>(1) 提出了交叉桶泛化算法。该算法结合泛化算法和桶算法的原理分别对用户身份和敏感属性进行相互独立的保护，从而解决了当使用泛化算法时对用户身份过度保护的问题。</p> <p>(2) 定义了个性化隐私保护的发布环境并提出了局部分解算法。在个性化隐私保护的发布环境中，用户可以在数据表中自由设置自身属性值的敏感性，并通过使用局部分解算法对所有的敏感值进行保护。</p> <p>(3) 提出了局部分解泛化算法。该算法通过在局部分解算法中加入泛化机制，使其可以在个性化隐私保护的发布环境中同时为用户身份和敏感值提供保护。</p>	
作者对博士学位论文有待改进之处的自我评述：	
<p>隐私保护数据发布技术的研究核心是在一定的发布环境以及面临一些匿名需求时，可以保障数据中隐私信息的安全并且尽可能提高匿名数据中的信息可利用性。本文提出的匿名保护算法虽然基于一定的发布环境前提，但是仍然可以应用于其他的发布环境中，所以，在未来的工作中我们将把提出的算法向其他发布环境中进行移植。除此之外，虽然传统的泛化算法可以对匿名数据表提供非常有效的隐私保护，但是也严重破坏了匿名数据表中的信息可利用性。因此，我们将研究一种新的匿名机制可以在一定的发布环境中取代泛化算法，使匿名数据表既可以得到安全的保护又可以保留更多的信息可利用性。</p>	

在学期间完成的与申请博士学位论文相关的科研成果（含获奖、专利）			
序号	署名排序	题名、刊物名称（获奖名称、专利号）、卷期号、起讫页码、发表时间	对应博士论文章节
1	第一作者	<p>题名 : Cross-Bucket Generalization for Information and Privacy Preservation</p> <p>刊物名称: IEEE Transactions on Knowledge and Data Engineering(SCI 检索, 1 区, CCF A 类)</p> <p>卷期号: 30</p> <p>起讫页码: 449-459</p> <p>发表时间: 2018.3</p>	第三章

注: 此表格应同“吉林大学博士学位论文”放在一起并位于首页, 以备专家评审时使用。

分类号: TP391
研究生学号:

单位代码: 10183
密 级: 公开

基于不同匿名需求的隐私保护数据发布算法研究



吉林大学

博士学位论文

基于不同匿名需求的隐私保护数据发布算法研究

Research on Privacy-Preserving Data Publishing Algorithms
Based on Different Anonymity Requests

作者姓名:

专 业: 计算机应用技术

研究方向: 隐私保护数据发布技术

指导教师:

培养单位: 计算机科学与技术学院

2018 年 月

吉林
大学

基于不同匿名需求的隐私保护数据发布算法研究

Research on Privacy-Preserving Data Publishing
Algorithms Based on Different Anonymity Requests

作者姓名:

专业名称: 计算机应用技术

指导教师:

学位类别: 工学博士

论文答辩日期: 2018 年 月 日

授予学位日期: 2018 年 月 日

答辩委员会主席:

论 文 评 阅 人:

**未经本论文作者的书面授权，依法收存和保管本论文学
面版本、电子版本的任何单位和个人，均不得对本论文的全
部或部分内容进行任何形式的复制、修改、发行、出租、改
编等有碍作者著作权的商业性使用（但纯学术性使用不在此
限）。否则，应承担侵权的法律责任。**

吉林大学博士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，
独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本
论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本
文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。
本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 2018 年 月 日

《中国优秀博硕士学位论文全文数据库》投稿声明

研究生院：

本人同意《中国优秀博硕士学位论文全文数据库》出版章程的内容，愿意将本人的学位论文委托研究生院向中国学术期刊（光盘版）电子杂志社的《中国优秀博硕士学位论文全文数据库》投稿，希望《中国优秀博硕士学位论文全文数据库》给予出版，并同意在《中国博硕士学位论文评价数据库》和 CNKI 系列数据库中使用，同意按章程规定享受相关权益。

论文级别： 硕士 博士

学科专业： 计算机应用技术

论文题目： 基于不同匿名需求的隐私保护数据发布算法研究

作者签名： 指导教师签名：

2018 年 月 日

作者联系地址（邮编）：

作者联系电话：

摘要

基于不同匿名需求的隐私保护数据发布算法研究

随着大数据和机器学习等技术的不断进步，各个行业对数据的需求量越来越大。行业之间的数据交换和共享逐渐成为了信息交流中越来越重要的活动，但是，在交流的数据中包含了大量的用户隐私信息。如果这些数据在没有经过隐私保护处理就对外进行发布或者交换，会非常容易造成用户的隐私泄露。因此，学者们通过提出隐私保护数据发布技术解决在数据发布和交换过程中用户的隐私泄露问题。本文主要研究了在特定的发布环境中面对一些匿名保护的需求时，提出适当的匿名算法为数据表中的隐私信息提供安全的保护并且尽可能保存数据中的信息可利用性，具体的主要工作包括以下三个方面：

1. 提出了交叉桶泛化算法。该算法结合泛化算法和桶算法的原理分别对用户身份和敏感属性进行相互独立的保护，从而解决了当使用泛化算法时对用户身份过度保护的问题。由于交叉桶泛化算法可以为用户身份和敏感属性提供独立的保护，所以我们通过提出并使交叉桶泛化算法遵循 (k, l) -anonymity 匿名原则将匿名数据表中用户身份暴露的概率和敏感属性值泄露的概率分别控制在 $1/k$ 和 $1/l$ 以内，并且参数 k 和 l 可以根据实际匿名需求自由设置。此外，还通过使用启发式将匿名数据表中各个等价组和桶包含的个体数量尽可能减少，并且尽量缩小等价组中 QI 泛化值的值域范围，从而进一步提高了匿名数据的信息可利用性。

2. 定义了个性化隐私保护的发布环境并提出了局部分解算法。在个性化隐私保护的发布环境中，用户可以在数据表中自由设置自身属性值的敏感性，并且根据包含数据值的类型将数据表中的属性分为 QI 属性、半敏感属性和敏感属性。局部分解算法基于桶算法的原理，在每个半敏感属性和敏感属性中将带有敏感值的用户划分为桶，从而在保障所有敏感值安全的同时还保留了所有原始 QI 值信息。局部分解算法不仅可以保留非常优秀的信息可利用性，还具有很好的可扩展性，它可以根据实际匿名需求或者不同属性的特点同时遵循不同的匿名原则对数据表中的敏感值进行保护。

3. 提出了局部分解泛化算法。该算法通过在局部分解算法中加入泛化机制，

使其可以在个性化隐私保护的发布环境中同时为用户身份和敏感值提供保护。局部分解泛化算法为用户身份提供保护的基本思想是根据数据表中个体携带 QI 值的情况将所有个体分为多个子集，并且在每个子集内将个体划分为等价组，从而使整个数据表满足 k -anonymity 匿名原则的条件。我们通过使用多维划分技术和 NCP 引导的启发式具体实现了两种局部分解泛化算法中的泛化机制。由于局部分解泛化算法对用户身份和敏感值的保护是相互独立的，所以使用不同的泛化机制不会降低对敏感值的保护效果。

综上，本文主要研究在一定的匿名需求和发布环境中实现为数据的隐私信息提供保护的方法，包括对用户身份和敏感属性进行独立的保护以及当允许用户自定义敏感值时为敏感值和用户身份提供保护。此外，相比于传统的匿名算法，本文提出的算法尽可能减少了数据在匿名过程中的信息损失，保留了更加优秀的信息可利用性。

关键词：

隐私保护，匿名原则， k -anonymity， l -diversity，匿名算法，泛化算法，桶算法

Abstract

Research on Privacy-Preserving Data Publishing Algorithms Based on Different Anonymity Requests

Along with the development of technologies of big data and machine learning, the demand for data becomes more heavily all over the trades. The data exchange and sharing between trades gradually turn into more and more important behaviors in the communication of information. However, the shared information contains amounts of privacy information of users. If these data are published or shared without any privacy protection, it is easy to bring on the disclosure of privacy information of users. Consequently, the scholars propose the technique of privacy-preserving data publishing to solve the problem of privacy disclosure during the process of data publishing and sharing. This paper mainly researches on presenting the proper algorithms to provide security protection for privacy information and preserve information utility as much as possible in the data when facing some anonymity requests in the certain publishing scenario. The specific contributions contain three parts as follows:

1. We propose a cross-bucket generalization algorithm. Cross-bucket generalization combines generalization and bucketization to separately protect user identity and sensitive attribute that solves the problem of overprotection for identity when using generalization algorithm. As cross-bucket generalization provides independent protections for user identity and sensitive attribute, we present and make cross-bucket generalization comply with (k,l) -anonymity principle to confine the disclosure probabilities of identity and sensitive values in the anonymous data under $1/k$ and $1/l$, respectively, and the parameters k and l can be adjusted according to the actual anonymity requests. In addition, we minimize the size of every equivalence group and bucket and the range of the generalized QI value in each equivalence group as far as possible by using heuristic that further improves the information utility of

anonymous data.

2. We define the publishing scenario of personalized privacy protection and propose a local anatomy algorithm. In the publishing scenario of personalized privacy protection, users can freely set sensibility for their attribute values in the data table, and the attributes contained in data table can be divided into the types of QI attribute, semi-sensitive attribute and sensitive attribute according to the varieties of values. Based on the rationale of bucketization, local anatomy divides the tuples in each semi-sensitive attribute and sensitive attribute into buckets who carries the sensitive value that guarantees all the sensitive values safe and preserves all the original QI values. Local anatomy not only preserves excellent information utility but also has great extendibility. It can satisfy different anonymity principles to protect the sensitive values in the data table at the same time according to the different characters of attributes or actual anonymity requests.

3. We present a local anatomy generalization algorithm. It combines generalization mechanism based on local anatomy algorithm that can provide both protections for user identiy and sensitive value in the publishing scenario of personalized privacy protection. The basic idea of protection of local anatomy generalization for user identity is that divides all the tuples into subsets according to their QI values, and divides the tuples into equivalence groups in each subset so that the whole data table satisfies the condition of k -anonymity priciple. We implement two local anatomy generalization algorithms by using multi-dimension division and the heuristic of NCP guidance to respectively achieve the generalization mechanisms. Since the protections for user identity and sensitive value are separate in the local anatomy generalization algorithm, using different generalization mechanisms does not reduce the protective effect of sensitive value.

In conclusion, this paper mainly researches on implementing the method of protection for privacy information in the certain anonymity request and publishing scenario that includes providing the separate protections for user identity and sensitive attribute and protecting the user identity and sensitive attribute when allows users to customize their sensitive values. Additionally, compared with the previous anonymity

algorithms, the algorithms proposed by this paper reduce the information loss during the anonymity process as far as possible that preserves better information utility.

Keywords:

Privacy Protection, Anonymity Principle, k -anonymity, l -diversity, Anonymity Algorithm, Generalization Algorithm, Bucketization Algorithm

目 录

摘 要.....	I
Abstract.....	III
第 1 章 绪 论	1
1.1 研究背景与意义	1
1.2 基本模型	2
1.3 研究现状	4
1.4 本文工作	5
1.5 论文结构	6
第 2 章 隐私保护数据发布技术概述.....	8
2.1 匿名原则	8
2.1.1 k -anonymity 匿名原则	8
2.1.2 l -diversity 匿名原则	9
2.1.3 差分隐私	11
2.2 匿名算法	11
2.2.1 泛化算法	12
2.2.2 桶算法	13
2.3 信息可利用性	14
2.3.1 信息损失量评估	15
2.3.2 查询分析评估	16
第 3 章 用户身份和敏感属性的独立保护.....	17
3.1 引言	17
3.1.1 问题的提出	17
3.1.2 本章工作	20
3.2 交叉桶泛化算法模型	22
3.2.1 基本概念	22
3.2.2 隐私保护分析	23
3.3 交叉桶泛化算法	26
3.3.1 计算敏感值集合	27
3.3.2 选择匿名个体	28
3.3.3 匿名个体数据	30
3.4 实验分析	33
3.4.1 敏感属性保护	34
3.4.2 信息可利用性	35
3.4.3 参数的影响	37
3.5 本章小结	39
第 4 章 个性化的隐私保护.....	40
4.1 引言	40
4.1.1 问题的提出	40
4.1.2 本章工作	42
4.2 局部分解算法模型	44
4.2.1 基本概念	44

4.2.2 敏感值保护分析	45
4.3 局部分解算法	48
4.4 实验分析	50
4.4.1 敏感值保护	51
4.4.2 信息可利用性	53
4.4.3 敏感值密度的影响	55
4.5 扩展讨论	57
4.6 本章小结	58
第 5 章 局部分解泛化算法	59
5.1 引言	59
5.1.1 问题的提出	59
5.1.2 本章工作	61
5.2 局部分解泛化算法模型	63
5.2.1 基本模型	63
5.2.2 隐私保护原理	64
5.3 局部分解泛化算法	66
5.3.1 基于多维划分技术的局部分解泛化算法	67
5.3.2 基于 NCP 引导的局部分解泛化算法	70
5.4 实验分析	72
5.5 本章小结	85
第 6 章 总结与展望	87
6.1 工作总结	87
6.2 研究展望	87
参考文献	89
作者简介及科研成果	100
致 谢	101

第1章 绪 论

1.1 研究背景与意义

随着人类社会进入大数据的时代，人们的各类信息被政府部门、企业组织甚至个人收集^[1]，如医疗档案、社会调查、人口普查和商业数据等。这些海量的数据信息被用于进行数据挖掘和机器学习等研究^[2]，从而帮助政府制定相关政策或者企业创造商业价值，同时也为人们提供更加丰富、智能和便捷的生活方式。但是，大数据技术是一把锋利的双刃剑，当它在推动社会进步的同时，如果不能对其善加利用或者被恶意使用，则极有可能对人们的日常生活甚至整个社会造成非常巨大的危害^[3]。

不同于将数据的二进制信息进行加密的传统信息安全技术，在大数据时代的背景下，由于数据信息的内容形式以及收集、发布和传播的方式变得多样化，信息安全问题涉及的范围更加广泛，并且研究的内容也更加复杂和多样^[4]。其中，人们的隐私安全问题一直是信息安全问题的研究核心之一^[5-6]。

在大数据技术的不断推动之下，各个行业和政府部门之间的信息壁障逐渐被打破。数据的交换和共享成为了信息交流中越来越重要的活动，但是，这些交流的数据中包含了大量个人隐私和敏感信息。如果这些数据在没有经过隐私保护处理之前就对外进行发布或者交换，会非常容易造成用户的隐私泄露^[7-8]。例如，在2006年，著名的因特网服务提供商美国在线(American Online)发布了一份2GB的数据文件，其中包括了关于65万用户的约2千万个查询词条信息。尽管这份数据使用了随机数作为假名代替数据中用户的ID，但是当数据发布之后，两名纽约时报的记者仍然通过数据中的信息准确地找到并且采访了其中的某位用户^[9]。因此，当需要发布和使用涉及用户隐私信息的数据时，如何有效地保障用户的隐私安全是一个非常重要的问题。

为了保护发布数据中的用户隐私信息，学者们提出了一种被称为隐私保护数据发布(Privacy-Preserving Data Publishing)的技术^[10]。该技术主要通过匿名或者添加噪音的方式对数据表中的信息进行一定程度的加密，将用户隐私暴露的概率降低在一定阈值以内，它可以帮助如社交网站、保险公司、医院和政府部门

门等在对外发布数据时，既不会泄露其中用户的隐私信息，还可以保证匿名数据的信息可利用性。

1.2 基本模型

在 1977 年，Dalenius 提出了一个比较严格的隐私保护定义^[11]：攻击者无法从获得的数据中得到任何额外信息。而在文献[12-13]中指出，由于攻击者可以根据背景知识进行推理的原因，只有在不发布数据的情况下才能实现完美隐私。隐私保护数据发布技术兼顾了隐私安全性和信息可利用性，既可以满足数据中用户的隐私安全需求，又使匿名之后的数据可以满足数据接收者进行分析的需求。

隐私保护数据发布的过程如图 1.1 所示。数据持有者首先收集和存储用户的数据信息，如网站数据、病例信息、薪水调查等，这些数据包含了人们大量的隐私信息。当收到外部机构的数据请求时，数据持有者将需要发布的原始数据进行一定程度的匿名处理，使得匿名之后的数据可以保证用户的隐私信息安全。最后，数据持有者将匿名之后的数据结果对数据接收者发布。

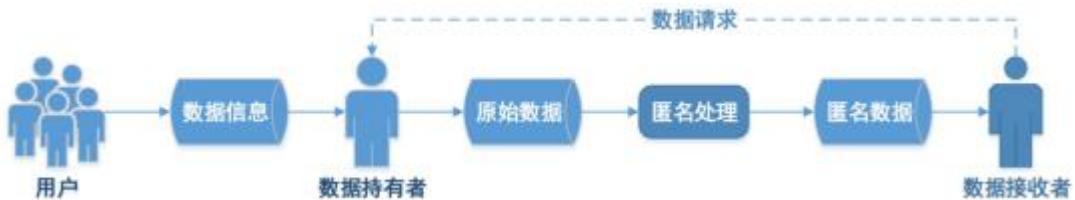


图 1.1 数据匿名及发布的过程

隐私保护数据发布技术研究的匿名对象主要分为两种。**第一种匿名对象为记录型数据**，如表 1.1 所示，当需要将这种类型的数据信息对外发布时，数据发布者通常使用泛化^[14]和**桶**^[15]等匿名算法将数据表中的内容进行匿名之后再对外发布；**第二种匿名对象是数据接收者向统计数据库发送查询语句之后得到的数据结果**。在很多情况下，数据发布者仅向数据接收者开放了数据访问的查询接口，但是，攻击者仍然可以通过使用连续特定的查询语句**攻击用户的隐私信息**。因此，数据发布者还需要使用**如满足差分隐私^[16]的噪音算法**，将数据接收者的查询结果加入一定程度的噪音，从而防止数据中用户的隐私信息泄露。

表 1.1 记录型数据

ID	Name	Age	Gender	Zip Code	Disease
1001	Neil	22	Male	13248	Pneumonia
1002	Mark	22	Male	13241	Dyspepsia
1003	Ella	24	Female	13247	Flu
1004	Sarah	25	Female	13242	Bronchitis
1005	Tina	26	Female	14553	Bronchitis
1006	Dean	34	Male	14423	Dyspepsia
1007	Dave	36	Male	14731	Hepatitis
1008	Daphne	38	Female	14417	Gastritis

一般情况下，记录型数据表包含三种属性类型^[17]：(1) 标识符属性 (Explicit-Identifier)，能够唯一或者在很大程度上识别用户身份的属性，并且在发布的数据中需要被移除；(2) 准标识符属性 (Quasi-Identifier，简称 QI)，作为用户的一般属性记录用户的非敏感信息，并且尽管单一项属性无法准确地识别用户的身份，但是当使用多项 QI 值作为条件在数据表中进行匹配时会有极大的概率辨认用户的真实身份；(3) 敏感属性 (Sensitive Attribute)，记录了用户的敏感信息。例如，在表 1.1 中，姓名属于标识符属性，年龄、性别和邮政编码属于 QI 属性，疾病属于敏感属性。

由于标识符属性对于用户身份具有很大的识别性，因此在数据进行发布时必须要将标识符属性删除。但是，仅仅在发布的数据表中删除标识符属性并不能有效地阻止攻击者识别目标用户的身份^[18-19]。如果攻击者具备了关于目标用户的背景知识，并在发布的数据表中通过使用 QI 信息进行匹配，攻击者仍然有很大的概率推断出目标用户在数据表中的个体标识，甚至得到目标用户的敏感信息。在文献[20]的研究中，87%的美国人可以通过邮政编码、性别和生日日期等 QI 属性值被唯一确定身份。因此，不仅需要将标识符属性从发布数据表中移除，还需要对数据表中的 QI 属性信息进行一定程度的匿名化，从而保证数据表中的隐私信息受到更加安全的保护。

例如，某攻击者已知他的邻居 Dave 的基本信息：年龄为 36、性别为男性、邮政编号为 14731，并且，攻击者知道 Dave 曾经去过发布数据的医院就医。当

数据表发布之后，如果攻击者得到了表中的数据信息，即使医院在发布的数据中没有加入姓名属性，攻击者仍然可以通过 Dave 的基本信息在数据表中进行匹配，从而推断出 Dave 在数据表中的 ID 为 1007，并且得知其病症为肝炎。

1.3 研究现状

近年来，随着隐私保护数据发布技术的不断成熟以及新技术对大数据隐私安全需求的不断增加，学者们对隐私保护数据发布技术的研究已经不仅局限于对静态数据表进行匿名，而是将丰富的匿名原则和匿名算法应用于更加广泛的隐私保护场景中^[21]。

目前，比较重要的应用场景包括：

(1) **社交网络隐私** (Social Network Privacy): 社交网络技术被用于描述人们的属性及人们之间的关系结构，并被用于理解这些关系的本质。随着互联网的发展，社交网络的数据信息已经被广泛地记录和收集，例如，电子邮件隐含地定义了人与人之间的关系，社交网站（如 Facebook、MySpace 和 Twitter）包含了用户之间的“朋友”关系等。尽管对这些社交数据进行广泛和深度的研究对社会的动态认知具有极高的价值，但是由于数据中包含了大量的个人及关系信息，使得攻击者可以通过如积极攻击^[22]和消极攻击^[23-24]等方式从数据中获取用户的隐私信息。为此，学者们基于隐私保护数据发布技术提出了多种匿名算法，可以防止用户的身份泄露^[25]、阻止近邻攻击^[26]和保护敏感关系^[27]等。

(2) **搜索日志隐私** (Search Log Privacy): 每当用户在搜索引擎或者购物网站中提交查询信息或者点击网页中的 URL 时，用户的 ID、查询内容和 URL 等数据信息都将被服务器记录下来。对这些搜索日志数据进行深入地分析，可以有效地提高搜索引擎的效率、改进网站的建设以及了解用户的喜好。但是，当没有对这些搜索日志数据进行合适的匿名处理时，发布和分析这些数据将导致大量用户隐私信息的泄露^[28-29]。并且，由于搜索日志中数据的内容和形式更加复杂以及考虑用户的使用习惯等因素，使得很多匿名算法^[30-31]无法有效地保障用户的隐私安全^[32]。目前，学者们开始尝试使用满足差分隐私的匿名算法^[32-33]对搜索日志数据进行匿名，并已经取得了一定的成果。

(3) **定位隐私** (Location Privacy): 随着现代科技的发展，许多移动设备

已经集成了全球定位系统，并且开发者通过推出基于位置服务的应用程序，为人们提供了更加智能和便捷的服务。一方面，对位置数据信息进行分析有助于提高应用程序的服务质量；另一方面，这些应用程序记录了人们的位置信息，对人们的隐私构成了潜在威胁。因此，对这些定位数据进行分析之前，必须首先保障用户的隐私信息安全。目前，基于隐私保护数据发布技术对定位隐私数据进行匿名保护的算法已经日趋成熟，主要包括：（1）对用户的[位置信息](#)进行匿名保护，如文献[34–38]分别提出了[不同的匿名算法将数据中的用户与周围其他用户的位置进行泛化](#)，使得每个用户与一定区域内其他用户的位置信息是无法辨别的；（2）对用户的[轨迹信息](#)进行匿名保护，如文献[39–41]在离线环境下将轨迹数据进行匿名并发布，文献[42–44]在在线环境下持续收集并更新匿名数据的动态。

1.4 本文工作

隐私保护数据发布技术不仅包括了很多[匿名原则](#)为用户的隐私安全提供保障，还包括了很多[遵循相应匿名原则的匿名算法](#)将数据表进行转化。目前，隐私保护数据发布技术面临的主要难题^[45]包括：一方面，研究更多的匿名技术使隐私保护数据发布技术在[更加广泛的场景](#)中应用，并且对隐私信息提供更加安全的保护；另一方面，提出更加高效的匿名算法使匿名之后的数据保留更多的信息可利用性，并且能快速处理规模庞大的数据。

本文的主要工作包括：

（1）提出[交叉桶泛化算法](#)并实现对[用户身份](#)和[敏感属性](#)提供相互独立的保护。在一般情况下，使用遵循 l -diversity 匿名原则的泛化算法会为用户身份提供过度的保护，从而导致匿名数据表损失大量的信息可利用性。通过结合泛化算法和桶算法的原理提出了交叉桶泛化算法，[将数据表中的个体划分为等价组和桶](#)，并且，通过使交叉桶泛化算法满足 (k, l) -anonymity 匿名原则的条件，将匿名数据表中用户身份暴露的概率和敏感属性值泄露的概率分别控制在 $1/k$ 和 $1/l$ 以内。除此之外，还通过在交叉桶泛化算法中使用启发式尽可能[减少匿名数据表中各个等价组和桶包含的个体数量以及所有泛化 QI 值的值域范围](#)，从而进一步提高了匿名数据表的信息可利用性。

（2）定义个性化隐私保护的发布环境并提出[局部分解算法](#)对数据表中的[敏](#)

感值进行保护。到目前为止，几乎所有匿名技术都没有考虑按照用户的个人意愿进行隐私保护的情况。一般的匿名技术只将组成数据表的属性作为最小的敏感单位，并分为如1.2节中介绍的标识符属性、QI属性和敏感属性。但是，数据的敏感性应该由用户决定而非数据发布者。当按照用户的个人意愿并以数据值作为最小的敏感单位时，一个属性可能同时包括了非敏感值和敏感值。这种类型的属性被称为半敏感属性。为了保护半敏感属性中的敏感值，我们基于桶算法的原理提出了一种局部分解算法。该算法允许用户在数据表中自由设置自己的敏感值，并且在每个半敏感属性和敏感属性中将带有敏感值的用户划分为桶，从而保证所有敏感值的安全。此外，局部分解算法的另一个优点是具有很强的扩展性，它可以根据不同的匿名需求和属性特点，同时遵循不同的匿名原则对数据中的敏感值进行保护。

(3) 在个性化隐私保护的发布环境下，提出局部分解泛化算法并实现对数据表中的用户身份和敏感值进行保护。局部分解算法为了保留尽可能多的信息可利用性，没有提供对用户身份保护的功能。在隐私泄露风险较高的发布环境中，为用户身份提供保护可以有效地降低未知攻击造成伤害的风险。因此，在局部分解算法的基础上，通过加入泛化机制提出了局部分解泛化算法同时保护数据表中的用户身份和敏感值。我们通过分别使用多维划分技术和NCP引导划分的启发式具体实现了两种局部分解泛化算法，并且，由于在局部分解泛化算法中泛化算法和桶算法对用户身份和敏感值的保护是相互独立的，所以可以根据实际需求使用合适的泛化机制与局部分解算法进行结合对数据中的隐私信息进行保护。

1.5 论文结构

本文共包括六个章节，每个章节的具体内容如下：

第1章：绪论。讨论了隐私保护数据发布技术的研究背景和意义，并且介绍了隐私保护数据发布技术的基本模型及研究现状；最后，给出了本文的研究工作和论文结构。

第2章：隐私保护数据发布技术概述。主要对本文所需要使用的隐私保护数据发布技术进行了较为详细的介绍，包括匿名原则、匿名算法以及信息可利用性的评估方法等。

第3章：用户身份和敏感属性的独立保护。首先，分析了泛化算法对用户身份产生过度保护的原因，并提出了交叉桶泛化算法对用户身份和敏感属性进行相互独立的保护；然后，分析了交叉桶泛化算法对用户身份和敏感属性的保护原理，提出了遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法的定义；最后，具体实现了遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法，并通过大量实验对算法的匿名效果进行了测试和分析。

第4章：个性化的隐私保护。首先定义了个性化隐私保护的发布环境，并提出局部分解算法对数据表中的敏感值进行保护；然后，分析了局部分解算法保护敏感值的工作原理，以及具体实现了遵循 l -diversity 匿名原则的局部分解算法；接下来，通过大量的实验证明局部分解算法可以有效地保护数据表中的所有敏感值，以及匿名之后的数据表具有非常优秀的信息可利用性；最后，还对同时遵循 l -diversity 和 t -closeness 匿名原则的局部分解算法进行实现，证明局部分解算法具有很高的可扩展性。

第5章：局部分解泛化算法。首先分析了局部分解算法无法为用户身份提供保护的原因，并提出局部分解泛化算法保护数据表中的用户身份和敏感值；然后，分析了局部分解泛化算法保护用户身份和敏感值的工作原理，以及具体实现了两种基于不同启发式的局部分解泛化算法；最后，对两种局部分解泛化算法的信息可利用性进行了大量的实验测试，并分析了实验结果。

第6章：总结与展望。主要对本文的工作进行全面总结，并指出未来的研究方向。

第2章 隐私保护数据发布技术概述

2.1 匿名原则

匿名原则是指对数据表提出的匿名要求。当一个数据表满足了某一匿名原则的条件时，数据表就会具备相应的隐私保护能力。在本节中，我们将介绍一些比较重要的匿名原则。

2.1.1 k -anonymity 匿名原则

k -anonymity 匿名原则^[20]可以有效地防止数据表中用户身份的泄露，它的具体定义如下：

定义 2.1：对于一个数据表 T ，且 T 包含 m 个 QI 属性 $A_1^{QI}, A_2^{QI}, \dots, A_m^{QI}$ ，如果 T 遵循 k -anonymity 匿名原则，当且仅当对于任意个体 $t \in T$ ，至少存在其他 $k-1$ 个个体 t_1, t_2, \dots, t_{k-1} ，满足

$$t(A_1^{QI}, \dots, A_m^{QI}) = t_1(A_1^{QI}, \dots, A_m^{QI}) = \dots = t_{k-1}(A_1^{QI}, \dots, A_m^{QI}) \quad (2.1)$$

其中， $t(A_1^{QI}, \dots, A_m^{QI})$ 表示个体 t 中各个 QI 属性的值。

如果匿名数据表遵循 k -anonymity 匿名原则，当攻击者使用目标用户的 QI 信息进行匹配时至少会得到 k 个无法区分的匹配个体。例如，将表 1.1 中的标识符属性删除并且将 QI 值转化使其遵循 2-anonymity 匿名原则的结果，如表 2.1 所示。在表 2.1 中，攻击者使用任意个体的 QI 值进行匹配将至少得到两个无法区分的个体，因此，攻击者无法准确地获得目标用户在数据表中的个体标识。

k -anonymity 匿名原则是目前隐私保护数据发布技术中应用最广泛的匿名原则，甚至在其他的安全领域中也发挥着重要作用。在不同的发布环境中，很多匿名原则在 k -anonymity 匿名原则的基础上被提出。例如， (X, Y) -anonymity 匿名原则^[46]、多重关系的 k -anonymity 匿名原则^[47] (MultiRelational k -anonymity) 和 Δ -growth 匿名原则^[48] 等。

表 2.1 遵循 2-anonymity 匿名原则的泛化数据表

ID	Age	Gender	Zip Code	Disease
1001	[22–24]	*	1324*	Pneumonia
1002	[22–24]	*	1324*	Dyspepsia
1003	[22–24]	*	1324*	Flu
1004	[25–26]	Female	1****	Bronchitis
1005	[25–26]	Female	1****	Bronchitis
1006	[34–38]	*	14***	Dyspepsia
1007	[34–38]	*	14***	Hepatitis
1008	[34–38]	*	14***	Gastritis

尽管 k -anonymity 匿名原则可以有效地防止用户身份的泄露，但是它却无法为数据表中的敏感属性提供足够的保护。例如，当攻击者的攻击目标为表 1.1 中的 Tina 时，假设攻击者已知 Tina 的所有 QI 值信息，攻击者可以通过在表 2.1 中匹配 Tina 的 QI 值推断出 Tina 的个体信息是 ID 为 1004 或 1005 的个体。但是，由于 ID 为 1004 和 1005 的个体对应的病症都为支气管炎，所以攻击者可以断定 Tina 的病症为支气管炎。

除此之外，虽然 k -anonymity 匿名原则可以在一定程度上阻止攻击者获知目标用户的信息存在于匿名表中，但是它并不能完全防止这种隐私泄露情况的发生。为此，Nergiz 提出了 δ -presence 匿名原则^[49]，即对于一个原始数据表 T 及匿名后的数据表 T' ，当攻击者具备一定的背景知识并且已知 T' 中所有信息时，对任意包含于数据表 T 中的个体，攻击者得知该个体信息存在于 T' 中的概率需要介于 $\delta = (\delta_{min}, \delta_{max})$ 之间。

2.1.2 l -diversity 匿名原则

通过学者们的研究发现，满足 k -anonymity 匿名原则的数据表仍然会受到同质攻击的危害，使得攻击者即使没有确定用户在数据表中的个体标识，仍然会获得目标用户的敏感属性值，因此，Machanavajjhala 提出了 l -diversity 匿名原则^[50–51] 用于增强匿名数据表保护敏感属性的能力。

l -diversity 匿名原则要求数据表，对数据表中的任意个体及与其 QI 信息相

同的所有个体，它们所携带的敏感属性值是“*l well-represented*”。值得注意的是，由于 Machanavajjhala 只对 *l*-diversity 匿名原则提出了一种抽象的定义，所以根据不同的实际匿名需求，*l*-diversity 匿名原则可以被进行不同的实例化，如熵 *l*-diversity 匿名原则 (Entropy *l*-diversity)、递归(*c, l*)-diversity 匿名原则 (Recursive (*c, l*)-diversity) 等。在本文中，*l*-diversity 匿名原则的定义如下：

定义 2.2：对于一个数据表 T ，如果 T 遵循 *l*-diversity 匿名原则，当且仅当对于任意个体 $t \in T$ ，满足

$$p(t, s) \leq \frac{1}{l} \quad (2.2)$$

其中， $p(t, s)$ 为 t 的敏感值 s 暴露的概率。

例如，表 2.2 为使用遵循 4-diversity 匿名原则的泛化算法对表 1.1 进行匿名之后的结果，并且匿名数据表中每个个体的敏感值暴露的概率均为 $1/4$ 。

表 2.2 遵循 4-diversity 匿名原则的泛化数据表

ID	Age	Gender	Zip Code	Disease
1001	[22–25]	*	1324*	Pneumonia
1002	[22–25]	*	1324*	Dyspepsia
1003	[22–25]	*	1324*	Flu
1004	[22–25]	*	1324*	Bronchitis
1005	[26–38]	*	14***	Bronchitis
1006	[26–38]	*	14***	Dyspepsia
1007	[26–38]	*	14***	Hepatitis
1008	[26–38]	*	14***	Gastritis

但是，*l*-diversity 匿名原则在一些发布环境中会受到倾斜攻击的危害。例如，假设攻击者已知在某数据表中，95%的用户疾病值为感冒，而仅有 5%的用户疾病值为癌症。当使用 2-diversity 匿名原则对数据表进行匿名时，攻击者却可以有 50%的概率确定患癌症的用户，因此，Li 提出了 *t*-closeness 匿名原则^[52] 用于弥补 *l*-diversity 匿名原则在某些发布环境中的不足，即数据表中每个等价组

的敏感属性值分布与整个数据表的敏感属性值分布之间的距离不超过上限值 t 。

与 k -anonymity 匿名原则一样, l -diversity 匿名原则的应用范围也非常广泛。并且,由于 l -diversity 匿名原则相比 k -anonymity 匿名原则为敏感属性提供了更加安全的保护效果,许多针对敏感属性保护的匿名原则在其基础上被提出。例如, (X, Y) -privacy 匿名原则^[46]、 (α, k) -anonymity 匿名原则^[53]、 (k, ϵ) -anonymity 匿名原则^[54]、FF-anonymity 匿名原则^[55]、 β -likeness 匿名原则^[56]、 (ϵ, m) -anonymity 匿名原则^[57]、 (n, t) -closeness 匿名原则^[58]和置信边界^[59] (Confidence Bounding) 等。

2.1.3 差分隐私

当使用发布的数据表进行分析时,数据研究者需要得到的是数据表整体或部分的统计信息,而不是针对某一特定用户的信息,因此,对于数据表中的任意用户,数据表匿名之后得到的统计结果应该与没有包含该用户时匿名之后得到的统计结果是几乎一样的。Dwork 基于上述原理提出了差分隐私保护技术^[60],它通过匿名或者添加噪音的方式,使得当添加或删除数据表中某个用户的信息时不会对数据的分析结果产生明显的影响。并且,即使攻击者通过目标用户的已知信息在数据表中进行匹配或者计算也无法进一步得到该用户的任何额外信息。通常情况下,差分隐私相比于一般的匿名保护技术具有更加可靠的保护效果,但是在不同环境下由噪音引起的误差失真更加难以控制^[61-62]。

目前,差分隐私技术的应用范围非常广泛并受到极大关注,主要包括:(1)基于差分隐私的数据发布,如直方图^[63-65]和数据立方发布^[66-67],数据划分发布^[68-72]、取样和过滤技术^[73]等;(2)面向数据挖掘和机器学习的隐私保护,如模式挖掘技术^[74-76]、分类分析^[77-79]、回归分析^[80-81]等;(3)基于差分隐私的查询处理技术,如 Laplace^[82]、NoiseFirst^[64]、Privlet^[83]和 AG^[84]等;(4)基于差分隐私的应用系统,如交互式数据分析系统^[85-86]、非交互式数据分析系统^[87]和基于 MapReduce 的聚集分析系统^[88]等。

2.2 匿名算法

匿名算法是为了使数据表满足匿名原则中的条件,对数据表进行匿名化的具

体实现。但是，即使遵循相同的匿名原则，不同的匿名算法也会对数据表产生不同的匿名效果。接下来，我们将对主要的匿名算法进行介绍。

2.2.1 泛化算法

泛化算法是匿名算法中最重要的算法，它通过将数据表中的 QI 属性值进行一定程度的泛化，即将具体的数据值转化为概括和抽象的形式，从而防止攻击者使用目标用户的 QI 值获取用户在数据表中的个体标识。对于数字类型的属性，数值将被转化为值域的类型，如将数值 24 转化为 [10-30]。而对于分类类型的属性，数值将根据用户自定义的泛化层次树进行泛化。如根据图 2.1，可以将工程师泛化为专业人员。

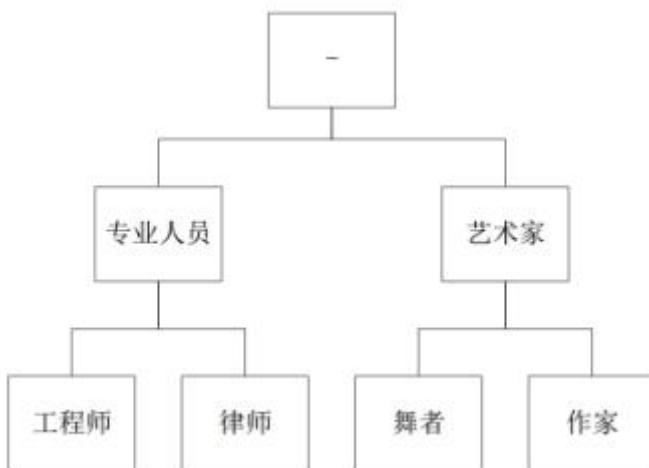


图 2.1 职业属性的泛化层次树

泛化算法的种类多种多样，根据数值泛化方式的不同主要分为以下四类：

(1) 全域泛化 (Full-Domain Generalization): 全域泛化^[89]是指对于某个属性的泛化层次树，所有叶子 QI 值均需泛化为同一层级。例如根据图 2.1 中，如果将工程师泛化为专业人员，则必须要将数据表中所有的律师泛化为专业人员，并将舞者和作家泛化为艺术家。

(2) 子树泛化 (Subtree Generalization): 子树泛化^[90-93]是指对于某个属性的泛化层次树，任意子树下的叶子 QI 值需要泛化为同一层级。例如根据图 2.1 中，如果将工程师泛化为专业人员，那么必须要将律师泛化为专业人员，但是舞

者和作家则不一定进行泛化。

(3) 兄弟泛化 (Sibling Generalization): 兄弟泛化^[89]与子树泛化相似，是指对于某个属性的泛化层次树，每个单独的叶子 QI 值需要泛化为同一层级。例如根据图 2.1 中，当将工程师泛化为专业人员时，律师不必与工程师一样泛化为专业人员。

(4) 细胞泛化 (Cell Generalization): 细胞泛化^[94]是指在某个属性的泛化层次树中，每个单独的叶子 QI 值在数据表中都可以泛化为不同层级。例如根据图 2.1 中，在将数据表进行转化时，可以将其中一部分工程师泛化为专业人员，同时将剩余的工程师保持原样。

其中，前三种泛化方式被称为全局泛化 (Global Generalization)，即对于泛化层次树中的某个叶子 QI 值，当将其在数据表进行匿名时都需要被转化为相同的形式；而最后一种泛化方式则被称为局部泛化 (Local Generalization)，即对于泛化层次树中的某个叶子 QI 值，当将其在数据表进行匿名时既可以转化为更加抽象的形式，也可以保持原样。

除此之外，泛化算法还可以被分为单维泛化和多维泛化两种类型。其中，单维泛化是指对于一个泛化算法，其转化的过程是依次将数据表中的每个 QI 值转化为相应泛化层次树中的值，如 Bottom-Up 算法^[95]；而多维泛化是指对于一个泛化算法，其转化的过程是同时将多个属性中的 QI 值以笛卡尔积的形式转化为相应属性的泛化层次树中的笛卡尔积形式，如 Mondrian 算法^[96-97]。

2.2.2 桶算法

不同于泛化算法需要对数据表中的 QI 值进行泛化处理，桶算法^[15]仅仅通过将 QI 属性和敏感属性的关联切断，从而保护数据表中用户的敏感属性值信息。通常情况下，泛化算法可以为数据表提供比较安全的保护，但是使用泛化算法需要损失大量的信息可利用性^[98-100]。相比之下，使用桶算法的匿名数据表获得的信息可利用性远远高于使用泛化算法，甚至十分接近原始数据表。但是，由于桶算法仅能为敏感属性提供保护，因此在匿名保护需求较高的场景中，桶算法的应用非常有限。

表 2.3 遵循 4-diversity 的桶数据表

ID	Bucket ID	Age	Gender	Zip Code	Disease
1001	1	22	Male	13248	Bronchitis
1002	1	22	Male	13241	Dyspepsia
1003	1	24	Female	13247	Flu
1004	1	25	Female	13242	Pneumonia
1005	2	26	Female	14553	Bronchitis
1006	2	34	Male	14423	Dyspepsia
1007	2	36	Male	14731	Gastritis
1008	2	38	Female	14417	Hepatitis

如表 2.3 所示，为使用遵循 4-diversity 匿名原则的桶算法对表 1.1 进行匿名的结果，其中，在每个桶中的 QI 属性与敏感属性之间的联系被切断，即任意用户对应于相应桶内每个敏感值的概率相等。值得注意的是，由于桶算法完整地保留了原始数据表中的 QI 值信息，一方面极大地有利于匿名数据表保留信息可利用性，另一方面也使得攻击者很容易从数据表中获得更多的信息。

为了提高桶算法的保护能力，Li 基于桶算法的原理提出了 Slicing 算法^[101]。Slicing 算法不仅将个体集合进行分组，还将属性集合进行划分，使得 QI 属性之间的关联被切断。当攻击者通过使用 QI 值在匿名数据表中进行匹配时，会得到更多的匹配个体，从而极大地提高了攻击者的匹配难度。相比于一般的桶算法，Slicing 算法不仅可以对敏感属性值进行保护，同时还在一定程度上可以防御身份识别和会员识别的攻击。此外，Slicing 算法还可以根据实际匿名需求调整属性划分的方式，从而使其应用更加灵活。值得注意的是，当 Slicing 算法仅将 QI 属性和敏感属性进行分割时，Slicing 算法变为一般的桶算法。

2.3 信息可利用性

匿名算法不仅需要保障用户的隐私安全，还需要尽可能地提高匿名数据中的信息可利用性。在本节中，我们将介绍目前比较常用的信息可利用性评估方法。

2.3.1 信息损失量评估

信息损失量评估是通过设置测量方式表示数据在匿名过程中产生的信息损失量，并且这种测量方式也可以被用于匿名算法中的启发式部分对算法进行优化。根据学者们的研究^[102-104]，在满足用户的隐私保护需求的前提下，根据给定的信息损失量评估方法将匿名数据表中的信息可利用性最大化是 NP-hard 问题。

目前，应用比较广泛的信息损失量评估方法包括：

(1) **最小的失真** (Minimal Distortion): 最小的失真^[104]是被用于计算单个属性的信息损失量。对于一个属性，每当某个用户在该属性中的一个具体值被泛化时，惩罚值就会加 1。文献[14]和[18]等使用最小的失真惩罚值作为算法中的启发式控制算法的分组策略。

(2) **损失度量** (Loss Metric): 损失度量^[105]被用于计算将一个具体的数据值进行泛化时产生的信息损失量。它的表达公式为：

$$ILoss(v_g) = \frac{|v_g| - 1}{|D[A]|} \quad (2.3)$$

其中， $|v_g|$ 表示数据值 v_g 在泛化树中包含子孙的数量， $|D[A]|$ 表示属性 A 中值域包含值的数量。

(3) **分类度量** (Classification Metric): 分类度量^[90]是用于评估对于一个特定属性，用户的分类是否正确的测量方法。当使用匿名算法将个体进行分组之后，对于数据表中的某个属性，如果一个个体在其分组中的属性值与分组中包含最多的属性值不同，则该个体的惩罚值即为 1，否则为 0。文献[97]和[106]等使用分类度量惩罚值作为算法中的启发式控制算法的分组策略。

(4) **鉴别力度量** (Discernibility Metric): 鉴别力度量^[91]主要用于对使用泛化算法匿名的数据表进行评估。它的表达公式为：

$$C_{DM} = \sum_{EG} |EG|^2 \quad (2.4)$$

其中， EG 表示匿名数据表中的等价组， $|EG|$ 表示等价组 EG 中包含个体的数量。文献[96]和[107]等使用鉴别力度量惩罚值作为算法中的启发式控制算法对个体进行分组。

2.3.2 查询分析评估

查询分析评估主要通过使用匿名之前和匿名之后的数据作为查询语句或分析任务的输入，并通过对两者之间的统计结果对匿名数据的信息可利用性进行评估。

目前，主要的查询分析方法包括：

(1) 概率性的查询处理^[108] (Probabilistic Query Processing)，是将匿名数据表以概率的形式重新展现，并对概率数据表使用查询语句统计数据表中的信息，例如，将表 2.3 转化为概率形式的部分结果，如表 2.4 所示。使用该技术的项目，包括 Avatar¹、MystiQ²、ORION³和 Trio⁴等。

表 2.4 概率形式

ID	Age	Gender	Zip Code	Disease	Probability
1001	22	Male	13248	Bronchitis	0.25
1001	22	Male	13248	Dyspepsia	0.25
1001	22	Male	13248	Flu	0.25
1001	22	Male	13248	Pneumonia	0.25
1002	22	Male	13241	Bronchitis	0.25
1002	22	Male	13241	Dyspepsia	0.25
...

(2) 对匿名数据直接使用查询语句统计信息，并与原始数据的统计结果进行对比。例如，文献[109]和[110]通过使用查询语句测量包含泛化值的数据表；文献[111]提出了一种使用查询语句对通过使用随机化技术匿名的数据表进行统计的方法；文献[54]和[112]提出了通过计算查询语句结果的上限和下限对桶数据表进行统计的方法。

¹ <http://www.almaden.ibm.com/cs/projects/avatar/>

² <http://www.cs.washington.edu/homes/suciu/project-mystiq.html>

³ <http://orion.cs.purdue.edu/>

⁴ <http://infolab.stanford.edu/trio/>

第3章 用户身份和敏感属性的独立保护

3.1 引言

泛化算法是隐私保护数据发布技术中应用最广泛的匿名算法，但是，遵循 k -anonymity 匿名原则的泛化算法无法保证敏感属性的安全，而遵循 l -diversity 匿名原则的泛化算法会对用户身份提供过度的保护，使得匿名数据表中的信息可利用性大大降低。在本章中，我们将改进经典的泛化算法，使得改进之后的算法可以分别对用户身份和敏感属性提供独立的保护，从而极大地提高匿名数据中的信息可利用性。

3.1.1 问题的提出

遵循 k -anonymity 匿名原则的泛化算法可以有效地防止用户身份的泄露，并且在一定程度上降低敏感属性值泄露和成员身份暴露的概率^[101]。例如，表 3.1 为攻击者的背景知识，表 3.2 为数据持有者需要发布的原始数据表，表 3.3 为遵循 2-anonymity 匿名原则的泛化数据表。在未经匿名处理的表 3.2 中，攻击者可以轻易地通过匹配性别、年龄和邮政编码的属性值判断 Helen 的病症为肺炎。但是在表 3.3 中，攻击者无法判断 Helen 的 ID 为 106、107 或 108，因此攻击者无法准确地推测出 Helen 的具体病症。此外，假设当攻击者不确定 Helen 的数据是否存在于数据表中时，表 3.3 也可以在一定程度上防止 Helen 的成员身份暴露，即攻击者无法确定 Helen 的信息存在于该数据表中。

表 3.1 攻击者的背景知识

Name	Gender	Age	Zip code
Tina	Female	26	22105
Helen	Female	31	43312
Louis	Male	42	01274
Rachel	Female	24	-

表 3.2 原始数据表

ID	Age	Gender	Zip code	Disease
101	16	Female	43307	Flu
102	22	Male	43302	Dyspepsia
103	24	Female	43306	Hepatitis
104	26	Male	43307	Bronchitis
105	29	Male	43309	Bronchitis
106	31	Female	43312	Pneumonia
107	34	Female	43312	Gastritis
108	35	Male	43309	Dyspepsia

表 3.3 遵循 2-anonymity 匿名原则的泛化数据表

ID	Group ID	Age	Gender	Zip code	Disease
101	1	[16–24]	*	[43302–43307]	Flu
102	1	[16–24]	*	[43302–43307]	Dyspepsia
103	1	[16–24]	*	[43302–43307]	Hepatitis
104	2	[26–29]	Male	[43307–43309]	Bronchitis
105	2	[26–29]	Male	[43307–43309]	Bronchitis
106	3	[31–35]	*	[43309–43312]	Pneumonia
107	3	[31–35]	*	[43309–43312]	Gastritis
108	3	[31–35]	*	[43309–43312]	Dyspepsia

但是，遵循 k -anonymity 匿名原则的泛化算法仍然存在比较严重的敏感属性值泄漏的问题。如在表 3.3 中，第 2 个等价组包含的疾病值全部为支气管炎。假设攻击者已知目标用户 ID 为 104 或 105 的 QI 值信息时，即使没有造成目标用户的身份泄露，攻击者仍然能推测出目标用户的病症为支气管炎。

目前，比较可行的解决办法是使泛化算法遵循 l -diversity 匿名原则对数据表进行泛化。如表 3.4 所示，在每个等价组中都包含了四个不同的敏感值，所以任意敏感值暴露的概率都为 $1/4$ 。但是，遵循 l -diversity 匿名原则的泛化数据表同时也必须遵循 k -anonymity 匿名原则，导致数据表的信息可利用性进一步减少。例如，泛化数据表 3.4 不仅遵循了 4-diversity 匿名原则，还同时遵循了 4-

anonymity 匿名原则，即每个等价组中都泛化了至少 4 个个体的 QI 信息。因此相比于表 3.3，表 3.4 中泛化 QI 值的精度进一步降低。

表 3.4 遵循 4-diversity 匿名原则的泛化数据表

ID	Group ID	Age	Gender	Zip code	Disease
101	1	[16–26]	*	[43302–43307]	Flu
102	1	[16–26]	*	[43302–43307]	Dyspepsia
103	1	[16–26]	*	[43302–43307]	Hepatitis
104	1	[16–26]	*	[43302–43307]	Bronchitis
105	2	[29–35]	*	[43309–43312]	Bronchitis
106	2	[29–35]	*	[43309–43312]	Pneumonia
107	2	[29–35]	*	[43309–43312]	Gastritis
108	2	[29–35]	*	[43309–43312]	Dyspepsia

表 3.5 遵循 4-diversity 匿名原则的桶数据表

ID	Bucket ID	Age	Gender	Zip code	Disease
101	1	16	Female	43307	Bronchitis
102	1	22	Male	43302	Dyspepsia
103	1	24	Female	43306	Flu
104	1	26	Male	43307	Hepatitis
105	2	29	Male	43309	Bronchitis
106	2	31	Female	43312	Dyspepsia
107	2	34	Female	43312	Gastritis
108	2	35	Male	43309	Pneumonia

不同于泛化算法可以为用户身份和敏感属性提供双重保护，桶算法仅能保护数据表中的敏感属性。但是与泛化算法相比，桶算法极大地提高了匿名数据表中的信息可利用性。如数据表 3.5 所示，使用桶算法的匿名数据表完整地保留了所有原始 QI 值，并且每个桶中的个体都对应了 4 个不同的敏感值。但是，由于桶算法不具备防止身份泄露的保护功能，使得桶数据表非常有可能成为攻击其他数据表的外部数据源。例如，攻击者在桶数据表 3.5 中匹配 Rachel 的 QI 值，可以

推测出 Rachel 很可能曾经生病并去过发布数据表的医院就医。同时，攻击者还获得了 Rachel 的邮政编码值 43306，从而增加了攻击者的背景知识。因此，尽管桶算法的信息可利用性十分出色，但是在隐私保护需求较高的场景中应用的范围非常有限。

3.1.2 本章工作

到目前为止，关于泛化算法的研究并没有区分对用户身份和敏感属性的保护需求，从而使泛化算法失去了很多信息可利用性和灵活性。由此，我们通过结合桶算法的特点，在传统泛化算法的基础上进行改进，提出一个被称为交叉桶泛化算法（Cross-Bucket Generalization）的新型算法。交叉桶泛化算法可以分别根据用户身份和敏感属性的匿名需求提供最适当的保护，使得算法具有更好的灵活性和信息可利用性，它首先通过泛化算法将数据表中的个体划分为多个等价组，从而满足用户身份的匿名需求，然后再将泛化之后的个体划分为多个桶，即通过桶算法对数据表中的敏感属性进行匿名保护。

表 3.6 遵循 2-anonymity 和 4-diversity 匿名原则的交叉桶泛化数据表

ID	Group ID	Bucket ID	Age	Gender	Zip code	Disease
101	1	1	[16–24]	Female	[43306–43307]	Dyspepsia
102	2	1	[22–26]	Male	[43302–43307]	Flu
103	1	2	[16–24]	Female	[43306–43307]	Bronchitis
104	2	2	[22–26]	Male	[43302–43307]	Hepatitis
105	3	3	[29–35]	Male	43309	Bronchitis
106	4	3	[31–34]	Female	43312	Pneumonia
107	4	4	[31–34]	Female	43312	Dyspepsia
108	3	4	[29–35]	Male	43309	Gastritis

通过交叉桶泛化算法对原始数据表 3.2 进行匿名之后的结果，如表 3.6 所示，其中，Group ID 指个体被划分为等价组的标识，Bucket ID 指个体被划分为桶的标识。由数据表中可以发现，每个个体都分别被划分到一个等价组和一个桶中。当攻击者对匿名数据表中的目标个体进行 QI 值匹配时，由于每个等价组中包含

个体属于多个不同的桶，因此攻击者在获取目标个体的敏感值时会更加困难。并且，匿名数据表3.6同时遵循了2-anonymity匿名原则和4-diversity匿名原则，相比于匿名数据表3.3和3.4，不仅分别达到了其相应的身份识别和敏感属性的匿名需求，还保留了更多的信息可利用性。例如，当攻击者使用Helen的QI值在匿名表3.6中进行匹配时，会得到Helen的信息在第四等价组中，但是第四等价组中包含的两个个体分别被划分在第三个和第四个桶中，因此，Helen的疾病值可能为第三和第四个桶中的任意值，即攻击者推测Helen的真实疾病值为肺炎的概率为1/4。

在本章的研究中，我们对数据发布的匿名需求进行假设：匿名数据表需要同时为用户身份和敏感属性提供保护，并且尽可能地提高匿名数据的信息可利用性。本章的具体工作如下：

首先，我们提出交叉桶泛化算法用于保护发布数据的敏感信息。该算法可以将数据中的个体划分为等价组和桶，从而同时防止用户身份和敏感属性值的泄露。并且，相比于传统的泛化算法，匿名之后的数据损失将大大减少。

其次，为了保证使用交叉桶泛化算法匿名的数据表提供安全的隐私保护，我们提出一个新的匿名原则，称为 (k, l) -anonymity匿名原则。当交叉桶泛化算法遵循 (k, l) -anonymity匿名原则时，对于匿名数据表中的任意个体，其用户身份和敏感属性值暴露的概率分别最多为 $1/k$ 和 $1/l$ 。

第三，我们通过结合泛化算法和桶算法的特点，具体实现遵循 (k, l) -anonymity匿名原则的交叉桶泛化算法。并且，为了尽可能提高匿名数据表中的信息可利用性，交叉桶泛化算法会尽量减少各个等价组和桶中包含的个体数量，同时尽可能缩小各个等价组中泛化QI值的值域范围。

最后，我们进行大量的实验，通过对比传统的泛化算法和桶算法测试交叉桶泛化算法的实际匿名效果。其中，对比的主要内容包括：(1) 交叉桶泛化算法比其他算法提供了更加安全的敏感属性保护；(2) 通过对比鉴别力度量和查询回答错误率的结果，证明交叉桶泛化算法相比传统的泛化算法极大地提高了信息可利用性。此外，还通过调整参数 k 和 l 的值，对遵循 (k, l) -anonymity匿名原则的交叉桶泛化算法产生的匿名影响进行研究。

本章的结构如下：在3.2节中，将介绍交叉桶泛化算法的基本概念并且分析

保护用户身份和敏感属性的工作原理；在3.3节中，将对遵循 (k, l) -anonymity匿名原则的交叉桶泛化算法进行具体实现；在3.4节中，将对交叉桶泛化算法的相关匿名效果进行实验测试和分析；在3.5节中，将对本章的内容进行总结。

3.2 交叉桶泛化算法模型

3.2.1 基本概念

在给出交叉桶泛化算法的定义之前，我们先介绍一些基本的概念。假设一个数据表 T 包含了 d 个 QI 属性 $A_1^{QI}, A_2^{QI}, \dots, A_d^{QI}$ 以及一个敏感属性 A^{SA} ，其中，每个属性都为数字类型或者分类类型。令 $D[A]$ 表示属性 A 的值域。对于任意个体 $t \in T$ ， $t[A]$ 表示个体 t 中属性 A 的值。

定义 3.1：一个划分是将数据表 T 中的个体划分为多个子集，每个子集被称为一个 QI 组，并且每个个体仅能属于一个 QI 组。假设数据表 T 包含 m 个 QI 组 $\{G_1, G_2, \dots, G_m\}$ ，那么有 $\bigcup_{i=1}^m G_i = T$ ，且对任意 $1 \leq i_1 \neq i_2 \leq m$ ，有 $G_{i_1} \cap G_{i_2} \neq \emptyset$ 。

根据不同的匿名算法，QI 组会具有不同的性质。在泛化算法中，每个 QI 组中所有个体的各个 QI 属性值都会转化为相等的泛化形式。而在桶算法中，每个 QI 组会被划分为两个部分，分别包含了 QI 值和敏感值。

定义 3.2：如果数据表 T 被划分为 m 个 QI 组，并且每个 QI 组都被称为一个等价组，当且仅当对于任意个体 $t \in T$ ，在数据表 T 中的泛化形式如下：

$$(G_j[1], G_j[2], \dots, G_j[d], t[A^{SA}]) \quad (3.1)$$

其中， $G_j (1 \leq j \leq m)$ 是唯一包含个体 t 的 QI 组，并且 $G_j[i] (1 \leq i \leq d)$ 是 G_j 组中所有个体在属性 A_i^{QI} 中的值。

定义 3.3：如果数据表 T 被划分为 m 个 QI 组，并且每个 QI 组都被称为一个桶，当且仅当每个 QI 组包含 $QIT(QI, GID)$ 和 $SAT(SA, GID)$ 两个部分，其中， QI 和 SA 分别为在 QI 组中所有个体的 QI 值和敏感值，并且 GID 表示 QI 组的 ID 标

识。

根据定义 3.2 和 3.3，对交叉桶泛化算法进行如下定义：

定义 3.4：对于一个数据表 T ，交叉桶泛化算法是将 T 划分为许多个等价组和桶，并且数据表中每个个体只能属于唯一的等价组和桶。假设数据表 T 被划分为 m 个等价组和 n 个桶，分别记为 $\{EG_1, EG_2, \dots, EG_m\}$ 和 $\{B_1, B_2, \dots, B_n\}$ ，并且有 $\bigcup_{i=1}^m EG_i = T$ 和 $\bigcup_{j=1}^n B_j = T$ ，以及对任意 $1 \leq i_1 \neq i_2 \leq m$ ，有 $EG_{i_1} \cap EG_{i_2} = \emptyset$ 和对任意 $1 \leq j_1 \neq j_2 \leq n$ ，有 $B_{j_1} \cap B_{j_2} = \emptyset$ 。

3.2.2 隐私保护分析

在本节中，我们将详细地分析交叉桶泛化算法对用户身份和敏感属性的保护原理，并且讨论交叉桶泛化算法遵循 (k, l) -anonymity 匿名原则所需要的条件。

首先，我们对用户身份的保护进行分析，并有如下结论。

定理 3.1：在一个使用交叉桶泛化算法进行匿名的数据表中，对于任意个体 $t \in T$ ，其身份泄露的概率最多为 $1/|G(t)|$ ，其中 $G(t)$ 为包含 t 的等价组。

证明：根据定义 3.4，原始数据表在使用交叉桶泛化算法进行匿名之后被划分为多个等价组。当攻击者通过目标用户的 QI 值进行匹配时，由于数据表中对应个体 t 已经与 $G(t)$ 中其他个体的 QI 值泛化为相同的形式，所以攻击者将最少获得 $|G(t)|$ 个匹配个体。因此，个体 t 的身份泄露的概率最多为 $1/|G(t)|$ 。

推论 3.1：对于使用交叉桶泛化算法进行匿名的数据表，如果匿名数据表遵循 k -anonymity 匿名原则，则匿名表中每个等价组需要至少包含 k 个个体。

证明：由定理 3.1 可知，当匿名数据表中每个等价组都至少包含 k 个个体，则任意个体都至少与其他 $k-1$ 个个体是无法识别的。因此，该匿名数据表遵循 k -anonymity 匿名原则。

接下来，我们分析交叉桶泛化算法对敏感属性的保护。根据定义 3.4，原始数据表在使用交叉桶泛化算法进行匿名之后被划分为多个桶，而在同一个等价组

中的个体也被分配到不同的桶中。因此，攻击者首先需要确定目标用户可能被划分在哪些桶内。

定义 3.5：对于任意个体 $t \in T$ ，其桶定位概率记为 $p(t, B)$ ，即个体 t 在桶 B 中的概率。

定义 3.6：在一个交叉桶泛化数据表 T 中，对于任意个体 $t \in T$ ， MB 为个体 t 的一个匹配桶，当且仅当 $t[A^{QI}] \in MB[A^{QI}]$ ，其中 $t[A^{QI}]$ 为 t 的 QI 值， $MB[A^{QI}]$ 为 MB 中 QI 值的集合。

定理 3.2：在一个使用交叉桶泛化算法进行匿名的数据表中，对于任意个体 $t \in T$ ，其敏感属性值 s 暴露的概率满足：

$$p(t, s) \leq \sum_{MB} p(t, MB) \frac{|MB(s')|}{|MB|} \quad (3.2)$$

其中， $|MB(s')|$ 为在匹配桶 MB 中出现次数最多的敏感值 s' 的数量，并且 $|MB|$ 为 MB 中包含个体的数量。

证明：为了得到目标个体 t 的敏感值 s ，攻击者首先需要计算个体 t 在匿名数据表中每个桶的定位概率，以及个体 t 的敏感值为 s 的概率。因此，攻击者有

$$p(t, s) = \sum_B p(t, B) p(s|t, B) \quad (3.3)$$

其中， $p(s|t, B)$ 为当个体 t 在桶 B 中时敏感值为 s 的概率。此外，攻击者可以排除不包含目标个体 t 的 QI 值的桶，即当 $t[A^{QI}] \notin B[A^{QI}]$ 时，有：

$$p(t, B) = 0 \quad (3.4)$$

因此，根据定义 3.6，有：

$$p(t, s) = \sum_{MB} p(t, MB) p(s|t, MB) \quad (3.5)$$

其中， MB 为个体 t 的匹配桶。对任意 MB 中出现次数最高的敏感值 s' ，有：

$$|MB(s)| \leq |MB(s')| \quad (3.6)$$

因此，有

$$p(s|t, MB) = \frac{|MB(s)|}{|MB|} \leq \frac{|MB(s')|}{|MB|} \quad (3.7)$$

根据式(3.5)，有

$$p(t, s) \leq \sum_{MB} p(t, MB) \frac{|MB(s')|}{|MB|} \quad (3.8)$$

推论 3.2: 对于使用交叉桶泛化算法进行匿名的数据表, 如果匿名数据表遵循 l -diversity 匿名原则, 则对数据表中的所有个体需要满足如下条件: (1) 在其所有匹配桶中, 每个敏感值仅出现一次; (2) 对于包含了目标个体的敏感值的匹配桶, 需要满足

$$\frac{p(t, MB)}{|MB|} \leq \frac{1}{l} \quad (3.9)$$

证明: 根据条件 (1), 对于任意个体 $t \in T$, 由于在其所有的匹配桶中, 每个敏感值只出现一次, 因此只有一个匹配桶 MB' 包含了 t 的敏感值 s , 即有:

$$p(t, s) = \sum_{MB} p(t, MB)p(s|t, MB) = p(t, MB')p(s|t, MB') \quad (3.10)$$

并且, 对任意 t 的匹配桶 MB , 有:

$$\frac{|MB(s')|}{|MB|} = \frac{1}{|MB|} \quad (3.11)$$

根据定理 3.2, 有:

$$p(t, s) \leq \sum_{MB} p(t, MB)p\left(\frac{|MB(s')|}{|MB|}\right) = \frac{p(t, MB')}{|MB'|} \quad (3.12)$$

在条件 (2) 的限制下, 有:

$$p(t, s) \leq \frac{p(t, MB')}{|MB'|} \leq \frac{1}{l} \quad (3.13)$$

综上, 当满足条件 (1) 和 (2) 时, 交叉桶泛化算法遵循 l -diversity 匿名原则。

推论 3.1 和 3.2 分别给出了交叉桶泛化算法满足 k -anonymity 和 l -diversity 匿名原则的理论条件。接下来, 我们定义 (k, l) -anonymity 匿名原则, 从而使交叉桶泛化算法同时遵循 k -anonymity 和 l -diversity 匿名原则。

定义 3.7: 对于使用交叉桶泛化算法进行匿名的数据表, 如果匿名数据表遵循 (k, l) -anonymity 匿名原则, 当对任意个体 $t \in T$, 其用户身份泄露的概率最多为 $1/k$, 并且有:

$$p(t, s) \leq \frac{1}{l} \quad (3.14)$$

其中, s 为 t 的敏感值。

3.3 交叉桶泛化算法

在本节中，我们结合泛化算法和桶算法的原理，对遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法进行具体实现。交叉桶泛化算法不仅需要满足 (k, l) -anonymity 匿名原则中对用户身份和敏感属性保护的匿名需求，还需要尽可能地减少匿名数据中数据的损失。因此，交叉桶泛化算法还包括了如下两个目标：(1) 等价组和桶中包含个体的数量尽可能最少，使其正好满足匿名原则中的要求；(2) 等价组中泛化 QI 值的值域范围尽可能缩小，使得泛化 QI 值更加精确。算法 3.1 给出了交叉桶泛化算法的主要描述。

Algorithm 3.1: 交叉桶泛化算法

```

function cross-bucket generalization( $T, k, l$ )
1  $T_{ori} = T$ 
2  $T_{anony} = \emptyset$ 
3  $tuple\_count = |T|$ 
4  $diversity\_num = cal\_diversity(k, l)$ 
5 while  $tuple\_count > 0$  do
6    $S_{set}, loop\_num = cal\_sen\_set(T_{ori}, k, diversity\_num)$ 
7   while  $loop\_num > 0$  do
8      $T_{gen} = pick\_tuples(T_{ori}, S_{set})$ 
9      $bucket\_set = divide\_tuples(T_{gen}, k)$ 
10     $T_{anony} = T_{anony} \cup bucket\_set$ 
11     $T_{ori} = T_{ori} - T_{gen}$ 
12     $tuple\_count = tuple\_count - |T_{gen}|$ 
13     $loop\_num = loop\_num - 1$ 
14  end while
15 end while
16 return  $T_{anony}$ 

```

图 3.1 交叉桶泛化算法

数据结构 T_{ori} 和 T_{anony} 分别用于存储未匿名和已匿名的个体信息（第 1 行和第 2 行）。变量 $tuple_count$ 记录未被匿名的个体数量（第 3 行）。函数 $cal_diversity(k, l)$ 用于计算并返回一个变量 $diversity_num$ （第 4 行），该变量将被用作计算敏感值集合的参数。在每一次循环中（第 5 行到第 15 行），算法首先计算出敏感值集合 S_{set} 和循环次数 $loop_num$ （第 6 行）；然后，在每次循环中（第 7

行到第 14 行), 算法根据 S_{set} 中敏感值的组成从 T_{ori} 中选取适当的个体集合 T_{gen} 进行匿名 (第 8 行); 接下来, 算法将 T_{gen} 中的个体进行泛化并将其划分为许多桶 (第 9 行), 并且将生成的桶集合加入至 T_{anony} 中 (第 10 行); 之后, 算法将泛化的个体集合 T_{gen} 从 T_{ori} 中删除(第 11 行), 以及更新变量 $tuple_count$ 和 $loop_num$ 的值 (第 12 行和第 13 行); 最后, 算法返回 T_{anony} 作为匿名结果 (第 16 行)。

根据算法 3.1 中的描述, 交叉桶泛化算法主要包括三个阶段: 计算敏感值集合阶段 (第 6 行)、选择匿名个体阶段 (第 8 行) 和匿名个体数据阶段 (第 9 行)。接下来, 我们将对每一个阶段进行详细的描述。

3.3.1 计算敏感值集合

在这个阶段中, 交叉桶泛化算法计算出一个敏感值集合及一个循环次数。计算出的敏感值集合结果将被用于在下一个阶段中选取进行匿名的个体集合, 并直到满足循环次数为止。我们对 m -invariance 算法^[113]中分配阶段的算法进行了修改, 用于实现算法 3.1 中的 $cal_sen_set(T, k, diversity_num)$ 函数。算法 3.2 给出了计算敏感值集合算法的主要描述。

Algorithm 3.2: 计算敏感值集合

```

function cal_sen_set( $T, k, diversity\_num$ )
1  $sen\_domain = \{value, count | value \in T[A^{SA}], order by count desc\}$ 
2 if  $|sen\_domain| < 2 * diversity\_num$  then
3    $\beta = |sen\_domain|$ 
4    $\alpha = the\ lease\ count\ in\ sen\_domain$ 
5 else
6    $\beta = diversity\_num$ 
7    $\alpha = calculate\_alpha(sen\_domain, \beta)$ 
8   while  $\alpha$  does not exist do
9      $\beta = \beta + k$ 
10    if  $|sen\_domain| - \beta < 2l$  then
11       $\beta = |sen\_domain|$ 
12       $\alpha = the\ lease\ count\ in\ sen\_domain$ 
13      break
14    end if
15     $\alpha = calculate\_alpha(sen\_domain, \beta)$ 

```

```

16 end while
17 end if
18  $S_{set} = \{the\ first\ \beta\ values\ in\ sen\_domain\}$ 
19 return  $S_{set}, \alpha$ 

```

图 3.2 计算敏感值集合的算法

在算法 3.2 中, 变量 α 和 β 与在 m -invariance 算法中的意义和计算方法相同。输入参数 $diversity_num$ 的值为大于或等于 1 并且能被 k 整除的数, 由算法 3.1 中的函数 $cal_diversity(k, l)$ 计算得出。第 2 行中的条件检查为判断本次计算敏感值集合函数的调用是否为在算法 3.1 中的最后一次。如果本次已经是最后一次调用, 则将所有剩余的敏感值作为敏感值集合的计算结果 (第 3 行和第 4 行)。如果不是, 则将 β 初始化为变量 $diversity_num$ 的值 (第 6 行), 并且 β 的递归增量为 k (第 9 行)。第 10 行中的条件检查为判断剩余敏感值的种类数量是否足够完成下一次的函数调用。如果剩余敏感值的种类数量少于 $2l$, 则将所有剩余的敏感值作为敏感值集合的计算结果 (第 11 行和第 12 行)。

根据算法 3.2 的描述, 敏感值集合 S_{set} 具有如下性质:

性质 3.1: 每个包含于敏感值集合 S_{set} 的敏感值都只出现且仅出现一次。

性质 3.2: 如果敏感值集合 S_{set} 中包含的个体数量能被 k 整除, 则 S_{set} 中包含的个体数量大于 l ; 否则, S_{set} 中包含的个体数量大于 $2l$ 。

3.3.2 选择匿名个体

在这个阶段中, 交叉桶泛化算法根据上一个阶段中计算的敏感值集合 S_{set} , 在还未匿名的个体集合 T_{ori} 中选出合适的个体进行匿名。选出的个体需要与 S_{set} 中的敏感值一一对应, 并且为了提高匿名数据的信息可利用性, 所选个体的 QI 值范围应该尽可能最小。我们对 Mondrian 算法^[96]进行了修改用于实现算法 3.1 中的 $pick_tuples(T, S_{set})$ 函数。算法 3.3 给出了选择匿名个体算法的主要描述。

Algorithm 3.3: 选择匿名个体

```

function pick_tuples( $T, S_{set}$ )
1  $attri\_QI\_set = \{attri | attri \in set\ of\ A^{QI}\}$ 

```

```

2 while attri_QI_set ≠ ∅ do
3   attribute ← choose_attri(T, attri_QI_set)
4   median ← cal_median(T, attribute)
5    $T_l \leftarrow \{t \in T : attribute(t) \leq median\}$ 
6    $T_r \leftarrow \{t \in T : attribute(t) > median\}$ 
7   if check_condition( $T_l, S_{set}$ ) then
8     pick_tuples( $T_l, S_{set}$ )
9     break
10  else if check_condition( $T_r, S_{set}$ ) then
11    pick_tuples( $T_r, S_{set}$ )
12    break
13  else
14    attri_QI_set = attri_QI_set - attribute
15  end if
16 end while
17 if attri_QI_set = ∅ then
18    $T_{gen} = select\_tuples(T, S_{set})$ 
19   return  $T_{gen}$ 
20 end if

```

图 3.3 选择匿名个体的算法

数据结构 $attri_QI_set$ 被初始化为数据表中 QI 属性的集合（第 1 行）。在接下来的每次循环中（第 2 行到第 16 行），交叉桶泛化算法选择一个 QI 属性（第 3 行），并计算出它的中间值 $median$ （第 4 行），然后将数据表 T 分为 T_l 和 T_r 两个子集（第 5 行和第 6 行）。在第 7 行和第 10 行中的函数 $check_condition(T, S_{set})$ 用来检测 T 中是否包含了 S_{set} 中的所有敏感值。如果 T_l 和 T_r 两个子集中任意一个子集满足了 $check_condition(T, S_{set})$ 函数中的条件，则对满足条件的子集调用函数递归（第 8 行和第 11 行）。否则，将该属性从 $attri_QI_set$ 中去除（第 14 行）。而当 $attri_QI_set$ 为空时（第 17 行），则证明已经没有 QI 属性可以将 T 分为更小的子集满足函数 $check_condition(T, S_{set})$ 中的条件。因此，算法通过调用函数 $select_tuples(T, S_{set})$ 从 T 中选取与 S_{set} 中敏感值一一对应的个体集合 T_{gen} （第 18 行）。在第 19 行中，算法将 T_{gen} 作为结果返回，用于在下一个阶段中进行匿名操作。

3.3.3 匿名个体数据

在算法 3.1 中, $divide_tuples(T_{gen}, k)$ 函数包含了两个部分。在第一个部分中, 交叉桶泛化算法将 T_{gen} 进行泛化并使其遵循 k -anonymity 匿名原则。算法 3.4 给出了使用泛化算法对 T_{gen} 进行匿名的主要描述。

Algorithm 3.4: 泛化个体数据

```

function generalize_tuples( $T_{gen}$ ,  $k$ )
1 if  $|T_{gen}|$  is divisible by  $k$  then
2   index = 0
3   while index <  $|T_{gen}|$  do
4     generalize_oper( $T_{gen}$ , index,  $k$ )
5     index = index +  $k$ 
6   end while
7 else
8   group_num =  $|T_{gen}|/k$ 
9   remainder_num =  $|T_{gen}| \% k$ 
10  per_base_num = remainder_num/group_num
11  per_remainder_num = remainder_num%group_num
12  index = 0
13  for i = 0 to per_remainder_num do
14    generalize_oper( $T_{gen}$ , index,  $k + per\_base\_num + 1$ )
15    index = index +  $k + per\_base\_num + 1$ 
16  end for
17  for i = per_remainder_num to group_num do
18    generalize_oper( $T_{gen}$ , index,  $k + per\_base\_num$ )
19    index = index +  $k + per\_base\_num$ 
20  end for
21 end if
22 return  $T_{gen}$ 
```

图 3.4 泛化个体数据的算法

在算法 3.4 中, 先判断 T_{gen} 包含个体的数量是否能被 k 整除 (第 1 行)。如果能被 k 整除, 则将 T_{gen} 平均划分为多个由 k 个个体组成等价组(第 2 行到第 6 行), 其中函数 $generalize_oper(T_{gen} , index, k)$ 表示对 T_{gen} 从下标为 $index$ 且步长为 k 的

个体进行泛化匿名。如果 $|T_{gen}|$ 不能被 k 整除，则先计算 T_{gen} 中可以被划分为等价组的数量、剩余个体的数量和每个等价组需要多包含个体的数量（第8行到第11行），再将 T_{gen} 中的个体尽可能平均地划分成等价组（第12行到第20行）。

命题3.1： T_{gen} 在经过算法3.4匿名之后遵循 k -anonymity匿名原则。

证明：根据算法3.4，每次调用函数 $generalize_oper(T_{gen}, index, step)$ 对 T_{gen} 进行泛化时，步长参数 $step$ 的值一直不小于 k 。因此，经过算法3.4匿名之后， T_{gen} 中每个等价组至少包含了 k 个个体。根据推论3.1， T_{gen} 遵循 k -anonymity匿名原则。

在 $divide_tuples(T_{gen}, k)$ 函数的第二个部分中，交叉桶泛化算法将 T_{gen} 进行桶划分并使其遵循 l -diversity匿名原则。算法3.5给出了使用桶算法对 T_{gen} 进行匿名的主要描述。

Algorithm 3.5: 桶划分算法

```

function partition_tuples( $T_{gen}$ )
1  $bucket\_set = \emptyset$ 
2 for  $i = 0$  to  $k$  do
3    $bucket = \emptyset$ 
4    $j = i$ 
5   while  $j < |T_{gen}|$  do
6     add  $T_{gen}[j]$  to  $bucket$ 
7      $j = j + k$ 
8   end while
9    $bucket\_set = bucket\_set \cup bucket$ 
10 end for
11 return  $bucket\_set$ 
```

图3.5 桶划分算法

数据结构 $bucket_set$ 用于存储被划分后的桶集合（第1行）。在每一次的循环中（第2行到第10行），算法首先生成一个空桶（第3行），并将从第 j 个个体开始并以步长为 k 的所有个体加入至空桶中（第5行到第8行）；之后，算法将

这次生成的桶加入至桶集合 $bucket_set$ 中（第 9 行）；最后，算法将 $bucket_set$ 作为最终的返回结果（第 11 行）。

命题 3.2： T_{gen} 在经过算法 3.5 进行匿名之后遵循 l -diversity 匿名原则。

证明：由于 T_{gen} 与 S_{set} 中的敏感值一一对应，因此 T_{gen} 同样具有性质 3.1 和 3.2。由于 T_{gen} 中任意个体的所有匹配桶都包含于 T_{gen} 中，并且根据性质 3.1， T_{gen} 中的敏感值都只出现且仅出现一次，所以 T_{gen} 中的每个个体都满足推论 3.2 中的第一个条件。

接下来，我们证明 T_{gen} 中的每个个体都满足推论 3.2 中的第二个条件。考虑 T_{gen} 的两种情况：(1) T_{gen} 中包含个体的数量能被 k 整除。根据算法 3.4，每 k 个个体组成一个等价组。在算法 3.5 中，**每个桶中正好包含了每个等价组中的一个个体**。由于每个等价组都包含了 k 个个体，因此有：

$$p(t, MB) = \frac{1}{k} \quad (3.15)$$

和

$$|MB| = \frac{|T_{gen}|}{k} \quad (3.16)$$

根据性质 3.2， T_{gen} 中包含个体的数量不少于 l ，因此有：

$$\frac{p(t, MB)}{|MB|} = \frac{1}{k|MB|} = \frac{1}{\frac{|T_{gen}|}{k}} \leq \frac{1}{l} \quad (3.17)$$

(2) T_{gen} 中包含个体的数量不能被 k 整除。根据算法 3.4， T_{gen} 中包含了两种个体数量不同的等价组，分别为 $k+per_base_num$ 和 $k+per_base_num+1$ 。并且，由于 $remainder_num$ 小于 k ，所以这两种等价组包含的个体数量都小于 $2k$ 。在算法 3.5 中，每个桶中最多包含两个个体在同一个等价组中，即有

$$p(t, MB) \leq \frac{2}{k} \quad (3.18)$$

和

$$|MB| \geq \frac{|T_{gen}|}{k} \quad (3.19)$$

根据性质 3.2， T_{gen} 中包含个体的数量大于 $2l$ ，因此有

$$\frac{p(t, MB)}{|MB|} \leq \frac{2}{k|MB|} \leq \frac{2}{|T_{gen}|} \leq \frac{1}{l} \quad (3.20)$$

综上所述, T_{gen} 中的每个个体都满足推论 3.2 中的两个条件, 因此 T_{gen} 在经过算法 3.5 进行匿名之后遵循 l -diversity 匿名原则。

3.4 实验分析

在本节中, 对第 3 节实现的交叉桶泛化算法的匿名效果进行测试和分析。实验数据取自最新的美国人口普查数据⁵, 并且从中删除了丢失属性值的个体, 随机选择了 22,517 个个体及 9 个属性作为实验对象, 其中, QI 属性包括了性别、年龄、家庭关系、婚姻状况、种族、教育情况、每周工作时长和职业, 敏感属性为薪水。表 3.7 详细地介绍了这些属性的相关信息。

表 3.7 属性的描述

序号	属性名称	数值类型	值域大小
1	Sex	分类类型	2
2	Age	数字类型	73
3	Relationship	分类类型	13
4	Marital status	分类类型	6
5	Race	分类类型	9
6	Education	分类类型	11
7	Hours per week	数字类型	90
8	Occupation	分类类型	216
9	Salary	数字类型	702

我们通过实现 Mondrian 算法^[96]和 Anatomy 算法^[15]与交叉桶泛化算法进行对比, 其中, 对比内容包括了敏感属性保护和信息可利用性两方面。除此之外, 还对 (k, l) -anonymity 匿名原则中参数 k 和 l 的变化对交叉桶泛化算法的匿名影响进行了研究。

⁵ <https://usa.ipums.org/usa/index.shtml>

3.4.1 敏感属性保护

在测试敏感属性保护的实验中，我们假定最坏的情况：攻击者已知目标用户的信息存在于匿名表中，并且已经获得目标用户的所有 QI 值信息。攻击者将通过目标用户的 QI 值在匿名表中进行匹配，并且试图获得目标用户的敏感属性值。我们将对原始数据中的所有用户进行测试，并且计算敏感属性值暴露概率的平均值进行比较。

在没有使用桶算法匿名的数据表中，对于任意一个个体 t ，计算 t 的匹配个体的数量，记作 $Num_{tuples}(t)$ ，以及在匹配个体中带有 t 的敏感值 s 的数量，记作 $Num_{sen}(t)$ ，有

$$p(t, s) = Num_{sen}(t) / Num_{tuples}(t) \quad (3.21)$$

而在使用桶算法匿名的数据表中，计算 $Num_{tuples}(t)$ ， MB 中 t 的匹配个体的数量，记作 $Num(t, MB)$ ，以及 t 的敏感值 s 在 MB 中出现的次数 $Num(s, MB)$ 。

根据式 (3.5)，有

$$p(t, s) = \sum_{MB} \frac{Num(t, MB) * Num(s, MB)}{Num_{tuples}(t) * |MB|} \quad (3.22)$$

其中， $|MB|$ 为 MB 包含个体的数量。

我们使 Mondrian 算法和 Anatomy 算法都遵循 l -diversity 匿名原则，交叉桶泛化算法遵循 (k, l) -anonymity 匿名原则，并将参数 k 的值固定为 3。图 3.6 显示了使用各个匿名算法时敏感属性值暴露概率的结果。

从图 3.6 中可以发现，Anatomy 算法发挥了遵循 l -diversity 匿名原则的预期表现。但是，由于遵循 l -diversity 匿名原则的 Mondrian 算法在等价组中包含了更多的个体数量，因此其保护效果要比 Anatomy 算法更加出色。交叉桶泛化算法的保护效果比前两者更加优秀，这是由于在匿名表中等价组包含的个体被分配至不同的桶中，所以当攻击者通过目标用户的 QI 值进行匹配时，会得到很多目标用户的匹配桶，从而增加了获取敏感值的难度。

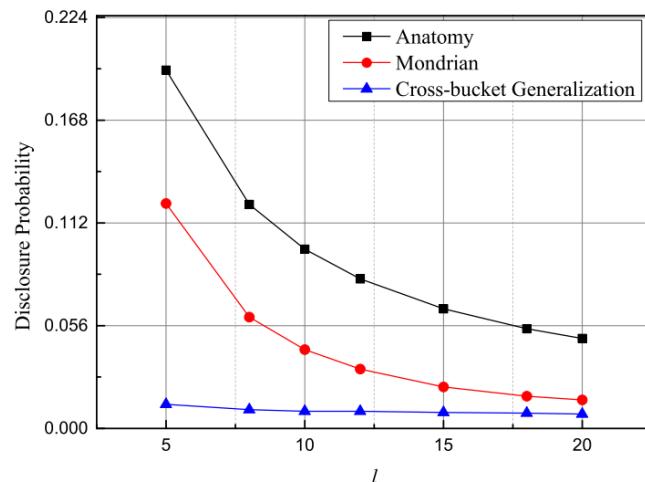


图 3.6 敏感属性值暴露的概率

3.4.2 信息可利用性

在本次实验中，采用了与之前相同的实验测试环境，即令 Mondrian 算法遵循 l -diversity 匿名原则，交叉桶泛化算法遵循 (k, l) -anonymity 匿名原则，并将参数 k 的值设置为 3。我们首先对交叉桶泛化算法和 Mondrian 算法的鉴别力度量进行测试。图 3.7 即为使用这两个算法对数据表进行匿名之后的鉴别力度量惩罚指数结果。

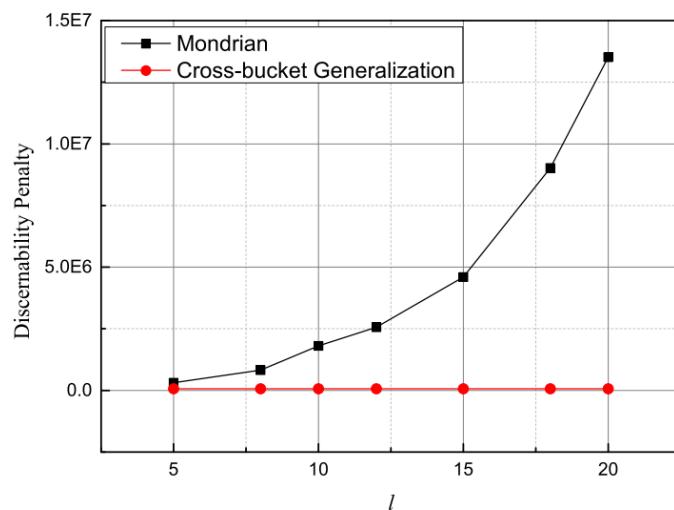


图 3.7 鉴别力度量惩罚指数

从图 3.7 中可以发现，随着参数 l 的增加，Mondrian 算法的鉴别力度量惩罚指数上升非常快，已经和交叉桶泛化算法的结果产生了量级的差距。因此，对于使用交叉桶泛化算法进行匿名的数据表，其等价组包含个体的数量远远小于使用 Mondrian 算法进行匿名时的情况。除此之外，在图 3.7 中我们还可以发现，交叉桶泛化算法的鉴别力度量惩罚指数一直保持不变，即对于使用交叉桶泛化算法进行匿名的数据表，其等价组包含个体的数量几乎不会被参数 l 所影响。因此，可以推断当改变对敏感属性的保护强度时，对用户身份的保护效果不会受到影响。该推断将在 3.4.3 节中的实验得到进一步的证实。

接下来，我们使用文献[54]中查询回答错误率的方法，对交叉桶泛化算法的匿名效果进行测试。通过随机生成 1000 条查询语句对每个匿名表计算查询回答错误率的平均值，并将结果进行对比。查询语句的形式如下：

```
SELECT SUM(salary) FROM Microdata
WHERE pred( $A_1^{QI}$ ) AND pred( $A_2^{QI}$ ) AND pred( $A_3^{QI}$ ) AND pred( $A_4^{QI}$ )
```

其中，查询语句的条件中随机包含了四个 QI 属性，并且将薪水的总和作为比较结果。对于分类 QI 属性， $\text{pred}(A^{QI})$ 表示为：

$$(A^{QI} = v_1 \text{ or } A^{QI} = v_2 \text{ or } \dots \text{ or } A^{QI} = v_m) \quad (3.23)$$

其中， $v_i (1 \leq i \leq m)$ 为在 $D[A^{QI}]$ 中的随机值。而对于数字 QI 属性， $\text{pred}(A^{QI})$ 表示为：

$$\begin{aligned} & (A^{QI} > v) \text{ or } (A^{QI} < v) \text{ or } (A^{QI} = v) \text{ or } (A^{QI} \geq v) \\ & \text{or } (A^{QI} \leq v) \text{ or } (A^{QI} \neq v) \end{aligned} \quad (3.24)$$

其中， v 为在 $D[A^{QI}]$ 中的随机值。

查询回答错误率的公式为：

$$Sum_{error} = (Sum_{upper} - Sum_{lower}) / Sum_{act} \quad (3.25)$$

其中 Sum_{upper} 和 Sum_{lower} 分别为薪水总和的上限和下限，并且 Sum_{act} 为薪水的实际总和。图 3.8 显示了使用各个匿名算法时查询回答错误率的结果。

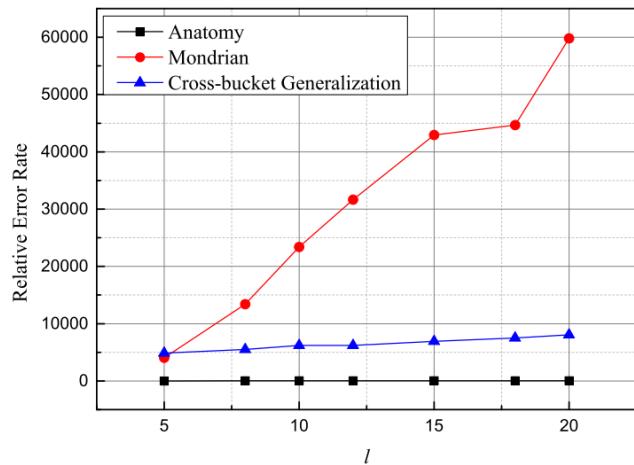


图 3.8 查询回答错误率

通过对图 3.8 的观察可以发现交叉桶泛化算法的查询回答错误率高于 Anatomy 算法。这是由于 Anatomy 算法完全保留了原始数据表中的 QI 值信息，使得匿名数据表获得了接近于原始数据表的信息可利用性。但是，交叉桶泛化算法不仅实现了保护用户身份的功能，而且当随着参数 l 的不断增加时，交叉桶泛化算法的查询回答错误率上升的非常缓慢，以至于远远低于 Mondrian 算法。因此，综合上述实验结果来看，当匿名数据表需要同时为用户身份和敏感属性提供保护时，交叉桶泛化算法是相对于 Mondrian 算法和 Anatomy 算法更加合适的选择。

3.4.3 参数的影响

之前的实验将交叉桶泛化算法的匿名效果与 Mondrian 算法和 Anatomy 算法进行了比较。而本节的实验将通过调整参数 k 和 l 的值，对遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法产生的影响进行测试和分析。参数 l 的值分别固定为 5、10、15 和 20，其他实验环境与之前的实验相同。相关的实验结果分别如图 3.9、3.10 和 3.11。

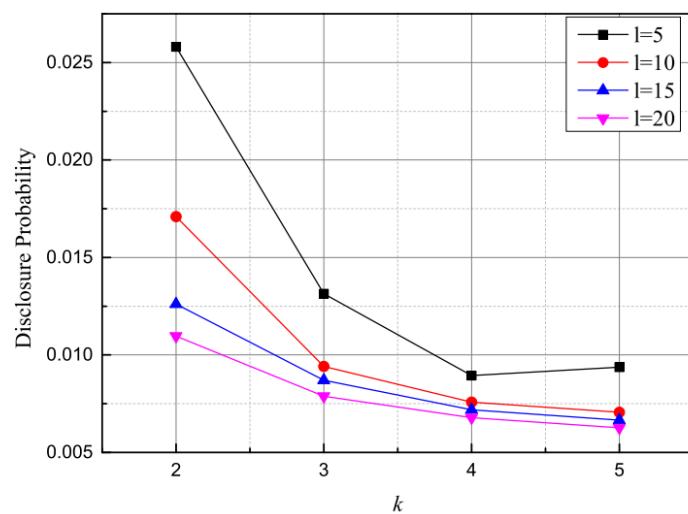


图 3.9 不同参数下敏感属性值暴露的概率

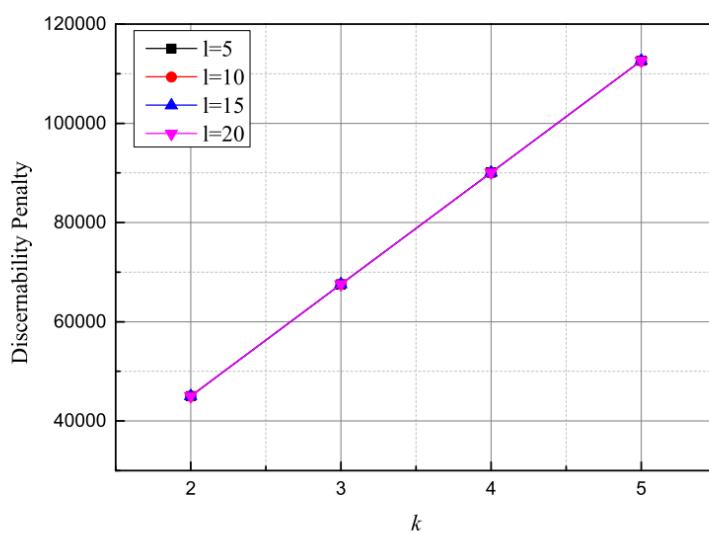


图 3.10 不同参数下辨别力度量惩罚指数

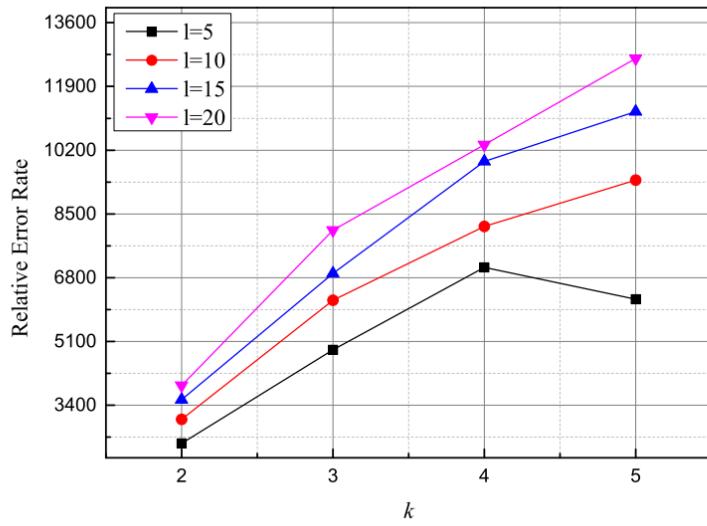


图 3.11 不同参数下查询回答错误率

由图 3.9 中可以发现当敏感值暴露的概率在 0.01 以上时，提高参数 k 的值对提高敏感属性的保护效果具有比较显著的作用。在图 3.10 中，四条线已经完全重合。这表明对于使用遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法进行匿名的数据表，等价组包含个体的数量完全取决于参数 k 的值，而与参数 l 的值无关。在图 3.11 中可以发现，当参数 l 的值越大时，随着参数 k 的值增加，查询回答错误率的上升也越快。

3.5 本章小结

在本章中，通过结合泛化算法和桶算法的原理提出了交叉桶泛化算法，用于保护数据表中的用户身份和敏感属性。并且，遵循 (k, l) -anonymity 匿名原则的交叉桶泛化算法可以分别为用户身份和敏感属性提供安全的隐私保护，从而解决了遵循 l -diversity 匿名原则的泛化算法对用户身份产生过度保护的问题。相比于一般的泛化算法，交叉桶泛化算法不仅对敏感属性具有更好的保护，而且具有更加出色的信息可利用性。在将来的工作中，我们将把交叉桶泛化算法应用于更加广泛的发布环境中，如多重发布^[114]、连续发布^[115]和个人隐私发布^[105]等。

第4章 个性化的隐私保护

4.1 引言

至今为止，虽然匿名保护技术已经被广泛地研究和应用，但是在几乎所有的匿名方法中都忽略了对于用户个性化隐私保护的问题。在一般的匿名方法中，属性的类型可以被分为标识符、准标识符和敏感属性三种，并且属性类型的划分是由数据发布者决定的，但事实上，一个数据值的敏感性应该由用户而不是数据发布者决定的。如果没有对用户的个性化隐私需求进行保护，则极有可能使用户的敏感信息泄露。在本章中，我们将基于桶算法的技术原理，提出一种新的匿名技术，称为局部分解算法，满足用户的个性化隐私保护需求。通过局部分解算法匿名之后的数据表，不仅可以对用户的所有敏感值提供有效地保护，并且尽可能保留了匿名数据中的信息可利用性。

4.1.1 问题的提出

不论是遵循 k -anonymity 匿名原则的泛化算法，还是遵循 l -diversity 匿名原则的桶算法，这些传统的匿名技术都是以属性作为数据敏感性的单位，但是，在实际生活中，人们是将数据值作为敏感性的单位而非属性。例如，对于职业而言，当某人的职业为程序员时，其职业为非敏感值，而当某人的职业为保镖时，其职业则为敏感值。因此，当考虑了个性化隐私保护的情况时，一个属性可能会同时包含敏感值和非敏感值。我们将这类属性称为半敏感属性 (Semi-Sensitive Attribute)。

如表 4.1 所示，在考虑了个性化隐私保护的情况下，数据表中除了姓名属性之外，其他属性都包含了一个用于记录该属性中的值是否为敏感值的 Flag 标识。值得注意的是，在表 4.1 的职业属性中，尽管 Mark 和 Sarah 的职业都为律师，但是 Sarah 并不关心她的职业是否会被其他人知道，而 Mark 则希望将他的职业信息作为个人隐私。

通过对表 4.1 的观察可以发现，年龄属性和职业属性都同时包含了敏感值和

非敏感值，因此它们都属于半敏感属性。但是，一般的泛化算法和桶算法只能将半敏感属性作为 QI 属性处理。通过泛化算法和桶算法对表 4.1 进行匿名之后的数据表，如表 4.2 和 4.3 所示。

表 4.1 原始数据表

Name	Age (Flag)	Gender (Flag)	Occupation (Flag)	Disease (Flag)
Mark	26 (No)	Male (No)	Lawyer (Yes)	Pneumonia (Yes)
Dave	35 (No)	Male (No)	Police (Yes)	Dyspepsia (Yes)
Ella	16 (No)	Female (No)	Student (No)	Flu (Yes)
Daphne	24 (Yes)	Female (No)	Guider (No)	Bronchitis (Yes)
Sarah	31 (Yes)	Female (No)	Lawyer (No)	Hepatitis (Yes)
Neil	22 (No)	Male (No)	Typist (No)	Dyspepsia (Yes)
Dean	29 (Yes)	Male (No)	Guard (Yes)	Bronchitis (Yes)
Tina	34 (Yes)	Female (No)	Scientist (Yes)	Gastritis (Yes)

表 4.2 遵循 4-anonymity 匿名原则的泛化数据表

ID	Group ID	Age	Gender	Occupation	Disease
1001	1	[16–35]	*	*	Pneumonia
1002	1	[16–35]	*	*	Dyspepsia
1003	1	[16–35]	*	*	Flu
1004	1	[16–35]	*	*	Bronchitis
1005	2	[22–34]	*	*	Hepatitis
1006	2	[22–34]	*	*	Dyspepsia
1007	2	[22–34]	*	*	Bronchitis
1008	2	[22–34]	*	*	Gastritis

由于泛化算法会将 QI 属性和半敏感属性中的值进行泛化，因此泛化算法会对半敏感属性中的敏感值提供一定程度的保护，然而，泛化算法会损失大量的数据信息，甚至包括很多敏感值信息。如表 4.2 所示，除敏感属性外，大部分的值已经被泛化而无法被用于进行数据分析。相比之下，桶算法只对敏感属性中的敏感值提供了匿名保护。因此，尽管桶算法保留了非常优秀的信息可利用性，却完

全忽略了个性化隐私保护的需求。当攻击者拥有足够的背景知识时，目标用户在半敏感属性中的敏感值几乎都会被泄露。例如，当攻击者得知 Mark 的信息存在于表 4.3 中，并且已知 Mark 的年龄和性别时，攻击者将推断出 Mark 的职业是一名律师。

表 4.3 遵循 4-diversity 匿名原则的桶数据表

ID	Bucket ID	Age	Gender	Occupation	Disease
1001	1	26	Male	Lawyer	Bronchitis
1002	1	35	Male	Police	Dyspepsia
1003	1	16	Female	Student	Flu
1004	1	24	Female	Guider	Pneumonia
1005	2	31	Female	Lawyer	Bronchitis
1006	2	22	Male	Typist	Dyspepsia
1007	2	29	Male	Guard	Gastritis
1008	2	34	Female	Scientist	Hepatitis

4.1.2 本章工作

为了在个性化隐私保护的发布环境中，对数据表中的敏感值提供可靠的保护，并且尽可能减少数据表在匿名过程中的信息损失，我们基于桶算法的原理提出一个被称为局部分解算法（Local Anatomy）的新型算法。局部分解算法的基本思想是以数据表中的属性值作为敏感单位，并在每个敏感属性和半敏感属性中，将把属性值视为敏感值的用户划分为桶，从而保护用户的敏感值。通过这种方法，局部分解算法可以只针对数据表中的敏感值进行匿名保护，并且保留原始的非敏感值。因此，局部分解算法不仅可以为敏感值提供安全的保护，还尽可能保留了数据表中的信息可利用性。通过局部分解算法对表 4.1 进行匿名后的结果，如表 4.4 所示。

在表 4.4 中，所有敏感属性和半敏感属性都具有“Bucket ID”标识，如年龄、职业和疾病。在这些属性中，所有将属性值当作敏感值的用户被划分为桶，而将该属性值当作非敏感值的用户则保留其原始值。例如，根据表 4.1，Sarah 将她的年龄和疾病当作敏感值，而性别和职业当作非敏感值。在表 4.4 中，Sarah

的 ID 标识为 1005，且年龄值和疾病值分别在相应属性中“Bucket ID”为 1 和 2 的桶内，即她的年龄值可能为 24、29、31 和 34，疾病值可能为支气管炎、消化不良、胃炎和肝炎。此外，由于 Sarah 的性别和职业属于非敏感值，所以这两个值在匿名数据表中被完整保留。

表 4.4 局部分解数据表

ID	Age (Bucket ID)	Gender	Occupation (Bucket ID)	Disease (Bucket ID)
1001	26 (-)	Male	Guard (1)	Bronchitis (1)
1002	35 (-)	Male	Lawyer (1)	Dyspepsia (1)
1003	16 (-)	Female	Student (-)	Flu (1)
1004	24 (1)	Female	Guider (-)	Pneumonia (1)
1005	29 (1)	Female	Lawyer (-)	Bronchitis (2)
1006	22 (-)	Male	Typist (-)	Dyspepsia (2)
1007	31 (1)	Male	Police (1)	Gastritis (2)
1008	34 (1)	Female	Scientist (1)	Hepatitis (2)

在本章的研究中，假设数据表的发布环境为：数据表中的任意属性可能为 QI 属性、半敏感属性或者敏感属性，并且一个原始数据表由多个 QI 属性和半敏感属性以及一个敏感属性组成。此外，考虑在最坏的情况下，攻击者已经知道目标用户的信息存在于数据表中，以及目标用户的所有 QI 值信息，但攻击者还没有获得目标用户在半敏感属性和敏感属性中的敏感值。我们需要对数据表中的所有敏感值进行保护，并且尽量保留更多的数据可利用性。本章的具体工作如下：

首先，我们提出局部分解算法用于保护数据表中用户的敏感值。该算法将每个半敏感属性和敏感属性中携带敏感值的用户划分为桶，从而为每个属性中的敏感值提供了独立的保护。此外，由于局部分解算法完整地保留了数据表中所有原始的 QI 值，所以相比于泛化算法，匿名之后的数据损失将大大减少。

其次，我们分析局部分解算法保护敏感值的原理，并且使局部分解算法遵循 l -diversity 匿名原则以保证匿名表中所有敏感值暴露的概率不高于 $1/l$ 。但是，由于 l -diversity 匿名原则具有局限性，所以遵循 l -diversity 匿名原则的局部分解算法并不一定适合所有匿名发布的情况。为此，我们基于局部分解算法可以

为每个属性提供相互独立保护的特性改进局部分解算法，使其可以同时遵循多个不同的匿名原则对数据表中的敏感值进行保护。

第三，我们基于桶算法的原理，具体实现遵循 l -diversity 匿名原则的局部分解算法。并且，为了增加匿名数据表中的信息可利用性，我们提出一种启发式将每个桶中包含的个体数量尽量降低，并且尽可能缩小每个桶中敏感值的值域范围。

最后，我们进行大量的实验，通过对比传统的泛化算法和桶算法测试局部分解算法的实际匿名效果。其中，对比的主要内容包括：(1) 局部分解算法相比桶算法可以更加安全地保护数据表中的所有敏感值；(2) 通过对查询回答错误率的结果，局部分解算法相比泛化算法极大地提高了信息可利用性。此外，我们还通过调整数据表中半敏感属性的敏感值密度，对遵循 l -diversity 匿名原则的局部分解算法产生的匿名影响进行研究。

本章的结构如下：在 4.2 节中，将介绍局部分解算法的基本概念并且分析对敏感值保护的工作原理；在 4.3 节中，将对遵循 l -diversity 匿名原则的局部分解算法进行具体实现，并提出一个启发式用于增加匿名数据表中的信息可利用性；在 4.4 节中，将对局部分解算法的相关匿名效果进行实验测试和分析；在 4.5 节中，将继续改进了局部分解算法，使其同时遵循了 l -diversity 和 t -closeness 匿名原则；在 4.6 节中，将对本章内容进行总结。

4.2 局部分解算法模型

4.2.1 基本概念

在对局部分解算法进行定义之前需要引入一些基本概念。首先，我们将根据包含数据值的类型对属性类型重新进行定义。

定义 4.1：如果一个属性是 QI 属性，记作 A^{QI} ，当且仅当这个属性只包含 QI 值。

定义 4.2：如果一个属性是敏感属性，记作 A^{SA} ，当且仅当这个属性只包含敏感值。

定义 4.3: 如果一个属性是半敏感属性, 记作 A^{SS} , 当且仅当这个属性同时包含了 QI 值和敏感值。

在实际应用中, 数据的敏感性应该由用户决定, 而数据发布者应该只负责满足数据表中用户的隐私保护需求, 并且尽可能提高匿名数据表的信息可利用性。根据定义 4.1、4.2 和 4.3 对属性类型的重新划分, 不仅使用户可以根据自身意愿自由设置敏感值, 还使得数据发布者可以根据属性的不同类型分别使用合适的算法进行匿名处理。

由定义 3.3 可以发现, 一般的桶算法只能用于保护敏感属性而无法保护半敏感属性中的敏感值。因此, 我们基于新的属性类型的定义提出局部分解算法对数据表中所有的敏感值进行保护。

定义 4.4: 对于数据表 T 中任意半敏感属性 A^{SS} 和敏感属性 A^{SA} , 所有将属性值当作敏感值的用户被划分为桶, 并且每个桶包含 $IDT(ID, BID)$ 和 $SAT(SA, BID)$ 两个部分, 其中, ID 和 SA 分别为在桶中所有个体的 ID 标识和敏感值, BID 表示在相应属性内桶的 ID 标识。

值得注意的是, 一般的桶算法可以被当作局部分解算法的一种特殊情况。当数据表中只包含多个 QI 属性和一个敏感属性时, 局部分解算法等价于一般的桶算法。

根据定义 4.4 可以发现, 在不同属性中的桶是相互独立的, 而且如果一个个体包含了多个敏感值, 则这个个体的各个敏感值会被分别包含在相应属性的桶中。由于各个桶之间是相互独立的, 因此即使目标用户的部分敏感值已被泄露, 其余的敏感值也不会受到影响。例如在表 4.4 中, 对于 ID 为 1005 的个体, 每个敏感值都被包含在相应属性中的一个桶中, 即年龄值和疾病值分别包含在相应属性内“Bucket ID”为 1 和 2 的桶中。即使攻击者已经知道了她的年龄值, 也不会推测出正确的疾病值。

4.2.2 敏感值保护分析

在本节中, 我们将详细地分析局部分解算法为匿名数据表中的敏感值提供保

护的工作原理。假设攻击者已经知道目标用户 t 的所有 QI 值信息，以及 t 的信息存在于匿名数据表 T 中，并且，攻击者试图通过 t 的 QI 值信息在匿名数据表 T 中进行匹配，从而得到 t 的所有敏感值信息。

定义 4.5：对于任意个体 $t \in T$ ，如果其每个 QI 值与另一个个体 $mt \in T$ 的相应 QI 值相等，则称 mt 为 t 的一个匹配个体。

定义 4.6：对于任意个体 $t \in T$ ，如果某属性中的一个桶 mb 中至少包含了 t 的一个匹配个体，则称 mb 为 t 的一个匹配桶。

值得注意的是，个体的匹配个体和匹配桶是随着攻击者的背景知识改变的。假设攻击者仅已知目标用户的相关信息。例如，在表 4.1 中，攻击者仅知道 Dean 的性别为男性，并且 Dean 的数据信息存在于数据表 4.4 中。攻击者通过匹配其 QI 值可以得到 ID 为 1001、1002、1006 和 1007 的个体为 Dean 的匹配个体。进一步，攻击者可以判断年龄和职业属性中的匹配桶分别是“Bucket ID”为 1 和 1 的桶，并且由于在疾病属性中“Bucket ID”为 1 和 2 的桶中均包含了 Dean 的匹配个体，因此在疾病属性中“Bucket ID”为 1 和 2 的桶都为 Dean 的匹配桶。

接下来，考虑一种更坏的情况，即攻击者已知数据表中所有用户的 QI 值信息，并且知道所有用户的信息均存在于匿名数据表中。在这种情况下，攻击者拥有更多的背景知识，从而可以帮助攻击者排除更多的匹配个体。例如，将表 4.1 中的所有 QI 信息都作为攻击者的背景知识，则攻击者可以通过在表 4.4 中匹配各个个体的 QI 值信息推断出 ID 为 1001、1002 和 1006 的个体分别为 Mark、Dave 和 Neil，所以 Dean 的匹配个体只剩下 ID 为 1007 的个体，并且 Dean 在疾病属性中的匹配桶仅为“Bucket ID”为 1 的桶。

在一般情况下，任意一个个体在匿名数据表中至少有一个匹配个体，并且在其敏感值相应的属性中至少有一个匹配桶。因此，有如下定义和定理。

定义 4.7：对任意个体 $t \in T$ ，其敏感值所在相应属性中桶 b 的概率，记为 $p(t, b)$ 。

定理 4.1：在一个使用局部分解算法进行匿名的数据表中，对于任意个体

$t \in T$, 其任意敏感值 s 暴露的概率满足:

$$p(t, s) \leq \sum_{mb} p(t, mb) \frac{|mb(s')|}{|mb|} \quad (4.1)$$

其中, $|mb(s')|$ 为在匹配桶 mb 中出现次数最多的敏感值 s' 的数量, 并且 $|mb|$ 为 mb 中包含个体的数量。

证明: 不失一般性, 目标个体 t 的敏感值 s 在敏感属性或者半敏感属性中。由于目标个体 t 的敏感值 s 包含于相应属性中的某个桶内, 攻击者首先需要计算个体 t 在 s 的相应属性中每个桶的定位概率, 以及个体 t 在每个桶中带有敏感值 s 的概率。因此, 攻击者有

$$p(t, s) = \sum_b p(t, b)p(s|t, b) \quad (4.2)$$

其中, $p(s|t, b)$ 为当个体 t 在桶 b 中时敏感值为 s 的概率。由于攻击者可以根据目标用户的 QI 值在数据表中进行匹配, 因此攻击者可以排除那些不包含个体 t 的匹配个体的桶, 即当不存在 $mt \in b$ 时, 有

$$p(t, b) = 0 \quad (4.3)$$

根据定义 4.6, 有

$$p(t, s) = \sum_{mb} p(t, mb)p(s|t, mb) \quad (4.4)$$

其中, mb 为个体 t 的匹配桶。对任意 mb 中出现次数最高的敏感值 s' , 有:

$$|mb(s)| \leq |mb(s')| \quad (4.5)$$

因此, 有

$$p(s|t, mb) = \frac{|mb(s)|}{|mb|} \leq \frac{|mb(s')|}{|mb|} \quad (4.6)$$

根据式 (4.6), 有

$$p(t, s) \leq \sum_{mb} p(t, mb) \frac{|mb(s')|}{|mb|} \quad (4.7)$$

推论 4.1: 对于使用局部分解算法进行匿名的数据表, 如果匿名数据表遵循 l -diversity 匿名原则, 则对数据表中的任意个体需要满足如下条件: (1) 在匿名数据表的所有匹配桶中, 每个敏感值仅出现一次; (2) 所有匹配桶至少包含 l 个个体。

证明: 根据条件 (1), 我们将每个桶中出现的敏感值的数量限制为 1, 因此对任意 $s \in mb$, 有

$$|mb(s)| = 1 \quad (4.8)$$

根据条件(2), 对 $\forall b \subseteq T$, 我们有

$$|b| \geq l \quad (4.9)$$

因此, 根据式(4.1)有

$$p(t, s) \leq \frac{1}{l} \sum_{mb} p(t, mb) = \frac{1}{l} \quad (4.10)$$

综上, 当满足条件(1)和(2)时, 局部分解算法遵循 l -diversity 匿名原则。

推论4.1给出了当局部分解算法遵循 l -diversity 匿名原则时需要满足的条件。接下来, 我们给出遵循 l -diversity 匿名原则的局部分解算法的正式定义:

定义4.8: 对于使用局部分解算法进行匿名的数据表, 如果匿名数据表遵循 l -diversity 匿名原则, 当对任意个体 $t \in T$, 其任意敏感值 s 泄露的概率满足:

$$p(t, s) \leq \frac{1}{l} \quad (4.11)$$

值得注意的是, 由于 l -diversity 匿名原则自身的局限性, 遵循 l -diversity 匿名原则的局部分解算法并非适用于所有发布环境。在4.5节中, 将对此情况进行扩展讨论, 并且提出一种可以根据数据表中不同属性的特性, 任意使用匿名原则对敏感值进行保护的方法。

4.3 局部分解算法

在本节中, 我们基于桶算法的原理对遵循 l -diversity 匿名原则的局部分解算法进行具体实现。局部分解算法需要将半敏感属性和敏感属性中所有携带敏感值的个体划分为桶, 并且每个桶之间是相互独立的。算法的主要描述如算法4.1所示。

Algorithm 4.1: 局部分解算法
function local_anatomy(T, l)
1 $Attri_{sen} = \{attributes\ including\ sensitive\ values\ in\ T\}$

```

2    $T_{anony} = T$ 
3   for each  $attribute \in Attris_{sen}$  do
4      $ValuePair_{sen} = \{(id, s) | s \text{ is a sensitive value in attribute}\}$ 
5      $anatomize(T_{anony}, ValuePair_{sen}, l)$ 
6   end for
7   return  $T_{anony}$ 

```

图 4.1 局部分解算法

数据结构 $Attris_{sen}$ （第 1 行）用于存储数据表 T 中包含敏感值的属性，即敏感属性和半敏感属性的集合。变量 T_{anony} （第 2 行）被初始化为 T ，用于存储匿名之后的结果。在每次循环中（第 3 行到第 6 行），局部分解算法首先依次统计 $Attris_{sen}$ 中带有敏感值的个体信息 $ValuePair_{sen}$ （第 4 行）；然后，根据参数 l 的值，在 T_{anony} 中对 $ValuePair_{sen}$ 包含的个体进行桶划分（第 5 行）；最后，返回 T_{anony} 作为匿名结果（第 7 行）。值得注意的是，根据算法 4.1，由于 $Attris_{sen}$ 中不包括 QI 属性，并且 $ValuePair_{sen}$ 中也没有包含带有 QI 值的个体，因此数据表中所有原始的 QI 值都将被完整地保留。

在算法 4.1 中，我们使用了 m -invariance 算法^[113]中的分配算法对函数 $anatomize(T, ValuePair, l)$ 进行具体实现将 $ValuePair$ 中的个体划分为多个桶。但是，为了在匿名数据表中保留更多的信息可用性，还可以通过在函数 $anatomize(T, ValuePair, l)$ 中加入启发式使每个桶中包含敏感值的值域范围尽可能缩小，并且减少每个桶中包含个体的数量。算法 4.2 给出了加入启发式版本的函数 $anatomize(T, ValuePair, l)$ 的主要描述。

Algorithm 4.2: 桶划分启发式

```

function  $anatomize(T, ValuePair, l)$ 
1    $value\_count = \{(value, number) | counter by ValuePair\}$ 
2    $median \leftarrow cal\_median(value\_count)$ 
3    $VP_{small} = \{(id, s) | (id, s) \in ValuePair \text{ and } s \leq median\}$ 
4    $VP_{big} = \{(id, s) | (id, s) \in ValuePair \text{ and } s > median\}$ 
5   if  $check\_condition(VP_{small}, l)$  and  $check\_condition(VP_{big}, l)$  then
6      $anatomize(T, VP_{small}, l)$ 
7      $anatomize(T, VP_{big}, l)$ 
8   else
9      $divide\_oper(T, ValuePair, l)$ 

```

```
10 end if
```

图 4.2 桶划分启发式

根据算法 4.2，启发式每次调用都将 $ValuePair$ 划分为更小的两个子集，并且这两个子集包含的敏感值集合是不相交的。数据结构 $value_count$ （第 1 行）用于存储 $ValuePair$ 中每个敏感值及数量。变量 $median$ （第 2 行）用于存储 $value_count$ 中敏感值的中间值。启发式通过中间值 $median$ 将 $ValuePair$ 分为 VP_{small} 和 VP_{big} 两个子集（第 3 行和第 4 行）。函数 $check_condition(ValuePair, l)$ 用于测试 $ValuePair$ 是否满足 l -diversity 匿名原则。如果 VP_{small} 和 VP_{big} 都满足条件，则 VP_{small} 和 VP_{big} 将分别继续递归调用 $anatomize(T, ValuePair, l)$ 函数（第 6 行和第 7 行）。否则，将 $ValuePair$ 中的个体划分为桶（第 9 行），其中，函数 $divide_oper(T, ValuePair, l)$ 可以通过 m -invariance 算法中的分配算法进行具体实现。

命题 4.1：在算法 4.1 中，匿名结果 T_{anony} 遵循 l -diversity 匿名原则。

证明：根据算法 4.1，所有携带敏感值的个体都在相应的属性中被划分为桶。并且，根据文献[113]，所有的桶都满足“ m -unique”，即每个桶中至少包含了 m 个个体，而且每个个体都带有不同的敏感值。在算法 4.1 中，无论函数 $anatomize(T, ValuePair, l)$ 是否加入算法 4.2 中的启发式，所有的桶划分算法都是由 m -invariance 算法中分配阶段的算法进行具体实现，并且将 l 作为参数赋值给 m 。因此，算法 4.1 中的匿名结果 T_{anony} 满足推论 4.1 中的两个条件，从而使其遵循 l -diversity 匿名原则。

4.4 实验分析

在本节中，我们将对 4.3 节实现的局部分解算法的匿名效果进行评估。实验数据下载于最新的美国人口普查数据，从中删除了丢失属性值的个体，并随机选择了 17,629 个个体及 9 个属性作为实验对象，其中，QI 属性包括了性别、家庭关系、婚姻状况、种族、教育情况和每周工作时长，半敏感属性包括年龄和职业，敏感属性为薪水。表 4.5 详细地介绍这些属性的相关信息。

表 4.5 属性的描述

序号	属性名称	数值类型	敏感类型	值域大小
1	Sex	分类类型	QI	2
2	Age	数字类型	半敏感属性	73
3	Relationship	分类类型	QI	13
4	Marital status	分类类型	QI	6
5	Race	分类类型	QI	9
6	Education	分类类型	QI	11
7	Hours per week	数字类型	QI	89
8	Occupation	分类类型	半敏感属性	187
9	Salary	数字类型	敏感属性	682

我们通过实现 Mondrian 算法^[96]和 Anatomy 算法^[15]与局部分解算法进行对比。其中，对比内容包括敏感值保护和信息可利用性两方面。除此之外，还通过调整半敏感属性中敏感值的密度，对遵循 l -diversity 匿名原则的局部分解算法产生的匿名影响进行研究。

4.4.1 敏感值保护

在测试敏感值保护的实验中，半敏感属性中敏感值的比例被设置为 20%。我们将对原始数据表中每个用户的所有敏感值都进行测试，并且通过计算敏感值暴露概率的平均值进行比较。由于 Mondrian 算法将半敏感属性中的敏感值进行了泛化，所以已经无法对其计算半敏感属性中敏感值暴露的概率。在本节实验中，只对 Anatomy 算法和加入启发式的局部分解算法的结果进行比较。除此之外，考虑了在 4.2.2 节中关于攻击者具备不同背景知识的情况，即攻击者仅知道目标用户的 QI 信息或者已知所有用户的 QI 信息，在数据结果中，将两种情况分别表示为 LocalAnatomy_1 和 LocalAnatomy_2。图 4.3 和 4.4 分别显示了使用不同的匿名算法时，半敏感属性和敏感属性中敏感值暴露概率的结果。

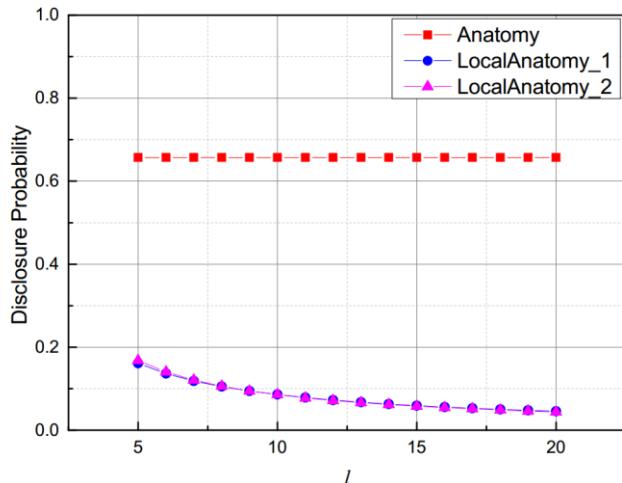


图 4.3 半敏感属性中敏感值暴露的概率

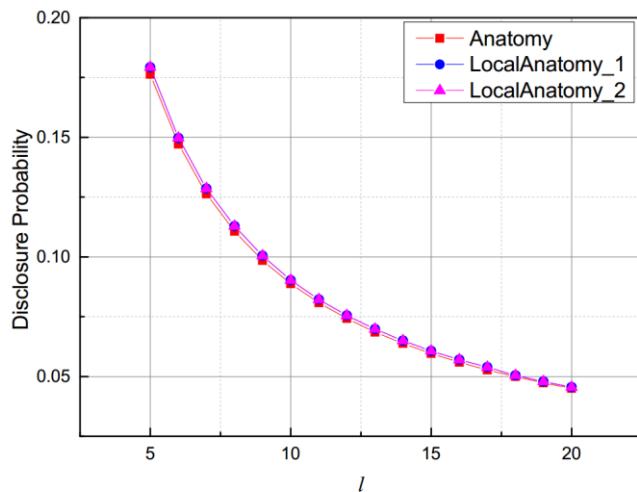


图 4.4 敏感属性中敏感值暴露的概率

由图 4.3 和 4.4 可以发现，Anatomy 算法和局部分解算法都可以为敏感属性中的敏感值提供可靠的保护，但是 Anatomy 算法却无法保护半敏感属性中的敏感值。在图 4.3 中，Anatomy 算法的敏感值暴露的概率是约为 0.65 的常数。这是由于通过 Anatomy 算法匿名的数据表没有为半敏感属性中的敏感值提供任何保护，因此攻击者可以直接通过使用目标用户的 QI 信息在匿名数据表中进行匹配获得目标用户在半敏感属性中的敏感值，并且增加参数 l 也不会提高对敏感值的保护。此外，尽管在 LocalAnatomy_2 中攻击者拥有更多的背景知识，但是局部分解算法仍然遵循 l -diversity 匿名原则保护了半敏感属性中的敏感值。

4.4.2 信息可利用性

在测试信息可利用性的实验中，同样将半敏感属性中敏感值的比例设为 20%，并且使用文献[54]中查询回答错误率的方法对局部分解算法与 Mondrian 算法和 Anatomy 算法分别进行比较。我们随机生成了 1000 条查询语句，并对每个匿名表计算平均查询回答错误率。查询语句的形式如下：

`SELECT SUM(salary) FROM Microdata`

`WHERE pred(A1) AND pred(A2) AND pred(A3) AND pred(A4)`

其中，查询语句的条件中随机包含了四个 QI 或半敏感属性，并且将薪水的总和作为结果进行比较。对于分类属性， $\text{pred}(A)$ 表示为：

$$(A = v_1 \text{ or } A = v_2 \text{ or } \dots \text{ or } A = v_m) \quad (4.12)$$

其中， $v_i (1 \leq i \leq m)$ 为在 $D[A]$ 中的随机值。而对于数字属性， $\text{pred}(A)$ 表示为：

$$\begin{aligned} & (A > v) \text{ or } (A < v) \text{ or } (A = v) \text{ or } (A \geq v) \\ & \text{or } (A \leq v) \text{ or } (A \neq v) \end{aligned} \quad (4.13)$$

其中， v 为在 $D[A]$ 中的随机值。

查询回答错误率的公式为：

$$Sum_{error} = (Sum_{upper} - Sum_{lower}) / Sum_{act} \quad (4.14)$$

其中 Sum_{upper} 和 Sum_{lower} 分别为薪水总和的上限和下限，并且 Sum_{act} 为薪水的实际总和。值得注意的是，在文献[54]中，当对桶匿名数据表执行查询时，需要先统计敏感属性中每个桶包含匹配个体的数量。但是，由于本节实验加入了半敏感属性的设定，因此需要将计算误差上限 Sum_{upper} 和误差下限 Sum_{lower} 的方法稍微进行修改。

我们首先计算满足 QI 属性条件的个体集合 ids_set ，然后计算在 ids_set 中包含于薪水属性内每个桶的个体数量。对任意个体 t ，将满足一个半敏感属性 A^{SS} 条件的概率记为 $Pro_t(A^{SS})$ 。当 t 在 A^{SS} 中没有被划分为桶时，如果其半敏感属性值满足条件，则

$$Pro_t(A^{SS}) = 1 \quad (4.15)$$

否则

$$Pro_t(A^{SS}) = 0 \quad (4.16)$$

而当 t 在 A^{SS} 中被划分为桶 b 时，则有

$$Pro_t(A^{SS}) = Num_b / |b| \quad (4.17)$$

其中， Num_b 为桶 b 中满足条件的敏感值数量。在 ids_set 中，每个个体满足所有条件的概率为所有半敏感属性中满足条件概率的乘积，记为 Pro_t ，有

$$Pro_t = \prod_{A^{SS}} Pro_t(A^{SS}) \quad (4.18)$$

根据薪水属性中按照桶的标识将满足所有条件的个体概率相加，记为 $Num_b(A^{SA})$ ，即对薪水属性中任意桶 b ，有

$$Num_b(A^{SA}) = \sum_t Pro_t \quad (4.19)$$

其中， t 为分配在桶 b 中的个体。 Sum_{upper} 为薪水属性中每个桶的 $Num_b(A^{SA})$ 向上取整对应的薪水上限的和， Sum_{lower} 为薪水属性中每个桶的 Num_b 向下取整对应的薪水下限的和。

为了测试算法 4.2 中启发式的效果，在将局部分解算法与 Anatomy 算法进行比较时，分别令 HeuristicLocalAnatomy 和 LocalAnatomy 表示局部分解算法使用和没有使用启发式的情况。图 4.5 和 4.6 分别为局部分解算法与 Mondrian 算法和 Anatomy 算法的比较结果。

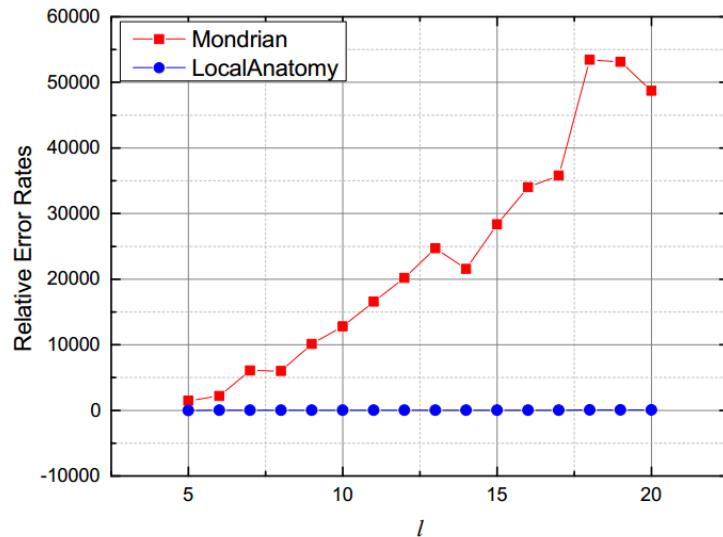


图 4.5 Mondrian vs LocalAnatomy

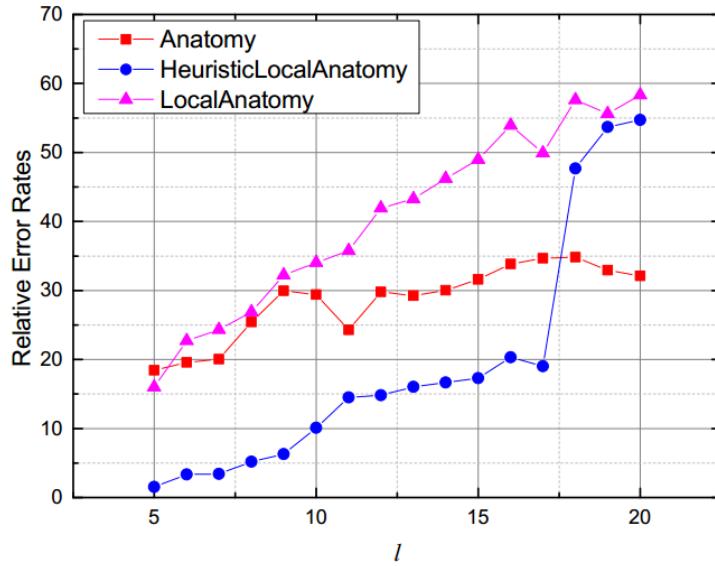


图 4.6 Anatomy vs LocalAnatomy

由图 4.5 可知，局部分解算法的查询回答错误率远低于 Mondrian 算法。这是由于局部分解算法仅对数据表中的敏感值进行匿名，从而最大限度地保留了原始数据表的内容。在图 4.6 中，没有加入启发式的局部分解算法 LocalAnatomy，其查询回答错误率一直高于 Anatomy 算法。但是，加入启发式的局部分解算法 HeuristicLocalAnatomy 在参数 l 的值小于 18 时小于 Anatomy 算法，而在 l 的值为 18 时查询回答错误率突然升高，并且非常接近于 LocalAnatomy。这是由于算法 4.2 中的启发式主要使用了二分法根据敏感值的值域对个体集合进行递归划分，而当 l 的值逐渐变大时，划分后的子集需要包含更多的敏感值类型，从而使划分后的子集值域范围变大。由此可知，当参数 l 的值超过一定阈值时，算法 4.2 中的启发式将失效。

4.4.3 敏感值密度的影响

在之前的实验中，半敏感属性中敏感值的比例被固定为 20%。而在本节实验中，我们将对半敏感属性中不同敏感值的密度对局部分解算法的匿名效果产生的影响进行研究，敏感值的比例被设置为 10%、20%、30% 和 40%，并在实验结果中分别表示为 LocalAnatomy_1、LocalAnatomy_2、LocalAnatomy_3 和 LocalAnatomy_4。图 4.7 和 4.8 分别显示了在不同敏感值密度下，局部分解算法

对敏感值的保护和查询回答错误率的实验结果。

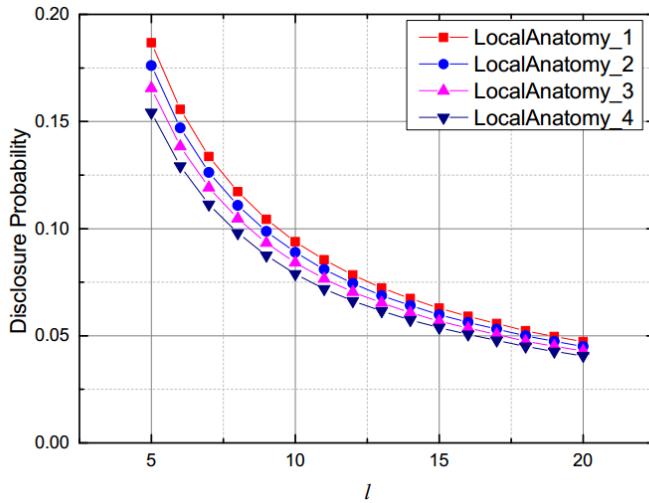


图 4.7 敏感值暴露的概率

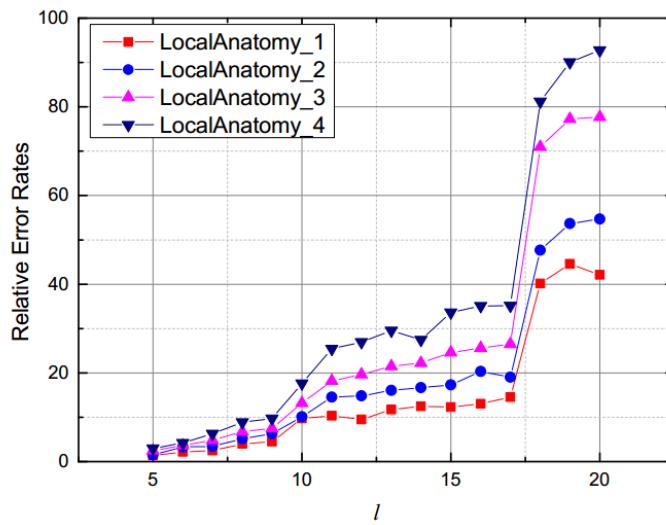


图 4.8 查询回答错误率

通过图 4.7 可知，尽管半敏感属性中敏感值的数量增加，但是局部分解算法仍然能遵守 l -diversity 匿名原则对数据表中的敏感值进行保护。由图 4.8 中可以发现，在敏感值密度越高的情况下，查询回答错误率的结果越会受到参数 l 的影响。此外，图 4.8 中所有的数据线都在 l 的值为 18 时大幅度上升。这是由于参数 l 的值过大，导致桶中敏感值的值域范围增加，导致启发式的效果大大减弱。

4.5 扩展讨论

至今为止，学者们已经提出了包括匿名原则和匿名算法在内的许多匿名技术。但是，由于攻击者具有不同的背景知识以及数据表具有不同的结构特点，使用单一的匿名原则和匿名算法无法完全保护数据表中用户的隐私信息。因此，将单一的匿名原则和匿名算法应用于不同的发布环境时会存在很大的局限性。例如，根据推论 4.1，在使用遵循 l -diversity 匿名原则的局部分解算法对敏感值进行保护时，匿名数据表中每个桶都必须至少包含 l 个个体，并且桶内所有个体的敏感值都不相同。在 4.4 节的实验中，数据表中的半敏感属性和敏感属性被设定为年龄、职业和薪水，而在实际应用中任意属性都可能是半敏感属性和敏感属性。例如，当性别作为半敏感属性或者敏感属性时，由于性别属性的值域中仅包含了两个值，所以无法使用遵循 l -diversity 匿名原则的局部分解算法对其进行保护。

尽管遵循 l -diversity 匿名原则的局部分解算法在一些发布环境中存在缺陷，但是由于局部分解算法对不同属性的保护是相互独立的，所以可以根据不同属性的特点使用不同的匿名原则对其进行保护。我们对 4.4 节中的实验环境进行修改，将性别属性加入至半敏感属性集合中。根据文献[52]， t -closeness 匿名原则相比 l -diversity 匿名原则可以为性别属性提供更加合适的隐私保护。因此，仅对性别属性使用了遵循 t -closeness 匿名原则的局部分解算法，而对其他属性仍然遵循 l -diversity 匿名原则，具体的匿名算法描述如算法 4.3 所示。

Algorithm 4.3: 满足多种匿名原则的局部分解算法

```

function local_anatomy( $T, l, t$ )
1  $Attri_{sen} = \{attributes \text{ including sensitive values in } T\}$ 
2  $T_{anony} = T$ 
3 for each  $attribute \in Attri_{sen}$  do
4    $ValuePair_{sen} = \{(id, s) | s \text{ is a sensitive value in attribute}\}$ 
5   if attribute is not Sex then
6      $anatomize\_l(T_{anony}, ValuePair_{sen}, l)$ 
7   else
8      $anatomize\_t(T_{anony}, ValuePair_{sen}, t)$ 
9   end if
10 end for
11 return  $T_{anony}$ 

```

图 4.9 满足多种匿名原则的局部分解算法

在算法 4.3 中, $\text{anatomize}_l(T,ValuePair, l)$ 为遵循 l -diversity 匿名原则的桶划分函数由算法 4.2 实现, 而 $\text{anatomize}_t(T,ValuePair, t)$ 为遵循 t -closeness 匿名原则的桶划分函数由 Sabre 算法^[116]具体实现。

我们将参数 t 的值固定为 0.2, 其他的实验环境与 4.4 节中相同, 并使用查询回答错误率的方法对匿名之后的数据表中信息可利用性进行测试, 结果如图 4.10 所示, 其中, LocalAnatomy_1、LocalAnatomy_2、LocalAnatomy_3 和 LocalAnatomy_4 仍然分别代表了半敏感属性中敏感值的比例为 10%、20%、30% 和 40%。

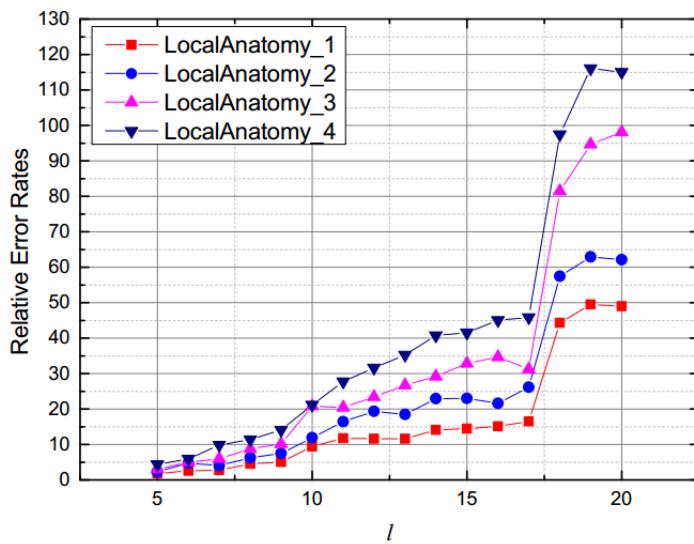


图 4.10 遵循多个匿名原则时的查询回答错误率

4.6 本章小结

在本章中, 我们基于个性化隐私保护的发布环境重新定义了属性的类型, 并且提出了局部分解算法对数据表中的敏感值进行保护。局部分解算法将敏感值作为数据表中敏感性的最小单位, 为各个半敏感属性和敏感属性提供相互独立的保护。此外, 我们分析了遵循单一匿名原则的局部分解算法的局限性, 并将局部分解算法改进为可以同时遵循多个匿名原则, 从而满足更加复杂的实际匿名需求。最后, 我们分别对相关算法的匿名效果进行了测试, 证明了局部分解算法具有可靠的敏感值保护能力、优秀的信息可利用性和灵活的可扩展功能。

第5章 局部分解泛化算法

5.1 引言

在第 4 章中，我们提出了一种基于个性化隐私保护需求的局部分解算法用于保护数据表中用户的敏感值。但是，由于局部分解算法缺乏泛化机制无法为用户身份提供保护，所以在隐私保护需求比较高的发布环境中，使用局部分解算法具有较高的隐私泄露风险。在本章中，我们仍然基于个性化隐私的保护需求，通过加入泛化机制使局部分解算法可以遵循 k -anonymity 匿名原则为数据表提供更加全面和安全的保护。

5.1.1 问题的提出

局部分解算法是一种轻量级的匿名算法，它仅将数据表中的敏感值在相应的属性下划分为桶，并且保留数据表中所有原始 QI 值。由此，局部分解算法使用了尽可能少的匿名操作来保护数据表中敏感值的安全。一方面，原始 QI 值极大地有利于保留数据表的信息可利用性。另一方面，攻击者可以更加容易地对匿名数据表进行攻击。因此，相对于数据的安全性，局部分解算法更加偏向于获得更多的信息可利用性。

尽管局部分解算法具有非常优秀的灵活性和可扩展性。例如，在 4.5 节中，局部分解算法同时遵循了 l -diversity 和 t -closeness 匿名原则分别对不同的属性进行保护。但是，由于局部分解算法缺乏泛化机制，导致其无法对数据表中的 QI 值进行泛化，所以局部分解算法仅能保障数据中敏感值的安全，却很难防止攻击者识别用户在数据表中的标识。

例如，表 5.1 为攻击者的背景知识，其中，假设攻击者不会提前知道关于攻击目标的任何敏感值。表 5.2 为原始数据表，其中，允许用户自由设置敏感值，并将除 ID 属性外的每个属性加入 Flag 标识用于区分数据表中的值是否为敏感值。表 5.3 为使用遵循 2-diversity 匿名原则的局部分解算法对表 5.2 进行匿名之后的数据表，其中除 ID 属性外，每个属性都包含 BID 标识用于标记每个属性

中携带敏感值的个体被分配至桶的标识。例如，当攻击者将 Mark 作为攻击目标时，攻击者最多只能推断出年龄值为 24 或 31、疾病值为支气管炎或消化不良。

表 5.1 攻击者的背景知识

Name	Age	Gender	Zip Code
Neil	22	Male	21358
Daphne	-	Female	-
Dean	16	Male	-
Mark	-	Male	21336

表 5.2 原始数据表

ID	Age (Flag)	Gender (Flag)	Zip Code (Flag)	Disease (Flag)
1001	28 (No)	Male (No)	21357 (Yes)	Bronchitis (Yes)
1002	25 (No)	Female (No)	21344 (Yes)	Gastritis (Yes)
1003	16 (No)	Male (No)	21352 (No)	Dyspepsia (Yes)
1004	24 (Yes)	Male (No)	21336 (No)	Bronchitis (Yes)
1005	31 (Yes)	Female (Yes)	21328 (No)	Hepatitis (Yes)
1006	22 (No)	Male (No)	21358 (No)	Flu (Yes)
1007	29 (Yes)	Female (No)	21340 (No)	Pneumonia (Yes)
1008	34 (Yes)	Male (Yes)	21328 (No)	Bronchitis (Yes)

表 5.3 遵循 2-diversity 匿名原则的局部分解数据表

ID	Age (BID)	Gender (BID)	Zip Code (BID)	Disease (BID)
1001	28 (-)	Male (-)	21344 (1)	Bronchitis (1)
1002	25 (-)	Female (-)	21357 (1)	Gastritis (1)
1003	16 (-)	Male (-)	21352 (-)	Bronchitis (2)
1004	24 (1)	Male (-)	21336 (-)	Dyspepsia (2)
1005	31 (1)	Female (1)	21328 (-)	Flu (3)
1006	22 (-)	Male (-)	21358 (-)	Hepatitis (3)
1007	29 (2)	Female (-)	21340 (-)	Bronchitis (4)
1008	34 (2)	Male (1)	21328 (-)	Pneumonia (4)

尽管表 5.3 可以为 Mark 的敏感值提供安全的保障，但是攻击者仍然可以准确地识别出 Mark 在匿名数据表中的 ID 为 1004。在一般情况下，隐藏用户在匿名数据表中的身份标识可以有效地降低隐私泄露的风险，并且在很大程度上提高对未知安全攻击的防御能力。因此，在隐私安全需求较高的发布环境中仍然需要匿名表具有保护用户身份的能力。

5.1.2 本章工作

在个性化隐私保护的发布环境中，我们通过在局部分解算法中加入泛化机制提出局部分解泛化算法（Local Anatomy Generalization）同时对数据表中用户身份和敏感值进行保护。局部分解泛化算法为用户身份提供保护的基本思想是根据数据表中个体携带 QI 值的情况将所有个体分为多个子集，并且在每个子集内再将个体划分为等价组，从而使整个数据表满足 k -anonymity 匿名原则的条件。此外，由于在局部分解泛化算法中对用户身份和敏感值的保护是相互独立的，所以可以直接使用局部分解算法对敏感值进行保护。通过局部分解泛化算法对表 5.2 进行匿名后的结果，如表 5.4 所示。

表 5.4 遵循 2-anonymity 和 2-diversity 匿名原则的局部分解泛化数据表

ID	GID	Age(BID)	Gender(BID)	Zip Code(BID)	Disease(BID)
1001	1	[25–28] (-)	*(-)	21344(1)	Bronchitis(1)
1002	1	[25–28] (-)	*(-)	21357(1)	Gastritis(1)
1003	2	[16–22] (-)	Male(-)	2135*(-)	Bronchitis(2)
1004	3	24(1)	*(-)	213**(-)	Dyspepsia(2)
1005	4	31(1)	Female(1)	21328(-)	Flu(3)
1006	2	[16–22] (-)	Male(-)	2135*(-)	Hepatitis(3)
1007	3	29(2)	*(-)	213**(-)	Bronchitis(4)
1008	4	34(2)	Male(1)	21328(-)	Pneumonia(4)

在表 5.4 中，将除 ID 属性外的其他属性加入 BID 标识，并且还增加一个 GID 属性用于标记数据表中每个个体被划分为等价组的标识。我们将带有相同的 QI 值属性的个体划分为子集，如 ID 为 1001 和 1002、1003 和 1006、1004 和 1007、

还有 1005 和 1008。在这些子集中，分别将包含的个体划分为等价组，并将 QI 值进行泛化。当攻击者仍然以 Mark 作为攻击目标，并使用其 QI 值在表 5.4 中进行匹配时，攻击者最多可以推断出 Mark 在表中的信息标识是 ID 为 1004 或 1007，其年龄值被包含于 BID 为 1 或 2 的桶内，可能为 24、31、29 或者 34，疾病值被包含于 BID 为 2 或 4 的桶内，可能为支气管炎、消化不良或者肺炎。由此可见，相比于仅使用局部分解算法的表 5.3，使用局部分解泛化算法的表 5.4 不仅防止了攻击者识别用户在数据表中的信息标识，还增强了对个体敏感值的保护。

在本章的研究中，我们仍将考虑用户的个性化隐私保护需求，并且数据表中包括了 QI 属性、半敏感属性和敏感属性。我们假设攻击者已经知道目标用户的信息存在于数据表中，以及目标用户的所有非敏感信息。此外，我们不仅需要保障数据表中敏感值的安全，还需要防止攻击者获取用户在数据表中的标识。本章的具体工作如下：

首先，我们提出局部分解泛化算法用于保护数据表中用户身份和敏感值。该算法通过在局部分解算法中加入泛化机制，将每个半敏感属性和敏感属性中携带敏感值的用户划分为桶，并且将带有相同的 QI 值属性的个体划分为等价组，从而在保护敏感值的同时为用户身份提供相互独立的保护。

其次，我们分析遵循 k -anonymity 匿名原则的局部分解泛化算法对用户身份和敏感值的保护原理。并且，我们证明当遵循 l -diversity 匿名原则的局部泛化算法加入遵循 k -anonymity 匿名原则的泛化算法时，局部分解泛化算法仍然遵循 l -diversity 匿名原则，即局部分解泛化算法同时遵循了 k -anonymity 和 l -diversity 匿名原则。

第三，我们具体实现两种局部分解泛化算法。由于局部分解泛化算法对用户身份和敏感值的保护是相互独立的，所以我们仍然使用局部分解算法对敏感值进行保护，并且分别通过使用多维划分技术和 NCP 引导的启发式实现两种局部分解泛化算法中的泛化机制对用户身份进行保护。

最后，我们对使用局部分解泛化算法匿名的数据表进行详细地实验测试和分析。我们通过调整半敏感属性中的敏感值密度、参数 k 和 l 的值分别研究了对两种局部分解泛化算法匿名数据表的 NCP 惩罚指数、鉴别力度量和查询回答错误率产生的影响。

本章的结构如下：在 5.2 节中，将介绍局部分解泛化算法的基本概念，并且分析对用户身份和敏感值保护的工作原理；在 5.3 节中，将具体实现两种遵循 k -anonymity 匿名原则的局部分解泛化算法；在 5.4 节中，将对局部分解泛化算法的匿名效果进行实验测试和分析；在 5.5 节中，将对本章内容进行总结。

5.2 局部分解泛化算法模型

5.2.1 基本模型

假设一个数据表 T 包含了 d 个属性 A_1, A_2, \dots, A_d ，其中，每个属性都为数字类型或者分类类型，并且每个属性都可能包含敏感值。属性 A 的值域，记为 $D[A]$ 。对于任意个体 $t \in T$ ， $t[A]$ 表示个体 t 中属性 A 的值。为了将数据表中的个体根据其携带 QI 值的组合类型进行划分，我们需要定义 QI 划分：

定义 5.1：对于任意个体 $t \in T$ ， $QI[t]$ 表示个体 t 携带的所有 QI 值对应的属性集合，即有：

$$QI[t] = \{A | A \in T \text{ and } t[A] \text{ is a QI value}\} \quad (5.1)$$

定义 5.2：对于一个数据表 T ，QI 划分是将 T 中的个体划分为许多互不相交的子集 $\{T_1, T_2, \dots, T_k\}$ ，其中，对于每个子集 $T_i (1 \leq i \leq k)$ ， $\forall t_1, t_2 \in T_i$ ，有 $QI[t_1] = QI[t_2]$ ，且有 $\bigcup_{i=1}^k T_i = T$ ，以及对任意 $1 \leq i_1 \neq i_2 \leq k$ ， $T_{i_1} \cap T_{i_2} = \emptyset$ 。

根据定义 5.2，局部分解泛化算法的定义如下：

定义 5.3：假设对于一个数据表 T ，其 QI 划分为 $\{T_1, T_2, \dots, T_k\}$ 。局部分解泛化算法是将每个子集 $T_i (1 \leq i \leq k)$ 划分为许多个等价组 $\{EG_1, EG_2, \dots, EG_{i_m}\}$ ，并且有 $\bigcup_{j=1}^{i_m} EG_j = T_i$ ，以及对任意 $1 \leq j_1 \neq j_2 \leq i_m$ ， $EG_{j_1} \cap EG_{j_2} = \emptyset$ 。此外，对于 T 中每个包含敏感值的属性，局部分解泛化算法将携带该属性敏感值的用户划分为桶，并且每个桶包含 $IDT(ID, BID)$ 和 $SAT(SA, BID)$ 两个部分，其中， ID 和 SA 分别为在桶中所有个体的 ID 标识和敏感值， BID 表示在属性内桶的 ID 标识。

5.2.2 隐私保护原理

在本节中，我们将详细地分析局部分解泛化算法在同时遵循 k -anonymity 和 l -diversity 匿名原则的情况下，分别对数据表中用户身份和敏感值的保护原理。首先，分析局部分解泛化匿名表对用户身份的保护原理，并有如下定理和推论：

定理 5.1：在一个使用局部分解泛化算法进行匿名的数据表中，对于任意个体 $t \in T$ ，其身份泄露的概率最多为 $1/|G(t)|$ ，其中 $G(t)$ 为包含 t 的等价组。

证明：根据定义 5.3，原始数据表被 QI 划分为互不相交的子集 $\{T_1, T_2, \dots, T_k\}$ ，并且每个子集包含的个体都被划分为等价组。当攻击者通过目标用户的 QI 值进行匹配时，由于数据表中对应个体 t 已经与 $G(t)$ 中其他个体的 QI 值泛化为相同的形式，所以攻击者最少获得 $|G(t)|$ 个匹配个体。因此，个体 t 的身份泄露的概率最多为 $1/|G(t)|$ 。

推论 5.1：对于使用局部分解泛化算法进行匿名的数据表，如果匿名数据表遵循 k -anonymity 匿名原则，则匿名表中所有的等价组需要至少包含 k 个个体。

证明：如果在使用局部分解泛化算法进行匿名的数据表中，所有等价组至少包含 k 个个体，则对任意个体 $t \in T$ ，有

$$|G(t)| \geq k \quad (5.2)$$

其中， $|G(t)|$ 为包含 t 的等价组中含有个体的数量。因此，有

$$\frac{1}{|G(t)|} \leq \frac{1}{k} \quad (5.3)$$

根据定理 5.1，个体 t 的身份泄露的概率最多为 $1/k$ 。因此，匿名数据表遵循 k -anonymity 匿名原则。

接下来，我们分析局部分解泛化算法对敏感值的保护原理。由于仍然考虑了个性化隐私保护的情况，所以对于攻击者背景知识的假设仍然会存在 4.2.2 节中的差别。但是，即使在最坏的情况下，任意一个个体在匿名数据表中至少包含于一个等价组中，并且等价组中包含的敏感值至少包含于一个桶中。

定义 5.4：在一个使用局部分解泛化算法进行匿名的数据表 T 中，对于任意个体 $t \in T$ ，以及 t 的任意敏感值 s 对应的属性 A^s ，所有包含于 $G(t)$ 中 A^s 的桶都为

t 在 A^S 下的匹配桶，记为 $MB_t[A^S]$ ，其中 $G(t)$ 为包含 t 的等价组。

定理 5.2：在一个使用局部分解泛化算法进行匿名的数据表 T 中，对于任意个体 $t \in T$ ，其任意敏感值 s 暴露的概率为：

$$p(t, s) \leq \sum_{MB_t[A^S]} p(t, MB_t[A^S]) \frac{|MB_t[A^S](s')|}{|MB_t[A^S]|} \quad (5.4)$$

其中， $|MB_t[A^S](s')|$ 为在匹配桶 $MB_t[A^S]$ 中出现次数最多的敏感值 s' 的数量，并且 $|MB_t[A^S]|$ 为 $MB_t[A^S]$ 中包含个体的数量。

证明：由于在局部分解泛化匿名表中所有的个体都被划分至等价组中，因此当攻击者使用目标用户 t 的 QI 值进行匹配时，会得到包含 t 的等价组 $G(t)$ 。对于 t 的任意敏感值 s ，攻击者会得到 $G(t)$ 在相应属性中包含的匹配桶。因此，有

$$p(t, s) = \sum_{MB_t[A^S]} p(t, MB_t[A^S]) p(s | t, MB_t[A^S]) \quad (5.5)$$

其中， $p(s | t, MB_t[A^S])$ 为当个体 t 在匹配桶 $MB_t[A^S]$ 中携带敏感值为 s 的概率。对任意 $MB_t[A^S]$ 中出现次数最高的敏感值 s' ，有：

$$|MB_t[A^S](s)| \leq |MB_t[A^S](s')| \quad (5.6)$$

所以，有

$$p(s | t, MB_t[A^S]) = \frac{|MB_t[A^S](s)|}{|MB_t[A^S]|} \leq \frac{|MB_t[A^S](s')|}{|MB_t[A^S]|} \quad (5.7)$$

并且，有

$$p(t, s) \leq \sum_{MB_t[A^S]} p(t, MB_t[A^S]) \frac{|MB_t[A^S](s')|}{|MB_t[A^S]|} \quad (5.8)$$

推论 5.2：如果局部分解算法遵循 l -diversity 匿名原则，则加入泛化机制后的局部分解算法仍然遵循 l -diversity 匿名原则。

证明：由于局部分解算法遵循 l -diversity 匿名原则，因此对于任意个体 $t \in T$ ，其任意敏感值 s 暴露的概率最多为 $1/l$ 。因此，对于匿名数据表中的任意桶 b ，有

$$\frac{|b(s')|}{|b|} \leq \frac{1}{l} \quad (5.9)$$

其中， $b(s')$ 为桶 b 中出现次数最多的敏感值 s' 的数量。根据式 (5.4)，有

$$p(t, s) \leq \frac{1}{l} \sum_{MB_t[A^S]} p(t, MB_t[A^S]) = \frac{1}{l} \quad (5.10)$$

因此，加入泛化机制后的局部分解算法仍然遵循 l -diversity 匿名原则。

推论 5.1 和 5.2 给出了局部分解泛化算法遵循 k -anonymity 和 l -diversity 匿名原则时需要满足的条件。我们对遵循 k -anonymity 和 l -diversity 匿名原则的局部分解泛化算法定义如下：

定义 5.5：对于使用局部分解泛化算法进行匿名的数据表，如果匿名数据表同时遵循 k -anonymity 和 l -diversity 匿名原则，当对任意个体 $t \in T$ ，其用户身份泄露的概率最多为 $1/k$ ，并且有：

$$p(t, s) \leq \frac{1}{l} \quad (5.11)$$

其中， s 为 t 的任意敏感值。

值得注意的是，由于局部分解泛化算法对用户身份和敏感值的保护是相互独立的，所以在实际应用中我们可以根据发布环境和匿名需求将 k -anonymity 或者 l -diversity 匿名原则替换为其他合适的匿名原则，并且不会降低另一方的隐私保护效果。

5.3 局部分解泛化算法

在本节中，我们在局部分解算法的基础上加入泛化机制，提出局部分解泛化算法，具体的描述如算法 5.1 所示。

Algorithm 5.1: 局部分解泛化算法

```

function local anatomy generalization( $T, k, l$ )
1  $Attri_{sen} = \{attributes \text{ including sensitive values in } T\}$ 
2  $T_{anony} = T$ 
3 for each  $attribute \in Attri_{sen}$  do
4    $ValuePair_{sen} = \{(id, s) | s \text{ is a sensitive value in attribute}\}$ 
5    $anatomize(T_{anony}, ValuePair_{sen}, l)$ 
6 end for
7  $generalize(T_{anony}, k)$ 
8 return  $T_{anony}$ 

```

图 5.1 局部分解泛化算法

局部分解泛化算法包括两个部分：首先，我们使用了第 4 章中提出的局部分解算法将数据表中的敏感值在相应的属性中划分为桶（第 1 行到第 6 行）；然后，再将匿名数据表 T_{anony} 划分为等价组并将 QI 值进行泛化（第 7 行），使其遵循 k -anonymity 匿名原则。

本文分别通过基于多维划分技术和基于标准化确定惩罚指数^[94]（Normalized Certainty Penalty，简称 NCP）引导的启发式具体实现 $generalize(T_{anony}, k)$ 函数。接下来，依次对两种局部分解泛化算法中的泛化算法部分进行描述。

5.3.1 基于多维划分技术的局部分解泛化算法

在使用多维划分技术实现算法之前，先简单介绍一下关于多维划分技术的相关概念，并且提出在个性化隐私保护发布环境下的多维划分技术。

定义 5.6：对于数据表中包含的 m 个 QI 属性 $A_1^{QI}, A_2^{QI}, \dots, A_m^{QI}$ ，多维划分是将 $D[A_1^{QI}] \times D[A_2^{QI}] \times \dots \times D[A_m^{QI}]$ 组成的多维空间划分为互相不重合的区域，并且根据个体的 QI 属性值 $(t[A_1^{QI}], t[A_2^{QI}], \dots, t[A_m^{QI}]) \in D[A_1^{QI}] \times D[A_2^{QI}] \times \dots \times D[A_m^{QI}]$ 通过映射函数 θ 将数据表中每个个体分配至其中一个区域内。

值得注意的是，由于定义 5.6 中的多维划分技术是将组成数据表的属性作为最小敏感性单位，所以在基于个性化隐私保护的发布环境中，定义 5.6 中的多维划分技术不仅无法安全地保护半敏感属性中的敏感值而且还会造成额外的信息损失。因此，需要基于个性化隐私保护的发布环境中，对多维划分技术重新进行定义。

定义 5.7：对于一个由 $D[A_1] \times D[A_2] \times \dots \times D[A_d]$ 组成的多维空间，它的一组子区域多维空间为 $\{S_1, S_2, \dots, S_m\}$ ，其中 $S_i (1 \leq i \leq m)$ 是由 $D[A_{i_1}] \times \dots \times D[A_{i_n}] (1 \leq i_n \leq d)$ 组成的子空间。

定义 5.8：基于个性化隐私保护的多维划分是在数据表由 QI 划分得到的子区域多维空间内进行划分。假设，对于组成数据表的 d 个属性 A_1, A_2, \dots, A_d 及由

$D[A_1] \times D[A_2] \times \dots \times D[A_d]$ 组成的多维空间，数据表根据 QI 划分得到的子区域多维空间为 $\{S_1, S_2, \dots, S_m\}$ ，将每个子区域多维空间 $S_i (1 \leq i \leq m)$ 划分为互相不重合的区域，並且根据 S_i 中个体的属性值 $(t[A_{i_1}], \dots, t[A_{i_m}]) \in D[A_{i_1}] \times \dots \times D[A_{i_m}]$ 通过映射函数 θ 将每个个体分配至其中一个区域内。

在使用多维划分技术对数据表进行分组泛化时，我们需要设定一个可划分的条件。当多维划分满足 k -anonymity 匿名原则的条件时，我们有如下定义：

定义 5.9：对于一个组成 d 维空间区域的个体集合 P ，如果对于属性 $A_i (1 \leq i \leq d)$ 中的值 x 是可允许的多维划分，当且仅当 $Count(P.A_i > x) \geq k$ 和 $Count(P.A_i \leq x) \geq k$ 。其中， $Count(P.A_i > x)$ 表示个体集合 P 中属性 A_i 的值大于 x 的个体数量， $Count(P.A_i \leq x)$ 表示个体集合 P 中属性 A_i 的值小于或等于 x 的个体数量。

定义 5.10：对于将 $D[A_1] \times D[A_2] \times \dots \times D[A_d]$ 进行多维划分得到的区域 R_1, R_2, \dots, R_n ，如果这个划分是最小的多维划分，则对每个区域 $R_i (1 \leq i \leq n)$ 包含的个体集合 P ，有 $|P| \geq k$ ，并且不再存在可允许的多维划分。

根据上述定义，我们给出基于多维划分技术且遵循 k -anonymity 匿名原则的局部分解泛化算法中的泛化算法部分，如算法 5.2 所示。

Algorithm 5.2: 基于多维划分技术的泛化算法

```

function generalize with multi-dimension( $T, k$ )
1  $T_{anony} = \emptyset$ 
2  $T_{set} = \{\text{subsets of } T \mid QI[t] \text{ is the same for every } t \text{ in the subset}\}$ 
3 for each  $t_{set} \in T_{set}$  do
4    $partition\_set = \{t_{set}\}$ 
5   while  $partition\_set \neq \emptyset$  do
6      $oper\_set = pick\_set(partition\_set)$ 
7      $partition\_set = partition\_set - oper\_set$ 
8      $attributes = QI[t_{set}]$ 
9      $partition\_flag = false$ 
10    while  $attributes \neq \emptyset$  do

```

```

11   attri = choose_dimension(oper_set, attributes)
12   split_value = find_median(oper_set, attri)
13    $T_l = \{t \in oper\_set : t[attri] \leq split\_value\}$ 
14    $T_r = \{t \in oper\_set : t[attri] > split\_value\}$ 
15   if satisfy_condition( $T_l$ ) and satisfy_condition( $T_r$ ) then
16       partition_set = partition_set +  $T_l$ 
17       partition_set = partition_set +  $T_r$ 
18       partition_flag = true
19       break
20   else
21       attributes = attributes - attri
22   end if
23 end while
24 if partition_flag is false then
25     gen_set = generalize_set(oper_set)
26      $T_{anony} = T_{anony} + gen\_set$ 
27 end while
28 end for
29 return  $T_{anony}$ 

```

图 5.2 基于多维划分技术的泛化算法

数据结构 T_{anony} 和 T_{set} 分别用于存储匿名之后的结果和数据表 T 的 QI 划分（第 1 行和第 2 行）。在接下来的每次循环中（第 3 行到第 28 行），算法将每个 T_{set} 中的子集进行多维划分。数据结构 $partition_set$ 用于存储需要划分的个体集合（第 4 行）。当 $partition_set$ 不为空时，将依次选取其中的集合进行递归划分（第 5 行到第 27 行）。在第 6 行中，算法在 $partition_set$ 中选取一个子集作为本次多维划分的对象，并在第 7 行中，将该选取的子集从 $partition_set$ 中移除。数据结构 $attributes$ 用于存储选取的子集中的划分维度，即该子集中个体的所有 QI 值对应属性的集合（第 8 行）。在第 9 行中，算法使用 $partition_flag$ 标记正在划分的子集 $oper_set$ 是否被划分的标识。当 $attributes$ 不为空时，算法将依次选取合适的维度进行空间划分（第 10 行到第 23 行）。在第 11 行中，我们使用了 kd -trees 算法^[117]选择进行划分的维度，并在第 12 行中，选取该维度的中间值作为划分值。通过选择的维度和划分值，将进行划分的子集分为两个更小的子集 T_l 和 T_r （第 13 行和第 14 行）。如果被划分的两个子集 T_l 和 T_r 都满足定义 5.4 中可划分的条件，则将两个子集都加入至 $partition_set$ （第 16 行和第 17 行），并在之后

的递归中进行划分。在第 18 行中，将 *partition_flag* 改为 true 标识，并在第 19 行中，退出选取维度的循环。如果两个子集 T_l 和 T_r 至少有一个不满足可划分的条件，则将该选取的维度从 *attributes* 中移除（第 20 行），并进行下次循环。在选取维度划分结束之后，算法需要判断 *partition_flag* 的标识（第 24 行），如果 *partition_flag* 为 false，则说明该选取的子集已经是最小的多维划分，因此将该选取的子集 *oper_set* 进行泛化（第 25 行），并将结果加入至 T_{anony} 中（第 26 行）。

当所有划分结束之后，算法将匿名结果 T_{anony} 返回（第 29 行）。

5.3.2 基于 NCP 引导的局部分解泛化算法

在提出基于 NCP 引导的启发式之前，先简单介绍一下关于 NCP 的相关概念。NCP 是一种信息损失率的计算方法，它可以直观地表示出个体 t 在进行泛化之后产生的信息损失量，其表达公式为：

$$NCP_A(t) = \frac{\text{size}(u)}{|A|} \quad (5.12)$$

其中， $\text{size}(u)$ 表示泛化值 u 在泛化层次树中子孙的数量， $|A|$ 表示属性 A 的值域包含值的数量。整个数据表 T 所产生的信息损失量，表示为：

$$NCP(T) = \sum_{t \in T} \sum_{i=1}^d (w_i \cdot NCP_{A_i}(t)) \quad (5.13)$$

其中， w_i 为属性 A_i 的权值。

为了使匿名之后的数据表通过 NCP 计算的信息损失量最小，我们使用 NCP 作为将个体划分为等价组过程中的启发式参数，从而尽可能保证每个等价组中泛化 QI 值的值域范围最小。基于 NCP 引导且遵循 k -anonymity 匿名原则的局部分解泛化算法中的泛化算法部分，如算法 5.3 所示。

Algorithm 5.3: 基于 NCP 引导的泛化算法

```

function generalize with NCP( $T, k$ )
1  $T_{anony} = \emptyset$ 
2  $T_{set} = \{\text{set of tuples} | QI[t] \text{ is the same for every } t \text{ in the set}\}$ 
3 for each  $t_{set} \in T_{set}$  do
4    $attributes = QI[t_{set}]$ 
5    $partition\_set = \{t_{set}\}$ 
6   while  $partition\_set \neq \emptyset$  do

```

```

7   oper_set = pick_set(partition_set)
8   partition_set = partition_set - oper_set
9   if |oper_set| < 2k then
10    gen_set = generalize_set(oper_set)
11    T_anony = T_anony + gen_set
12 else
13  node_1, node_2 = find_nodes(oper_set, attributes)
14  T1, T2 = generate_subsets(oper_set, node_1, node_2, attributes)
15  if |T1| ≥ k and |T2| ≥ k then
16    partition_set = partition_set + T1
17    partition_set = partition_set + T2
18  else
19    gen_set = generalize_set(oper_set)
20    T_anony = T_anony + gen_set
21  end if
22 end if
23 end while
24 end for
25 return T_anony

```

图 5.3 基于 NCP 引导的泛化算法

数据结构 T_{anony} 和 T_{set} 分别用于存储匿名之后的结果和数据表 T 的 QI 划分（第 1 行和第 2 行）。在接下来的每次循环中（第 3 行到第 24 行），算法依次将 T_{set} 中的子集划分为等价组。数据结构 $attributes$ 和 $partition_set$ 分别用于存储计算 NCP 的属性和需要划分的个体集合（第 4 行和第 5 行）。当 $partition_set$ 不为空时，将依次选取其中的集合进行递归划分（第 6 行到第 23 行）。在第 7 行中，算法在 $partition_set$ 中选取一个子集 $oper_set$ 作为本次划分的对象，并在第 8 行中，将 $oper_set$ 从 $partition_set$ 中移除。在第 9 行中，算法将判断 $oper_set$ 中包含的个体数量是否小于 $2k$ 。如果是，则证明 $oper_set$ 已经无法被划分两个满足 k -anonymity 匿名原则条件的子集，因此将 $oper_set$ 进行泛化并加入至 T_{anony} 中（第 10 行和第 11 行）。否则，算法将根据 $attributes$ 中的属性在 $oper_set$ 中寻找两个 NCP 距离最远的个体。其中，我们使用了文献[94]中的 Top-down 算法具体实现函数 $find_nodes(oper_set, attributes)$ 。接下来，算法通过依次将 $oper_set$ 中的个体计算与个体 $node_1$ 和 $node_2$ 的 NCP 距离，并将个体加入至 NCP 距离最短的个体集合中（第 14 行）。如果被划分的两个子集 T_1 和 T_2 包含的个体数量都大于

或者等于 k (第 15 行), 则将两个子集都加入至 $partition_set$ (第 16 行和第 17 行)。否则, 将 $oper_set$ 进行泛化并加入至 T_{anony} 中 (第 19 行和第 20 行)。在第 25 行中, 算法将匿名结果 T_{anony} 返回。

5.4 实验分析

在本节中, 我们对 5.3 节实现的两种局部分解泛化算法的匿名效果进行测试。实验数据下载于最新的美国人口普查数据, 并从中删除了丢失属性值的个体, 随机选择了 31,055 个个体及 9 个属性作为实验对象, 其中, QI 属性包括了家庭关系、婚姻状况、种族、教育情况和每周工作时长, 半敏感属性包括性别、年龄和职业, 敏感属性为薪水。表 5.5 详细地介绍这些属性的相关信息。

表 5.5 属性的描述

序号	属性名称	数值类型	敏感类型	值域大小
1	Sex	分类类型	半敏感属性	2
2	Age	数字类型	半敏感属性	73
3	Relationship	分类类型	QI	13
4	Marital status	分类类型	QI	6
5	Race	分类类型	QI	9
6	Education	分类类型	QI	11
7	Hours per week	数字类型	QI	93
8	Occupation	分类类型	半敏感属性	257
9	Salary	数字类型	敏感属性	719

在本节的实验中, 将从三个方面对局部分解泛化算法的信息可利用性进行评估, 包括鉴别力度量、NCP 度量和查询回答错误率。由于局部分解泛化算法可以同时为数据表中的用户身份和敏感值提供保护, 所以通过遵循 k -anonymity 匿名原则保护用户身份、遵循 l -diversity 匿名原则保护年龄、职业和薪水属性、遵循 t -closeness 匿名原则保护性别属性。并且, 将参数 k 和 l 作为变量, 将参数 t 固定为 0.2, 以及考虑半敏感属性中的敏感值密度对匿名算法的影响。

首先，我们将测试两种局部分解泛化算法的鉴别力度量惩罚指数，如图 5.4 到 5.11 所示。其中，参数 k 被分别设置为 5、8、10，参数 l 被分别设置为 5、8、10、12、15、18、20，半敏感属性中敏感值的密度分别为 10%、20%、30%、40%。此外，将使用多维划分技术的局部分解泛化算法记为 LAG_1，并将使用 NCP 引导启发式的局部分解泛化算法记为 LAG_2。

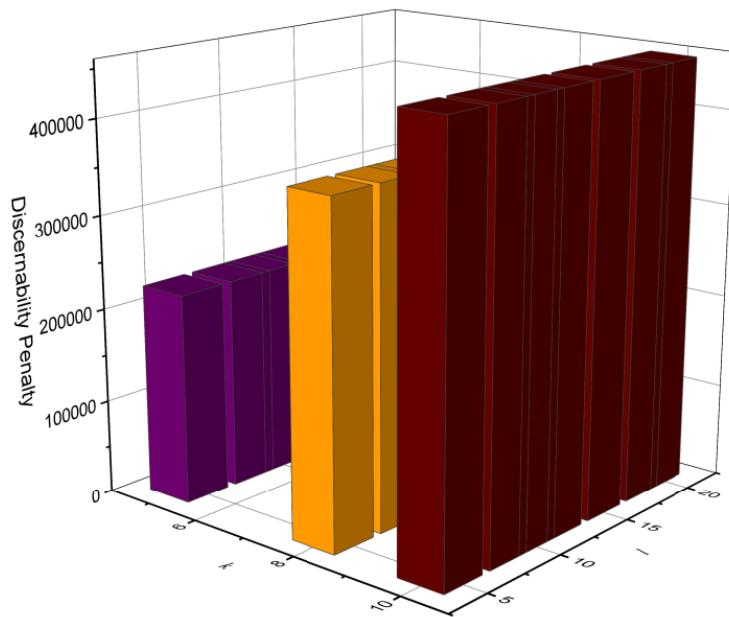


图 5.4 密度为 10% 时 LAG_1 的鉴别力度量惩罚指数

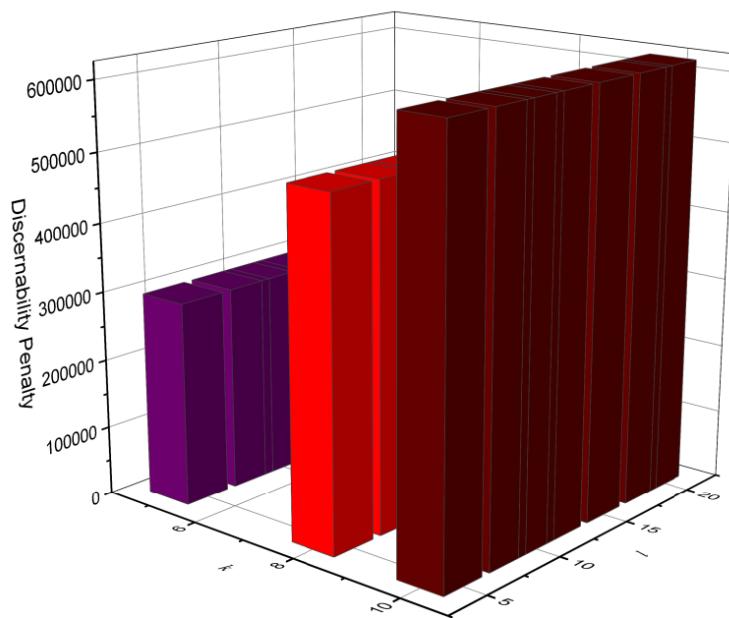


图 5.5 密度为 10% 时 LAG_2 的鉴别力度量惩罚指数

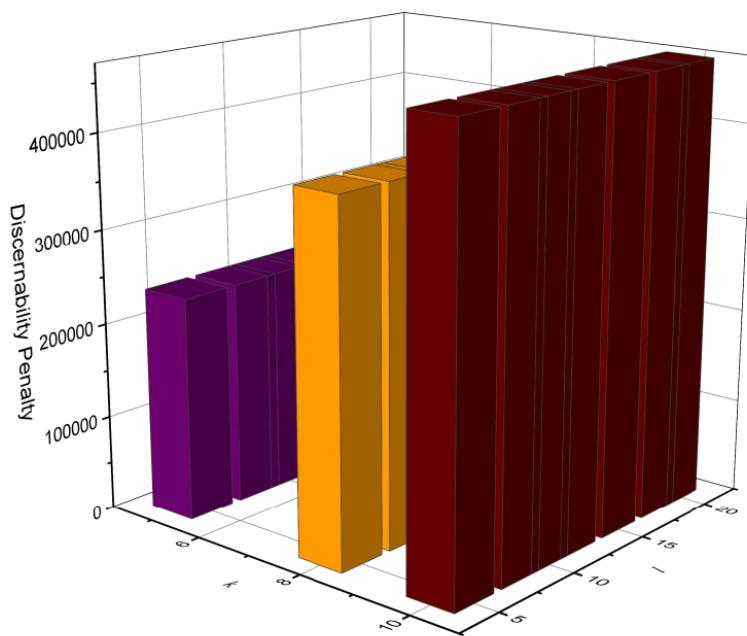


图 5.6 密度为 20% 时 LAG_1 的鉴别力度量惩罚指数

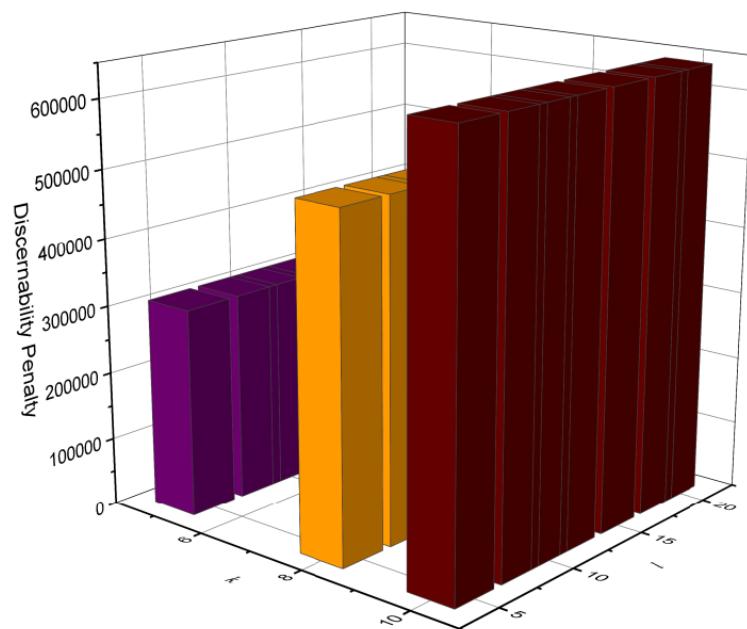


图 5.7 密度为 20% 时 LAG_2 的鉴别力度量惩罚指数

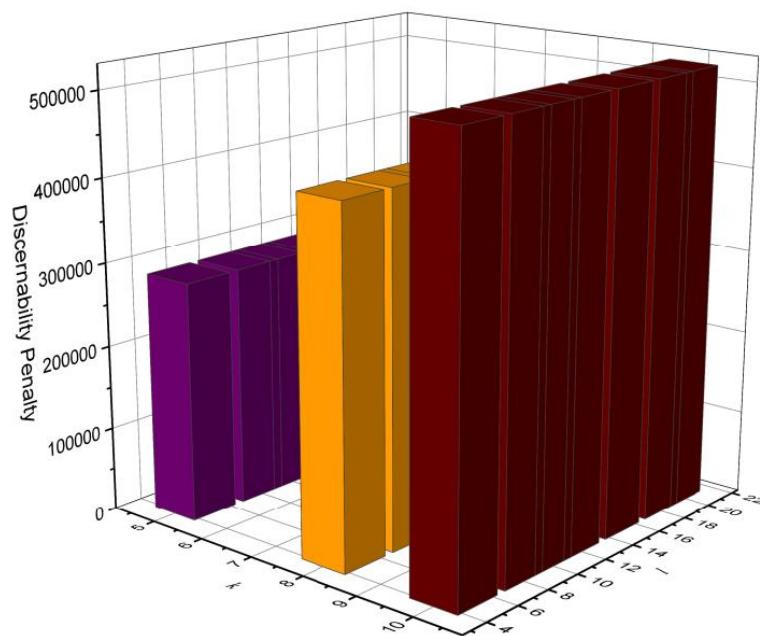


图 5.8 密度为 30% 时 LAG_1 的鉴别力度量惩罚指数

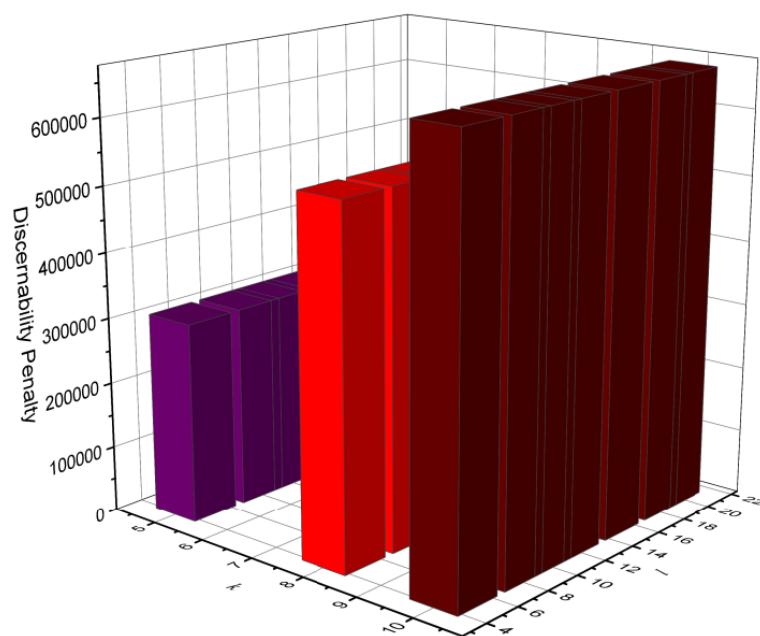


图 5.9 密度为 30% 时 LAG_2 的鉴别力度量惩罚指数

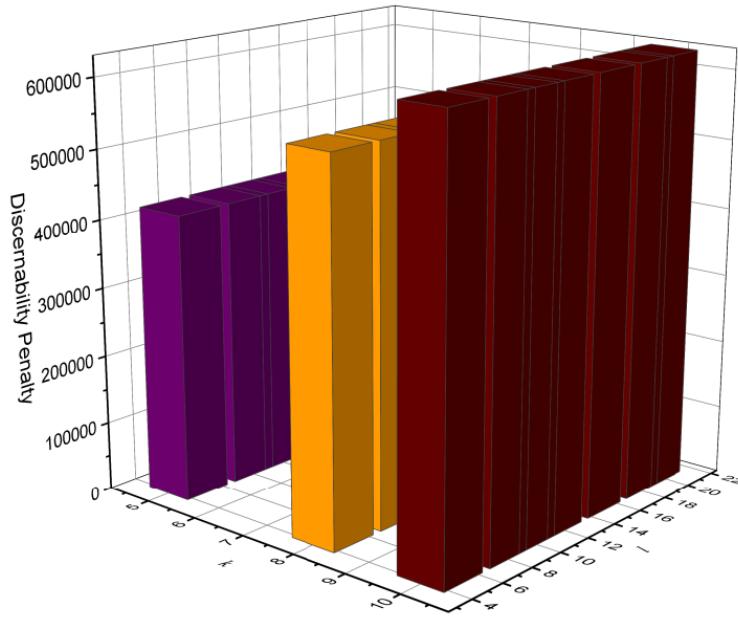


图 5.10 密度为 40% 时 LAG_1 的鉴别力度量惩罚指数

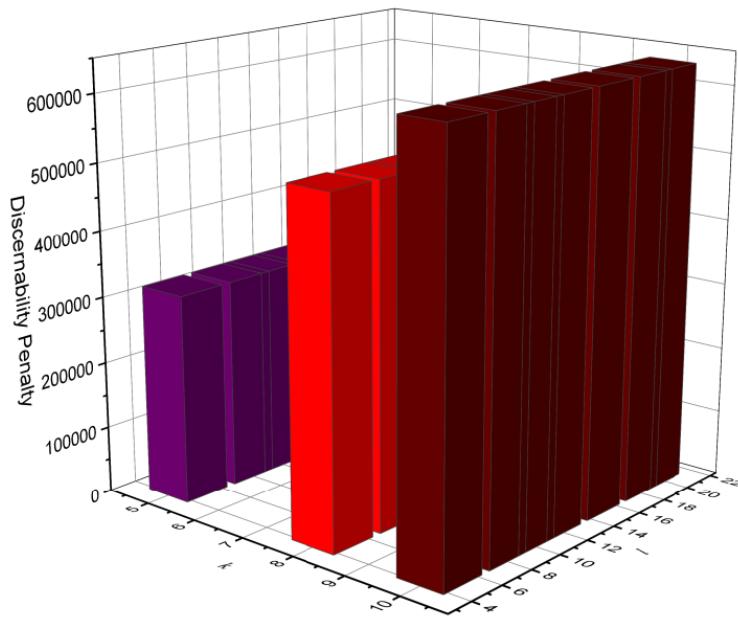


图 5.11 密度为 40% 时 LAG_2 的鉴别力度量惩罚指数

通过对实验结果的观察可知，参数 l 的值并不能影响鉴别力度量惩罚指数的结果，所以局部分解泛化算法对用户身份的保护与对敏感值的保护是相互独立的。并且，两种局部分解泛化算法的鉴别力度量惩罚指数都随着参数 k 的值和半敏感

属性中的敏感值密度的增加而升高。此外，LAG_1 的鉴别力度量惩罚指数略低于 LAG_2，证明在通过使用多维划分技术的局部分解泛化算法匿名的数据表中等价组包含的个体数量分配的更加平均。

接下来，我们使用 NCP 度量对使用两种局部分解泛化算法匿名之后的数据表进行评估。本次实验的设置与之前的实验设置相同。实验结果如图 5.12 到 5.19 所示。

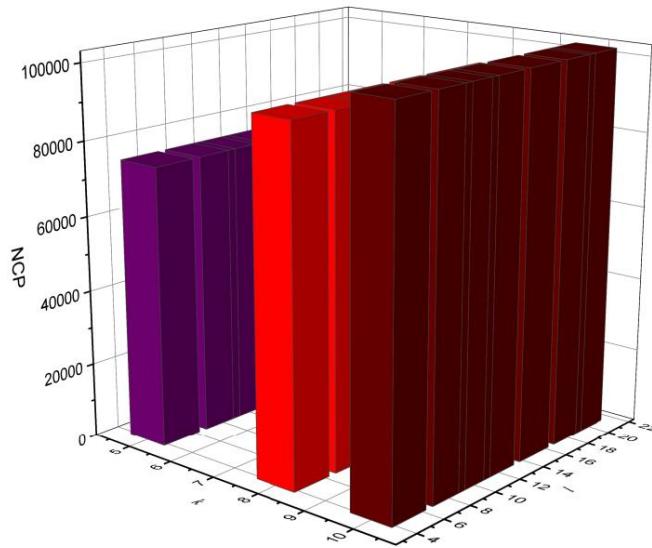


图 5.12 密度为 10% 时 LAG_1 的 NCP 指数

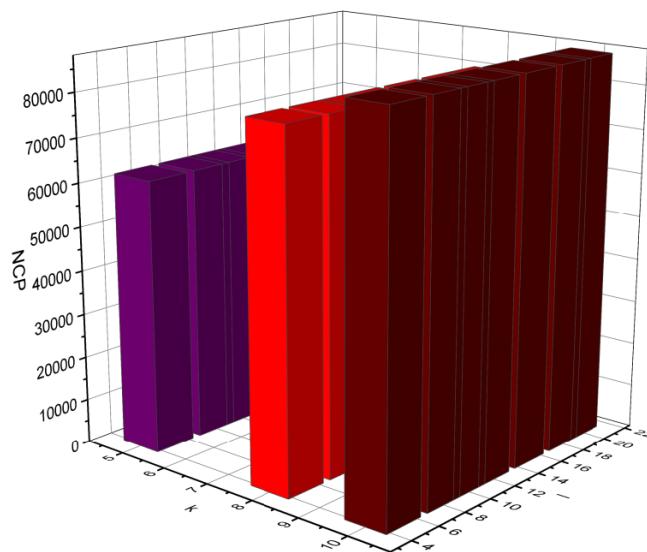


图 5.13 密度为 10% 时 LAG_2 的 NCP 指数

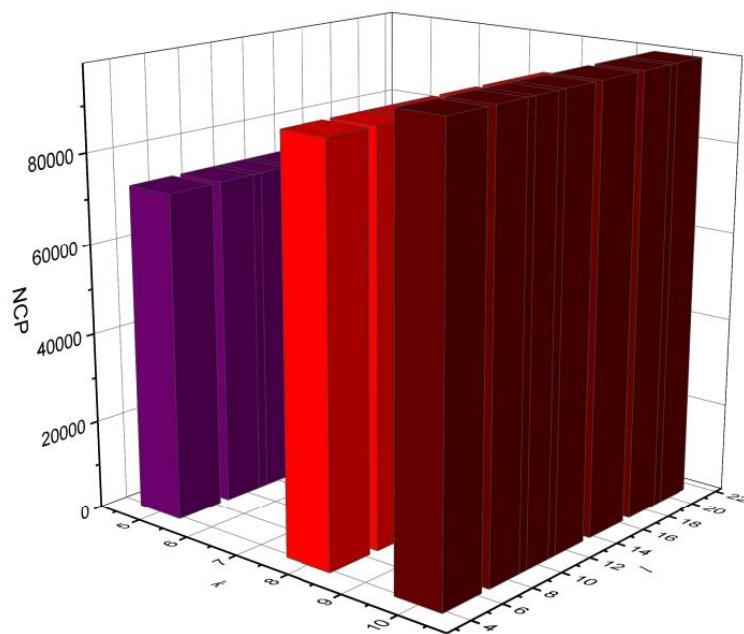


图 5.14 密度为 20% 时 LAG_1 的 NCP 指数

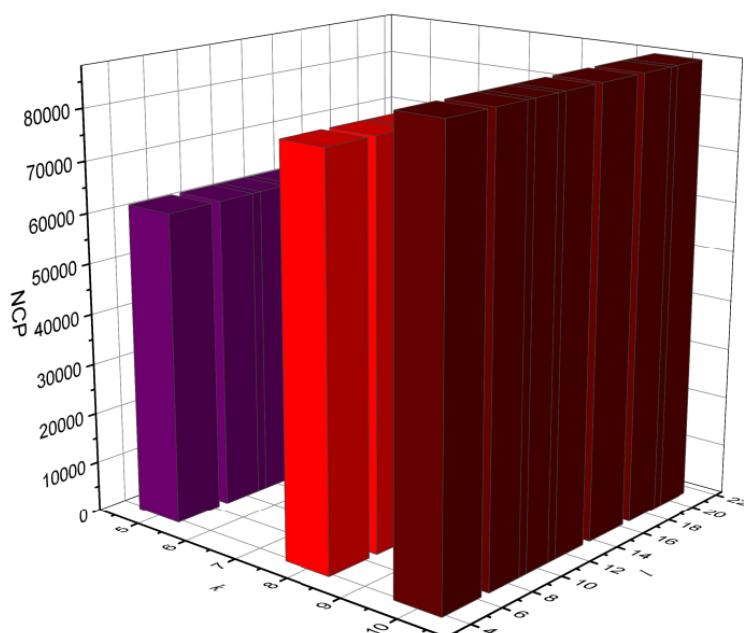


图 5.15 密度为 20% 时 LAG_2 的 NCP 指数

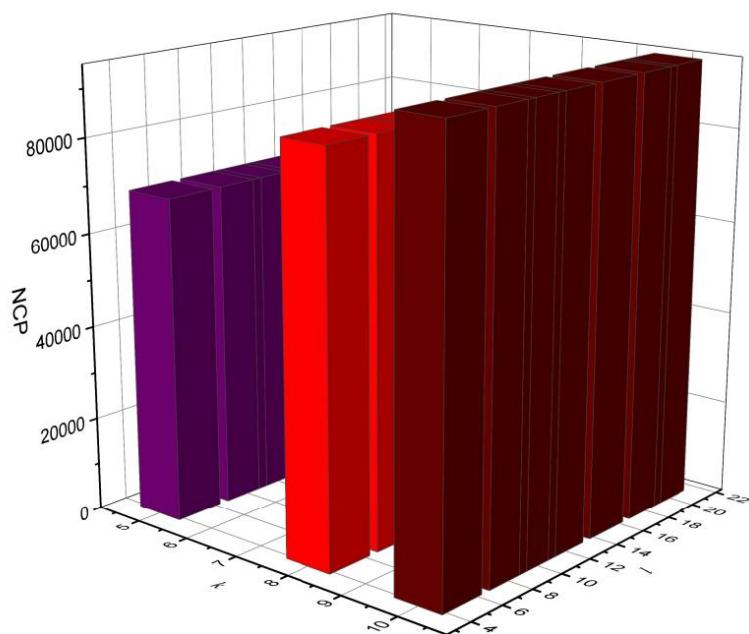


图 5.16 密度为 30% 时 LAG_1 的 NCP 指数

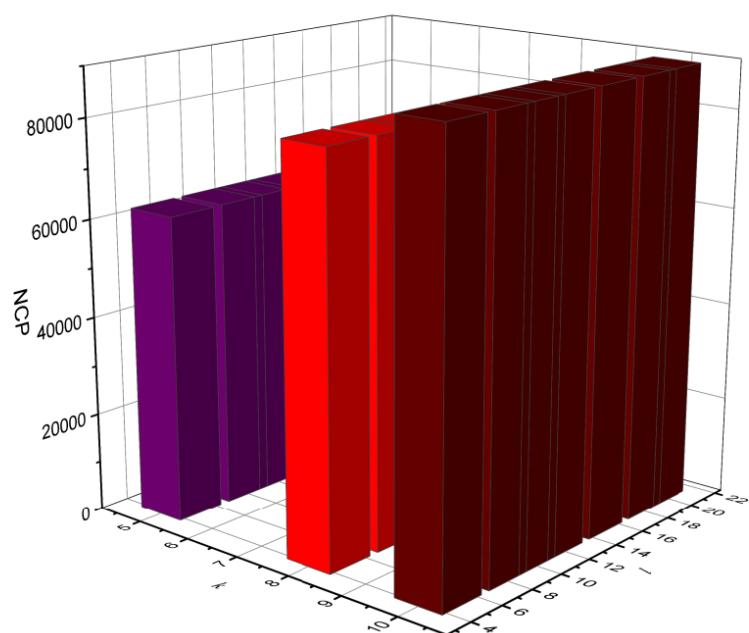


图 5.17 密度为 30% 时 LAG_2 的 NCP 指数

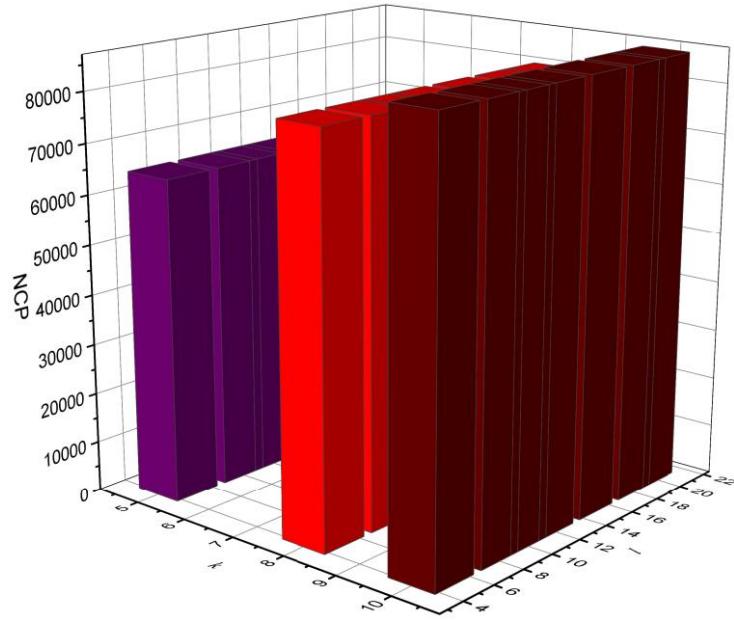


图 5.18 密度为 40% 时 LAG_1 的 NCP 指数

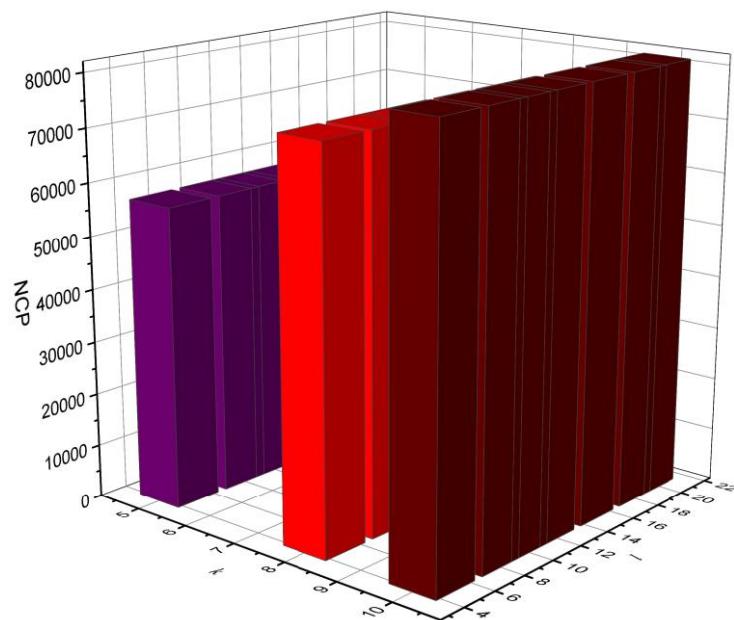


图 5.19 密度为 40% 时 LAG_2 的 NCP 指数

通过对比数据结果可以发现，当参数环境相同时，LAG_2 的 NCP 指数都低于 LAG_1，因此 LAG_2 中泛化算法部分的启发式发挥了作用。同时，由于参数 l 的值没有影响 NCP 度量指数的结果，所以本次实验再次证明了局部分解泛化算法对

用户身份的保护与对敏感值的保护是相互独立的。此外，通过对 LAG_1 的数据观察，随着半敏感属性中的敏感值密度不断增加，NCP 度量指数不断下降。这是由于当敏感值不断增加时泛化 QI 值的数量越来越少，所以降低了局部分解泛化算法在匿名时的信息损失。

最后，我们使用查询回答错误率的方法对匿名数据表进行评估，通过随机生成了 1000 条查询语句对每个匿名数据计算平均查询回答错误率作为比较结果。由于本节实验中的数据表同时包含了 QI 属性、半敏感属性和敏感属性，所以需要使用 4.4.2 节中描述的查询语句配置和查询回答错误率的计算方法，并且，各个参数的设置与之前的实验相同。查询回答错误率的测试结果，如图 5.20 到 5.27 所示。

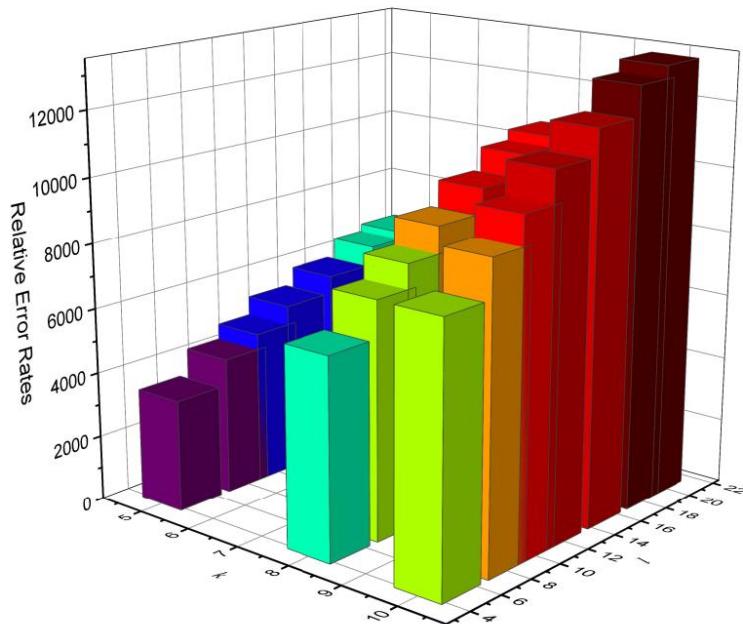


图 5.20 密度为 10% 时 LAG_1 的查询回答错误率

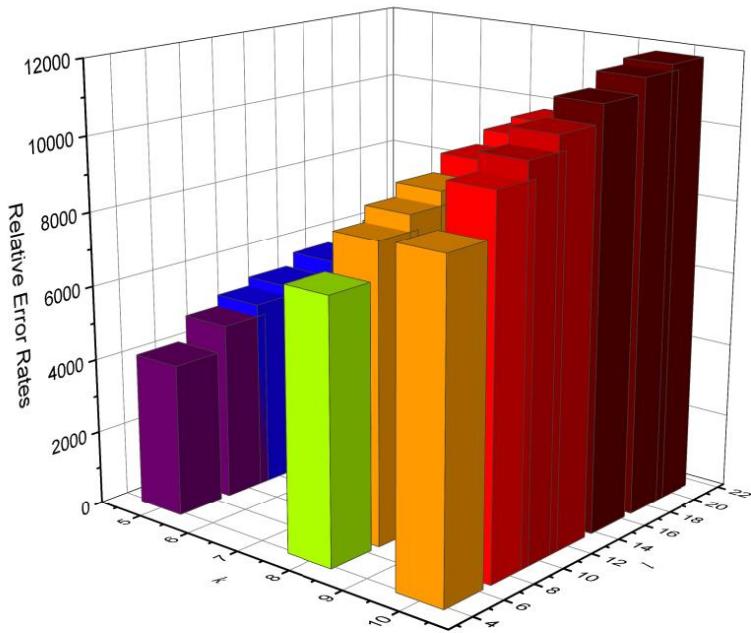


图 5.21 密度为 10% 时 LAG_2 的查询回答错误率

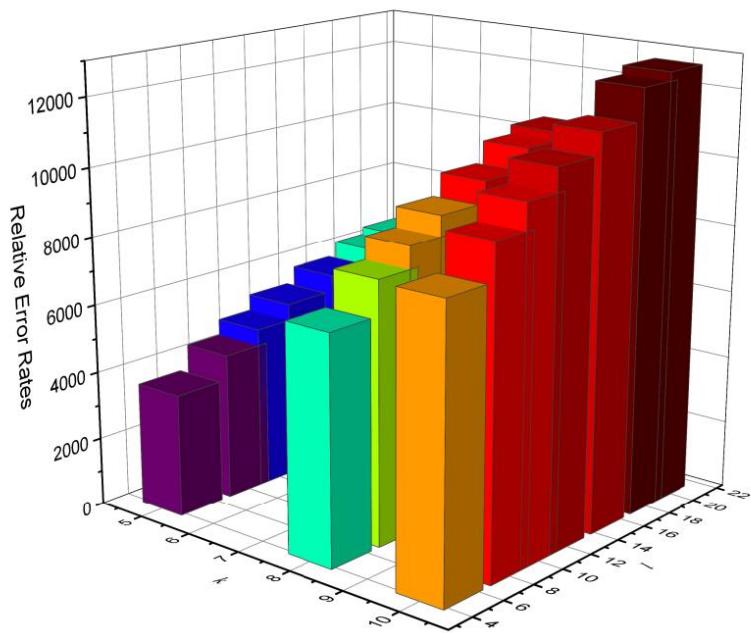


图 5.22 密度为 20% 时 LAG_1 的查询回答错误率

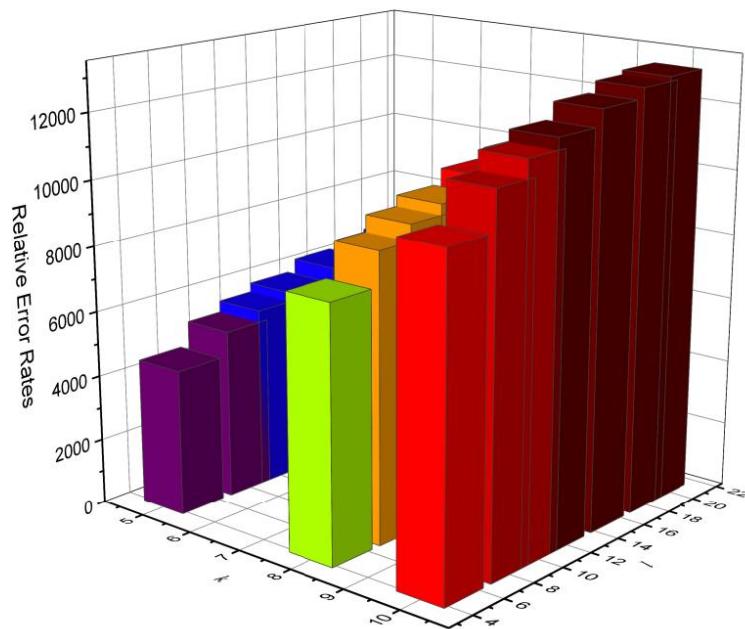


图 5.23 密度为 20% 时 LAG_2 的查询回答错误率

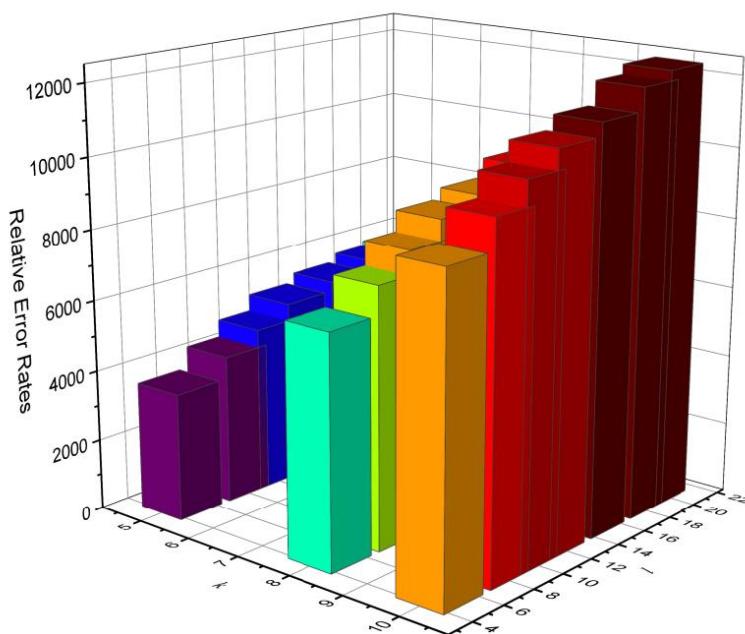


图 5.24 密度为 30% 时 LAG_1 的查询回答错误率

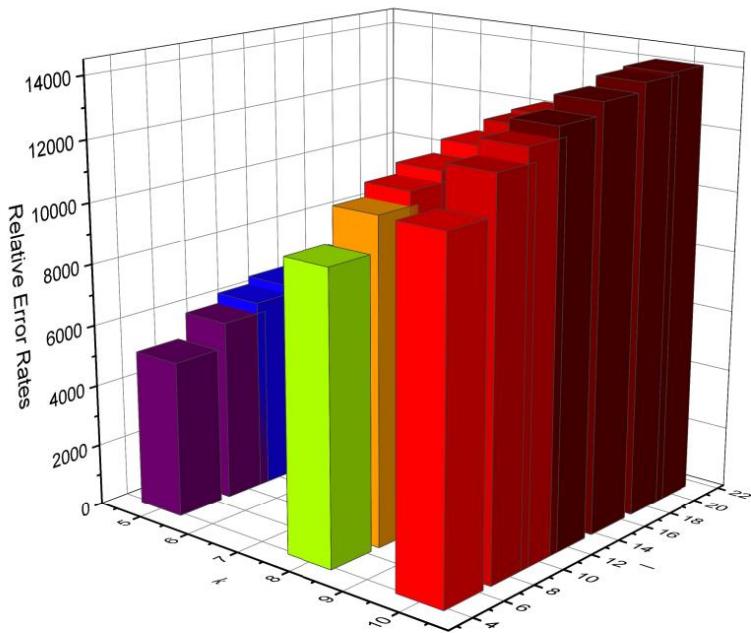


图 5.25 密度为 30% 时 LAG_2 的查询回答错误率

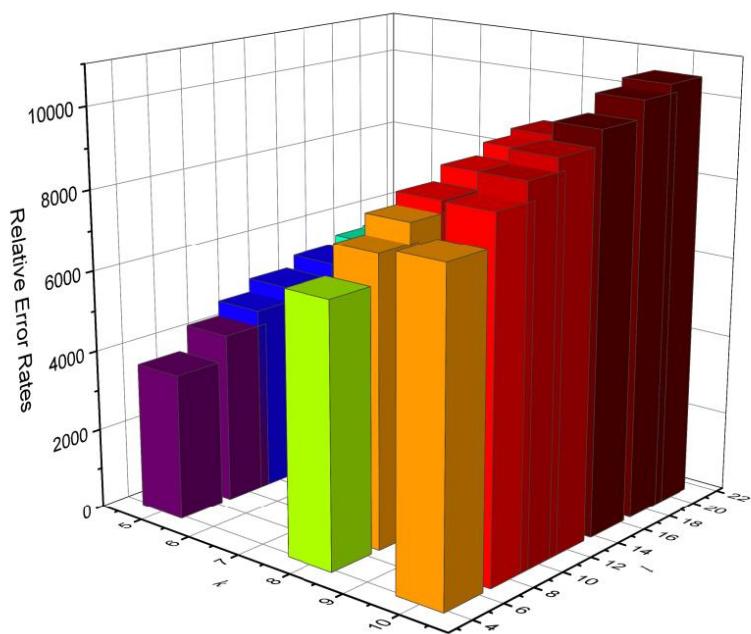


图 5.26 密度为 40% 时 LAG_1 的查询回答错误率

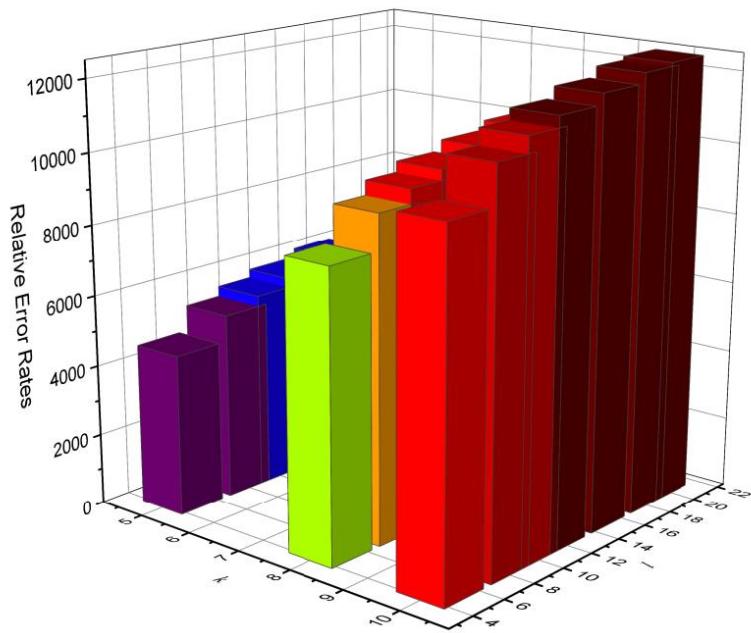


图 5.27 密度为 40% 时 LAG_2 的查询回答错误率

通过对实验数据的观察可以发现，在相同参数的环境下，LAG_1 的查询回答错误率略低于 LAG_2。因此，多维划分技术相比于 NCP 引导的启发式可以保留更加准确的 QI 值信息。此外，参数 k 对查询回答错误率的影响比参数 l 更加明显，并且当参数 k 的值较高时参数 l 对查询回答错误率的影响更加明显。

值得注意的是，尽管在局部分解泛化算法中的泛化机制中加入了启发式以尽可能减少数据在匿名过程中的信息损失，但是相比于 4.4 和 4.5 节中的实验结果，局部分解泛化算法的信息可利用性远远小于局部分解算法。因此，虽然局部分解泛化算法具有更加完善的保护功能，但是在隐私保护需求较低并且信息可利用性需求较高的场景中，局部分解算法可能是更好的选择。

5.5 本章小结

在本章中，我们通过在局部分解算法中加入泛化机制提出了局部分解泛化算法，用于在个性化的隐私保护发布环境中保护数据表中的用户身份和敏感值。局部分解泛化算法具有很高的灵活性，它继承了部分局部分解算法的特性，可以根据实际匿名需求和数据表中属性的特点，同时遵循不同的匿名原则保护数据表中

的隐私信息。并且，局部分解泛化算法为用户身份和敏感值提供了相互独立的保护，因此修改任意方面的保护机制不会降低另一方隐私保护的效果。

相比于局部分解算法，局部分解泛化算法将数据表中的 QI 值进行泛化。一方面，局部分解泛化算法为用户身份提供了可靠的保护，使局部分解泛化算法可以应用于隐私保护需求更高的环境中；另一方面，局部分解泛化算法中的泛化机制使匿名数据表损失了大量的信息可利用性。在未来的工作中，我们将研究使用其他匿名方法替换泛化机制，从而既可以保护数据表中的用户身份，又可以降低信息可利用性的损失。

第6章 总结与展望

6.1 工作总结

本文主要研究了在隐私保护数据发布的过程中，当面对不同的匿名需求时提出适当的匿名算法为数据中的隐私信息提供可靠的保护并且尽可能地减少信息可利用性的损失，具体的主要研究工作如下：

(1) 为用户身份和敏感属性提供相互独立的保护。通过结合泛化算法和桶算法的原理提出了交叉桶泛化算法，将数据表中的个体划分为等价组和桶，从而解决了当使用泛化算法对敏感属性进行保护时匿名数据表对用户身份产生过度保护的问题。并且，由于交叉桶泛化算法对用户身份和敏感属性的保护是相互独立的，所以交叉桶泛化算法可以根据实际匿名需求自由调整对用户身份和敏感属性的保护程度。

(2) 定义个性化的隐私保护发布环境并为数据中的敏感值提供安全的保护。首先定义了个性化的隐私保护发布环境，并且将数据表中的属性划分为 QI 属性、半敏感属性和敏感属性；然后，基于桶算法的原理提出了局部分解算法，在每个包含敏感值的属性内将带有敏感值的个体划分为桶，从而保护数据表中所有的敏感值，并且，局部分解算法具有很高的灵活性，它可以根据不同的匿名需求和数据表中属性的特点，同时遵循不同的匿名原则对数据表中的敏感值进行保护。

(3) 在个性化的隐私保护发布环境中为数据中的用户身份和敏感值提供安全的保护。通过将局部分解算法中加入泛化机制提出了局部分解泛化算法，将数据表中的个体根据携带 QI 值的情况划分为多个子集，然后在每个子集内将其中的个体划分为等价组，从而为用户身份提供独立的保护。此外，由于局部分解泛化算法对用户身份和敏感值的保护是相互独立的，所以使用不同的泛化机制不会降低对敏感值的保护效果。

综上，本文主要在一定的匿名需求和发布环境中为数据表中的敏感信息提供保护方法进行了深入地研究，并且取得了一定显著的研究成果。

6.2 研究展望

近年来，人工智能技术得到了突破性的发展。学者们主要通过使用数据样本训练不同模型的机器学习算法，使计算机可以模拟人类大脑对信息进行处理。但是，训练算法模型的成功与否在很大程度上取决于数据样本的容量和质量，并且，由于未来的人工智能模型需要处理更加复杂和多样化的信息，所以行业之间大规模和高质量的数据交互必将成为未来的发展趋势。

为了防止数据在进行交换或者发布时造成用户的隐私泄露，并且同时在匿名数据中保留更多的信息可利用性，需要考虑更加复杂的发布环境和更加有效的匿名算法。例如，本文提出的个性化隐私保护发布环境从用户的角度出发允许用户在数据表中自由设置敏感值从而更加贴近于实际应用，然而，目前绝大多数的匿名算法仍然停留于只对单一静态的数据表进行保护而忽略了动态应用环境。随着时间的推移，在多重发布和动态发布等数据发布环境中进行隐私保护将逐渐成为匿名保护的研究热点。

在未来的工作中，我们将主要研究在个性化的隐私保护发布环境中进行动态发布使得匿名算法的应用更加广泛，并且，使用新的匿名算法替代泛化算法用于对用户身份进行保护，从而降低数据在匿名过程中的损失。

参考文献

- [1] 李国杰, 程学旗. 大数据研究: 未来科技及经济社会发展的重大战略领域——大数据的研究现状与科学思考[J]. 中国科学院院刊, 2012, 27(06): 647–657.
- [2] 何清, 李宁, 罗文娟, 等. 大数据下的机器学习算法综述[C]. 中国计算机学会人工智能会议, 2013.
- [3] 孟小峰, 慈祥. 大数据管理: 概念、技术与挑战[J]. 计算机研究与发展, 2013, 50(01): 146–169.
- [4] 冯登国, 张敏, 李昊. 大数据安全与隐私保护[J]. 计算机学报, 2014, 37(1): 246–258.
- [5] Walker S J, Viktor M-S, Kenneth C. Big Data: A Revolution That Will Transform How We Live, Work, and Think[J]. International Journal of Advertising, 2014.
- [6] 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.
- [7] 刘向宇, 王斌, 杨晓春. 社会网络数据发布隐私保护技术综述[J]. 软件学报, 2014, 25(3): 576–590.
- [8] 张啸剑, 孟小峰. 面向数据发布和分析的差分隐私保护[J]. 计算机学报, 2014(4): 927–949.
- [9] Barbaro M, Zeller T. A face is exposed for AOL Searcher no. 4417749[J]. New York Times, 2006.
- [10] Cox L. Suppression Methodology and Statistical Disclosure Control[J]. Publications of the American Statistical Association, 1980, 75(370): 377–385.
- [11] Dalenius T. Towards a methodology for statistical disclosure control[J]. Statistik Tidskrift, 1977, 15: 429–444.
- [12] Miklau G, Suciu D. A formal analysis of information disclosure in data exchange[J]. Journal of Computer and System Sciences, 2007, 73(3): 507–534.
- [13] Machanavajjhala A, Gehrke J. On the efficiency of checking perfect privacy[C]. Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on

- Principles of database systems. ACM, 2006: 163–172.
- [14] Latanya S. Achieving k-anonymity privacy protection using generalization and suppression[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05):571–588.
- [15] Xiao X, Tao Y. Anatomy: simple and effective privacy preservation[C]. International Conference on Very Large Data Bases. VLDB Endowment, 2006:139–150.
- [16] Dwork C. Differential privacy: A survey of results[C]. International Conference on Theory and Applications of Models of Computation, 2008: 1–19.
- [17] Dalenius T. Finding a needle in a haystack—or identifying anonymous census record[J]. Journal of Official Statistics, 1986, 2(3):935–936.
- [18] Samarati P. Protecting respondents identities in microdata release[J]. IEEE Transactions on Knowledge & Data Engineering, 2001, 13(6):1010–1027.
- [19] Samarati P, Sweeney L. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression[R]. Technical report, SRI International, 1998.
- [20] Latanya S. k-Anonymity: a model for protecting privacy[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 2002, 10(05):557–570.
- [21] Chen B C, Kifer D, Lefevre K, et al. Privacy-Preserving Data Publishing[J]. Foundations & Trends in Databases, 2009, 2(1 – 2):1–167.
- [22] Backstrom L, Dwork C, Kleinberg J. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography[C]. International Conference on World Wide Web. ACM, 2007:181–190.
- [23] Hay M, Miklau G, Jensen D, et al. Resisting structural re-identification in anonymized social networks[J]. VLDB Journal, 2010, 19(6):797–823.
- [24] Narayanan A, Shmatikov V. De-anonymizing Social Networks[J]. IEEE Symposium on Security and Privacy, 2009:173–187.
- [25] Liu K, Terzi E. Towards identity anonymization on graphs[C]. ACM SIGMOD International Conference on Management of Data. ACM, 2008:93–106.

- [26] Zhou B, Pei J. Preserving Privacy in Social Networks against Neighborhood Attacks[C]. IEEE, International Conference on Data Engineering. IEEE, 2008: 506–515.
- [27] Zheleva E, Getoor L. Preserving the privacy of sensitive relationships in graph data[C]. ACM SIGKDD International Conference on Privacy, Security, and Trust in Kdd. Springer-Verlag, 2007:153–171.
- [28] Jones R, Kumar R, Pang B, et al. "I know what you did last summer": query logs and user privacy[C]. Proceedings of the 16th Conference on Information and Knowledge Management, 2007.
- [29] Kumar R, Novak J, Pang B, et al. On anonymizing query logs via token-based hashing[C]. World Wide Web Conference Series. 2007:629–638.
- [30] Adar E. User 4xxxxx9: Anonymizing query logs[J]. Workshop on Query Log Analysis at WWW, 2007, 43: 71–77.
- [31] Hong Y, He X, Vaidya J, et al. Effective anonymization of query logs[C]. Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009: 1465–1468.
- [32] Gotz M, Machanavajjhala A, Wang G, et al. Publishing search logs — a comparative study of privacy guarantees[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(3): 520–532.
- [33] Korolova A, Kenthapadi K, Mishra N, et al. Releasing search queries and clicks privately[C]. Proceedings of the 18th international conference on World wide web. ACM, 2009: 171–180.
- [34] Gedik B, Liu L. Location Privacy in Mobile Systems: A Personalized Anonymization Model[C]. IEEE International Conference on Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. IEEE, 2005: 620–629.
- [35] Gedik B, Liu L. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms[J]. IEEE Transactions on Mobile Computing, 2007, 7(1): 1–18.
- [36] Gruteser M, Grunwald D. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking[C]. International Conference on Mobile Systems,

- Applications, and Services. DBLP, 2003: 31–42.
- [37] Kalnis P, Ghinita G, Mouratidis K, et al. Preventing Location-Based Identity Inference in Anonymous Spatial Queries[J]. IEEE Transactions on Knowledge & Data Engineering, 2007, 19(12): 1719–1733.
- [38] Mokbel M F, Chow C Y, Aref W G. The new Casper:query processing for location services without compromising privacy[C]. International Conference on Very Large Data Bases. VLDB Endowment, 2006: 763–774.
- [39] Abul O, Bonchi F, Nanni M. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases[C]. IEEE, International Conference on Data Engineering. IEEE Computer Society, 2008: 376–385.
- [40] Nergiz M E, Ercan M, Atzori, et al. Towards trajectory anonymization: a generalization-based approach[J]. Transactions on Data Privacy, 2009, 2(1): 52–61.
- [41] Terrovitis M, Mamoulis N. Privacy Preservation in the Publication of Trajectories[C]. International Conference on Mobile Data Management. IEEE, 2008: 65–72.
- [42] Gruteser M, Hoh B. On the Anonymity of Periodic Location Samples[C]. Security in Pervasive Computing, Second International Conference, SPC 2005, Boppard, Germany, April 6–8, 2005, Proceedings. DBLP, 2005: 179–192.
- [43] Krumm J. Inference Attacks on Location Tracks[C]. Pervasive Computing, International Conference, Pervasive 2007, Toronto, Canada, May 13–16, 2007, Proceedings. DBLP, 2007: 127–143.
- [44] Hoh B, Gruteser M, Xiong H, et al. Preserving privacy in GPS traces via uncertainty-aware path cloaking[C]. ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October. DBLP, 2007: 161–171.
- [45] Fung B, Wang K, Chen R, et al. Privacy-preserving data publishing: A survey of recent developments[J]. ACM Computing Surveys (CSUR), 2010.
- [46] Wang K, Fung B C M. Anonymizing sequential releases[C]. Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia,

- Pa, Usa, August. DBLP, 2006:414–423.
- [47] Nergiz M E, Clifton C, Nergiz A E. MultiRelational k-Anonymity[J]. IEEE Transactions on Knowledge & Data Engineering, 2009, 21(8):1104–1117.
- [48] Tao Y, Xiao X, Li J, et al. On Anti-Corruption Privacy Preserving Publication[C]. IEEE, International Conference on Data Engineering. IEEE, 2008:725–734.
- [49] Nergiz M E, Atzori M, Clifton C. Hiding the presence of individuals from shared databases[C]. ACM SIGMOD International Conference on Management of Data. ACM, 2007:665–676.
- [50] Machanavajjhala A, Gehrke J, Kifer D, et al. l-Diversity: privacy beyond k-anonymity[C]. International Conference on Data Engineering. IEEE, 2006.
- [51] Machanavajjhala A, Kifer D, Gehrke J, et al. l-Diversity: Privacy beyond k-anonymity[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2007.
- [52] Li N, Li T, Venkatasubramanian S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity[C]. IEEE, International Conference on Data Engineering. IEEE, 2007:106–115.
- [53] Wong R C-W, Li J, Fu A W-C, et al. (α, k) -anonymity: an enhanced k-anonymity model for privacy preserving data publishing[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006:754–759.
- [54] Zhang Q, Koudas N, Srivastava D, et al. Aggregate Query Answering on Anonymized Tables[C]. IEEE, International Conference on Data Engineering. IEEE, 2007:116–125.
- [55] Wang K, Xu Y, Fu A W C, et al. FF-Anonymity: When Quasi-identifiers Are Missing[C]. IEEE, International Conference on Data Engineering. IEEE, 2009:1136–1139.
- [56] Cao J, Karras P. Publishing Microdata with a Robust Privacy Guarantee[J]. Proceedings of the Vldb Endowment, 2012, 5(11):1388–1399.
- [57] Li J, Tao Y, Xiao X. Preservation of proximity privacy in publishing numerical sensitive data[C]. Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008: 473–486.

- [58] Li N, Li T, Venkatasubramanian S. Closeness: A new privacy measure for data publishing[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(7) : 943–956.
- [59] Wang K, Fung B C M, Philip S Y. Handicapping attacker's confidence: an alternative to k-anonymization[J]. Knowledge and Information Systems, 2007, 11(3) : 345–368.
- [60] Dwork C. Differential privacy[J]. Lecture Notes in Computer Science, 2006, 26(2) : 1–12.
- [61] Han C, Wang K. Sensitive Disclosures under Differential Privacy Guarantees[C]. IEEE International Congress on Big Data. IEEE, 2015:110–117.
- [62] Wang K, Han C, Fu A W, et al. Reconstruction Privacy: Enabling Statistical Learning[C]. Proceedings of 18th International Conference on Extending Database Technology, 2015: 469–480.
- [63] Hay M, Rastogi V, Miklau G, et al. Boosting the accuracy of differentially private histograms through consistency[J]. Proceedings of the VLDB Endowment, 2010, 3(1–2) : 1021–1032.
- [64] Xu J, Zhang Z, Xiao X, et al. Differentially private histogram publication[J]. The VLDB Journal, 2013, 22(6) : 797–822.
- [65] Chaudhuri K, Monteleoni C, Sarwate A D. Differentially private empirical risk minimization[J]. Journal of Machine Learning Research, 2011, 12(Mar) : 1069–1109.
- [66] Li C, Miklau G. An adaptive mechanism for accurate query answering under differential privacy[J]. Proceedings of the VLDB Endowment, 2012, 5(6) : 514–525.
- [67] Yaroslavtsev G, Cormode G, Procopiuc C M, et al. Accurate and efficient private release of datacubes and contingency tables[C]. Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013: 745–756.
- [68] Cormode G, Procopiuc C, Srivastava D, et al. Differentially private spatial decompositions[C]. Data engineering (ICDE), 2012 IEEE 28th international conference on. IEEE, 2012: 20–31.

- [69] Chen R, Mohammed N, Fung B C M, et al. Publishing set-valued data via differential privacy[J]. Proceedings of the VLDB Endowment, 2011, 4(11): 1087–1098.
- [70] Chen R, Fung B, Desai B C, et al. Differentially private transit data publication: a case study on the montreal transportation system[C]. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012: 213–221.
- [71] Inan A, Kantarcioglu M, Ghinita G, et al. Private record matching using differential privacy[C]. Proceedings of the 13th International Conference on Extending Database Technology. ACM, 2010: 123–134.
- [72] Peng S, Yang Y, Zhang Z, et al. DP-tree: indexing multi-dimensional data under differential privacy[C]. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012.
- [73] Cormode G, Procopiuc C, Srivastava D, et al. Differentially private summaries for sparse data[C]. Proceedings of the 15th International Conference on Database Theory. ACM, 2012: 299–311.
- [74] Bhaskar R, Laxman S, Smith A, et al. Discovering frequent patterns in sensitive data[C]. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010: 503–512.
- [75] Li N, Qardaji W, Su D, et al. Privbasis: Frequent itemset mining with differential privacy[J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1340–1351.
- [76] Zeng C, Naughton J F, Cai J Y. On differentially private frequent itemset mining[J]. Proceedings of the VLDB Endowment, 2012, 6(1): 25–36.
- [77] Blum A, Dwork C, McSherry F, et al. Practical privacy: the SuLQ framework[C]. Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2005: 128–138.
- [78] Friedman A, Schuster A. Data mining with differential privacy[C]. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010: 493–502.

- [79] Mohammed N, Chen R, Fung B, et al. Differentially private data release for data mining[C]. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 493–501.
- [80] Smith A. Privacy-preserving statistical estimation with optimal convergence rates[C]. Proceedings of the forty-third annual ACM symposium on Theory of computing. ACM, 2011: 813–822.
- [81] Lei J. Differentially private m -estimators[C]. Advances in Neural Information Processing Systems. 2011: 361–369.
- [82] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis[C]. TCC. 2006: 265–284.
- [83] Xiao X, Wang G, Gehrke J. Differential privacy via wavelet transforms[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(8): 1200–1214.
- [84] Qardaji W, Yang W, Li N. Differentially private grids for geospatial data[C]. Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013: 757–768.
- [85] McSherry F D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis[C]. Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009: 19–30.
- [86] Peng S, Yang Y, Zhang Z, et al. Query optimization for differentially private data management systems[C]. Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013: 1093–1104.
- [87] Mohan P, Thakurta A, Shi E, et al. GUPT: privacy preserving data analysis made easy[C]. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012: 349–360.
- [88] Roy I, Setty S T V, Kilzer A, et al. Airavat: Security and privacy for MapReduce[C]. NSDI. 2010, 10: 297–312.
- [89] Lefevre K, Dewitt D J, Ramakrishnan R. Incognito: efficient full-domain K-anonymity[C]. ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June. DBLP, 2005:49–60.
- [90] Iyengar V S. Transforming data to satisfy privacy constraints[C]. Eighth

- ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
ACM, 2002:279–288.
- [91] Bayardo R J, Agrawal R. Data privacy through optimal k-anonymization[C]. Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. IEEE, 2005: 217–228.
- [92] Fung B C M, Wang K, Yu P S. Top-down specialization for information and privacy preservation[C]. Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. IEEE, 2005: 205–216.
- [93] Fung B C M, Wang K, Yu P S. Anonymizing Classification Data for Privacy Preservation[J]. IEEE Transactions on Knowledge & Data Engineering, 2007, 19(5):711–725.
- [94] Xu J, Wang W, Pei J, et al. Utility-based anonymization using local recoding[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006:785–790.
- [95] Wang K, Yu P S, Chakraborty S. Bottom-up generalization: A data mining solution to privacy protection[C]. Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. IEEE, 2004: 249–256.
- [96] Lefevre K, Dewitt D J, Ramakrishnan R. Mondrian Multidimensional K-Anonymity[C]. International Conference on Data Engineering. IEEE, 2006.
- [97] LeFevre K, DeWitt D J, Ramakrishnan R. Workload-aware anonymization[C]. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006: 277–286.
- [98] Aggarwal C C. On k-anonymity and the curse of dimensionality[C]. International Conference on Very Large Data Bases, Trondheim, Norway, August 30 – September. DBLP, 2005:901–909.
- [99] Li T, Li N. On the tradeoff between privacy and utility in data publishing[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2009:517–526.
- [100] Kifer D, Gehrke J. Injecting utility into anonymized datasets[C]. ACM SIGMOD International Conference on Management of Data. ACM, 2006:217–228.

- [101] Li T, Li N, Zhang J, et al. Slicing: A New Approach for Privacy Preserving Data Publishing[J]. IEEE Transactions on Knowledge & Data Engineering, 2012, 24(3):561–574.
- [102] Meyerson A, Williams R. On the complexity of optimal k-anonymity[C]. Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2004: 223–228.
- [103] Aggarwal G, Feder T, Kenthapadi K, et al. Approximation algorithms for k-anonymity[J]. Journal of Privacy Technology, 2005.
- [104] Sweeney L. Datafly: A system for providing anonymity in medical data[M]. Database Security XI. Springer US, 1998: 356–381.
- [105] Xiao X, Tao Y. Personalized privacy preservation[C]. Proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM, 2006: 229–240.
- [106] Wang K, Fung B C M, Yu P S. Template-based privacy preservation in classification problems[C]. Data Mining, Fifth IEEE International Conference on. IEEE, 2005.
- [107] Vinterbo S A. Privacy: a machine learning view[J]. IEEE Transactions on knowledge and data engineering, 2004, 16(8): 939–948.
- [108] Dalvi N, Suciu D. Efficient query evaluation on probabilistic databases[J]. The VLDB Journal—The International Journal on Very Large Data Bases, 2007, 16(4): 523–544.
- [109] Burdick D, Deshpande P M, Jayram T S, et al. OLAP over uncertain and imprecise data[C]. Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005: 970–981.
- [110] Burdick D, Deshpande P M, Jayram T S, et al. Efficient allocation algorithms for OLAP over imprecise data[C]. Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006: 391–402.
- [111] Agrawal R, Srikant R, Thomas D. Privacy preserving OLAP[C]. Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, 2005: 251–262.

- [112] Muralidhar K, Sarathy R. A theoretical basis for perturbation methods[J]. Statistics and Computing, 2003, 13(4): 329–335.
- [113] Xiao X, Tao Y. m-invariance: towards privacy preserving re-publication of dynamic datasets[C]. Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007: 689–700.
- [114] Yao C, Wang X S, Jajodia S. Checking for k-anonymity violation by views[C]. Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005: 910–921.
- [115] Byun J W, Sohn Y, Bertino E, et al. Secure anonymization for incremental datasets[J]. Secure Data Management, 2006, 6: 48–63.
- [116] Cao J, Karras P, Kalnis P, et al. SABRE: a Sensitive Attribute Bucketization and Redistribution framework for t-closeness[J]. The International Journal on Very Large Data Bases, 2011, 20(1): 59–81.
- [117] Friedman J H, Bentley J L, Finkel R A. An algorithm for finding best matches in logarithmic expected time[J]. ACM Transactions on Mathematical Software (TOMS), 1977, 3(3): 209–226.

作者简介及科研成果

1. 作者简介:

2. 发表论文:

- [1] **第一作者**. Cross-Bucket Generalization for Information and Privacy Preservation[J]. IEEE Transactions on Knowledge and Data Engineering(SCI, 1 区, CCF A 类), 2018, 30(3):449–459.
- [2] **第一作者**. Reverse twin plant for efficient diagnosability testing and optimizing[J]. Engineering Applications of Artificial Intelligence(SCI, 2 区, CCF C 类), 2015, 38:131–137.
- [3] **非第一作者**. An Antenna Array Sidelobe Level Reduction Approach through Invasive Weed Optimization[J]. International Journal of Antennas and Propagation(SCI), 2018.

致 谢