北京郵電大學

本 科 毕 业 设 计

题目：满足本地化差分隐私的频繁项挖掘算法设计与实现

姓　　名：张俊

学　　院：电子工程学院

专　　业：电子信息科学与技术

班　　级：2014211201

学　　号：2014210808

班内序号：11

指导教师：程祥

2018 年 6 月

满足本地化差分隐私的频繁项挖掘算法设计与实现

摘　要

随着信息技术的发展、智能终端的普及和传感器的广泛使用，全球数据量爆炸式地增长。当今社会已然进入大数据时代。集值数据是一类典型的大数据。通过挖掘用户集值数据的频繁项，数据收集者可以了解用户的偏好，为决策提供支持。然而，集值数据中含有用户的大量敏感信息，直接发布频繁项及相应的计数或频率有可能导致用户隐私的泄露。本地化差分隐私模型作为当前最为先进的隐私保护模型，为解决频繁项挖掘中的隐私安全问题供了一种可行的方案。

本文提出了一种满足 ε-本地化差分隐私的频繁项挖掘算法 GFIM（Group-based Frequent Items Mining）。该算法的主要思想是首先把用户随机等分成两组，基于第一组用户的数据我们得到了频繁项的候选集，再利用第二组用户的数据我们得到候选集各项更精确的估计频率，最后综合两个阶段的成果得到最终的估计频繁项及对应频率。本文从理论上证明了 GFIM 满足 ε-本地化差分隐私。并且，基于合成数据集和真实数据集的实验结果证明了该算法的合理性和优越性。

关键词　集值数据　频繁项挖掘　本地化差分隐私

Design and Implementation of Frequent Item Mining with Local Differential Privacy

ABSTRACT

With the development of information technology, popularity of smart devices and the extensive uses of sensors, the data volume in the world presents explosively increasing. Nowadays, our society has entered the era of big data. Set-valued data is one kind of classical big data. By mining the frequent items of users' set-valued data, data aggregator can learn about the preferences of users, which can provide support for decisions. However, set-valued data contains a great number of sensitive information of users, directly reporting the frequent items and the corresponding counts or frequencies could lead to the leakage of users' privacy. As the state-of-the-art privacy protection model, local differential privacy(LDP) provides a feasible solution for such problem.

In this paper, we propose a frequent items mining algorithm with ε-LDP, named Group-based Frequent Items Mining (GFIM). The main idea of this algorithm is to first split the users randomly into two groups with the same size. Basing on the user data of the first group we gather the candidate set of possible frequent items, then we refine the candidate set using the user data of the second group. Finally, we obtain the estimated frequent items and the corresponding frequencies by combining the results of these two phases. This paper theoretically proves that GFIM satisfies ε-LDP. In addition, extensive experiments on synthetic dataset and real dataset demonstrate its effectiveness and superiority over existing method.

Key words set-valued data local differential privacy frequent item mining

目 录

第一章　绪论

1.1　研究背景

当今社会，信息技术融入到人们生活的方方面面。从日常出行，购物，饮食，工作，社交到住房，医疗，教育，研究，人类生活的几乎每个方面都离不开现代电子设备和信息技术。这些方面产生的海量数据被以数字化的形式存储起来，构成了人们常说的大数据。大数据里面蕴藏着巨大的价值，研究和挖掘大数据能帮助我们加深对现状的理解和更准确地预测未来。譬如，通过研究某商场一段时期的销售记录，我们能了解到该段时期内销售商品的种类构成，所占比例，消费人群等信息，同时我们能根据研究结果预测未来一段时间的畅销产品和增长趋势，进而帮助商家更好地决策以增加销售和利润。

在数据挖掘领域中，频繁项挖掘是一项非常重要的话题。频繁项挖掘的目的是找出频繁出现在数据集中的项，它是关联规则、相关性分析、分类、聚类和其他数据挖掘任务的基础[1]。然而，对于包含用户敏感信息的数据集，直接发布由源数据集得到的频繁项及其对应计数或频率都有可能导致用户隐私的泄露。为此，我们必须采用可靠的隐私保护技术来保护频繁项挖掘过程中用户隐私的安全。

最早期的隐私保护技术采用置空和模糊化等手段保护用户的隐私。这类手段简单直接，至今仍被广泛使用。如火车票上被星号隐去的部分身份证号码，就是采用置空来保护用户的隐私。这类方法的缺点也很明显，保护手段过于简单粗暴，攻击者在获取关于用户的足够背景知识后能轻易推出被置空的信息。仍以被置空保护的身份证号为例，身份证号的组成是有规律的，其中有一部分是用户的出身年月日，假若被置空的字段恰好是该部分，攻击者已知用户姓名和被置空的身份证号，再通过其他途径得知用户的出身年月日（这个并不困难，可从用户的社交平台得知）后，便可获得用户的完整身份证号码。这种通过把关于用户的多方面的信息链接起来以推出用户隐私的方法称为链接攻击[2]。而被用来链接两方面信息，被置空的身份证号和出生年月日的信息，这个例子中的姓名被称为准标识符。1998 年，Samaratip 和 Sweeney 首次提出匿名化技术 k-anonymity[3, 4]。简单而言，k-anonymity 的原则是，在同一数据库中，具有某一准标识符的记录至少有 k 条，准标识符可以为出生年月，性别，邮编等能用于链接其他数据表的信息。因为具有某一准标识符的记录至少有 k 条，假若攻击者仅知道用户的该个准标识符，则他至多有 1/k 的概率能准确识别的具体用户并获取其隐私。k-anonymity 是各种匿名化的基础，它能一定程度上防止攻击者使用链接攻击获取用户隐私。但匿名化技术仍不是完美的，它们对用户隐私的保护仍然极其有限。匿名化技术与上文提到的简单隐私保护技术一样，有两个突出的缺陷：一是它们对隐私的保护程度与攻击者的背景知识相关。换言之，只要攻击者掌握的关于用户的背景知识足够多，攻击者仍能以较高的概率获得用户的敏感信息。二是上述技术并没有提供一个严谨的数学意义上的隐私保护模型，无法对隐私保护的能力进行定量分析。差分隐私（Differential Privacy, DP）[5-7]的出现很好地解决了这两个缺陷。

差分隐私由 Dwork 于 2006 年首次提出。它最初是针对中心化的数据库的隐私问题定义的。根据该定义，对于数据库的查询计算对单个数据不敏感。单条数据的加入或删去对查询结果的影响极其微小。攻击者无法通过观察新纪录加入后查询结果的变化而获得准确的个体信息。

差分隐私能够解决上述简单隐私保护技术和匿名化技术共同拥有的两个缺陷。首先，差分隐私不关心攻击者拥有的背景知识。它假设攻击者掌握除目标记录外的所有信息，这也是攻击者所能拥有的最大背景知识。因而它对隐私的保护是极其严密的。其次，通过引入隐私参数（又称隐私预算），差分隐私定义了严格的隐私保护模型，能对隐私保护的能力进行量化评估。ε 越大，隐私保护能力越弱，同时数据的效用性越高，ε 越小，隐私保护能力越强，数据的效用性越低。通过设置合理的隐私参数大小，我们能获得隐私保护能力和数据效用性的平衡。经过十数年的发展，由传统的中心化差分隐私模型发展出了本地差分隐私模型[8, 9]。不同于中心化差分隐私模型，本地化差分隐私模型中用户首先在本地设备上对个人数据进行隐私化处理，然后再把扰动数据发送给数据收集者。如此，本地化差分隐私规避了由不可靠的数据收集者带来的泄露风险，能够提供更高程度的隐私保护。在本文的工作中，我们采用本地化差分隐私技术保护频繁项挖掘过程中的用户隐私安全。

1.2 研究的意义

现代信息社会于 2015 年进入大数据时代。此后，大数据的价值被学界和业界广泛接受。各种产品的提供商都试图通过收集和挖掘大量用户数据以帮助决策，提高收益。然而，用户隐私泄露事件层出不穷，民众对个人隐私的保护需求远远得不到满足。今年 3 月，Facebook 曝出史上最大规模用户数据外泄事件，高达 8700 万用户的个人数据被滥用。同年 5 月 25 日，史上最严格的数据保护条例《一般数据保护条例》（General Data Protection Regulation）在欧盟地区正式生效。可见，隐私问题已经成为了用户和企业、政府不得不考虑的问题，如何在满足一定程度的隐私保护的同时尽可能准确地挖掘有价值信息正是本文要解决的问题。

本次毕业设计的题目为“满足本地化差分隐私的频繁项挖掘算法设计与实现”。频繁项挖掘为数据挖掘领域的重要话题，在实际中有十分广泛的应用，如可以了解一段时期的畅销商品，流行音乐/明星，热度最高的话题等。频繁项挖掘经过多年的研究，已有 Apriori 等多种算法。而对于满足当前最为先进的隐私模型——本地化差分隐私模型的频繁项挖掘，学界中的研究还很有限。尤其是对于集值数据的满足本地化差分隐私的频繁项挖掘这一话题，目前仅有 Zhan Qin 等人提出的 LDPMiner[10]等研究成果。LDPMiner 提出了一种两阶段的频繁项挖掘算法，在第一阶段中数据收集者首先确定一个频繁项的候选集并发送给用户，数据收集者然后利用第二阶段的用户数据对候选集中的各项进行更精确的频率估计。由于 LDPMiner 在第一和第二阶段中对相同的数据集采用两次随机算法，根据隐私参数的序列组合性，它不得不把隐私参数进行划分，由此也限制了 LDPMiner 的准确性。针对这个问题，本文提出的 GFIM（Group-based Frequent Items

Mining）算法通过把用户划分成不相交的两组，避免了划分隐私参数，实现了优于LDPMiner 的频繁项挖掘，具有重要的研究意义。

1.3 毕业设计所完成工作

在本次毕业设计中，我的任务主要包括： 1)学习理解差分隐私的概念及相关背景知识；2）学习满足本地化差分隐私的频繁项挖掘的主要算法并总结其优劣；3）根据之前的研究，设计并实现满足本地化差分隐私的频繁项挖掘算法；4）对算法进行测试和评估。具体内容如下：

一、学习理解差分隐私的概念及相关背景知识：由于我之前没有了解过相关的背景，首先我需要从阅读学习文献综述开始对差分隐私有一个大概的认识，然后通过查阅各种文献对差分隐私的具体概率、数学模型、性质等有深刻的理解。同时，我还要理解本地化差分隐私和传统的中心化差分隐私的区别和优劣。

二、学习满足本地化差分隐私的频繁项挖掘的主要算法并总结其优劣：差分隐私最早于 2006 年提出，此后，业内学者们对它的研究主要集中在隐私保护数据发布和隐私保护数据挖掘两个方向。本文属于隐私保护数据挖掘方向，前人已有一些研究成果。通过学习和总结前人成果，我发现已有的解决方案仍然存在准确率不够高等问题。解决这个问题将是本文的最终目的。

三、设计并实现满足本地化差分隐私的频繁项挖掘算法：设计出相应算法并用 Python 实现，其中包括加噪算法和解码算法。

四、对算法进行测试和评估：用合成数据和真实数据对算法进行测试。真实数据将采用学界常用的公共数据集。测试得到结果将与现有算法进行比较，总结评价其优劣。

1.4 本文组织结构

本文旨在解决满足一定隐私保护能力的数据挖掘问题，设计并实现满足本地化差分隐私的频繁项挖掘算法。该算法在输入较大规模数据集的情况下，能够准确挖掘用户的频繁项信息，并且保护用户的记录不能被准确识别。根据本次毕业设计的研究内容，本文将按以下结构分为六章：

第一章为绪论，简要介绍差分隐私的起源，研究的意义及本文的主要工作内容；

第二章为基础知识和相关技术，此处将详细介绍差分隐私的数学定义，种类，性质，相关技术，为下一章介绍所设计算法的主要内容奠定基础；

第三章是本文的重点，这章首先得对问题进行详细的数学上的描述，然后提出一种使用抽样思想的两阶段的频繁项挖掘算法，并将从理论上对其进行分析；

第四章是测试部分，通过使用合成数据集和公有数据集，对算法的准确度，性能进行了量化分析，并与已有算法进行对比；

第五章是全文总结，总结全文的设计思路和实现过程，对毕业设计过程遇到的问题进行归纳，对未解决的问题进行了初步探讨并提出了部分未来工作可以入手的方向。

第二章　基础知识及相关技术

2.1　差分隐私概念及性质

经过十数年的发展，差分隐私模型发展出了两种主要的模型，中心化差分隐私和本地化差分隐私。两者的主要区别在于隐私化处理场所的不同。在中心化差分隐私模型中，用户发送未经过隐私化处理的源数据给第三方的数据收集者。隐私化处理发生在数据收集者的服务器上，由数据收集者在其服务器的数据集上加入噪声扰动，并把由扰动后数据得到的统计报告发布出来。而在本地化差分隐私模型中，隐私化处理发生在用户的本地设备中，用户把个人数据加噪处理后再发给数据收集者。数据收集者得到是不是准确的用户数据。后者假设数据收集者是不可信的，存在着泄露用户隐私的风险。

2.1.1 中心化差分隐私

定义 2.1 相邻数据集：假设数据集 D 和数据集 D′ 具有相同的属性结构，如果 D 和 D′ 所拥有的记录数量只差为 1，则称 D 和 D′ 为相邻数据集。例如，表 2-1 和表 2-2 具有相同的属性结构：Name，Age，Major，ZipCode 和 Disease，表 2-1 有 3 条记录，表 2-2 有 4 条记录，两者相差 1，则称表 2-1 和表 2-2 为相邻数据集。

表2-1

Name	Age	Major	ZipCode	Disease
Bob	23	A	11000	Pneumonia
Ken	27	B	13000	Dyspepsia
Linda	65	D	25000	Gastritis

表2-2

Name	Age	Major	ZipCode	Disease
Bob	23	A	11000	Pneumonia
Ken	27	B	13000	Dyspepsia
Linda	65	D	25000	Gastritis
Alice	65	D	25000	Flu

定义 2.2 中心化差分隐私：假设有一个随机算法 A，A 的所有输出构成集合 O，A 的所有取值构成集合 I，如果对于任意两个相邻数据集 D∈I 和 D′ ∈I 和任意一个输出 S∈O，存在

$$\Pr(\mathrm{A}(\mathrm{D}) \in \mathrm{S}) < e^{\varepsilon} \times \Pr(A(D') \in S) \qquad \text{式(2-1)}$$

则称算法 A 满足中心化 ε -差分隐私，其中 ε 称为隐私参数或隐私预算。

中心化差分隐私的意思是两个相邻数据集通过随机算法 A 后获得同样输出的概率之比被控制在e^{ε}以内，其意义是数据集新增或减少一条记录，对 A 的输出影响微乎其

微。换言之，攻击者无法通过观察数据集新增记录前后 A 的输出的变化来推测出与个体用户相对的隐私信息。隐私参数 ε 表征了算法 A 隐私保护的能力大小。ε 越大，隐私保护能力越大，ε 越小，隐私保护能力越小。极端情况下，当 ε 取 0 时，任意两个相邻数据集通过 A 算法后获得相同输出的概率一致，A 算法相当于随机从输出集合 O 中选出一个 S。此时，经过 A 处理后数据的效用性近乎为 0，因为从输出根本没有任何把握判断出输入，也就无法对源数据进行频率，中位数等统计估计。这也与我们的初衷的相悖。我们的目的是，在保证一定隐私保护程度的条件下，尽可能提高数据的效用性，挖掘出有效信息。在中心化差分隐私模型中，ε 一般取 0.2，0.5 或者 ln2，ln3 等。

关于隐私参数 ε 有以下两个重要的性质:

性质 2.1 序列组合性：假设有随机算法$A_1,A_2,A_3,\dots,A_n$，其隐私参数分别为$\varepsilon_1,\varepsilon_2,\varepsilon_3,\dots,\varepsilon_n$，当这些算法作用于同一数据集时，这些算法构成的组合算法 A$(A_1(I),A_2(I),A_3(I),\dots A_n(I))$提供$(\sum_{i=1}^{n}\varepsilon_i)-$差分隐私保护。

该性质表明，当多个随机算法作用于同一个数据集时，它们共同提供的隐私保护水平为各算法提供的隐私参数的总和。特殊地，当$A_1=A_2=A_3=\cdots=A_n$时，$\varepsilon=\varepsilon_i\times n$，即当同一个算法多次作用于一个数据集时，总的隐私参数成倍增大，隐私保护水平成倍下降。

性质 2.2 并行组合性：假设有随机算法$A_1,A_2,A_3,\dots,A_n$，其隐私参数分别为$\varepsilon_1,\varepsilon_2,\varepsilon_3,\dots,\varepsilon_n$，当这些算法作用于不相交的数据集$D_1,D_2,D_3,\dots,D_n$，时，这些算法构成的组合算法 A$(A_1(I),A_2(I),A_3(I),\dots A_n(I)$对这些数据集提供$(\max(\varepsilon_i))-$差分隐私保护。

此性质表明，当多个随机算法作用的数据集两两之间互不相交时，它们对所有目标数据集构成的总数据提供的隐私保护水平取决于其中最大的ε_i，即取决于保护水平最差者。特殊地，当$A_1=A_2=A_3=\cdots=A_n$时，$\varepsilon=\varepsilon_i$,即当同一个算法多次作用于不相交的数据集时，隐私保护水平不变。

2.1.2 本地化差分隐私

定义 2.3 本地化差分隐私：假设有一个随机算法 A，A 的所有输出构成集合 O，A 的所有取值构成集合 I，如果对于任意两条记录 R∈I 和 R′ ∈I 和任意一个输出 S∈O，存在

$$\Pr(\mathrm{A}(\mathrm{R})\in \mathrm{S})<e^{\varepsilon}\times\Pr(A(R')\in S) \qquad \text{式(2-2)}$$

则称算法 A 满足 ε -本地化差分隐私，其中 ε 称为隐私参数或隐私预算

不同于中心化差分隐私通过相邻数据集来定义，本地化差分模型因把隐私化处理场所转移到用户的本地设备中，通过随机算法的是单条用户记录而非整个数据集，因而在本地化差分隐私中任意两条记录满足上述不等式。

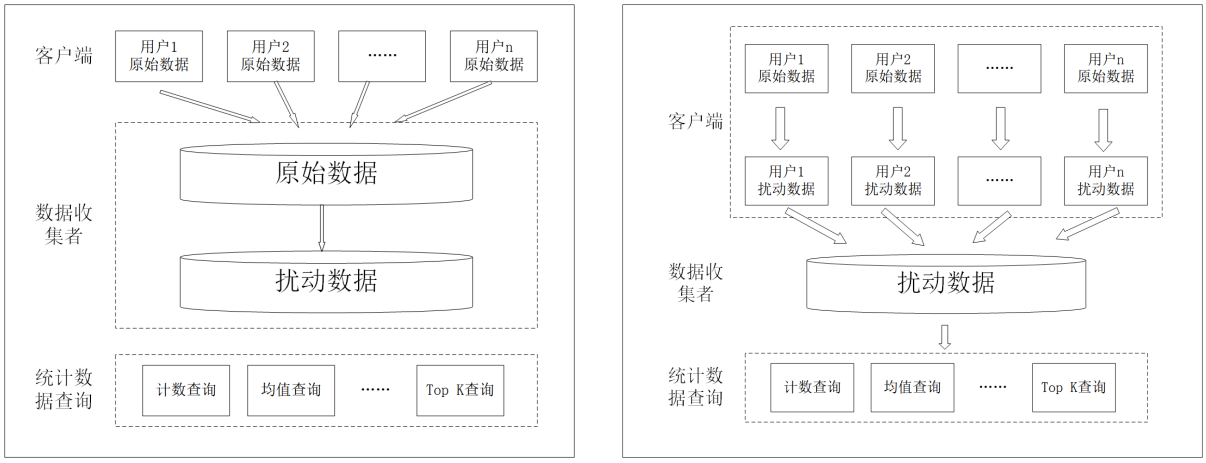
中心化差分隐私模型满足的关于隐私参数的序列组合性和并行组合性，在本地化差分隐私中同样满足。

2.1.3 中心化差分隐私和本地化差分隐私的区别

本地化差分隐私由中心化差分隐私发展而来，其继承了中心化差分隐私的许多性质，但在隐私化处理场所等方面区别于前者，这使得本地化差分隐私具有更强的隐私保护能力，能够避免不可信的第三方数据收集者造成的隐私泄露。下面将归纳两者的区别并对比其优劣。

（1） 隐私化处理的场所不同

中心化差分隐私与本地化差分隐私的数据处理结构如图 2-1。



(a) 中心化差分隐私处理框架　　(b) 本地化差分隐私处理框架

图 2-1 中心化和本地化的数据处理框架[11]

由上图可知，在中心化差分隐私模型中，用户数据在数据收集者的数据库中进行隐私化处理，数据收集者从客户端处收集到用户的准确的原始数据。而在本地化差分隐私模型中，用户数据在本地客户端处进行隐私化处理，数据收集者收集到的是经过加噪处理后非完全准确的扰动数据。

（2） 对隐私保护能力不同

一般而言，本地化差分隐私模型的隐私保护能力更强。中心化差分隐私模型相比本地化差分隐私模型，有两个可能导致隐私泄露的缺陷，如图 2-2 所示：首先，中心化差分隐私模型假设第三方数据收集者是可信的，不会主动泄露用户数据而且有能力保护好数据不被窃取，而事实告诉我们这个假设往往并不成立；其次，用户数据在从客户端到数据收集者的传输过程中可能受到攻击，由于这个过程中传输的原始数据，一旦收到攻击，用户隐私将会完全被暴露。

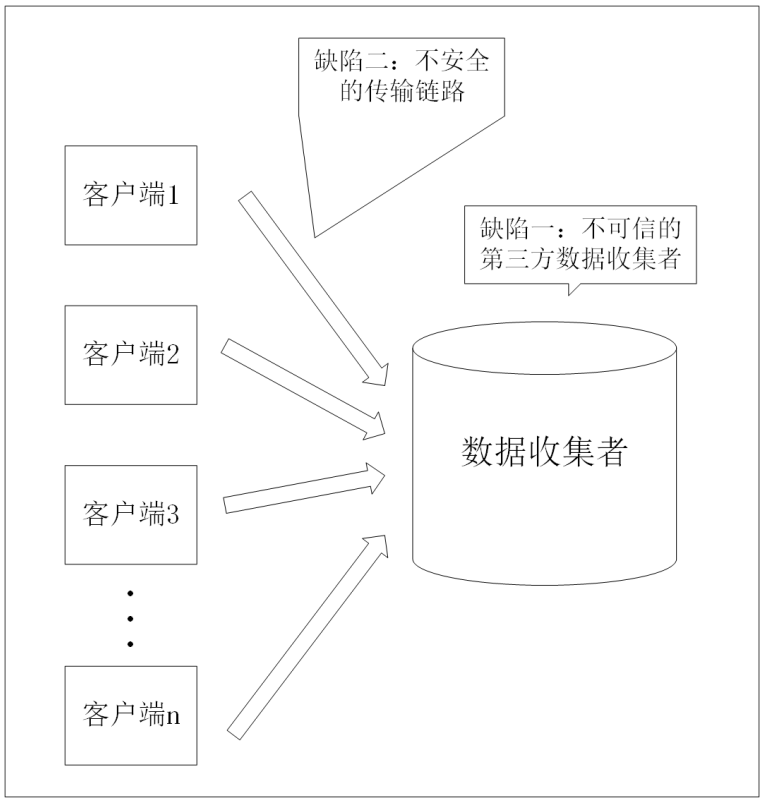


图2-2 中心化差分隐私的缺陷

本地化差分隐私提前在本地设备上对数据进行隐私化处理，传输的就是满足ε-差分隐私保护水平的扰动数据，能有效解决以上两个问题。因而，本地化差分隐私模型对隐私的保护程度更高，这也是本文选择本地化差分隐私模型的原因。

（3） 扰动机制不同

在中心化差分隐私模型中，常用的扰动机制为拉普拉斯机制[12]和指数机制[13]。这两者分别面向连续性数据的查询和离散型数据的查询。这两种扰动机制均涉及到全局敏感度。全局敏感度指的是两个相邻数据集通过同一个随机算法后的两个输出值的最大 1-阶范数距离。因为本地化差分隐私模型中各个用户各自扰动自己的数据，任意两个用户并不知晓对方的数据，故不存在相邻数据集也不存在全局敏感度，所以拉普拉斯机制和指数机制并不适用于本地化差分隐私。目前，本地化差分隐私最常用的扰动机制是随机响应技术。该技术将会在 2.2 节中详细阐述。

（4） 适用场景不同

本地化差分隐私适用于个性化隐私保护场景而中心化差分隐私不适用。本地化差分隐私中每个用户对个人数据的隐私处理过程有充分的把握，可以自定义隐私参数来控制隐私保护水平；而中心化差分隐私中数据交由第三方数据收集者进行扰动，隐私参数由数据收集者设定。

本地化差分隐私模型适用于大数据场景而中心化差分隐私对数据量没有特别

要求。本地化差分隐私模型对每一条记录进行正向和负向的扰动，要得到有效的统计信息必须通过累积大量的扰动结果以抵消其中的正向负向噪音。这只有当数据量足够大时才能有较好的结果。与此不同，中心化差分隐私模型通过定义全局敏感度来添加噪音，再以统计的手段来限制隐私泄露的边界。该手法对数据集的数据量不作特殊要求。

2.2 满足本地化差分隐私的扰动机制及其解码方案

2.2.1 随机响应（Random Response）

一个完整的差分隐私解决方案包括一个用户侧的对原始数据的扰动算法和一个数据收集者侧的对扰动数据的解码算法。前者用以加入噪声以提供隐私参数为 ε 的隐私保护，后者是为了从扰动数据中提取有效的统计信息并不侵犯用户隐私。目前，在本地化差分隐私保护技术中，随机响应（Random Response）[14]是各主流解决方案的基石。随机响应技术由 Warner 于 1965 年提出。其原理相当于让每个用户投掷一个有偏的硬币，当正面向上时用户回答真实答案，反之用户回答与真实值相反的答案。因为硬币的投掷是随机的，攻击者没法仅从用户的回答准确地获取具体用户的真实答案。下面将从一个具体的场景来详细阐述整个解决方案。

假设我们要统计 n 个用户中乙肝患者的比例 π，我们向每个用户问一个问题，“您是否为乙肝患者”。显然，这是一个敏感的问题，每个用户的真实答案是我们要保护的隐私。我们的目标是，保护每个用户的真实答案不被窃取的同时，获得一个尽可能准确的 π。为了达到这个目标，每个用户在回答问题之前，先投掷一个有偏的硬币，当下落硬币正面朝上时，用户回答真实的答案，这个概率为 p，当下落硬币反面朝上时，用户回答与真实值相反的答案，概率为 1-p。其中，p>0.5，第 x 个用户的回答为x_i。

然后我们对每个用户的回答进行统计。假设回答“是”的用户人数为n_1，则回答“否”的用户人数为$n-n_1$。由此可得回答为“是”和“否”的用户比例为：

$$\Pr(x_i = \text{“是”}) = \pi p + (1-\pi)(1-p) \quad \text{式(2-3)}$$

$$\Pr(x_i = \text{“否”}) = (1-\pi)p + \pi(1-p) \quad \text{式(2-4)}$$

显然， 这两个比例并不等于真实结果的比例。要据此对真实值进行估计，我们还需要进行矫正。构建似然函数如下：

$$L = [\pi p + (1-\pi)(1-p)]^{n_1}[(1-\pi)p + \pi(1-p)]^{n-n_1} \quad \text{式(2-5)}$$

由式（2-5）求得 π 的极大似然估计：

$$\hat{\pi} = \frac{p-1}{2p-1}n + \frac{n_1}{(2p-1)n} \quad \text{式(2-6)}$$

求$\hat{\pi}$的数学期望得：

$$E(\hat{\pi}) = \frac{1}{2(p-1)}\left[p - 1 + \frac{1}{n}\sum_{i=1}^{n} x_i\right] = \frac{1}{2(p-1)}[p - 1 + \pi p + (1-\pi)(1-p)] = \pi \quad \text{式(2-7)}$$

由此可知$\hat{\pi}$即为π的无偏估计。同时，根据本地化差分隐私的定义有$\frac{p}{1-p} = e^{\varepsilon}$，即p

需设置为p＝ln$\frac{e^{\varepsilon}}{e^{\varepsilon}+1}$。

2.2.2 S-Hist 算法（Succinct Histogram）

2.2.1 节的随机响应技术仅可用于取值范围为 0，1 的二值型数据。对于取值超过两种的数据，必须对变量进行一定预处理或者对随机响应技术进行修改才能应用随机响应技术。前者的思路是对变量的不同取值进行编码得到一个二进制字符串，然后再对字符串的每一位进行随机响应。应用此思路的算法有 S-Hist[15]和 RAPPOR[16] 。本节将详细阐述 S-Hist 算法（Succinct Histogram）的原理。

S-Hist 算法的应用场景如下：假设有 n 位用户，每位用户仅有一个项（Item），用户u_i的项设为s_i，所有项的取值集合为 D，大小为 d。SH 算法能准确估计 n 位用户中拥有 D 中各个项的比例。SH 算法由两部分组成，数据收集者的服务器端部分和用户的客户端部分，分别如图 2-3 和图 2-4 所示。

（1） 服务器端部分：输入为用户的项，隐私参数 ε 和对估计准确性有重要影响的置信参数 β，输出为每个项的估计频率。S-Hist 算法的主要原理是生成一个尺寸为 m×d 的随机矩阵，如图 2-3 第一步至第三步所示，矩阵中的每个元素从集合$\left\{\frac{1}{\sqrt{m}},-\frac{1}{\sqrt{m}}\right\}$中随机选取，矩阵的每一列代表不同项的码字。第四行初始化了两个向量 **z** 和 **f**，**z** 的长度为 m，用来存取中间信息，f 的长度为 d，用以存储每个项的最终估计频率。接着，服务器从矩阵中随机选取第 j 行发送给用户客户端，客户端接收后找出与所持有的项对应的其中一位码字并应用随机响应进行处理，如第六到第八步所示，服务器收集到用户的返回值并累加至向量 **z** 的第 j 位，最后通过公式$<\phi e_{i_k}, \mathrm{z}>/\mathrm{n}$，求出每个项$i_k$的估计频率。其中，$\phi e_{i_k}$求出项$i_k$的码字，$<\phi e_{i_k}, \mathrm{z}>$求出项$i_k$的估计数量。需要说明的是，随机矩阵ϕ的生成有多种方法。此处假设$\mathrm{d} \gg \mathrm{m}$,故需对生成矩阵进行降维，算法 1 采用 Johnson-Lindenstrauss 定理进行降维，使 m 满足以下 Johnson-Lindenstrauss 定理。

Johnson-Lindenstrauss 定理：给定$0<\delta<1$和 t 个长度为 d 的向量点，所有向量点组成集合$\mathcal{V}$，位于 d 维实数空间$\mathcal{R}^d$中。若$\mathrm{m}=O(\frac{\ln t}{\delta^2})$,则对所有$\mathbf{u}$，$\mathbf{v}\in\mathcal{V}$均存在从 d 维实数空间到 m 维实数空间的映射ϕ，且满足：

$$(1-\delta)||\mathbf{u}-\mathbf{v}||_2^2 \leq \left||\phi\mathbf{u}-\phi\mathbf{v}\right|_2^2 \leq (1+\delta)||\mathbf{u}-\mathbf{v}||_2^2 \qquad \text{式(2-8)}$$

定理中的 δ 对应图 2-3 中的 β，Johnson-Lindenstrauss 定理表明，当$\mathrm{m}=O(\frac{\ln t}{\delta^2})$时，至少有 1-δ 的概率使得任$\mathbf{u}$，$\mathbf{v}\in\mathcal{V}$完成从 d 维实数空间到 m 维实数空间的映射。δ 越小，准确率越高。一般 δ 取 0.01 左右。

算法 1 Succint Histogram(数据收集者服务器端)

输入: 用户项$\{s_i \subseteq D : 1 \le i \le n\}$
输入: 隐私参数：ϵ
输入: 置信参数：$\beta, 0 < \beta < 1$
输出: 每个项出现的频率$\{i_j \in D : 1 \le j \le d\}$

1: 计算$\delta = \sqrt{\frac{\ln(2d/\beta)}{n}}$
2: 计算$m = \frac{\ln(d+1)\ln(2/\beta)}{\delta^2}$
3: 生成一个随机矩阵$\phi \in \{-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\}^{m \times d}$
4: 初始化向量**z**和向量**f**
5: **for** 每个用户u_i **do**
6: 随机生成$j \in \{1, ..., m\}$
7: 发送矩阵ϕ的第j行ϕ_j给用户u_i
8: u_i返回$z_i = LR(\phi_j, s_i, \epsilon)$给服务器
9: 服务器把z_i加到向量**z**的第j位
10: **end for**
11: **for** 每个项$i_k \in D$ **do**
12: e_{i_k}为长度为d的单位向量，第i_k位1，其余全为0
13: 把向量**f**的第k位设为$\langle \phi e_{i_k}, \mathbf{z} \rangle / n$
14: **end for**
15: **return** 向量**f**,其中的每一位代表了每个项的估计概率

图2-3 算法1 Succinct Histogram[15](数据收集者服务器端

(2)客户端部分：输入为 d-bit 字符串（即图 2-3 中第 7 步得到的ϕ_j），隐私参数和用户的项 ID，输出为扰动后的 1 比特数据z_i。本地随机器（Local Randomizer，LR）是 S-Hist 算法在用户客户端上的实现部分，它保证了整个算法能对用户数据提供充分的隐私保护能力。该部分的第一、第二步实现了取出与项 ID 对应码字的第 m 位，第三步采用了随机响应技术使其满足本地化差分隐私。下面证明对任意$\mathbf{x} \in \left\{\frac{-1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\right\}^d$，LR 满足 ε-本地化差分隐私。

证：根据本地化差分隐私的定义，要证 LR 满足 ε-本地化差分隐私，需证对任意$i_i, i_j \in D$，满足

$$\frac{\Pr[\mathrm{LR}(\mathbf{x}, \mathrm{i_i}, \varepsilon) = \mathrm{z_i}]}{\Pr[\mathrm{LR}(\mathbf{x}, \mathrm{i_j}, \varepsilon) = \mathrm{z_i}]} \le e^{\varepsilon} \qquad \text{式}(2-9)$$

可知x_{i_i}，$x_{i_j} \in \left\{\frac{-1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\right\}$，$z_i \in \{-c_\varepsilon\sqrt{m},\ c_\varepsilon\sqrt{m}\}$。当$x_{i_i} = x_{i_j}$,有

$$\frac{\Pr[LR(\boldsymbol{x}, i_i, \varepsilon) = z_i]}{\Pr[LR(\boldsymbol{x}, i_j, \varepsilon) = z_i]} = 1 \qquad \text{式}(2-10)$$

当$x_{i_i} \neq x_{i_j}$，存在 4 种情况：

情况 1：$z_i = -c_\varepsilon\sqrt{m}$，$x_{i_i} = -\frac{1}{\sqrt{m}}$，$x_{i_j} = \frac{1}{\sqrt{m}}$，有

$$\frac{\Pr[LR(\boldsymbol{x}, i_i, \varepsilon) = z_i]}{\Pr[LR(\boldsymbol{x}, i_j, \varepsilon) = z_i]} = \frac{\frac{e^\varepsilon}{e^\varepsilon + 1}}{\frac{1}{e^\varepsilon + 1}} = e^{\varepsilon} \qquad \text{式}(2-11)$$

情况 2：$z_i = -c_\varepsilon\sqrt{m}$，$x_{i_i} = \frac{1}{\sqrt{m}}$，$x_{i_j} = -\frac{1}{\sqrt{m}}$，有

$$\frac{\Pr[LR(\boldsymbol{x}, i_i, \varepsilon) = z_i]}{\Pr[LR(\boldsymbol{x}, i_j, \varepsilon) = z_i]} = \frac{\frac{1}{e^{\varepsilon}+1}}{\frac{e^{\varepsilon}}{e^{\varepsilon}+1}} = e^{-\varepsilon} \qquad 式(2-12)$$

情况 3：$z_i = c_\varepsilon\sqrt{m}$，$x_{i_i} = -\frac{1}{\sqrt{m}}$，$x_{i_j} = \frac{1}{\sqrt{m}}$，有

$$\frac{\Pr[LR(\boldsymbol{x}, i_i, \varepsilon) = z_i]}{\Pr[LR(\boldsymbol{x}, i_j, \varepsilon) = z_i]} = \frac{\frac{1}{e^{\varepsilon}+1}}{\frac{e^{\varepsilon}}{e^{\varepsilon}+1}} = e^{-\varepsilon} \qquad 式(2-13)$$

情况 4：$z_i = c_\varepsilon\sqrt{m}$，$x_{i_i} = \frac{1}{\sqrt{m}}$，$x_{i_j} = -\frac{1}{\sqrt{m}}$，有

$$\frac{\Pr[LR(\boldsymbol{x}, i_i, \varepsilon) = z_i]}{\Pr[LR(\boldsymbol{x}, i_j, \varepsilon) = z_i]} = \frac{\frac{e^{\varepsilon}}{e^{\varepsilon}+1}}{\frac{1}{e^{\varepsilon}+1}} = e^{\varepsilon} \qquad 式(2-14)$$

综上所述，对对任意$i_i, i_j \in D$，均存在

$$\frac{\Pr[\mathrm{LR}(\mathbf{x}, \mathrm{i_i}, \varepsilon) = \mathrm{z_i}]}{\Pr[\mathrm{LR}(\mathbf{x}, \mathrm{i_j}, \varepsilon) = \mathrm{z_i}]} \leq e^{\varepsilon} \qquad 式(2-15)$$

所以，LR 满足 ε-本地化差分隐私。

算法 1 Local Randomizer LR(用户客户端部分)

输入： d-bit字符串$\mathbf{x} \in \{\frac{-1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\}$

输入： 隐私参数：ϵ

输入： 用户u_i的项ID：i_i

输出： 干扰比特z_i

1: 生成一个标准基向量$e_{i_i} \in \{0,1\}$

2: 计算$x_{i_i} = \mathbf{x}\top e_{i_i}$

3: 计算$c_\epsilon = \frac{e_\epsilon+1}{e_\epsilon-1}$

4: 对x_{i_i}使用随机响应技术

$$概率 p = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+1}, when & z_i = c_\epsilon m x_{i_i} \\ \frac{1}{e^{\epsilon}+1}, when & z_i = -c_\epsilon m x_{i_i} \end{cases}$$

5: **return** z_i

图2-4 算法1 Succinct Histogram–Local Randomizer[15](用户客户端部分)

上述两部分的关系如下图 2-5。数据收集者把码字矩阵的其中一行发送到用户的客户端，客户端找出与所拥有项对应的字节运用随机响应技术进行扰动，再把扰动比特发送给数据收集者。数据收集者根据统计到的信息估计各项所占比例。在这个体系中，隐私保护能力由客户端提供，无论是数据收集者还是攻击者都无法从获得的扰动比特准确推断中个体用户所拥有的项。

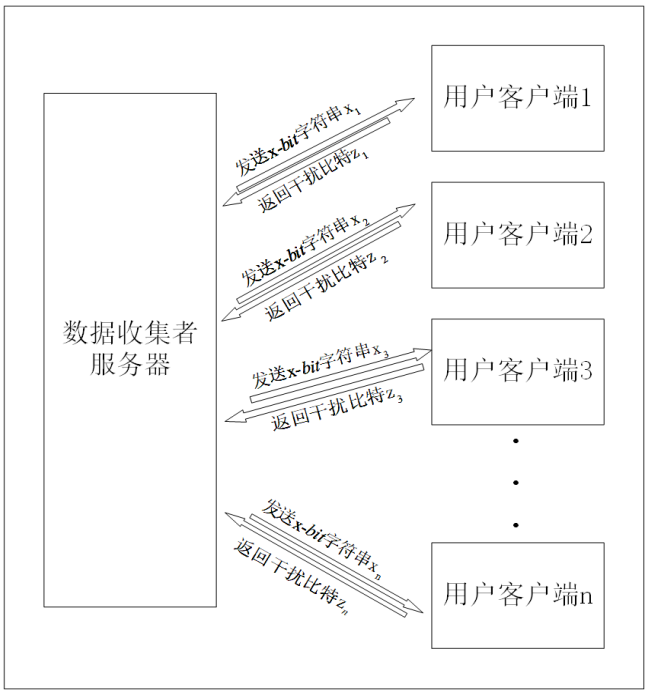


图2-5 S-Hist 算法中两者的关系

2.2.3 RAPPOR 算法（Randomized Aggregatable Privacy-Preserving Ordinal Response）

RAPPOR[14]由谷歌研究员 Erlingsson Ú 等人于 2014 年提出，最初用于调查 Chrome 浏览器用户最喜爱的浏览器主页及其所占比例。RAPPOR 的典型应用场景是分类数据的频率估计。与 S-Hist 不同，RAPPOR 用一个长度为 d 的二进制码字代表项i_i，其中第 i 比特为 1，其余为 0。对码字的每一个比特，用户用概率为 p 的随机响应模型进行扰动，然后把长度为 d 的码字发送给数据收集者，如图 2-6 所示。

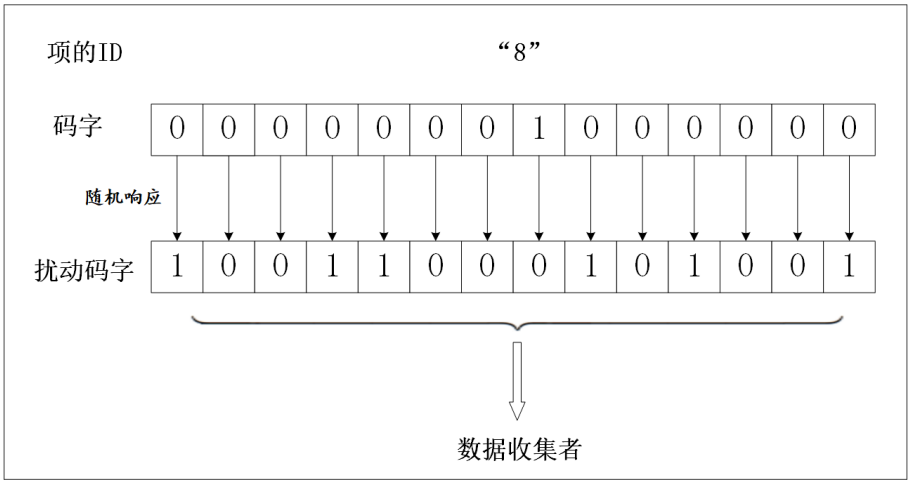


图2-6 RAPPOR[14]：对项ID的编码与扰动

数据收集者根据收集到的扰动码字用随机响应模型进行解码。参考文献[15]证明了为了保证 RAPPOR 满足 ε-本地化差分隐私，须把 p 设置为

$$\mathrm{p} = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}} + 1} \qquad 式(2-16)$$

2.2.4 S-Hist 和 RAPPOR 的比较

本节主要是对 S-Hist 和 RAPPOR 在通信开销、计算开销、估计精度上进行比较。

（1） 通信开销

S-Hist 中用户发送给数据收集者的是单个经过扰动的比特，通信开销是 O(1)；RAPPOR 中用户发送给数据收集者的是经过扰动的长度为 d 的数组，d 为项的取值集合大小，通信开销为 O(d)。当项的取值集合较大时，如取值集合是某个地区的地点时，RAPPOR 的通信开销极大，甚至于不可用；而 S-Hist 将始终保持极低的通信开销。

（2） 计算开销

S-Hist 通过计算随机矩阵和向量 $\mathbf{z}$ 的内积$< \phi e_{i_k}, \mathrm{z} >$求得项的估计频数，计算开销与矩阵大小即 d 有关。当 d>>n 时，可以通过 Johnson-Lindenstrauss 定理进行降维降低计算开销。RAPPOR 估计项的频数需要构造一个$\mathrm{d} \times \mathrm{n}$的矩阵，然后通过拉索回归和最小二乘法求出估计频数，计算开销远大于 S-Hist。

（3） 估计精度

S-Hist 和 RAPPOR 的渐进误差边界是$O(\frac{d}{\varepsilon\sqrt{n}})$。S-Hist 因引入了随机矩阵，在项的取值比较少时误差稍大于 RAPPOR，但项的取值较大时两者的误差非常接近。

本文旨在解决 d 和 n 都较大时的集值数据的频繁项挖掘问题，出于节约通信开销和计算开销的考虑，本文选择 S-Hist 作为本文提出算法的基本技术。

2.3 本章小结

本章介绍了本文工作所需了解的基础概念和本文将会用到的基础知识。首先我们在第一部分对差分隐私的基本概念和性质进行了介绍，通过对中心化差分隐私和本地化差分隐私进行详细对比分析，我们得出了本地化差分隐私能提供优于中心化差分隐私的隐私保护能力的结论。这也突出了本文题目的重要研究价值。在本章的第二部分，本文重点介绍了当前本地化差分隐私模型中最常用的扰动机制——随机响应技术，以及 S-Hist 和 RAPPOR 等两种由随机响应技术发展而来的扰动技术。通过对比 S-Hist 和 RAPPOR，我们给出了本文采用 S-Hist 作为基础技术的原因。

第三章　满足本地化差分隐私的频繁项挖掘算法设计与实现

3.1　问题描述

本文聚焦在集值数据的频繁项挖掘，即每个用户拥有一个项集而非一个项。项集中每个项互不相同，项集的大小可以大于 1。我们的目标是找出在用户中出现频率为前 k 名的项 ID 及其频率，并使整个过程满足隐私参数为 ε 的本地化差分隐私要求。

关于问题详细的数学描述如下：假设有 n 个用户，第 i 个用户u_i拥有项集S_i，大小为$|S_i|$，每个用户的项集内部的项各不相同，所有用户拥有的项的取值集合为 D，大小为$|D| = d$，要求的是出现频率为前 k 位的项及其频率，其中，项 j 的频率表示为f_j，频率定义如下：

$$f_j = \frac{\left|\{u_i | i_j \in s_i, 1 \le i \le n\}\right|}{n} \qquad 式(3-1)$$

即拥有项 j 的用户比例。一般地，我们假设k ≪ d，d ≪ n。

另外，本文假设我们已知参数ℓ，ℓ为用户项集大小的第 90 百分位数所对应的长度。每个用户在发送扰动数据之前，先对自身拥有的项集进行预处理使其大小变为ℓ。预处理的具体办法为：对于项集大小大于ℓ的用户项集，用户从原项集中随机选取ℓ个项组成新的项集；对于项集大小小于ℓ的用户项集，用户给该项集增加一定数量的冗余项使其大小增至ℓ，冗余项是“无用”的项，数据收集者并不统计冗余项的频率。这样处理的原因是：假若每个用户发送给数据收集者的项集大小各异，一方面会对数据收集者挖掘频繁项带来巨大困难；另一方面可能会带来潜在的隐私泄露风险。因为每个用户拥有的项集大小这个数据本身就隐含了用户的隐私信息，所以，用户对自己的项集进行预处理是必不可少的。ℓ值的选取应该适中，若ℓ太大，则引入过多噪音，会降低前 k 项频率估计的准确性，若ℓ太小，则大量用户项集被截断，有效信息大量流失，同样会降低估计频率的准确性。本文取ℓ为用户项集大小的第 90 百分位数所对应的长度。

3.2　算法设计

3.2.1 总体框架

过去关于本地化差分隐私下的频繁项挖掘的研究大多仅针对于每个用户仅有一个项的情况，对于每个用户拥有一个项集的情况研究得很少。Zhan Qin 等学者提出的 LDPMine 是目前关于该问题的主要研究成果。该办法把隐私参数进行切分，一部分用于寻找频繁项的候选集，另一部分用于从候选集中计算出较为准确的估计频率。该方法能实现在满足本地化差分隐私的情况下较为准确地挖掘频繁项。然而，由于该办法需要切分隐私预算，LDPMiner 的准确率仍有待提高。特别是在隐私参数较小的情况下，LDPMiner 的准确率会极大降低。本文创新性地提出了一种基于分组思想的两阶段解决

方案，名为 GFIM（Group-based Frequent Items Mining）。GFIM 不划分隐私参数而划分用户，它利用隐私参数的并行组合性，实现了在隐私参数较小的情况下仍能较准确地挖掘频繁项，在现实应用场景中更有优势。本节将首先介绍该解决方案的总体框架。

GFIM 把用户随机划分为不相交且大小相等的两部分，整个运行过程也分为两个阶段。在第一阶段，GFIM 根据第一部分用户提交的信息挖掘出一个大小为$k_{max} = O(k)$的关于频繁项的候选集 C。在第二阶段，收据收集者先把候选集发送给第二部分每个用户，接着用户把在自身拥有的项集中却不在候选集内的项置为冗余项，把自身的项集缩小为 O(k)后发送给数据收集者。这样做是基于频繁项主要出自于候选集的假设，目的是最大化减少候选集外的项对第二阶段估计结果的影响，提高候选集内各项估计频率的准确性。当然，这样做也会存在着一定的风险，假如第一阶段得到的候选集不完全精确，第二阶段如果只计算候选集内的项的估计频率并以此为最终结果的话，误差将会很大。为了解决这个问题，最终的估计结果 top-k 频繁项及其对应频率将会利用上两个阶段的结果，具体的处理方法将会在 3.2.3 节详述。GFIM 的总体框架如图 3-1 所示。

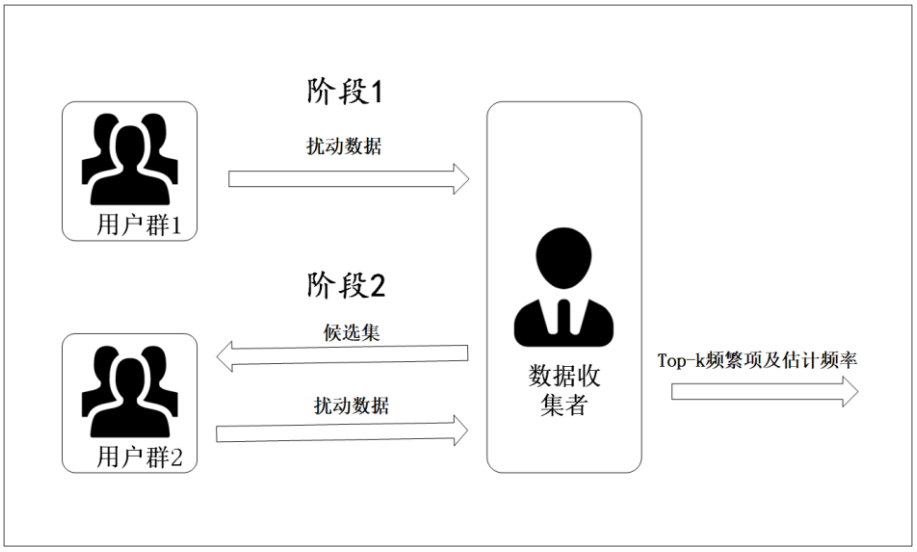


图3-1 GFIM 整体框架

如果我们细看的这两个阶段的话，容易发现每个阶段解决的问题与整个方案要解决的原问题几乎完全一致，都是每个用户拥有一个项集，要从一定数量的用户中挖掘出其中出现最为频繁的 k 个项及其对应频率，每个阶段都可以看为一次完整的集值数据 top-k 频繁项挖掘过程。不同的是，与原问题相比的话，两个阶段处理的用户数量都减半，并且，第一阶段求解的频繁项取值范围为 D，大小为 d，第二阶段求解的频繁项取值范围为 C，大小为k_{max}。

GFIM 满足 ε-本地化差分隐私。这是因为了 GFIM 利用了隐私参数的并行组合性，多个随机算法作用于不相交的数据时它们的隐私保护水平取决于其中隐私参数最大者。只要每个阶段都分别满足 ε-本地化差分隐私，则由它们组成的 GFIM 满足 ε-本地化差分隐私。以下两节将会证明这两个阶段各自满足 ε-本地化差分隐私。

3.2.2 阶段一设计

阶段一的重点是找出频繁项的候选集，本质上也是一个找 top-k 项的问题。已有的研究成果中，RAPPOR 和 S-Hist 是两个最为经典的频繁项挖掘算法。但是它们不能直接运用于阶段一的场景，因为传统的 RAPPOR 和 S-Hist 均要求每个用户的输入为一个项，而在本文的场景中用户输入的是一个经过处理过后的大小为ℓ的项集。一个直观的解决方案是调用ℓ次 RAPPOR 或 S-Hist，再把每次调用得到的估计频率累加得到最终的估计频率。这个想法是简单，直接，可行的，但也是低效的。对这种想法的一种改进方案是从每个用户的项集S_i中随机选取一个项输入到随机算法，然后采用已有的算法。已有的工作[10]证明了该想法的合理性，证明了其优于前面所说的调用ℓ次 RAPPOR 或 S-Hist。一个需要注意的点是，直接的随机抽取会导致有偏频率估计，因为每个用户只向数据收集者发送一个随机的项而不是所有ℓ个项。为了达到无偏估计，需要使估计频率乘以ℓ，这样计算得到的最终频率将是真实频率的无偏估计。根据第三章第四节对 RAPPOR 和 S-Hist 算法的对比分析，出于节约通信带宽和提高计算效率的考虑，这里将对以 S-Hist 算法为基础对其进行改造分析，改造后的算法称为抽样 S-Hist 算法，抽样 S-Hist 算法将作为阶段一和阶段二的基本算法。抽样 S-Hist 算法的服务器端部分与传统 S-Hist 算法大体一致，但本文采用了另外一种随机矩阵的生成方法。本文假设 d ≪ n，参考文献[17]证明了在d ≪ n的情况下采用正交矩阵取代完全随机矩阵能够提高 S-Hist 的准确性。正交矩阵的生成见图 3-2[17]。

算法 2 抽样S-Hist 正交矩阵的生成

输入： 项的取值集合大小：d

输出： 正交矩阵ϕ

计算$m = 2^{\lceil \log_2 d \rceil}$
$S = \{[1,-1],[1,1]\}$
while $|S| < m$ **do**
    $S' = \emptyset$
    for $v \in S$ **do**
        $S' \leftarrow S' \bigcup \{v||v, v||(-v)\}$
    end for
    $S \leftarrow S'$
end while
N=S[1:m]
$\phi = N^{\top}$
return ϕ

图3-2 抽样S-Hist 正交矩阵的生成

客户端部分改动较大，具体实现的伪代码见图 3-3。抽样 S-Hist 与传统的 S-Hist 的不同主要体现在第一和第三第四步。当随机选取的项为冗余项时，LR 从$\{c_\varepsilon\sqrt{m}, -c_\varepsilon\sqrt{m}\}$中随机选取一个发送给数据收集者。这样做的目的是期望在用户基数很大的情况下，由冗余项引入的噪声能够达到正负抵消。下面将证明抽样 S-Hist 算法满足 ε-本地差分隐私。

算法 2 抽样S-Hist LR(用户客户端部分)

输入: d-bit字符串$\mathbf{x}\in\{\frac{-1}{\sqrt{m}},\frac{1}{\sqrt{m}}\}$

输入: 隐私参数：ϵ

输入: 用户u_i的项集：S_i

输出: 干扰比特z_i

从项集S_i中随机选取一个项i_i

计算$c_\epsilon=\frac{e_\epsilon+1}{e_\epsilon-1}$

if $i_i=\perp$ **then**

　　随机选取$z_i\in\{c_\epsilon\sqrt{m},-c_\epsilon\sqrt{m}\}$

else

　　生成一个标准基向量$e_{i_i}\in\{0,1\}$

　　计算$x_{i_i}=\mathbf{x}\top e_{i_i}$

$$\text{概率}p=\begin{cases}\frac{e^\epsilon}{e^\epsilon+1},when & z_i=c_\epsilon m x_{i_i}\\ \frac{1}{e^\epsilon+1},when & z_i=-c_\epsilon m x_{i_i}\end{cases}$$

end if

return z_i

图3-3 抽样S-Hist算法LR

证：使S_1和S_2为任意两个用户项集，有$|S_1|=|S_1|=\ell$。使$\mathcal{A}$表示抽样 S-Hist 算法，z_i表示$\mathcal{A}$的任意可能的输出，$z_i\in\{c_\varepsilon\sqrt{m},-c_\varepsilon\sqrt{m}\}$。要证 S-Hist 满足 ε-本地差分隐私，即证

$$\frac{\Pr(\mathcal{A}(S_1)=z_i)}{\Pr(\mathcal{A}(S_2)=z_i)}\le e^\varepsilon \qquad \text{式}(3-2)$$

使$\Pr(i_i'|s_i)$表示输入为S_i时，算法 2 第一步的输出为i_i'的条件概率。可知对任意i_i'，$\Pr(i_i'|S_i)$=1/L。使 $\Pr(z_i|i_i'\wedge S_i)$表示输入为S_i，随机选取到i_i'，最终输出为z_i的条件概率，有

$$\frac{1}{e^\varepsilon+1}\le\Pr(z_i|i_i'\wedge S_i)\le\frac{e^\varepsilon}{e^\varepsilon+1} \qquad \text{式}(3-3)$$

进而有

$$\frac{\Pr(\mathcal{A}(S_1)=z_i)}{\Pr(\mathcal{A}(S_2)=z_i)}=\frac{\Pr(z_i|i_i\ \wedge S_1)\cdot\Pr(i_i\ |S_1)}{\Pr(z_i|i_i\ \wedge S_2)\cdot\Pr(i_i\ |S_2)}=\frac{\Pr(z_i|i_i\ \wedge S_1)}{\Pr(z_i|i_i\ \wedge S_2)}\le e^\varepsilon \qquad \text{式}(3-4)$$

所以抽样 S-Hist 算法满足 ε-本地差分隐私，即阶段一满足 ε-本地差分隐私。

阶段一产出一个大小为k_{max}的候选集。候选集大小的选取至关重要。候选集选的太小，真实的频繁项没有落在候选集内，估计的误差急剧增大；候选集过大，k_{max}有可能会大于ℓ，同样会降低估计频率的准确性。本文取候选集大小为 k 的一阶线性函数。

3.2.3 阶段二设计

阶段二面对的场景与阶段一有所不同，不同之处有二：一是项的取值集合从 D 变为 C，取值的范围大为缩小；二是每个用户的项集也发生了变化。用户在收到来自数据收集者的候选集后，修剪原项集得到新的大小为k_{max}的项集。

针对这两点不同，我们需要设计出相应的解决方案。阶段二中我们同样采用采样 S-

Hist 作为算法的基础。显然，当项的取值集合缩小时，S-Hist 中的随机矩阵会大为缩小。这一方面减少了噪音另一方面也降低了运算开销，最终导致了估计频率准确性的提高。针对第二点不同，我们在客户端上的 LR（Local Randomizer）做了以下调整：①LR 收到候选集 C 后，先求出用户原本的项集S_i与候选集 C 的交集T_i；②如果交集 N 的大小小于k_{max}，补充若干个冗余项使其大小变为k_{max}，得到新项集N_i；③从N_i中随机选取一个项i_i应用随机响应技术。（详细步骤见图 3-4 第一步到第三步）这样处理的好处是能有效增大候选集中的项被抽中的概率，进而提高候选集中的项的估计频率准确性。一个例子能很好解释其中的原因：假设用户u_i的原项集S_i真包含候选集 C，S_i的大小为ℓ=50，C 的大小为k_{max}=20；在未经过以上处理前，从s_i随机抽取一个项，该项属于候选集的概率为 2/5，但经过①～③步处理后，被抽中的项属于候选集的概率为 1。可见，这样的处理是很有意义的。另一方面，如果不对长度小于的交集T_i填充冗余项，T_i的大小直接暴露给用户，会给攻击者提供额外的信息，存在着隐私泄露的风险。

阶段二中的抽样 S-Hist 只能计算候选集中各项的估计频率。然而，真实的 top-k 项并不一定恰好都落到候选集中，为了缓解这个问题，需要把两个阶段的结果利用起来。这里我们按以下公式得到最终各项的估计频率：

$$\hat{f} = \begin{cases} \widehat{f_1} & ,i_i \notin C \\ (\widehat{f_1} + (\ell - 1)\widehat{f_2})/\ell & ,i_i \in C \end{cases} \qquad \text{式}(3-5)$$

需要说明的是，阶段一处理的数据与阶段二处理的数据并不相交，为了保证阶段一、阶段二的数据与总数据满足同分布，必须保证划分数据时是随机划分的

算法 3 GFIM LR(用户客户端部分)

输入： k_{max}-bit字符串$\mathbf{x}\in\{\frac{-1}{\sqrt{m}},\frac{1}{\sqrt{m}}\}$
输入： 隐私参数：ϵ
输入： 用户u_i的项集：S_i
输入： 候选集：C
输出： 干扰比特z_i

求交集T_i=$S_i \bigwedge C$
if $|T_i| < k_{max}$ **then**
　往T_i中加入若干个冗余项得到新的用户项集$|N_i| = k_{max}$
end if
从项集N_i中随机选取一个项i_i
计算$c_\epsilon = \frac{e_\epsilon+1}{e_\epsilon-1}$
if $i_i = \perp$ **then**
　随机选取$z_i \in \{c_\epsilon\sqrt{m}, -c_\epsilon\sqrt{m}\}$
else
　生成一个标准基向量$e_{i_i} \in \{0,1\}$
　计算$x_{i_i} = \mathbf{x}\top e_{i_i}$

$$\text{概率}p = \begin{cases} \frac{e^\epsilon}{e^\epsilon+1}, when & z_i = c_\epsilon m x_{i_i} \\ \frac{1}{e^\epsilon+1}, when & z_i = -c_\epsilon m x_{i_i} \end{cases}$$

end if
return z_i

图3-4 GFIM 第二阶段

以下证明阶段二满足 ε-本地差分隐私。

证：使$\mathcal{B}$代表整个阶段二对用户数据的处理算法。$\mathcal{B}$的输入是（1）隐私参数 ε，（2）用户u_i的项集S_i，（3）候选集 C。$\mathcal{B}$首先对S_i进行修剪得到N_i。之后$\mathcal{B}$对N_i采用抽样 S-Hist 算法，该处理过程用$\mathcal{C}$表示。由 3.2.2 节知，$\mathcal{C}$满足ε-本地差分隐私。

使S_1，S_2表示任意两个未修剪前的用户项集，o 表示$\mathcal{B}$的任意输出。要证阶段二满足 ε-本地差分隐私，即证：

$$\frac{\Pr(\mathcal{B}(S_1,\varepsilon,\mathrm{C})=\mathrm{o})}{\Pr(\mathcal{B}(S_1,\varepsilon,\mathrm{C})=\mathrm{o})}\leq e^{\varepsilon} \qquad 式(3-6)$$

因为修剪过程是确定的，所以有

$$\Pr(\mathcal{B}(\mathrm{S},\varepsilon,\mathrm{C})=\mathrm{o})=\Pr(\mathcal{C}(N_1,\varepsilon)=\mathrm{o}) \qquad 式(3-7)$$

又$\mathcal{C}$满足 ε-本地差分隐私，有

$$\frac{\Pr(\mathcal{B}(S_1,\varepsilon,\mathrm{C})=\mathrm{o})}{\Pr(\mathcal{B}(S_1,\varepsilon,\mathrm{C})=\mathrm{o})}=\frac{\Pr(\mathcal{C}(N_1,\varepsilon)=\mathrm{o})}{\Pr(\mathcal{C}(N_2,\varepsilon)=\mathrm{o})}\leq e^{\varepsilon} \qquad 式(3-8)$$

所以阶段二满足 ε-本地差分隐私。

由于阶段一，阶段二分别满足 ε-本地差分隐私，又阶段一、阶段二面向的是数据互不相交，由差分隐私的并行组合性知，GFIM 满足 ε-本地差分隐私。

3.3 本章小结

在这一章中，我们首先对本次毕设要解决的问题进行了严格细致的数学描述，明确了最终要达到的目标。接着，我们从整体框架，具体每个阶段的设计对本文提出的算法 GFIM 进行了非常详细的阐述。GFIM 的创新点归纳如下:

（1） 两阶段设计，通过提前筛选出候选集的办法提高频繁项估计频率的准确性；

（2） 通过把用户划分为不相交的两部分分别对应两个阶段，避免划分隐私参数。

此外，本章从数学上证明了 GFIM 满足 ε-本地差分隐私，并且给出了关键部分的伪代码。读者可以自己动手实现 GFIM，加深对本文算法的理解。

第四章　算法性能测试与指标

我们设计了广泛的实验来研究 GFIM 挖掘频繁项的准确性。实验的目的主要有两个：(1)我们想了解 GFIM 能否准确地挖掘出 top-k 频繁项；（2）关键参数的变化会在多大程度上影响 GFIM 结果的准确性；（3）GFIM 相对已有的解决方案有没有改善。为此，我们在合成数据集和真实数据集上进行测试。在合成数据集中测试能够帮助我们快速系统地了解 GFIM 的效果，而在真实数据集上测试能展示 GFIM 在真实应用场景中的可用性。

4.1　数据集

（1） 合成数据集

我们生成了两种分布的合成数据集以测试 GFIM 的性能，分别是满足拉普拉斯分布的数据集和满足正态分布的数据集。一般而言，各项所占频率的差异越大，曲线越陡，GFIM 越容易准确挖掘出频繁项。在期望和方差相同的情况下，拉普拉斯分布的曲线比正态分布的曲线更陡。通过观察在不同分布的数据集的结果，我们能更深刻理解这个特点。在两种不同分布的数据中，我们均取数学期望为 500，方差为 1800。所以对这两种数据而言，频繁项均在 500 附近。对应每个用户的项集的大小和项的取值大小，我们取ℓ=50，d=1000。

（2） 真实数据集

本文选取了 US Census 1990 dataset[18]数据集作为测试用的真实数据集。US Census 1990 dataset 是从 1990 年美国人口统查数据得到的公共使用微数据样本（Public Use Microdata Samples）中的部分样本组成的。数据集中的每条记录与 47 个属性值相连，整个数据集共包 1000000 条记录和 396 个不同的项。

4.2　实验参数

GFIM 的性能受以下参数的影响：

（1） n, 用户数量。一般而言，在差分隐私模型中，用户数量越多，数据越多，数据中的统计信息就越容易被准确地估计。本地化差分隐私模型中每个用户都对自己的数据进行加噪，引入的噪音量远大于中心化差分隐私模型，因而足够大量的数据对于准确挖掘频繁项意义重大。

（2） ε,隐私参数。根据差分隐私的定义，ε 越大，隐私保护能力越高，数据的效用性越低；ε 越小，隐私保护能力越低，数据的效用性越高。与传统的中心化差分隐私模型一般取 1 以内的隐私参数不同，由于本地化差分隐私模型自身的优势，我们可以采用更大的隐私参数以便更准确地挖掘频繁项。

（3） k，要求出的频繁项的数量。容易理解，随着 k 的增大，整体的估计频率的准确性会随之下降，这是因为出现频率较低的项更不容易被发现。

（4） k_{max}，候选集的大小。在阶段一中，我们求得一个频繁项，假设最终估计得到的频繁项主要出自于候选集。这就要求候选集的大小选择得当。候选集要是太小，频繁项落到候选集外的概率增大；候选集过大，阶段二中生成的随机矩阵增大，对候选集中的项的频率估计的准确性下降。如何选择合适的

k_{max}不在本文的研究范围中。毕业设计的各次实验均取取1`。

在我们的实验中，我们将会更改上述的（1）到（3）个参数，研究它们对 GFIM 准确性的影响。

4.3 评判标准

回顾我们的目标是要挖掘出出现最频繁的 k 个项以及其频率，因此，评价算法的好坏应该从两个方面进行考核：①真实的 top-k 项是不是都被挖掘出来；②估计的 top-k 项频率有多准确。我们用以下两个指标来分别评估 GFIM 在这两方面的表现。

（1） 准确率（Precision）

假设V为真实的 top-k 项构成的集合，$\hat{V}$为估计的 top-k 项构成的集合，有

$$\text{Precision} = \frac{\left|V \cap \hat{V}\right|}{k} \qquad \text{式}(4-1)$$

Precision 计算的$\hat{F}$中是真正的 top-k 项的比例。Precision 的取值范围为[0,1]，Precision 越接近 1，挖掘出的真实 top-k 项越多。

（2） 相对误差（Relative Error，RE）

假设$V = \{v_1, v_2, v_3, \ldots, v_n\}$为真实的 top-k 项构成的集合，$\hat{f}(v_i)$为$v_i$的估计频率，$f(v_i)$为$v_i$的真实频率，有

$$\text{RE} = Median_{v_i \in V} \frac{\left|\hat{f}(v_i) - f(v_i)\right|}{f(v_i)} \qquad \text{式}(4-2)$$

RE 衡量的是估计频率与真实频率之间的误差。RE 的取值范围为[0,1]，RE 越接近 0，估计频率越准确。

综合使用 Precision 和 RE 能较为全面地衡量频繁项挖掘算法的真实表现，这两者缺一不可。如果只用 Precision 评价算法的性能，将难以发现 Precision 接近 1 但估计频率远远偏离真实频率的情况；如果只用相对误差评价算法的性能，则难以发现相对误差接近 0 但是真实的 top-k 项却并不是估计的 top-k 项的情况。

对照算法：

目前，关于满足本地化差分隐私的集值数据的频繁项挖掘的研究非常有限，主要的算法仅有抽样 S-Hist，抽样 RAPPOR，LDPMiner-RAPPOR，LDPMiner-SH 等几种。其中，LDPMiner-PAPPOR 和 LDPMiner-SH 均是抽样 S-Hist 的改进算法，LDPMiner-RAPPOR 在算法的第一部分采用抽样 S-Hist，第二部分采用抽样 RAPPOR；LDPMiner-SH 在算法的第一、第二部分都采用抽样 S-Hist。本文的贡献在于创新性地提出了一个基于抽样思想的两阶段的满足本地化差分隐私的频繁项挖掘算法。为了验证该思想的可行性和合理性，我们选取了 LDPMiner-SH 作为本文的对比算法。

4.4 基于合成数据集的实验

我们首先研究了样本数量对 GFIM 估计准确性的影响。需要说明的是，在基于合成数据集的一系列实验中，我们均设置隐私参数 ε=2，项的取值集合设为 1000。图 6-1 展

示了 GFIM 在满足拉普拉斯分布的小数据集上的准确率表现。该图对应的数据集满足数学期望为 500，方差为 1800 的拉普拉斯分布，共有 50000 用户，其中每个用户拥有 50 个项集。由于数学期望设置在 500，故该数据集对应的真实的 top-30 项为项 486 到项 515。图中实心柱形代表各频繁项的真实频率，空心边框代表对应的估计频率。为了便于观察 GFIM 的准确率表现，此处把 GFIM 没有识别出来的频繁项的估计频率置为 0，哪怕原本 GFIM 对该项的估计频率不为 0。可以看到，当数据集的大小为 50000 条记录时，GFIM 的 Precision 仅为 0.4，超过一半的频繁项没有被准确识别。而且，估计频率和真实频率之间的误差较大。图 6-2 说明了对于符合正态分布的小数据集，该问题同样存在。

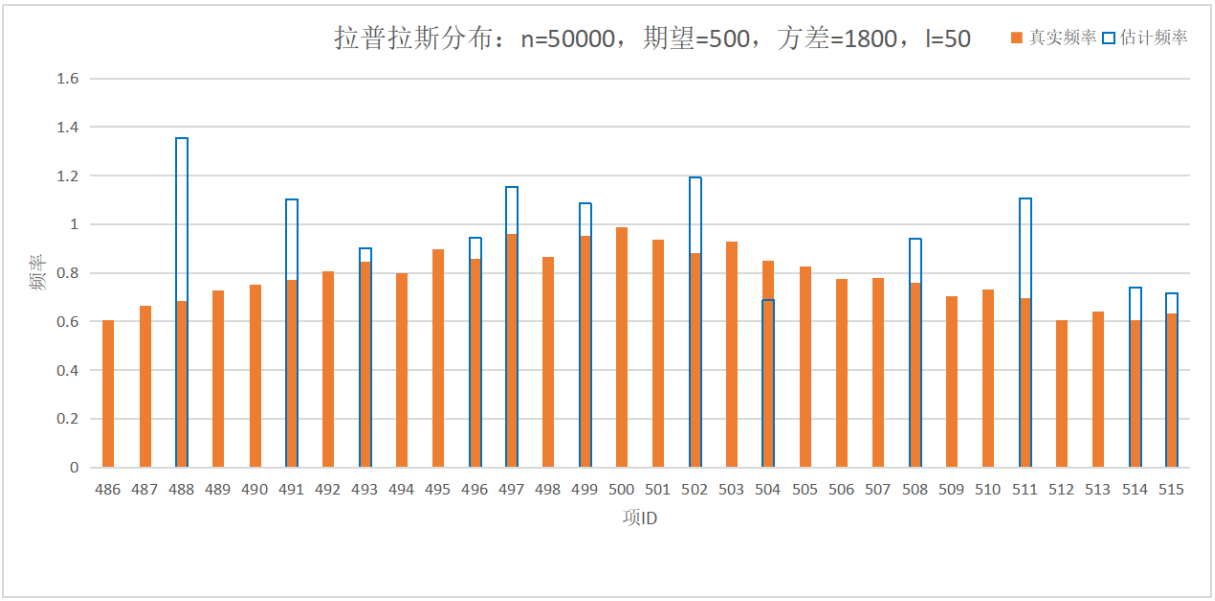


图5-1 GFIM 在满足拉普拉斯分布的小数据集的合成数据集上的实验结果

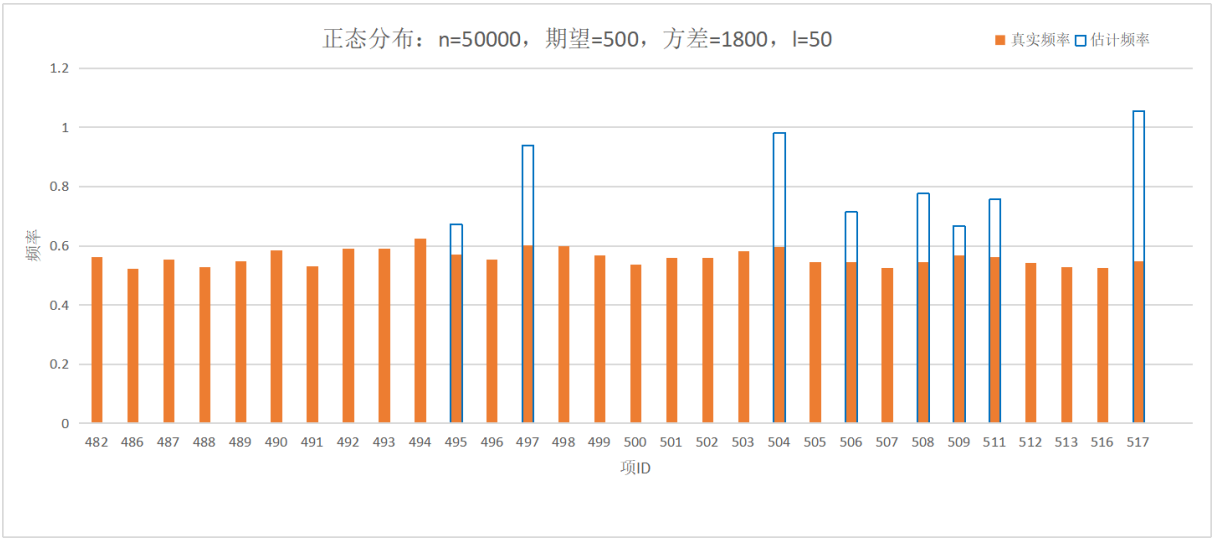


图5-2 GFIM 在满足正态分布的小数据集的合成数据集上的实验结果

当数据集增大至 n=500000 时，由 GFIM 得到的估计结果大为改善。观察图 6-3 可

知，对于同样满足方差为 500，期望为 1800 的拉普拉斯分布的大数据集，GFIM 得到的结果非常准确，对于 top-30 频繁项估计的 Precision 高达 0.967，仅有一个频繁项没有被识别出来。并且，估计频率和真实频率之间的误差也大为缩小。同样，对于符合正态分布的数据集，当数据集的大小从 50000 扩大至 500000 时，估计的准确率从 0.267 上升到 0.7，估计频率和真实频率之间的误差也缩小了。

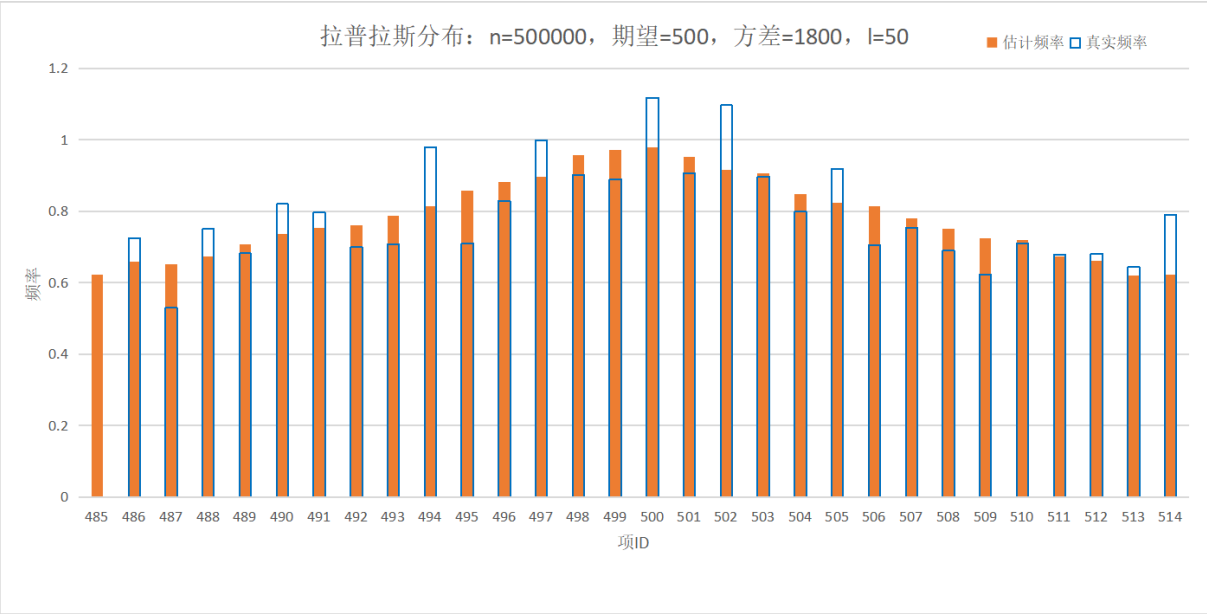


图5-3 GFIM 在满足拉普拉斯分布的大数据集的合成数据集上的实验结果

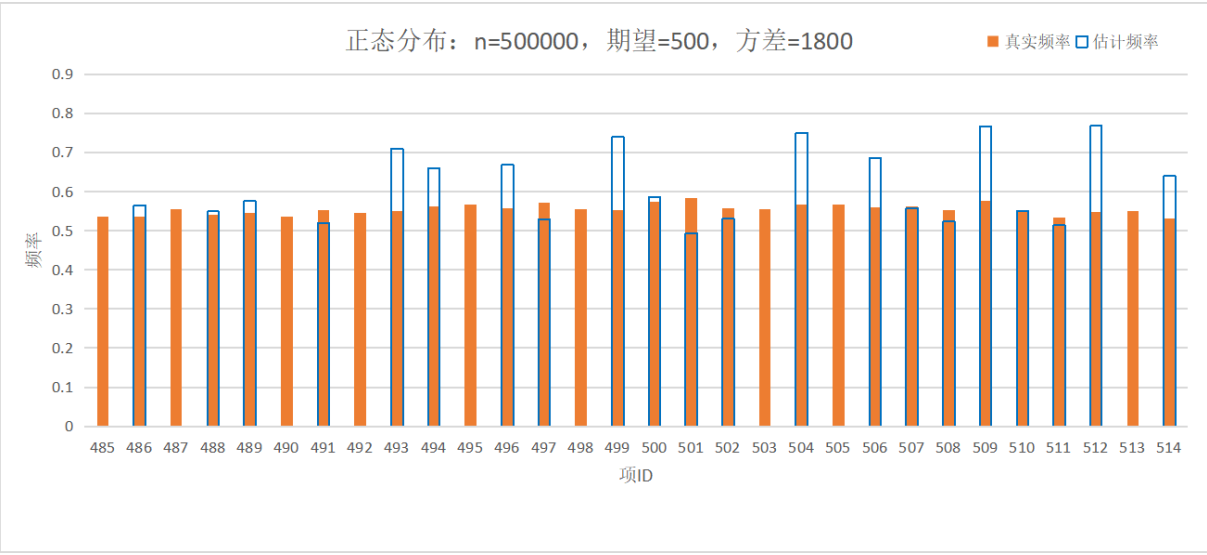


图5-4 GFIM 在满足正态分布的大数据集的合成数据集上的实验结果

如果我们对比一下 GFIM 在满足拉普拉斯分布的数据集和满足正态分布分布的数据集这两种数据集上的表现，容易看出当数据集的大小一致，期望和方差一致时，GFIM 在满足拉普拉斯分布的数据集上能取得更好的效果。对比图 5-3 和图 5-4，图 5-3 的准确率为 0.967 而图 5-4 的准确率仅为 0.7，图 5-3 的估计频率与真实频率的误差更

小。这说明了，数据集分布模型的概率密度曲线斜率会影响 GFIM 算法的准确性。曲线越陡，即各个频繁项之间的频率之差越大时，数据集的频繁项越容易被挖掘出来。

我们同样用对比算法 LDPMiner 在以上四个数据集上做了同样的实验。实验结果分别如图 5-5，图 5-6，图 5-7，图 5-8，Precision 分别是 0.267，0.3，0.933，0.667。实验结果图样反映了上述提到的两个趋势：①随着数据集的增大，估计的准确率上升；②估计的准确率与数据集的分布有关，分布模型的概率密度曲线斜率越大，越容易准确估计频繁项。可见，这两个趋势并不是 GFIM 算法独有的，而是满足本地化差分隐私的频繁项挖掘共有的特点。GFIM 在上面四个合成数据集上的准确率分别是 0.4，0.267，0.967，0.7。与 LDPMiner 得到的结果相比差距并不明显，下一节将基于真实数据集用更丰富的实验详细分析对比两者的优劣。

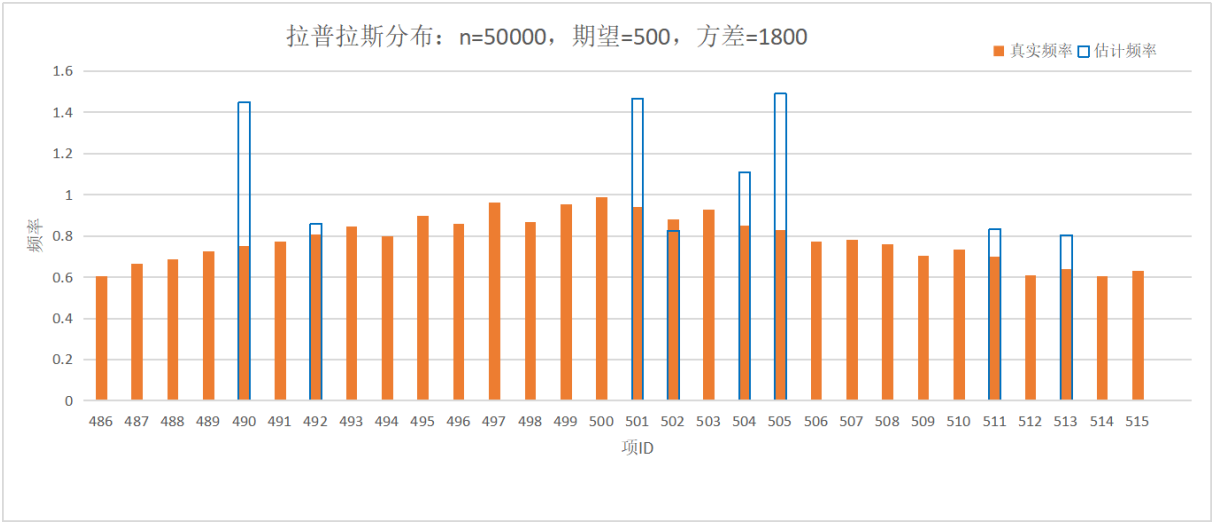


图5-5 LDMPMiner 在满足拉普拉斯分布的数据集合成数据集上的实验结果

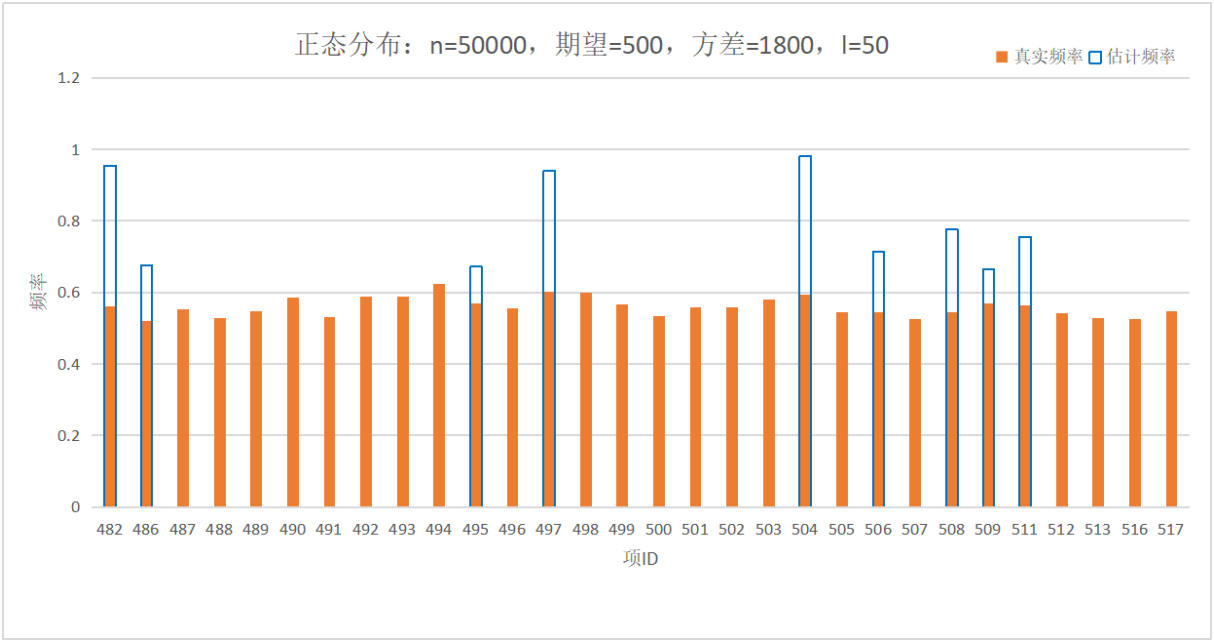


图5-6 LDMPMiner 在满足正态分布的小数据集的合成数据集上的实验结果

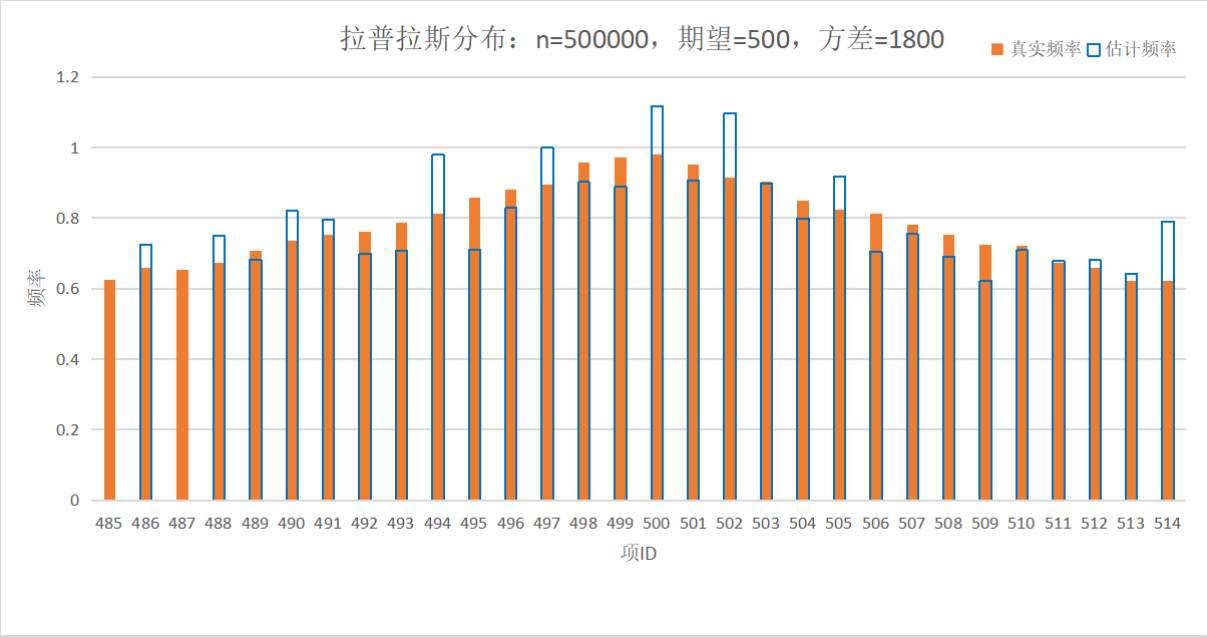


图5-7 LDMPMiner 在满足拉普拉斯分布的大数据集的合成数据集上的实验结果

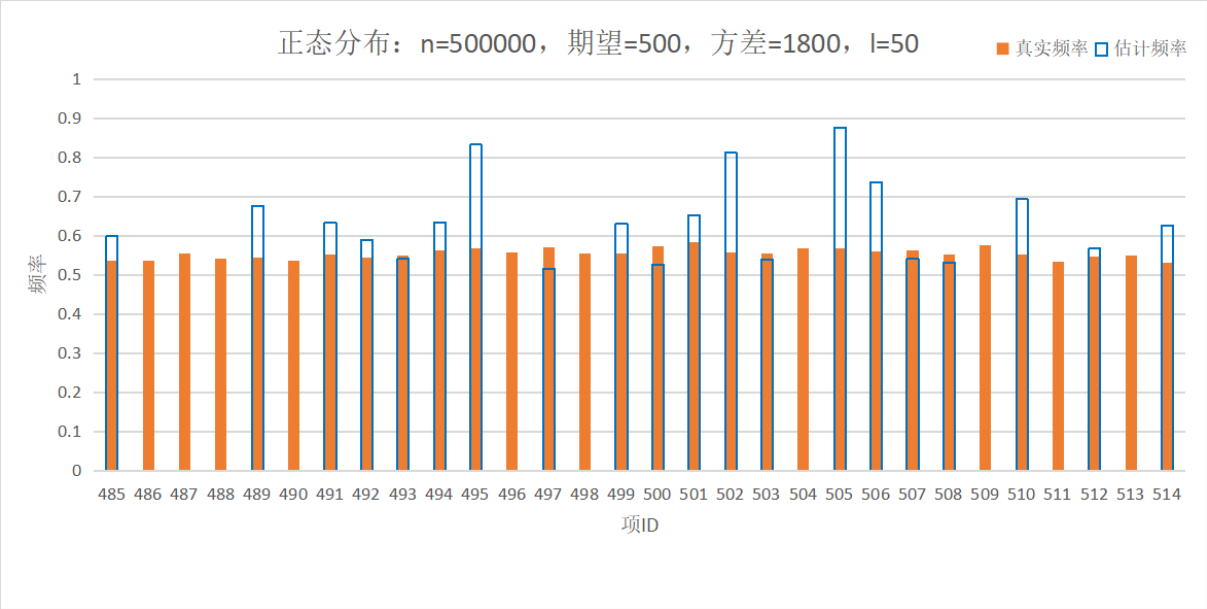


图5-8 LDMPMiner 在满足正态分布的大数据集的合成数据集上的实验结果

4.5 基于真实数据集的实验

本节我们基于真实数据集 US Census 1990 dataset，通过分别单独修改 k 和隐私参数的大小，观察比较 LDPMIner 和 GFIM 在不同情况下的性能表现。注意，由于本地

化差分隐私模型中多处引用随机噪声，由 GFIM 和 LDPMiner 得到的结果均有一定波动性，所以下文展示的关于相对误差和精确率的结果均是重复实验 20 次取平均值得到的。

（1） k 对相对误差和准确率的影响

实验结果如图 5-9 和图 5-10 所示。由图 5-9，图 5-10 可知，总体而言，当 ε 取 1 时，随着 k 的增大，LDPMiner 和 GFIM 的相对误差均不断增大，准确率不断提高。注意当 k 在 3 附近时，两条相对误差曲线均存在一个极小点。此时观察图 5-10，当 k<3 时，准确率急剧下降。进一步分析程序的详细运行结果可知，当 k<3 时，候选集大小$k_{max} < 6$，候选集过小导致频繁项落到候选集的概率下降，即准确率下降。而由于真实的频繁项没有每次都落到候选集内，又导致对频繁项的估计频率误差增大。由此解释了 k=3 附近两条相对误差曲线存在极小点的原因。

对比 LDPMiner 和 GFIM，当$\varepsilon = 1$，$k \in [1,20]$时，GFIM 在相对误差和准确率上的表现均优于 LDPMiner。

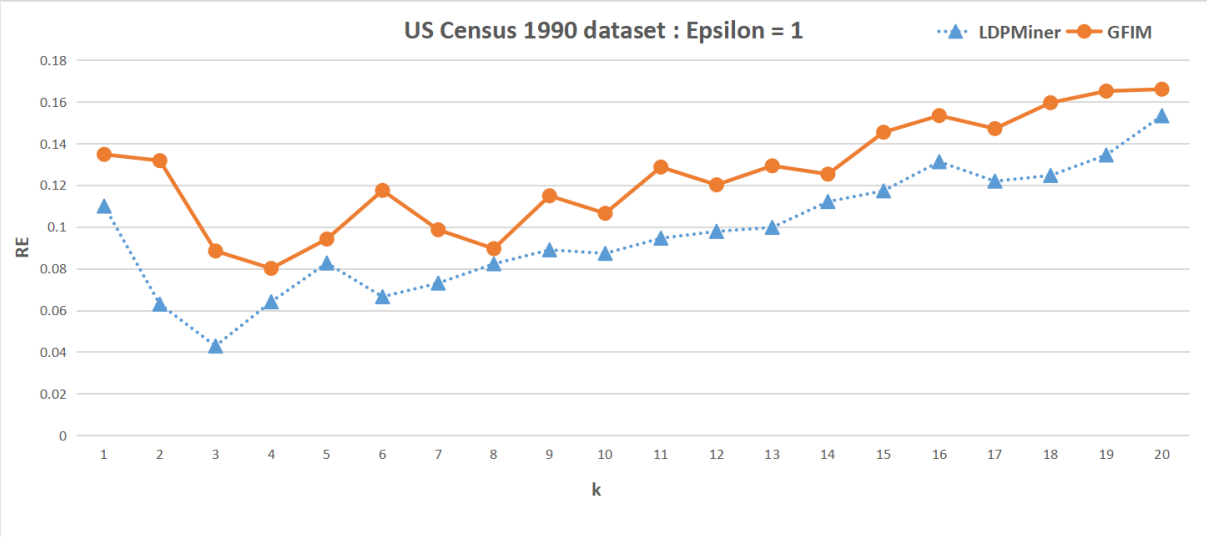


图5-9 随着k变化的相对误差结果

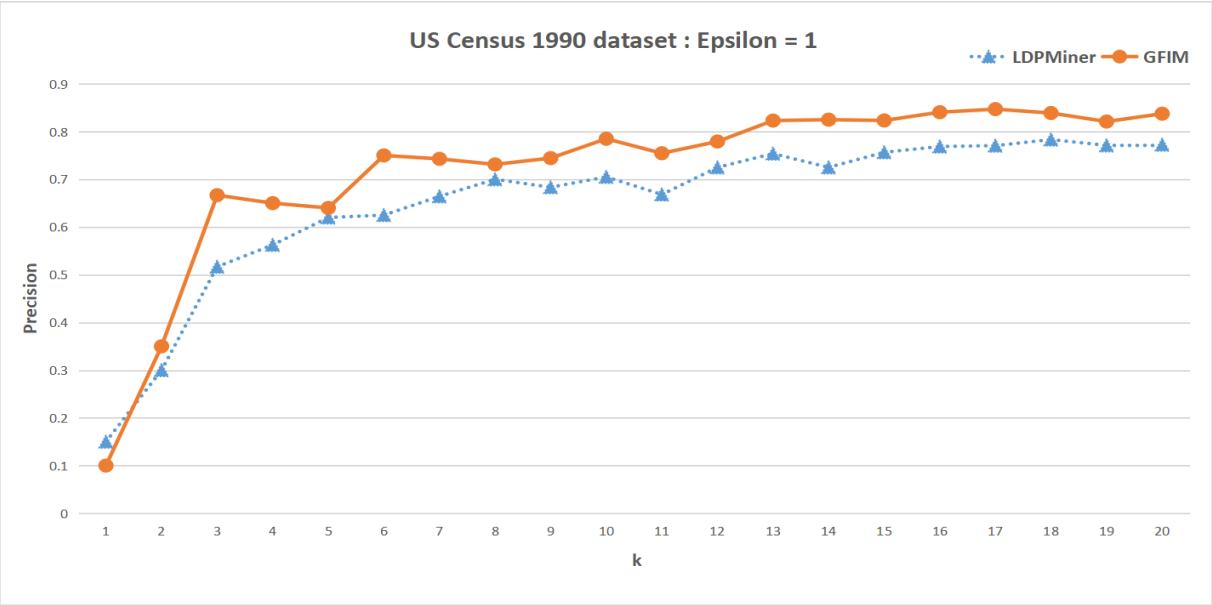


图5-10 随着k变化的准确率结果

（2） ε 对相对误差和准确率的影响

接着我们来看一下 LDPMiner 和 GFIM 在不同隐私参数大小下的表现。

显然，随着隐私参数的不断增大，LDPMiner 和 GFIM 在相对误差和准确率都持续改善。观察图 5-11 和图 5-12 可知，当隐私参数小于 1.5 时，GFIM 在 RE 和 Precision 上的表现都要优于 LDPMiner。这是因为 LDPMiner 和 GFIM 均采用了两阶段求候选集的思想，但 LDPMiner 把隐私参数平均划分给两个阶段而不划分数据集大小，GFIM 不划分隐私参数而随机等大划分数据集。当数据集足够大而隐私参数较小时，GFIM 的准确率更高。在具体应用场景中，数据集一般在百万量级，为了给用户隐私提供充分的保护，隐私参数越小越好，所以可以说，本文提出的 GFIM 在实际应用场景中将能获得更好的表现。

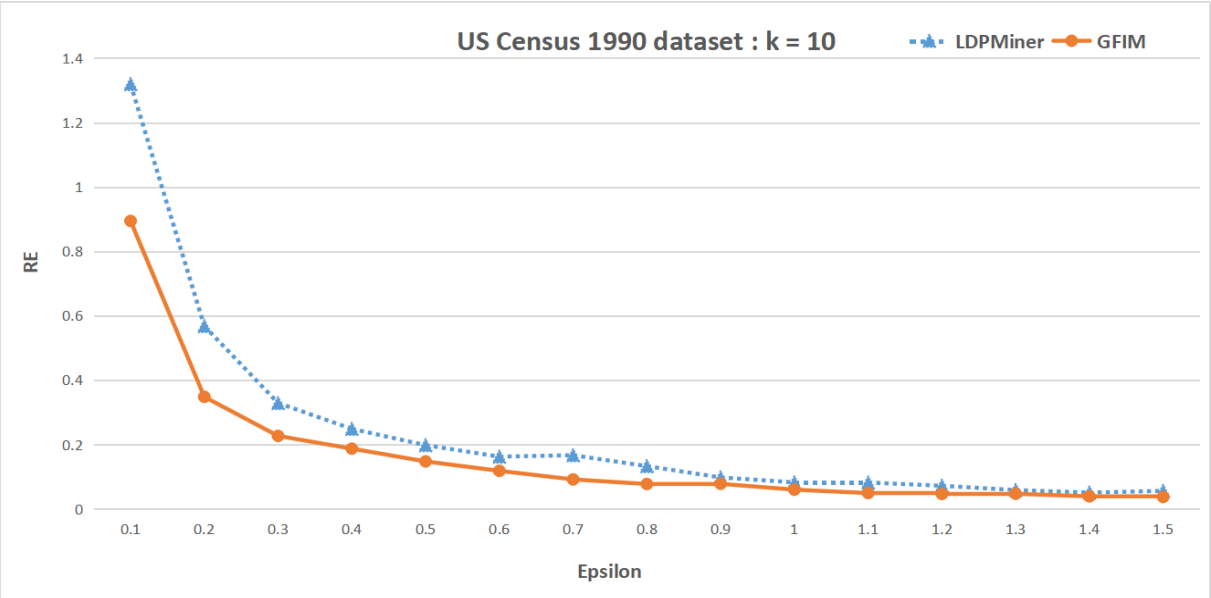


图5-11 随着ε变化的相对误差结果

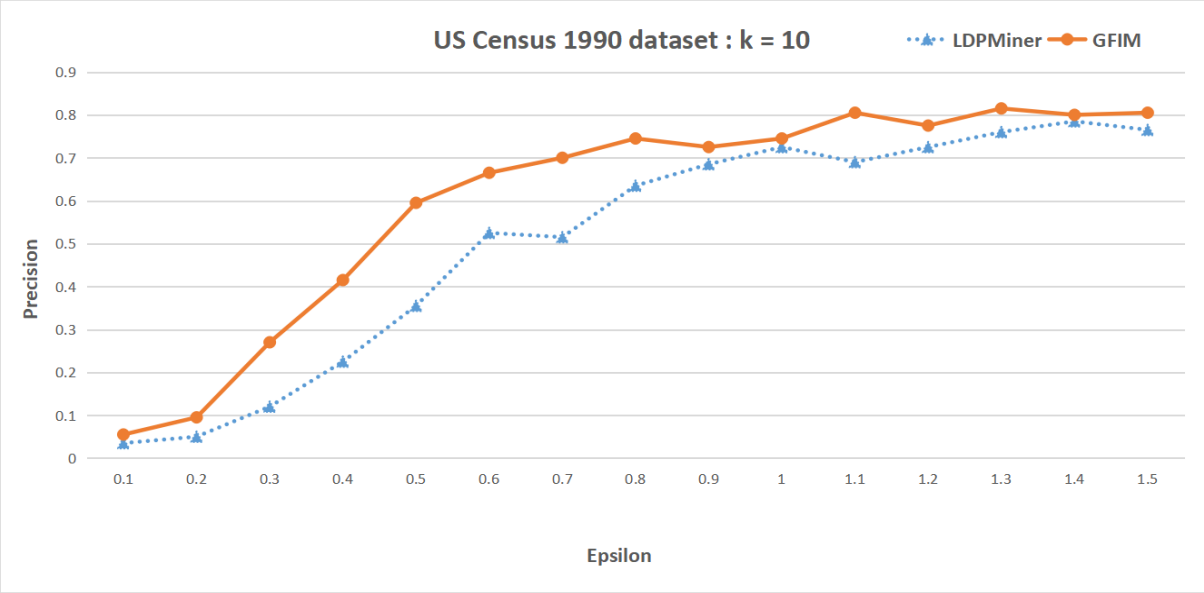


图5-12 随着ε变化的准确率结果

4.6 本章小结

本章为实验部分。本文对 GFIM 及其对比算法 LDPMiner 进行了广泛的实验，包括基于合成数据集上的实验与基于真实数据集上的实现，具体的实验结果证明了 GFIM 算法的优越性。本章的 4.1 节，4.2 节，4.3 节分别对实验的数据集，影响实验结果的关键参数，实验结果的评判标准作了明确而清晰的说明，4.4 节和 4.5 节为实验结果说明。

由实验结果归纳得到的结论如下：

（1） 随着数据集的增大，频繁项越容易被挖掘出来；

（2） 数据集的分布会影响频繁项挖掘的准确性，分布模型对应的概率密度曲线越倾斜，频繁项挖掘的准确性越高；

（3） 随着 k 的增大，GFIM 和 LDPMiner 的相对误差不断增大，准确率不断提高；

（4） 随着 ε 的增大，GFIM 和 LDPMiner 的相对误差不断减小，准确率不断提高；

（5） 总体而言，GFIM 的性能优于 LDPMiner。

由于隐私参数越小，对用户隐私的保护水平越高，结论（5）说明了 GFIM 相对 LDPMiner 更有实用价值。

第五章　结束语

5.1　论文工作总结

近年来，隐私安全事故层出不穷。越来越大规模的隐私泄露唤起了公众对个人隐私保护的强烈欲望。与此同时，以 Google，Facebook，Alibaba 等为首的科技巨头并不愿意放弃对用户数据蕴含价值的挖掘。如何在保证用户一定程度隐私保护的同时尽可能准确地挖掘有价值信息成了这些企业亟需解决的问题。本地化差分隐私模型作为当前最为先进的隐私保护模型，是解决该问题的首选。本文聚焦于数据挖掘中最经典的频繁项挖掘问题，创造性的提出了一种满足本地化差分隐私的基于抽样思想的两阶段的频繁项挖掘算法，并从理论和实验的角度证明了该算法的可行性和优越性。

文章的主要工作始于对背景知识的学习。差分隐私模型提出于 2006 年。起初，它主要针对于中心化数据库的隐私保护，能够实现在数据库发生变化的时候不泄露个体用户的隐私。慢慢地，该模型的弊端逐渐显现，原因在于一个实际上的“完全可靠”的第三方数据收集者是不存在的。中心化的数据收集者总会面临有意无意的数据泄露问题。于是乎，更为先进的本地化差分隐私模型被提出，该模型把隐私化处理的场所转移到用户的本地客户端，既提高了隐私保护水平，又规避了数据收集者因数据保护不力导致的数据泄露风险。基于本地化差分隐私模型，我们详细讨论了几种最常用的扰动和解码方案——随机响应，S-Hist 和 RAPPOR，并在对比中选择了通信开销和计算开销都更小的 S-Hist 作为本文提出算法 GFIM（Group-based Frequent Items Mining）的基础技术。GFIM 的主要思想是首先把用户随机划分为等大的两份，假设每一份子数据集均与划分前的整体数据集同分布，然后把整个加噪和挖掘频繁项的过程划分为两个阶段。在第一个阶段，GFIM 实现对第一部分用户的加噪保护和频繁项候选集的筛选；在第二个阶段，GFIM 把候选集发送给用户，让用户对自身的项集重新进行打包和加噪，重点挖掘候选集内各项的频率。最后，GFIM 综合两个阶段的成果得到最后估计的候选集和对应频率。

为了验证算法 GFIM 的可行性和对已有方案的改善，我们选取了当前关于同样问题的最好的解决方案 LDPMiner 进行对比，并在合成数据集和真实数据集进行多次实验。实验表明，GFIM 实现了准确识别大部分真实的频繁项，并且在 Precision 和 RE 这两个指标上的表现均优于 LDPMiner。

5.2　问题与展望

本文的贡献在于提出了一个基于抽样思想的两阶段的频繁项挖掘算法，这在之前的关于本地化差分隐私的研究是没有的，但本文也并非解决了关于此话题的所有问题。一个悬而未决的问题是如何选择合适的候选集大小k_{max}。在关于本文的所有实验中，我们均设置$k_{max} = 2\mathrm{k}$。这是一个直观上的选择。事实上，在实验中我们发现，针对不同的数据集和不同的 k 值，k_{max}有对应的最适合的大小。k_{max}的大小会对实验结果有重要的影

响，一个合适的k_{max}将极大提高实验结果的准确性。如何让算法针对不同的数据集自动地选取一个合适的k_{max}值而不是人为地依靠经验和对数据集的先验知识来手工设置k_{max}值是本文未解决的问题。我们期待这个问题能在未来的工作中得到妥当的解决。

参考文献

[1] 丁丽萍, 卢国庆. 面向频繁模式挖掘的差分隐私保护研究综述[J]. 通信学报, 2014,35(10):200-209.

[2] 王平水, 王建东. 匿名化隐私保护技术研究综述[J]. 小型微型计算机系统, 2011,32(2):248-252.

[3] Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information (abstract): Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, Seattle, Washington, USA, 1998[C]. ACM.

[4] SWEENEY L. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002,10(05):557-570.

[5] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis: Theory of Cryptography Conference, 2006[C]. Springer.

[6] Dwork C. Differential privacy: a survey of results: Proceedings of the 5th international conference on Theory and applications of models of computation, Xi'an, China, 2008[C]. Springer-Verlag.

[7] Dwork C, Lei J. Differential privacy and robust statistics: Proceedings of the forty-first annual ACM symposium on Theory of computing, Bethesda, MD, USA, 2009[C]. ACM.

[8] Kasiviswanathan S P, Lee H K, Nissim K, et al. What Can We Learn Privately?[J]. SIAM Journal on Computing, 2011,40(3):793-826.

[9] Duchi J C, Jordan M I, Wainwright M J. Local Privacy and Statistical Minimax Rates: Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, 2013[C]. IEEE Computer Society.

[10] Qin Z, Yang Y, Yu T, et al. Heavy hitter estimation over set-valued data with local differential privacy: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016[C]. ACM.

[11] 叶青青, 孟小峰, 朱敏杰, 等. 本地化差分隐私研究综述[J]. 软件学报, 2017.

[12] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis: Theory of Cryptography Conference, 2006[C]. Springer.

[13] McSherry F, Talwar K. Mechanism Design via Differential Privacy: 48th Annual IEEE Symposium on Foundations of Computer Science, Washington, DC, USA, 2007[C]. IEEE.

[14] Warner S L. Randomized response: A survey technique for eliminating evasive answer bias[J]. Journal of the American Statistical Association, 1965,60(309):63-69.

[15] Bassily R, Smith A. Local, private, efficient protocols for succinct histograms: Proceedings of the forty-seventh annual ACM symposium on Theory of computing, 2015[C]. ACM.

[16] Erlingsson Ú, Pihur V, Korolova A. Rappor: Randomized aggregatable privacy-preserving ordinal response: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, 2014[C]. ACM.

[17] Nguyên T T, Xiao X, Yang Y, et al. Collecting and analyzing data from smart device users with local differential privacy[J]. arXiv preprint arXiv:1606.05053, 2016.

[18] Spmf: An open-source data mining library.[EB/OL]. http://www.philippe-fournier-viger.com/spmf.

致　谢

此次毕业设计工作从寒假就开始了。因为之前没有接触过差分隐私这个领域，我在学习和理解背景知识上花了不少功夫，在设计改进算法和做实验过程中也遇到了很多困难，甚至一度停滞不前，辛亏在学长和老师的指导下最终解决了问题。在这里我要特别感谢我的指导老师程祥老师和一直指导我的学长杨健宇师兄。没有他们就没有本文的研究成果。感谢他们一直对我进行无私的指导和帮助，一次次帮助我找出算法设计和实现上的问题，以及提出论文修改的建议。

同时，要感谢本文参考论文的学者，我们都是站在前人的肩膀上。正是有了他们的努力，我才有了设计本文算法的灵感。从他们的身上我也学会了很多做研究和写论文的方法和技巧。

最后要感谢我身边的同学和朋友。在完成本论文的过程中我受到了很多人的帮助和支持。作为一个本科生，本人自觉自己的学术水平还非常有限，本文难免有不当之处，在此恳请各位老师和学者评判指正！