剪烛漫记:Differential Privacy

差分隐私学习笔记

作者：谢天

时间：Dec 29, 2022, 大二寒假

版本：1.0

学校：中科大网安学院

临渊羡鱼，不如退而结网

目录

序言

致亲爱的读者：

好吧，或许此份笔记的读者寥寥无几，但是我还是想写下来，因为我觉得这是一件很有意思的事情。

在一切开始之前，请先谅解一下作者的水平，我还是一名“头脑空空”的大二本科生，抱着 Dwork 差分隐私洋文书独自猛啃。文中英文术语是完全凭借自己理解翻译的，其他图表之类则是参照了相关论文和资料。

在做笔记的过程中，越发意识到自己的不足，向上空间还是很大的。这份笔记是第一版，日后如果对差分隐私有更深刻的了解，我也会继续更新第二版，第三版…

你问我写这本书的意义？理由呢？总得有个坚持下去的动力吧。呃，我觉得其实万事不必需要那么多理由。就像有人得癌症，有人不想吃河鱼，有人喜欢把脑袋枕在胳膊上，就当上帝像一个小男孩，喜欢没事就在草地上踩蘑菇。你要问为什么？就是因为刚刚下过雨，草地里钻出无数的蘑菇，而他脚上正好有双不错的运动鞋而已。

阅读愉快！

谢天

第一章 The Spark of Differential Privacy

内容提要

- 历史背景
- Dinur & Nissim
- reconstruction attack
- Blatant Non-Privacy(BNP)

1.1 简述

本节从回溯 Dinur 的一篇论文开始，该文被认为是启发了 Dwork 差分隐私的灵感源泉。我暂时不会阐释怎样使得一个算法有多隐私，恰恰相反，我想说的是大部分方案都是“全然非隐私”的（BNP）。同时，我们会重点分析一种称为重构攻击（Reconstruction Attacks）的攻击形式，并阐述了为了保护隐私，我们应该增加的噪声理论上的值应该是多少。

注 本节均参照此论文 Revealing Information while Preserving Privacy Policy

1.2 简化版模型与基本表述

我们从这里严谨完备一个数据库模型，以下做了部分简化以便描述

- 定义每条数据为一行
- 每列为一个属性/特征
- 我们假定 Name,Postal Code,Date of Birth,Sex 等属性是不敏感的，是公开数据
- 只认为一种属性是敏感的，即 Has Disease？我们将其简化为 1bit(取值只能为 0 或 1)
- 令 $d \in \{0,1\}^n$ 表示隐私 bit 的向量

Name	Postal Code	Date of Birth	Sex	Has Disease?
Alice	K8V7R6	5/2/1984	F	1
Bob	V5K5J9	2/8/2001	M	0
Charlie	V1C7J	10/10/1954	M	1
David	R4K5T1	4/4/1944	M	0
Eve	G7N8Y3	1/1/1980	F	1

图 1.1: 数据库模型实例

命题 1.1 (模型概述)

这里假定有攻守双方，一方是查询者，一方是管理者。

查询者被允许提的问题形如：“数据库里有多少行在条件 X 下满足’Has Disease=1’?”

条件 X 可以是”Name= Alice OR Name = Charlie OR Name = David”，那么问题答案就是 2

更抽象起见，我们令查询向量为 $S = \{0,1\}^n$,0 表示该行不满足条件，1 表示满足条件，例如条件 X 可以表示为 $S = \{1,0,1,1,0\}$. 真实结果用 A(S) 表示，$A(S) = d \cdot S$ (例中 $A(S) = \{1,0,1,0,1\} \cdot \{1,0,1,1,0\} = 2$)

当然，这极大侵犯了隐私，所以管理者的响应 r(S) 会在 A(S) 的基础上加上一些噪声，界限小于等于 E

$$|r(S) - A(S)| \leq E \tag{1.1}$$

1.3 重构攻击的数学表达

定义 1.1 (全然非隐私 BNP)

我们称如下算法是全然非隐私的（blatantly non-private）：
如果攻击者能够重构一个数据库 $c \in \{0,1\}^n$, 使得其与真实数据库 d 几乎完全匹配, 或者说偏移量不超过 $o(n)$.

♣

如果一个算法是 BNP 的，那么这个算法之下毫无隐私可言。这就是重构攻击（reconstruction attack），随后我们可以证明出一般的方案都是 BNP 的。

定理 1.1

如果查询者被允许进行 2^n 次子集查询，并且管理者添加了 E 的噪声，那么根据响应结果，查询者可以重构出偏移量为 4E 的数据库。

♡

证明

由于查询者可以进行 2^n 次子集查询，那么他可以得到 2^n 个响应, 记响应总体为 $r(S)$
遍历所有 $c \in \{0,1\}^n$，剔除掉所有的 $|\sum c_i - r(S)| > E$，剩下的 c 就是可能的情况。
显然，真正的数据库 d 不会被剔除，由是我们考察两个集合 $I_0 = \{i|d_i = 0\}, I_1 = \{i|d_i = 1\}$

$$\begin{cases} |\sum_{i\in I_0} c_i - r(I_0)| \le E \\ |\sum_{i\in I_0} d_i - r(I_0)| \le E \end{cases}$$

由三角不等式得 $|\sum_{i\in I_0}(d_i - c_i)| \le 2E$, 对于 I_1 同理，故总偏移量为 $4E$
简要来说，这证明了该算法就是全然不隐私的。当然一个现实的问题是，查询者不可能进行指数量级的查询，这是一种低效的攻击，下面的攻击更高效，更具有现实意义。

定理 1.2 (Dinur-Nissim attack)

如果查询者被允许进行 O(n) 次子集查询，管理者加入噪声界限 $E = O(\alpha\sqrt{n})$, 那么根据响应结果，查询者可以重构出偏移量为 $O(\alpha^2)$ 的数据库。

♡

证明 具体的证明是困难的，这里只给出一些直觉上的分析

由于查询者只能进行 O(n) 次随机且均匀子集查询，仿照上文的 $c \in \{0,1\}^n$，利用线性回归剔除掉不合理的值。此处我们做一个转换，令 $c, d, S \in \{-1,+1\}^n$...

假设可能的 c(candidates) 与真实的数据库 d 在 $\Omega(n)$ 的数据上是不重合的, 考察 c 与 d 的差异到底有多大，$(c-d)\cdot S = \sum (c_i - d_i)\cdot S_i$

1. 如果 $c_i = d_i$，那么 $(c-d)\cdot S$，不影响结果。
2. 如果 $c_i \neq d_i$，那么

$$(c-d)\cdot S = \begin{cases} 2 & \text{w.p.}\frac{1}{2} \\ -2 & \text{w.p.}\frac{1}{2} \end{cases}$$

所以 $(\sum (c_i - d_i)\cdot S_i) \sim Bin(\Omega(n), \frac{1}{2})$，放缩移位后它取得 $\Omega(\sqrt{n})$ 的概率还是很高的。

由于管理者加入的噪声被限制在 $E = O(\sqrt{n})$，这使得不少 c 可以被排除。将偏移量太远的 c 排出后，我们对可能情况取一个并集，这样就十分接近真实情况了。严谨的数学证明还是参照 Dwork书里定理 8.2 的证明吧。

让我们回顾一下本小节，第一种攻击需要 2^n 次的查询以消除 $O(n)$ 的噪音, 第二种攻击需要 $\Omega(n)$ 次的查询以消除 $O(\sqrt{n})$ 的噪音. 而差分隐私允许在 $O(\sqrt{n})$ 的噪声条件下，进行 $O(n)$ 次查询，一般是通过拉普拉斯或者高斯机制去实现。其实隐私上限还是有”缩水“的空间的。比方说，我们查询的数据量远小于 n, 例如仅请求 $m << n$ 次的查询，那么相应的，管理者只需要加入 $O(\sqrt{m})$ 噪音就可以了。

正式的讨论，还是留给我们进入差分隐私的时候再说吧。

1.4 悬赏

到目前为止，所有的讨论看起来都像是在纸上谈兵。你是否在想，在现实生活中，这些攻击是不是真的可以实现呢？Aircloak 公司表示，它可以一战。

2017 年，Aircloak 公司发布了新系统 Diffix，它是一个数据库查询系统。不过，它的工作方式，看上去完全违反了上一节为了安全加以的诸多限制，首先它允许不限次数的查询，其次，它加入的噪声大小远远小于差分隐私保护机制起效所需的噪声量。尽管如此，他们还是大大咧咧的开出了 5000 美元的悬赏：给出一个攻击 Diffix 的方法，并重构数据库。

与以前类似，数据分析人员可以进行子集查询。主要的区别是，随着每次查询进行，噪声的大小呈条件集本身大小的平方根形式增加。其他防止攻击的方法包括限制计数量，调整极端值并且，禁用了"OR" 操作符。

回顾我们此前攻击做法，Dinur-Nissim 要求进行随机式地查询。这里先是获取随机的查询子集总体，然后利用一些条件集使得查询子集的特征得以显现。即便我们忽略不得使用"OR" 操作符的限制，看上去为了使得 k 个查询子集得以区分，我们似乎不得不加入 k 个条件集 (诸如此类的其他要求)。

Cohen 和 Nissim 天才式地绕过了这些限制，他们提出一个有效的新思路：相比与之前先选择查询子集再加入条件集使之凸显，他们在查询时就加入了极少量的条件。经历了精心设计的查询条件后，得到响应数据的随机性足以重构整个数据库。

具体方法解释如下。每个数据库中的用户对应唯一的 client-id, 问题是是否有一个函数以 client-id 作为参数，并将它“足够随机地”地隐含在查询集中？他们使用由四个变量指定的函数：mult、exp、d、pred。前三个变量是数字，第四个变量是一个真伪判断 (T/F),。即 $(mult * client - id)^{exp}$ 这个表达式的 d 位是否满足某个条件？它的 SQL 实现如下：

```
SELECT count(clientId)
FROM loans
WHERE floor(100 * ((clientId * 2)^0.7) + 0.5) = floor(100 * ((clientId * 2)^0.7))
AND clientId BETWEEN 2000 and 3000
AND loanStatus = 'C'
```

最后一位 loanStatus 就是他们尝试攻击的隐私位，他们攻击的用户范围是 2000 到 3000 之间的用户。如上所示，他们只是使用了一个条件，这给每次查询得到响应数据里，只带来了常数量的噪声，比上文所需的 $O(\sqrt{n})$ 噪声要远远小得多。最后，他们稍作调整除杂就 100% 地重构了整个数据库，拿走了奖金！详情参照此论文

第二章 Introduction to Differential Privacy

内容提要

- ❑ 差分隐私定义
- ❑ 相关性质

2.1 差分隐私定义

为了精准描述隐私程度，我们引入差分隐私的概念，这有时也叫中心化差分隐私模型 (central Differential Privacy) 或者置信管理者模型 (Trusted Curator)

我们假定有 n 个个体，从 $X_1 \sim X_n$, 他们将各自的数据传递给置信管理者。管理者将数据利用算法 M 输出一个公开结果。所谓差分隐私就是这个算法 M 的性质：单个个体的数据不会对算法整体输出造成太大的影响。

定义 2.1

给定算法 $M : \mathrm{X}^n \to \mathrm{Y}$, 现在我们考察 X^n 中任意两个数据集 X, X', 它们两个仅在某一条目上不一致，我们称其为“邻近数据集”。由是，我们称如下算法算法 M 是 ϵ–(纯) 差分隐私的, 如果：

对于任意邻近数据集 $X, X' \in \mathrm{X}^n$, 以及任意 $T \in \mathrm{Y}$, 都满足:

$$Pr[M(X) \in T] \leq e^{\epsilon} Pr[M(X') \in T] \tag{2.1}$$

2.2 相关性质与解释

1. ϵ 越小，意味着隐私性越强
2. 隐私性与准确性始终是矛盾的一对命题。ϵ 的取值也大有讲究，一般在在 $0.1 \sim 5$ 为宜
3. 为什么采用 e^{ϵ}? 好吧，当 ϵ 足够小的时候，你不妨考虑泰勒展开，$e^{\epsilon} \approx \epsilon + 1$, 这个看上去自然多了。当然，这种 e 指数在后续考虑“群体差分隐私”的时候还是十分便捷的，利用两个同底指数相乘化简为指数相加可以轻松化简！...

结论

让我们回顾一下：差分隐私意味着什么？简单地重复这个定义：无论单个个体包含或不包含在数据集内，算法输出结果的概率都是相近的。这意味着差分隐私可以做一些事，也说明了它所不能做的事。

首先，它可以阻止我们此前提到的诸多攻击类型。比方说 linkage attack——通过比对包含相同条目的几个数据库从而识别隐私。它还可以防止重建攻击，在某种意义上“匹配”了 Dinur-Nissim 攻击中显示的噪声边界，我们将在后续章节中对此进行量化分析。

不过，差分隐私并不妨碍对个体做出的推断。换句话说：差分隐私并不妨碍统计数据和机器学习。以经典的“吸烟会导致癌症”为例。假设一个吸烟的人正在权衡他们选择参加医学研究的选择，该研究调查吸烟是否会导致癌症。他们知道，这项研究的积极结果将对他们有害，因为它会导致他们的保险费上升。他们也知道，这项研究是通过不同的私人方式进行的，所以他们选择参与研究，他们也知道他们的隐私将会得到尊重。不幸的是，研究表明吸烟确实会导致癌症！这是一种侵犯隐私的行为，对吧？不过，差分隐私确保研究结果不会受到他们参与的显著影响。换句话说，不管他们是否参与其中，结果无论如何都会显现出来。

差异隐私也不适用于目标是识别特定个人的情况，而这与该定义是截然相反的。举个当下的例子，尽管人们强烈要求追踪 COVID-19 感染者的行踪，但目前还不清楚如何利用差异隐私来促进个人层面的接触者追踪。这需要考虑一个特定个体在哪里，以及他们与哪些特定个体互动过。另一方面，如果许多检测呈阳性的人都参加了同一项活动，那么就有可能促进总体水平的跟踪。

2.3 差分隐私的优势与局限性

根据上文分析，我们可以总结差分隐私所具有的六大优势如下:

1. 对于任意先验知识风险的防范性。在上文中我们提到，由于数据分析师的先验知识背景非常多样，因此数据分析师的辅助信息也会将一些不破坏隐私的查询变得破坏隐私。但是差分隐私与先验知识完全无关，可以抵御所有依靠辅助信息所进行的攻击。
2. 对于链式攻击的防范。将隐私攻击手段拆成多个问题，或者对数据集进行多次差分都不会泄露隐私。
3. 传递性带来的隐私闭包。由于差分隐私的传递特性，数据分析师在没有其他有关私有数据库的知识的情况下，其所做的任何进一步处理也具有差分隐私特性。也就是说，数据分析师不能仅仅通过坐在角落里思考算法的输出，得到任何会泄露个人隐私的结论。
4. 对隐私损失的量化。差分隐私利用 (ϵ,δ) 的大小对隐私损失进行度量。这就允许在不同技术之间进行比较：对于固定的隐私损失 (ϵ,δ)，哪种技术可以提供更好的准确性？为了达到固定的精度，哪种技术可以提供更好的隐私保护？
5. 对团体数据隐私的保护。差分隐私对于团体数据（例如家庭数据）所带来的隐私损失也进行了分析与控制。
6. 对于数据分析 Pipeline 的组合（Composition）隐私进行度量。一般而言，数据分析师会用多种随机算法进行分析，而不同的随机算法会带来不同的隐私损失。例如，深度学习模型训练的过程中，每一个 Epoch 都会发布一个中间模型，随着获取的中间模型越来越多，我们需要度量所有中间模型所带来的隐私损失；又例如一个可视分析系统往往有多个视图，每个视图的底层算法也不一样（如 t-sne 和热力图）。对于一个由多种随机算法组合系统如何度量隐私，以及如何设计每一个模块的 (ϵ,δ) 使得整个系统的隐私损失最小，差分隐私给出了理论思路，能够通过多个简单差分隐私模块设计和分析复杂的差分隐私算法。

但是，差分隐私不是万能的，也有其局限性。

正如上文在“吸烟引起癌症”示例中看到的，差分隐私不能保证参与人无条件不受伤害，换句话说，差异性隐私并不能保证人们认为属于自己的特性不因他人接受的调查所泄露，它只是确保不会透露自己参与了数据分析，也不会披露个体参与数据分析的任何细节。从群体调查中得出的统计结论很有可能反映出个体的统计信息，例如，通过他人数据训练的算法推荐给用户的视频往往是符合用户独特品味的，但这并不表示存在隐私权的侵害，因为用户数据甚至可能没有参加过模型训练。也就是说，差分隐私确保无论个人是否参加数据分析，都将以非常相似的概率符合统计结论性的结果。而如果分析结果告诉我们，某些特定的私有属性与公开可观察的属性密切相关，这并不违反差分隐私，因为这种相同的相关性将以几乎相同的概率被独立观察到，无论任意个体是否出现在分析所用的数据库中。

第三章 Lapalce Mechanism

内容提要

- ❏ l_1 灵敏度
- ❏ Laplace 分布
- ❏ Laplace Mechanism
- ❏ Counting queries & Histograms

3.1 基础知识

定义 3.1 (l_1 灵敏度) 我们令 $f:\mathrm{X}^n \to \mathrm{R}^k$，$X, X'$ 为邻近数据集，定义 l_1 灵敏度为：

$$\Delta(f) = max_{X,X'}||f(X) - f(X')||_1 \tag{3.1}$$

注 下面以 Δ 简代 l_1 灵敏度。差分隐私总是试图掩盖数据集里单个样本的贡献。因此，考量“函数在仅仅改变一处样本时变化的上界”看上去是十分符合直觉的

举一个简单的例子，考虑函数 $f(x) = \frac{1}{n}\sum_{i=1}^{n} X_i$, 其中 $X_i \in \{0,1\}$, 很容易推知 $\delta = \frac{1}{n}$

定理 3.1 (拉普拉斯分布) 位置参数为 0，尺度参数为 b 的拉普拉斯分布概率密度函数为

$$p(x) = \frac{1}{2b}exp(-\frac{|x|}{b})$$

它的方差是 $2b^2$,(图像上相当于 x>0 时的指数分布函数沿着 y 轴对称得到)

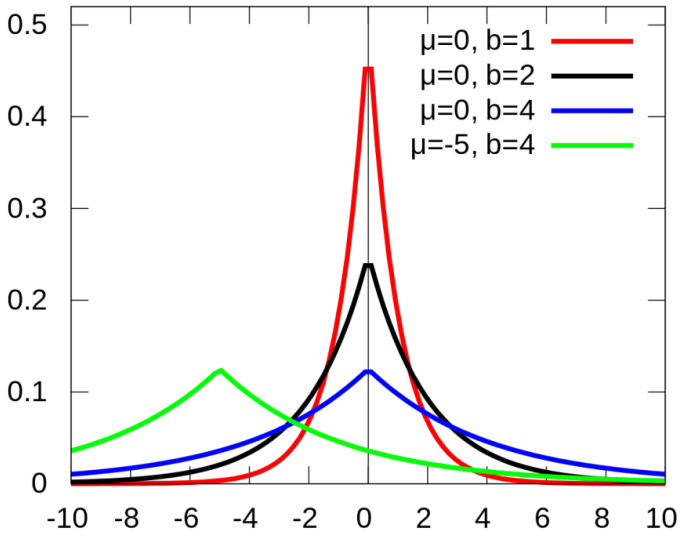

图 3.1: 拉普拉斯分布示意图

3.2 拉普拉斯机制定义

定义 3.2 (Laplace Mechanism)

令 $f:\mathrm{X}^n \to \mathrm{R}^k$, 拉普拉斯噪声机制如下：

$$M(x) = f(x) + (Y_1, Y_2, \ldots, Y_k), \text{其中 } Y_i \text{ 是属于 Laplace}(\Delta/\epsilon) \text{ 的随机变量} \tag{3.2}$$

例如 $f(x) = \frac{1}{n}\sum_{i=1}^{n} X_i$

由上文我们得到的 $\delta = \frac{1}{n}$，因此拉普拉斯机制下，$\tilde{p} = f(X) + Y$, 其中 $Y \sim Laplace(\frac{1}{n\epsilon})$

回顾前文定义的 $p = f(x)$，由于此处拉普拉斯函数位置参数为 0，所以 $E[\tilde{p}] = p$,

而 $Var[\tilde{p}] = Var[Y] = O(\frac{1}{n^2\epsilon^2})$, 由切比雪夫不等式做出估计 $|\tilde{p} - p| \leq O(\frac{1}{n\epsilon})$(在此范围内概率极高)

定理 3.2

拉普拉斯机制是 $\epsilon - DP$ 的。

证明

我们令 X, Y 邻近数据集，令 $p_X(z), p_Y(z)$ 为在某处 $z \in R^k$时$M(X), M(Y)$ 的概率密度函数，只要证明两者比值上界为 e^ϵ 即可。

$$\begin{aligned}
\frac{p_X(z)}{p_Y(z)} =& \frac{\prod_{i=1}^{n} exp(-\frac{\epsilon|f(X_i - z)|}{\Delta})}{\prod_{i=1}^{n} exp(-\frac{\epsilon|f(Y_i - z)|}{\Delta})} \\
=& \prod_{i=1}^{n} exp(-\frac{\epsilon}{\Delta}(|f(X_i - z)| - |f(Y_i - z)|)) \\
\leq& \prod_{i=1}^{n} exp(-\frac{\epsilon}{\Delta}|f(X_i) - f(Y_i|)) \\
=& exp(-\frac{\epsilon}{\Delta}\sum_{i=1}^{n}|f(X_i) - f(Y_i)|) \\
=& exp(-\frac{\epsilon}{\Delta}||f(X) - f(Y)||_1) \\
\leq& exp(\epsilon)
\end{aligned}$$

第一处不等号使用了三角不等式放缩，第二处是利用了 l_1 灵敏度的定义。

3.3 应用

3.3.1 Counting Queries

来看一下拉普拉斯机制在一些场景中的应用。我们可以问这样一个问题："数据集中有多少人具有属性 P?"与之前分析非常相似。每个个体都会有一个小的 $X_i \in \{0, 1\}$，表示它们是否具有 P，我们考虑的函数 f 是它们的和。易得，l_1 灵敏度为 1，因此该统计量 $\epsilon - DP$ 值为 $f(X) + Laplace(1/\epsilon)$。这将导致 $O(1/\epsilon)$ 的查询错误，(与数据库的大小无关)。

如果我们想回答很多查询呢？假设我们有 k 个计数查询 $f = (f_1, \ldots, f_k)$，这些都是预先指定的。我们将输出向量 $f(X)+Y$,其中 Y_i 是 i.i.d. 拉普拉斯分布的随机变量。但是我们对 Y_i 应该使用什么尺度参数呢？每个单独的计数查询f_j 的敏感性为 1, 但是我们使用相同的数据集来回答所有的查询, 因此更改单个个体可能会同时影响多个查询的结果。例如，考虑两个个体的交换：一个不满足任何属性，另一个满足任何每个属性。这个交换将使每个查询的结果改变 1, 因此总体 l_1 敏感度为 k。让我们用数学方法来分析一下。由于 $f(X) = \sum P(f1(X_i), \ldots, fk(X_i))$, 如果相邻的数据集 X 和 Y 不同，一个包含 x，另一个包含 y，那么灵敏度可以被写成 $\sum_j |f_j(x) - f_j(y)|$。它的上界可以确定：$\sum_j |f_j(x) - f_j(y)| \leq \sum_j 1 = k$。

有了这个灵敏度约束的 $\Delta = k$，我们可以在每个坐标中添加 $Y_i \sim Laplace(k/\epsilon)$ 噪声，以 $O(k/\epsilon)$ 级的误差回答每个计数查询。

总结一下。首先，这种回答 k 个计数查询的方法要求我们预先指定所有的查询——换句话说，这是一个非自适应 (non-adaptive) 的设置。我们稍后将看到，在自适应设置中，类似的保证是可以实现的，其中查询的选择可能取决于以前的查询。其次，让我们将其与之前讨论的 Dinur-Nissim 攻击进行比较。

如果查询者进行 $\Omega(n)$ 计数查询，由管理者添加 $O(\sqrt{n})$ 的噪声来防御，查询者可以重建数据库 (BNP)。

形成比较的，如果查询者进行 $O(n)$ 计数查询，并且管理者添加了 $O(n/\epsilon)$ 的噪声，那么隐私就得到了保护。

这似乎是两个结果中的一个巨大差距. 是否有更强大的攻击，让对手在更多的噪音下成功？或者我们可以减少噪音，同时保持隐私吗？幸运的是，后者是可行的，我们接下来会进一步讨论。

3.3.2 Histogram

另一种查询类型是 histogram queries。对于计数查询，改变单个个体可能会同时影响每个查询的结果。但是 histogram queries 不是这样。假设数据集中的每个人都有一些分类特征，例如，某人的年龄。我们想回答诸如“数据集中有多少人 X 岁了”这样的问题?”虽然这类似于计数查询示例，但这里每个人年龄显然只有一种。定义函数 $f : (f_0, \ldots, f_{k-1})$, 其中 f_i 问有多少人是 i 岁。这个函数的敏感度是 2：改变任何一个人的年龄都会导致一个计数减少，另一个计数增加。所以最终结果是 $f(X) + Y$, 其中 $Y_i \sim Laplace(2/\epsilon)$.

这导致了多少误差？与之前一样，我们观察到任何单个计数都会有 $O(1/\epsilon)$ 的误差。我们使用如下定理3.3来考量整体计数的误差。

定理 3.3

如果 $Y \sim Lapalce(b)$，那么

$$Pr[|Y| \geq tb] = exp(-t)$$

♡

现在，对于直方图里第 i 个 bin，计数中的误差正好是 Y_i，我们有

$$Pr[|Y_i| \geq 2\log(k/\beta)/\epsilon] \leq \beta/k$$

换句话说：误差的大小只与 bin 的数量成对数关系，而不是当我们的计数查询的线性关系

3.4 差分隐私的性质

3.4.1 传递性

差分隐私是一个强大的隐私保护工具，但是要实现强差分隐私约束（即 ϵ,δ→0）往往需要增加大量噪声。一般而言，数据的分析包括多个环节，如果在每个环节上都满足差分隐私约束，势必会令数据分析结果失真。差分隐私的传递性给出了一个强有力的断言：只要在数据分析的任何一个环节，随机算法 M 满足 (ϵ,0)-DP，那么在仅用该环节的结果作为输入的任意之后的数据处理过程都满足 (ϵ,0)-DP，

定理 3.4 (Post-Processing)

令 $M : X^n \to Y$ 是 $\epsilon - DP$ 的，令 $F : Y \to Z$ 为任意映射，那么 $F \circ M$ 也是 $\epsilon - DP$ 的

♡

证明

对邻进数据集中任意一对 $x, y, \|x - y\|_1 \leq 1$, 取 $S \subseteq Y$, 令 $T = \{r \in R : f(r) \subseteq S\}$

$$\begin{aligned} Pr(f(M(x)) \in S) &= Pr(M(x) \in T) \\ &\leq exp(\epsilon) Pr(M(y) \in T) \\ &= exp(\epsilon) Pr(f(M(y)) \in S) \end{aligned}$$

3.4.2 群差分隐私

定理 3.5 (Group Privacy)

令 $M : X^n \to Y$ 是 $\epsilon - DP$ 的, 假定两个数据集 X, X' 在 k 个位置上的数据都不同，那么对于所有的 $T \subseteq Y$, 都有

$$Pr(M(X) \in T) \leq exp(k\epsilon)Pr(M(X') \in T)$$

♡

证明 构造 $X^{(0)} = X, \sim X^{(k)} = X'$ 上进行插值，构造一系列临近数据集即可证明

$$\begin{aligned} Pr(M(X) \in T) &= Pr(M(X^{(0)}) \in T) \\ &\leq exp(\epsilon)Pr(M(X^{(1)}) \in T) \\ &\leq exp(2\epsilon)Pr(M(X^{(2)}) \in T) \\ &\leq \cdots \\ &\leq exp(k\epsilon)Pr(M(X^{(k)}) \in T) \end{aligned}$$

3.4.3 Basic Composition

定理 3.6 (Composition)

$M = (M_1, M_2, \ldots, M_k)$ 是一系列 $\epsilon - DP$ 函数 (选取是顺序的，自适应的)，那么最终 M 是 $k\epsilon - DP$ 的

♡

证明 考虑临近数据集 X, X', 对于函数顺序输出的 $y = (y_1, \ldots, y_k)$, 我们有：

$$\begin{aligned} \frac{Pr(M(X) = y)}{Pr(M(X') = y)} &= \prod_{i=1}^{k} \frac{Pr(M_i(X) = y_i) | M_1(X) = y_1, \ldots, M_i(X) = y_i}{Pr(M_i(X') = y_i) | M_1(X') = y_1, \ldots, M_i(X') = y_i} \\ &\leq \prod_{i=1}^{k} exp(\epsilon) \\ &= exp(k\epsilon) \end{aligned}$$

第四章 Approximate Differential Privacy

内容提要

- ❏ $(\epsilon, \delta) - DP$
- ❏ Privacy loss
- ❏ Gausssian Mechanism

4.1 两个定义

松弛化差分隐私将拥有稍弱的隐私保护，但允许我们添加显著更少的噪音来实现它

定义 4.1 (松弛 DP)

我们称如下算法 M 是 $(\epsilon, \delta) - DP$, 如果对于任意邻近数据集 $X, X' \in \mathrm{X}^n$, 以及任意 $T \in \mathrm{Y}$, 都满足:

$$Pr[M(X) \in T] \leq e^{\epsilon} Pr[M(X') \in T] + \delta \tag{4.1}$$

♣

定义 4.2 (Privacy Loss)

Y,Z 为两个随机变量。隐私损失随机变量定义为 $\mathcal{L}_{Y||Z} = \ln(\frac{Pr(Y=t)}{Pr(Z=t)})$

♣

4.2 Gaussian Mechanism

定义 4.3 (l_2 灵敏度)

我们令 $f : \mathrm{X}^n \to \mathrm{R}^k$，$X, X'$ 为邻近数据集，定义 l_2 灵敏度为:

$$\Delta_2^{(f)} = max_{X,X'} ||f(X) - f(X')||_2 \tag{4.2}$$

♣

定义 4.4 (Gaussian distribution)

高斯分布 $N(\mu, \sigma^2)$ 的概率密度函数是:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{4.3}$$

一个有用的性质: $X, Y \sim N(0,1)(i.i.d.)$, 那么 $aX + bY \sim N(0, a^2 + b^2)$

♣

定义 4.5 (Gaussian Mechanism)

令 $f : \mathcal{X}^n \to \mathcal{R}^k$，$\Delta_2^{(f)}$ 是 f 的 l_2 灵敏度，高斯机制 M_f 是一个随机函数，其输出是: $M(X) = f(X) + Y$, 其中 $Y_i \sim N(0, 2\ln(1.25/\delta)\Delta_2^2/\epsilon^2)$

♣

定理 4.1

Gaussian Mechanism 是 $(\epsilon, \delta) - DP$ 的

♡

定理 4.2

X, X' 是 $\mathcal{X}^n$ 上的临近数据集，对于 $f : \mathcal{X}^n \to R^k, M(Y) = f(Y) + N(0, \sigma^2 I)$, 隐私损失

$$\mathcal{L}_{M(X)||M(X')} \sim N(\frac{\|f(X) - f(X')\|_2^2}{2\sigma^2}, \frac{\|f(X) - f(X')\|_2^2}{\sigma^2})$$

♡

4.3 松弛差分隐私的性质

4.3.1 传递性

定理 4.3 (Post-Processing)

令 $M : X^n \to Y$ 是 $(\epsilon, \delta) - DP$ 的，令 $F : Y \to Z$ 为任意映射，那么 $F \circ M$ 也是 $(\epsilon, \delta) - DP$ 的

♡

4.3.2 群差分隐私

定理 4.4 (Group privacy)

令 $M : X^n \to Y$ 是 $(\epsilon, \delta) - DP$ 的, 假定两个数据集 X, X' 在 k 个位置上的数据都不同，那么对于所有的 $T \subseteq Y$, 都有

$$Pr(M(X) \in T) \leq e^{k\epsilon} Pr(M(X') \in T) + ke^{(k-1)\epsilon}\delta$$

♡

4.3.3 组合

定理 4.5 (Composition)

令 $M_1 : X^n \to Y$ 是 $(\epsilon_1, \delta_1) - DP$ 的, 令 $M_2 : Y \to Z$ 是 $(\epsilon_2, \delta_2) - DP$ 的, 那么 $M_2 \circ M_1$ 也是 $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2) - DP$ 的

♡