论文阅读笔记：K-Anonymity

纳文琪

2019 年 4 月 22 日

1 Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data[1]

1.1 引言

最近研究表明，在大数据时代简单地匿名化数据集已无法再抵御针对隐私的攻击。保护隐私最直接的办法就是移除数据集中的 ID，然而，事实表明这并不奏效。本质上，一个 ID 表示的是被描述对象的一组特征，但是实际上，我们依靠一个人的多种特征而不是 ID 来识别这个人。

隐私研究的两个方面 隐私研究集中在两个方面：内容隐私（content privary）和行为隐私（interaction privacy）。

内容隐私 攻击者根据受害者的一些知识背景从匿名或加密的数据集中识别受害者身份。

行为隐私 攻击者更关注受害者的行为。

现代隐私研究主要关注两个方面：数据聚类（data clustering）和隐私框架（privacy framework）。

1.2 基础知识

隐私参与者 包括：数据生产者、使用者、管理者、攻击者。

数据操作 包括：收集、清洗（anonymizing）、交换（communicating）。

数据属性的类型 包括：

显示标识（explicit identifier**）** 比如身份证号、特定学校里的学号等；

准标识（quasi-identifier**）** 通过关联其他数据集就可以确定用户的属性，比如性别、年龄等。我们一般将一个记录里面的所有准标识字段称为“qid”。拥有相同 qid 的值的一组记录被称为一个等价类（equivalent class）。

敏感信息（sensitive information**）** 用户希望保护的那部分数据；

其他 用户的其他信息。

1.3 隐私研究的成果

1.3.1 数据聚类方面

数据聚类方面的研究成果主要包括：k-anonymity、l-diversity、t-closeness 等。

k-anonymity 首先，为保护隐私，我们必须将数据的 ID 全部移除，这样才能避免特定用户被识别。然而即便所有 ID 被移除，攻击者还是有可能通过诸如链接外部数据库等方式，根据 qid 来识别用户。k-anonymity 方法的基本原则是：确保含有相同 qid 组数据的记录在数据集中至少出现 k 次，也就是说，每个等价类至少有 k 个记录。这样就可以使得攻击者通过 qid 识别特定用户的概率变为 $\frac{1}{k}$，当存在一个很大的 k 值的时候，会对用户的识别产生一个很大的信息损失，从而达到隐私保护的作用。k-anonymity 方法主要是用于处理准标识字段上的隐私保护问题, 但不能处理敏感数据。攻

击者可能使用同质攻击（homogeneity attach）或背景知识攻击（backgroud knowledge attack）来破解。

l-diversity　l-diversity 方法可以克服 k-anonymity 方法的缺点。它要求数据集“对每一个 qid 的值，确保敏感数据字段至少有 l 个不同的值”。为了实现它，我们需要增大（还是减小?）敏感信息字段的颗粒度或增加噪声。某些特殊的时候，l-diversity 会起到反作用，他会释放更多的信息增益给攻击者。

t-closeness　可以修复 l-diversity 的脆弱性。它的基本思想是：对于任何一个等价类，保证它的值的分布被限定在 t 范围内。

1.3.2　隐私框架方面

微分隐私（differential privacy**）**　在了解用户背景知识的情况下，攻击者可能会通过多次进行统计查询来获得期望的信息。防范策略是：对两个差别很小的数据集进行查询，其结果差别也应该很小，这样就可以限制攻击者获得的信息增益。

微分可识别性（differential identifiability**）**

成员隐私（membership privacy**）**

1.4　隐私研究的学科

1.5　隐私研究的数学描述

匿名系统　是一个映射函数：$F = X \to Y$，$X = \{X_1, X_2, ..., X_n\}$ 是原始数据，$Y = \{Y_1, Y_2, ..., Y_m\}$ 是系统的输出，对于攻击者，其目的是建立一个映射：$G : Y \to \hat{X}$，尽可能地从输出还原原始数据。

隐私保护系统的两个目的　被描述为 utility 和 privacy，这也是匿名系统 F 的两个关键指标。

utility　使用 distorion D 来度量，抽象地表示为：

$$D = \lambda(X;Y) \tag{1}$$

D 有很多种度量方法，例如，可以使用均方来表示。

privacy　使用 leakage L 来度量，抽象地表示为：

$$L = \lambda(X;\hat{X}) \tag{2}$$

L 通常使用互信息来度量，即：

$$L = I(X,\hat{X}) \tag{3}$$

给定两个阈值 D_0 和 L_0，匿名系统可作为一个优化问题：

$$\begin{aligned} &optimize F \\ &\quad s.t.\ D \le D_0 \\ &\quad\quad L \le L_0 \end{aligned} \tag{4}$$

1.6 隐私度量（Privacy Measurements）

隐私的度量至今都没有太清晰的方法。现有以下几种度量方法：

1.6.1 相对度量（Relative Measurement）

其思想是：首先给定一个标准（benchmark），再度量研究对象到此标准的距离。比较流行的距离计算方式是 Kullback-Leibler 距离（相对熵）。

KL 距离是基于一阶统计的度量方法，而二阶方法可以度量得更加精确。

1.6.2 信息论度量（Information Theoretic Measurement）

对于一个投票系统，定义三个随机变量 V、S、E，分别表示投票者所投的票、来自投票系统以外的信息、攻击者由投票系统中获得的信息。

完美隐私（perfectly privacy） 定义为：在 S 条件下，V 和 E 独立。即：$p_{V|S}(v;s) = p_{V|S,E}(v;s,e)$。

隐私损失总量（amount of privacy loss） 定义为：

$$L = \max(H(V|S) - H(V|S,E)) \tag{5}$$

度量隐私可能纰漏的程度。

1.6.3 Unicity Measure

1.7 隐私数学模型

1.7.1 k-anonymity 模型

定义数据表 $T = \{t_1, t_2, ..., t_n\}$ 是数据行的集合，$A = \{A_1, A_2, ..., A_n\}$ 是数据的属性集，$t_i[A_j]$ 表示元组 t_i 的属性 A_j 的值，$C = \{C_1, C_2, ..., C_k\} \subseteq A$ 是子属性集。定义 $T[C] = \{t[C_1], t[C_2], ..., t[C_k]\}$ 是 t 在 C 上的映射，QI 为所有准标识符的集合。

我们说一个表 T 满足 k-anonymity，如果它满足，对于每一个元组 $t \in T$ 都存在 $k-1$ 个其他的元组 $t_{i_1}, t_{i_2}, ..., t_{ik-1} \in T$ 使得 $t[C] = t_{i_1}[C] = t_{i_2}[C] = ... = t_{i_{k-1}}[C], C \subseteq QI$。也就是说，任何一组具有相同属性值的准标识符在表中至少出现 k 次。

1.7.2 l-diversity 模型

1.7.3 t-closeness 模型

1.7.4 Differential Privacy 框架

2 k-anonymity: A model for protecting privacy [2]

2.1 背景

人们希望大量数据用于研究和分析，但同时又保证不泄露隐私，即同时满足“数据利用 (utility)”和“保护隐私 (privacy)”两个要求。通常，人们在发

布数据时，会将数据集的标识符（如姓名、身份证号等）删去，从而使得数据集中的个体 (individual) 不能够被攻击者“重新识别 (re-identify)”。然后，在很多情况下，利用非标识符字段仍然可以重新识别出个体。例如，研究显示，87% 的美国人可以通过邮编、性别和生日被识别出来。k-anonymity 提供了一种避免此问题的框架。

2.2 定义

纰漏（disclosure**）** 指的是数据明显地或通过推理被意外地泄露。

纰漏控制（disclosure control**）** 的目的是去识别或限制发布数据中的纰漏，也就是说，确保发布数据具有充分的匿名性。

准标识符（Quasi-identifier**）** 如果数据表中的一组属性，可以通过与外部属性连接重新识别数据表中的个体，则称这组属性为准标识符。

准标识符的形式化定义 令 U 为数据个体的全集，$T(A_1, ..., A_n)$ 为数据表，存在两个映射 $f_c : U \to T$ 和 $f_g : T \to U'$，其中 $U' \subseteq U$。表 T 的准标识符 Q_T 是一个属性集 $\{A_1, ..., A_j\} \subseteq \{A_1, ..., A_n\}$，满足 $\exists p_i \in U$ 使得 $f_g(f_c(pi)[Q_T]) = p_i$。

2.3 模型

k-anonymity 有表 T 和它的准标识符 Q_T，当且仅当 $T[Q_T]$ 中的每一行在 $T[Q_T]$ 中至少出现 k 次时，我们说 T 满足 k-anonymity。

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

Figure 2 Example of *k*-anonymity, where *k*=2 and QI={*Race*, *Birth*, *Gender*, *ZIP*}

图 1:

2.4 攻击 k-anonymity

未排序匹配 (Unsorted matching) **攻击** 如果发布的表的数据的顺序固定，就可以根据这种固定顺序，关联多个发布的表来获取敏感信息。避免这种攻击很简单，只需要对数据进行乱序操作后再发布即可。

互补发布攻击（Complementary release attack） 通常情况下，我们不会将所有发布的字段都列为准标识符，这样的话，如果同意数据表被发布了多个版本，攻击者就可能会通过关联多个发布的数据表来攻击 k-anonymity。

时间攻击 (Temporal attack) 如果数据集是动态变化的，由于每个时间的数据表都可能不一样，而不同时间点之间的数据表没有规则限制，因此攻击者有可能通过关联两个不同时间点数据集来识别数据表中的个体，从而实现对 k-anonymity 的攻击。两个不同时间点的数据表就与上面所讲的两个不同版本的数据集类似。

3 Protecting respondents identities in microdata release [3]

本文提出了一种实现 k-anonymity 的泛化算法。

3.1 **泛化数据** (Generalizing data)

3.1.1 **泛化关系** (Generalization relationships)

泛化关系 是关于全体值域的集合 Dom 上的一个偏序，记作 $\leq_D$。它满足两个条件：

- $\vee D_i, D_j, D_z \in Dom : D_i \leq_D D_j, D_i \leq_D D_z \Rightarrow D_j \leq_D D_z \vee D_z \leq_D D_j$ (每个 D_i 最多只有一个直接泛化域)
- 域中的所有值最终都被泛化到一个单一值（单一最大值）

域泛化结构 (domain generalization hierarchy, DGH_D 由泛化关系得到的一个全序结构。

值泛化关系 (value generalization relationships) 是与 DGR 类似的，关于某个值域 D 下的所有值的一个偏序，记作 $\leq_V$

值泛化结构 (value generalization hierarchy, VGH_D) 与 DGH_D 类似，可以用树来表示，其叶子节点就是域 D 中的值。

值泛化关系和值泛化结构可以看做是域泛化关系和域泛化结构更低层次的抽象。其示例如图 2 。

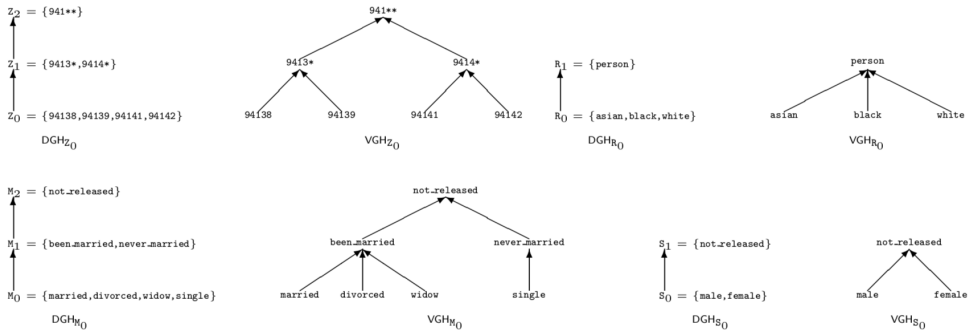

Figure 2: Examples of domain and value generalization hierarchies

图 2:

DGH **的格表示**　给定一个域元组 $DT =< D_1, ..., D_n >$，我们可以定义关于 DT 的域泛化结构为 $DGH_{DT} = DGH_{D_1} \times ... \times DGH_{D_n}$，即 DT 的 DGH 是其每个元素的 DGH 的笛卡尔积。由于每个 DGH_{D_i} 都是全序的，因此 DGH_{DT} 是一个关于 DT 的格 (lattice)，其最小值 (DT) 在底部，最大值在顶部。如图 3所示。

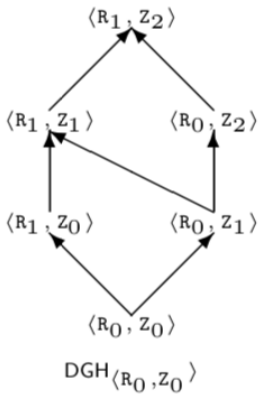

图 3: DGH 的格表示

泛化策略 (generalization strategy)　DGH_{DT} 格中的一组边及其节点组成的集合成为泛化策略。如图 4 所示。

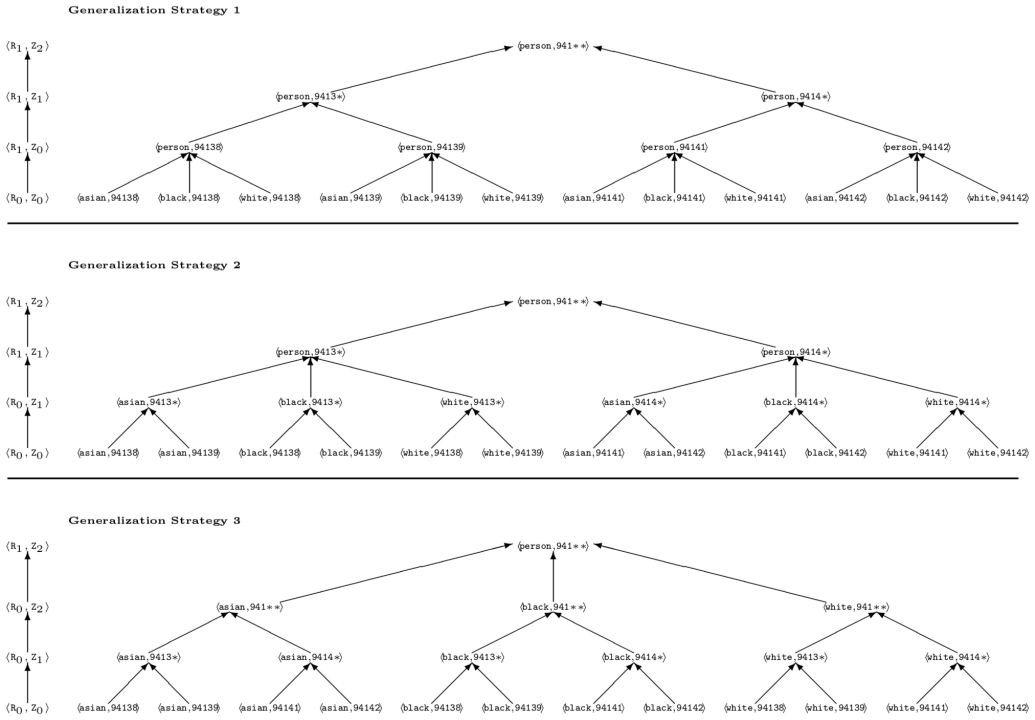

图 4: DGH 的格表示

3.1.2 Generalized table and minimal generalization

属性的值域 定义 $dom(A_i, T)$ 表示属性 A_i 在 T 中的值域。

泛化表 (Generalized Table) 令 $T_i(A_1, ..., A_n)$ 和 $T_j(A_1, ..., A_n)$ 是具有相同属性 (字段) 的两个表, 若满足以下条件则称 T_j 是 T_i 的泛化, 记作 $T_i \preceq T_j$:

- $|T_i| = |T_j|$(两个表中的数据量一致)
- $\forall A_z \in \{A_1, ..., A_n\} : dom(A_z, T_i) \leq dom(A_z, T_j)$ (每个属性满足一致的域泛化关系)
- 每个属性值满足一致的值泛化关系

距离向量 (distance vector) 两个表 T_i、T_j 满足 $T_i \preceq T_j$，从 T_i 到 T_j 的距离向量记作 $DV_{i,j} = [d_1, ..., d_n]$，其中 d_z 是 DGH_{D_z} 中 D_i 到 D_j 的距离。

距离向量的偏序关系　两个距离向量 DV、DV'，如果所有的 $d_i \leq d'_i$，则有 $DV \leq DV'$。因此，域泛化结构可以看做是距离向量的格，如图 5 所示。

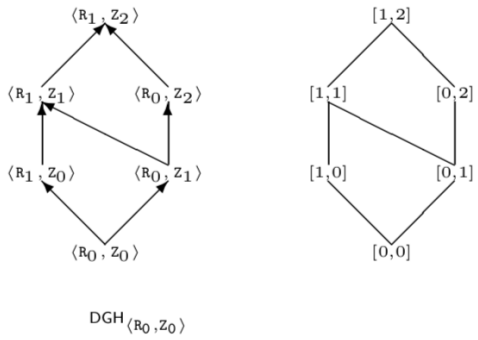

图 5: DGH 的距离向量格表示

k-minimal **泛化** (generalization)　两个表 T_i、T_j 满足 $T_i \preceq T_j$，如果 T_j 满足下列两个条件就称 T_j 是 T_i 的一个 k-minimal 泛化：

- T_j 满足 k-anonymity
- $\forall T_z : T_i \preceq T_z satisfies k-anonymity \Rightarrow \neg(DV_{i,z} \leq DV_{i,j})$(不存在任何一个比 T_j 小，而且满足 k-anonymity 的泛化表)。

3.2 Suppressing data

泛化的好处是可以在满足 k-anonymity 的条件下发布表中所有的数据条目，而 suppression 则通过移除某些条目来满足 k-anonymity，同时减少泛化所需的步骤。

带 suppression 的泛化表 (generalized table with suppression)　与泛化表相比，它不要求 $|T_i| = |T_j|$，而是要求 $|T_i| \geq |T_j|$，也就是说，泛化表的条目可以比原表少。

最小要求抑制 (minimal required suppression)　T_j 是 T_i 的满足 k-anonymity 的泛化，满足下列条件时，我们称对 T_j 执行了最小要求抑制 (也就是说，T_j

是满足 k-anonymity 的抑制中条目最多的):

$$iff\ \ \forall T_z : T_i \preceq T_j, DV_{i,z} = DV_{i,j}, T_z\ \ satisfies\ \ k-anonymity \Rightarrow |T_j| \geq |T_z|. \tag{6}$$

对任意一个表，若想要它满足 k-anonymity，最简单的办法是移除 QI 重复次数少于 k 的数据条目。这样并不是好的做法，我们最希望得到的是一个执行了最小要求抑制的泛化结果，尽可能保留数据条目。泛化和抑制应当交叉使用，而问题是：以损失数据进度为代价的泛化和以损失数据完整性为代价的抑制，哪种更好。

我们假设一个可接受的抑制值 MaxSup，当被移除的数据条目少于它时，我们认为这是可接受的，也就是说，此时抑制比泛化更好。这是由于抑制只影响一条数据，而泛化影响整个数据表的某个属性。

带 suppresion 的 k-minimal 泛化　两个表 T_i、T_j，满足 $T_i \preceq T_j$，若满足以下条件则称 T_j 是 T_i 的 k-minimal 泛化：

- T_j 满足执行最小要求抑制的 k-anonymity。
- $|T_i| - |T_j| \leq MaxSup.$（不需要执行比要求的更多的抑制操作）
- $\forall T_z : T_i \preceq T_z$ and satisfies conditions 1 and 2 $\Rightarrow \neg(DV_{i,z} < DV_{i,j})$（不存在满足以上条件，且比它 DV 更小的泛化）。

3.3 Computing a k-minimal generalization

定理 5.1　表 $T_i = PT[QI]$，DT 是其值域元组，T_i 的每一个 k 最小泛化都是 DGH_{DT} 中的泛化策略的局部最小泛化。

由定理 5.1 可知，可以这样计算一个 k 最小泛化：遍历每一个泛化策略，沿着 DT 到 DGH 的最大值的方向出发，直到找到一个泛化结果，是的其满足 k-anonymity 和抑制阈值为止。给定 D_i 由 DT 到 DGH 的顶部的高度为 h_i，DT 在 DGH 中的不同策略的数量为 $\frac{(h_i+...+h_n)!}{h_1!...h_n!}$。

定理 5.2　表 $T_i = PT[QI]$，T_j、T_z 是它的两个满足最小要求抑制的泛化，则有：$DV_{i,j} < DV_{i,z} \Rightarrow |T_j| \leq |T_z|$。直观来看，由于 T_z 更加“泛化”，它要满足抑制要求就只需要移除更少的条目。

推理 5.1　由定理 5.1 可知，$|T_j| \leq |T_z| \Rightarrow (|T_i| - |T_j|) \geq (|T_i| - |T_z|)$，则：$|T_i| - |T_j| \leq MaxSup \Rightarrow |T_i| - |T_z| \leq MaxSup$，反之，$|T_i| - |T_z| > MaxSup \Rightarrow |T_i| - |T_j| > MaxSup$。因此，如果一个具有 $DV_{i,z}$ 的表 T_z 不能在抑制阈值 MaxSup 范围内提供 k-anonymity，则所有具有 $DV_{i,j} < DV_{i,z}$ 的表 T_j 也不能。

DV **的高度**　定义为在一个 DV 格 VL 中，从结点 DV 到格 VL 最小元素的距离。表示为 $height(DV, VL)$。

引理 5.1　格 $VL =< DV, \leq >$，$\forall DV_{i,j}, DV_{i,z} \in DV$，有：$DV_{i,j} < DV_{i,z} \Rightarrow height(DV_{i,j}, VL) < height(DV_{i,j}, VL)$。

通过以上引理，可以推理出实现 k-anonymity 的策略。设 $\top$ 是 VL 的最大值，对每一个高度 h，如果在高度 h 对应的 DV 不满足指定的泛化，那么任何小于高度 h 的 DV 都不满足指定的泛化。因此，我们可以通过二分搜索的方法找到满足指定泛化的最小高度 h，并找到相应的 DV。假设 $h = height(\top, VL)$，我们首先评估高度 $\lfloor \frac{h}{2} \rfloor$ 处的所有 DV 是否满足指定的泛化，如果满足，则记录下这些 DV，然后在小于 $\lfloor \frac{h}{2} \rfloor$ 高度处递归寻找是否有满足指定泛化的更小的 DV。反之，则在大于 $\lfloor \frac{h}{2} \rfloor$ 高度处寻找。此过程类似二分查找。

此算法基于预定义的 DGH，采用二分法查找 DV。

4 Incognito: Efficient full-domain k-anonymity[4]

4.1 基本概念

频率集 (frequency set) 表 T 关于属性集 Q 的频率集是一个映射，从 Q 的一个值到此值在表 T 中的数量。在 SQL 中，类似于 COUNT(*) 子句。

值泛化函数 (value generalization function) 定义为：$\gamma : D_i \to D_j$，是一个多对一的函数，用于表示值泛化关系，例如：$\gamma(53712) = 5371*$。

值泛化函数组合 我们使用 γ^+ 表示一个或多个值泛化函数的组合，它返回一个集合，包含多个泛化函数的值。例如，$537** \in \gamma^+(53712)$。

全域泛化 (full-domain generalization [3]所提的泛化只针对一个 QI 属性，而全域泛化指的是有一组 QI 属性 $Q_1, ...Q_n$ 和一组对应的值泛化函数 $\phi_1, ...\phi_n$，将表 T 的 QI 属性的值 q 替换为 $a = \phi_i(q)$ 就可以得到其全域泛化 V。

文献 [3]**的局限性** 文章定义了 DV 最小的泛化是最优泛化，有些时候这并不适用于某些应用，他们需要的是灵活性。例如，有的应用希望 Sex 字段不做任何泛化，而 ZipCode 则无所谓。而文章中的算法是无法满足这种灵活性的。文章中基于二分查找的算法不能保证找到最小的泛化，因此文本采用一个自底向上的宽度优先搜索算法，检查每一种可能的泛化，这样就可以找到最小的泛化，或找到格上的所有泛化。

4.2 Incognito

全域泛化通过连接表 T 和它的维度表，再对域属性进行恰当的投影来产生。例如，表 $T =< A, ... >$ 和属性 A 的维度表 $< A_0, A_1, ... >$，泛化表 $T_1 =< A_1, ... >$ 可通过连接和投影操作来得到。

关于泛化维度，有几个关键的性质：

泛化性质 (generalization property)　有表 T，P、Q 是表 T 的属性集，且 $D_P <_D D_Q$。如果表 T 满足关于 P 的 k-匿名，则也满足关于 Q 的 k-匿名。

汇总性质 (rollup property)　有表 T, P、Q 是表 T 的属性集, 且 $D_P <_D D_Q$。如果已知关于 P 的频率集 f_1，则由 f_1 和 γ 可生成关于 Q 的频率集 f_2。

子集性质 (subset property)　有表 T, P、Q 是 T 的属性集, 且 $P \subseteq Q$。如果 T 满足关于 Q 的 k-匿名，则 T 也满足关于 P 的 k-匿名。(此性质有个推论: 如果 T 不满足关于 P 的 k-匿名, 则 T 也不满足关于 Q 的 k-匿名)

4.2.1　基本 Incognito 算法

此算法从包含单个属性的 QI 集合的子集开始，逐个检查是否满足 k-anonymity，并不断迭代增大子集。迭代过程如下:

- 从 $i = 1$ 开始，考虑包含 i 个属性的 QI 集的子集，以及由这些子集出发的 DGH。定义 C_i 为由包含 i 个属性的子集出发所构成的这些 DGH 中顶点的集合，E_i 为边的集合。使用一个宽度优先搜索对 C_i 中的元素进行检查，并将生产的结果保存在 S_i 中。
- 得到 S_i 之后，迭代进行第 $i+1$ 次计算，直到 $i > n$ 为止。

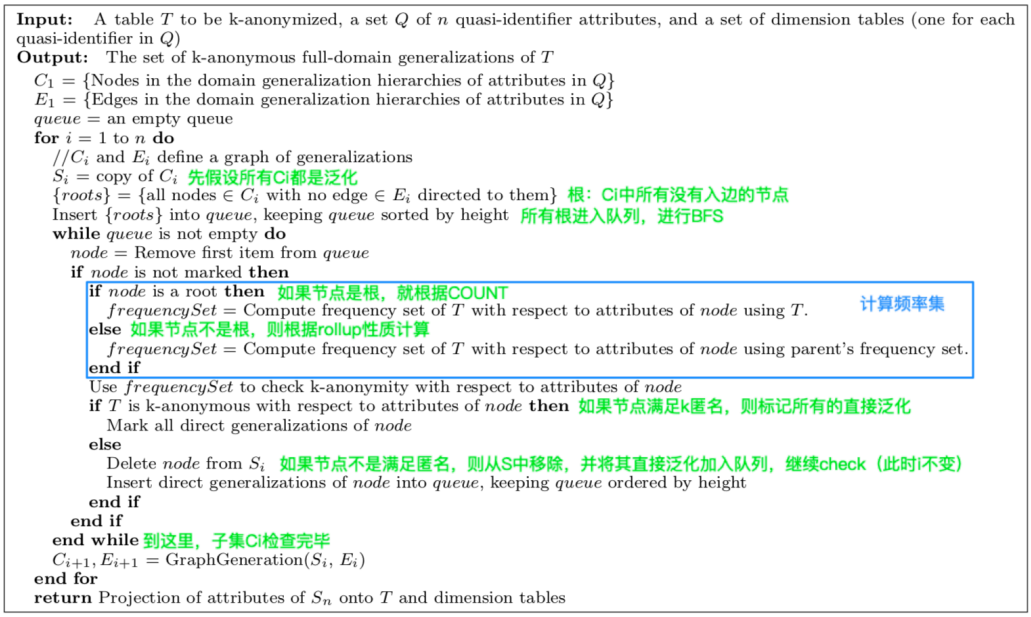

图 6: 基本 Incognito 算法

算法过程见图 6 所示。以图所示数据为例，计算关于 $\{Birthdate, Sex, Zipcode\}$ 的 2-匿名泛化过程如下：

1. i=1 时

   (a) 包含 1 个属性的 QI 集的子集是：$\{< B >, < S >, < Z >\}$（为描述方便，以下省略“<>”）

   (b) DGH：$B_0 \to B_1 \to B_2 \to B_3$($B_0$ 表示原始日期，B_1 表示只使用“年月”表示的日期，B_2 表示只使用“年”表示的日期，以此类推)、$S_0 \to S_1$、$Z_0 \to Z_1 \to Z_2 \to Z_3$。

   (c) 计算 C_1：$\{B_0, ...B_3, S_0, S_1, Z_0, ...Z_3\}$。

   (d) 以广度优先搜索方法检查泛化是否满足 2 匿名：

      i. 搜索开始前：令结果集 $S_1 = \Phi$（此处与文中算法相反，但原理一致)，将所有 DGH 的根节点加入到搜索队列中：$queue = \{B_0, S_0, Z_0\}$，下面开始迭代：

      ii. 取队列中第一个节点 $node = B_0$ 得到：根据 Count 计算频率集 $\Rightarrow$ B_0 不满足 2 匿名 $\Rightarrow$ 将 B_0 的直接泛化 B_1 加入到

队列中：$queue = \{S_0, Z_0, B_1\}$

iii. 取队列中下一个节点 $node = S_0$ 得到：根据 Count 计算频率集 $\Rightarrow$ S_0 满足 2 匿名 $\Rightarrow$ 将 S_0 的所有直接泛化加入到结果集 S_1 中得到：$S_1 = \{S_0, S_1\}$。

iv. 取队列中下一个节点 $node = Z_0$ 得到：根据 Count 计算频率集 $\Rightarrow$ Z_0 满足 2 匿名 $\Rightarrow$ 将 Z_0 的所有直接泛化加入到结果集 S_1 中得到：$S_1 = \{S_0, S_1, Z_0, Z_1, Z_2, Z_3\}$。

v. 取队列中下一个节点 $node = B_1$ 得到：根据父节点和 γ 函数计算频率集 $\Rightarrow$ B_1 满足 2 匿名 $\Rightarrow$ B_1 的所有直接泛化加入到结果集 S_1 中得到：$S_1 = \{S_0, S_1, Z_0, Z_1, Z_2, Z_3, B_1, B_2, B_3\}$。

2. i=2 时

   (a) 包含 2 个属性的 QI 集的子集是：$\{< B, S >, < S, Z >, < B, Z >\}$

   (b) 计算 C_2，C_i 是 S_{1-1} 的超集，可由 S_{1-1} 扩展而来，另外，C_i 中的元素应当按 DGH 的 height 排序。

   (c) 以下的步骤与 $i = 1$ 时类似。

3. $i = 3$ 的步骤与前面一致。
4. 此算法的最终结果 S_3 是最终泛化结果的泛化过程。

4.2.2 完备性

对于 k-匿名全域泛化来讲，基本 Incognito 算法是完备的。

4.3 k-匿名模型的分类

4.3.1 全局重编码模型 (global recoding models)

基于结构的单维重编码 (hierarchy-based single-demension recoding) 单维重编码模型为每一个 QI 集的属性 Q_i 定义一个泛化函数 ϕ_i。不同的单维重编码模型的区别在于泛化函数的不同。

基于分区的单维重编码 (Partition-based single-dimension recoding)　在单维有有序分区模式下，我们假设每一个属性的值域 D_{Q_i} 都能表示为一个全序集，ϕ_i 将属性值映射到这个全序集。

基于结构的多维重编码 (hierarchy-based multi-dimension recoding)　多维重编码模型定义一个单一的函数 $\phi :< D_{Q_1} \times ... \times D_{Q_n} \rightarrow D'$，用来重编码 QI。

基于分区的多维重编码　同样，这也是单维模型的扩展。

4.3.2　**局部重编码模型** (local recoding models)

5　Transforming data to satisfy privacy constraints[5]

5.1　Usage Based Metrics

我们考虑构造一个分类模型，表格的其中一个列用于存储分类标签。用 $G(r)$ 表示行 r 所属的等价类。由于泛化的原因，一个等价类内的行有可能分属于不同的分类类别。因此，我们可以通过惩罚包含不同分类类别的等价类来度量数据质量。

分类度量　定义为对每一个行的惩罚均值：

$$CM = \frac{\sum_{\text{all rows}} penalty(r)}{N} \tag{7}$$

$$penalty(r) = \begin{cases} 1 & \text{if r is suppressed} \\ 1 & \text{if} class(r) \neq magority(G(r)) \\ 0 & otherwise \end{cases} \tag{8}$$

如果行 r 被隐匿或它的分类类别 $class(r)$ 其所在等价类的主分类类别 (占多数的分类类别)，则其惩罚为 1，否则为 0。如图 7 所示，行 $(X = 9, Y = 9)$ 的惩罚为 1，因为它的分类类别不是所在等价类的主分类类别。

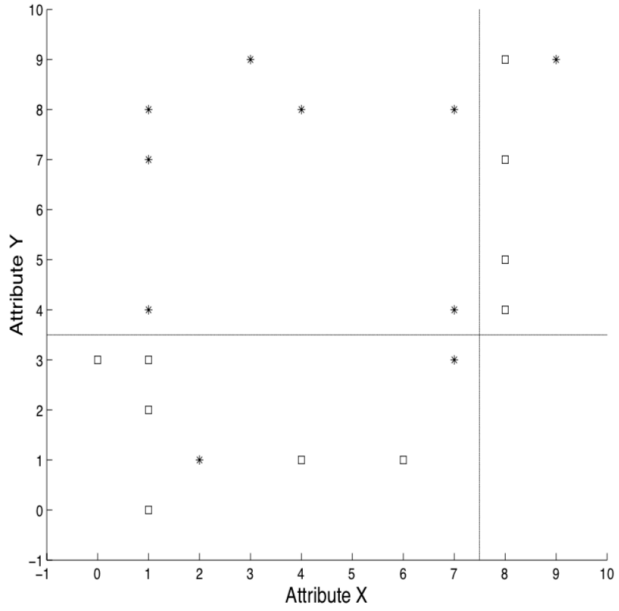

Figure 4: Example with numeric attributes X and Y

图 7: DGH 的格表示

6 Data privacy through optimal k-anonymization[6]

6.1 Introduction

本文提出一种发现给定数据集的最优化 k-匿名的实用方法。此方法与传统方法的不同之处在于:

1. 一般方法通常对给定数据集系统地或使用贪心方法进行泛化，直到满足 k-匿名。而本文的算法则是首先给定一个完全泛化的数据集（所有数据条目都一样），然逐渐具体化数据，直到满足最小 k-匿名。
2. 本文基于树搜索策略进行基于成本的剪枝和动态搜索安排 (dynamic search rearrangement)。
3. 提出一种新颖的数据管理策略，以减少评估匿名状态 (anoymization) 成本。

6.1.1 基础

属性的值域 数据元组中，第 i 个属性的值域表示为 Σ_i。

属性泛化 (attribute generalization) 对一个值域为 Σ 的有序属性来说，其属性泛化是将其值域划分为 $< I_1, ...I_m >$，这种划分使得属性的每个值都出现在某个划分中，并且 I_i 中的每个值都出现在其前一个划分 I_j 的每个值之前，也就是说，划分互不交叠。例如，对于属性 age，其值域是 [1,30] 的整数，其属性泛化可以表示为 $< [1, 10], [11, 20], [21, 30] >$。为方便，我们通常使用每个区间的最小值表示这个区间，上述例子就可以表示为：$\{1, 11, 21\}$。

属性泛化的应用 属性泛化 a 可以应用于属性值 v 上，表示为 $a(v)$，其返回的是值 v 所属的区间。例如，对于属性 age，a(2) 返回的是 [1,10]。数据集 D 的匿名化可以由 D 中各个属性的一组属性泛化 $\{a_1, ..., a_m\}$ 组成，可看作是对 D 转换为另一个数据集：$D' = \{< a(v_{1,1}), ..., a(v_{1,m}) >, ..., < a(v_{n,1}), ..., a(v_{n,m}) >\}$，$v_{i,j}$ 表示第 i 条数据的第 j 个属性值。

度量方法 每一种匿名化方法都需要有一个成本度量方法。成本度量一般是对泛化或隐匿操作进行统计信息损失。有两种主要度量方法：

分辨度量 (discernibility metric) 它针对每一条数据的辨别度进行其进行惩罚。对结果表中的每一个等价类，若它的大小为 j，则其中每一条数据的惩罚也为 j，若一条数据被隐匿，则其惩罚为 $|D|$。即：

$$C_{DM}(g, k) = \sum_{\forall E \ \ s.t. \ \ |E| \geq k} |E|^2 + \sum_{\forall E \ \ s.t. \ \ |E| < k} |D||E| \tag{9}$$

分类度量 (classification metric) 它根据等价类中每条数据的分类标签进行惩罚，具体见 [5]：

$$C_{CM}(g, k) = \sum_{\forall E \ \ s.t. \ \ |E| \geq k} (|minority(E)|) + \sum_{\forall E \ \ s.t. \ \ |E| < k} |E| \tag{10}$$

6.2 匿名化的集合表示

参照属性泛化，我们可以在所有属性上强加一个全排序 (total order)，使得当 $i > i$ 时，任何一个 Σ_i 中的值都在 Σ_j 中的值之前。如图8 所示。根据属性泛化的表示方法，由于每个值域中的最小值都必须出现泛化器 (generalizer) 中，因此这些值可以被省略。如图8 中标注“*”的那些。

Figure 1. Example total ordering of the value domains for a simple 3-attribute/9-value table. Given a total ordering, values can be identified by their position along the ordering. The least value from each attribute domain (flagged with a *) must appear in any valid generalization and hence can be treated as implicit.

图 8: 所有属性的全序表示

匿名化集合表示　我们用 v_{l_i} 表示第 i 个属性的最小值，我们可以这样表示匿名化：

$$\Sigma_{all} = (\Sigma_1 \setminus v_{l_1}) \cup ... \cup (\Sigma_m \setminus v_{l_m}) \tag{11}$$

特别地，空集合 {} 表示最一般 (most general) 匿名化，它只包含一个等价类；Σ_{all} 表示最具体 (most specific) 匿名化

示例　如图8 ，考虑匿名化 $\{2, 7, 9\}$，加上蕴含的（被省略的）最小值就是：$\{1, 2, 4, 6, 7, 9\}$，也就是说，将三个属性分别划分为：Age:<[10-29],[30-49]>

6.3 A Systematic Search Strategy

我们将确定最优化 k 匿名的问题界定为一个以最小代价在匿名化集合 Σ_{all} 中进行搜索的问题。集合枚举 (set-enumeration) 搜索策略是从给定的集合中以树形结构系统地枚举所有子集的最直接的一种方法。

头集 (head set)　在集合枚举树中，任何节点都能够被所枚举的集合节点表示，这些节点成为头集。

尾集 (tail set)　能够被加入到头集中的值的集合成为尾集。

全集 (allset)　由头集和尾集组成的集合成为全集。

由于集合枚举是系统和完全的，通过此算法我们可以保证得到最佳的匿名化。不幸的是，匿名化集合的大小是 $2^{|\Sigma|}$。

6.3.1 Pruning Overview

要减小搜索空间，最好的办法是剪枝。搜索树首先尝试减掉节点，如果不行则尝试减掉节点的尾集元素。是否剪枝则通过计算成本下界 (lower-bound) 来决定。(如果成本下界比目前最优成本还高，则可以剪去)

6.3.2 Computing Cost Lower-Bounds

隐匿产生的 penalty　如果匿名方案 H 隐匿了元组集 S，则 H 的任何后代节点所隐匿的元组集一定保护 S:

观察 5.1　在搜索树中，节点 H 隐匿元素是 H 的后代节点所隐匿节点的子集。对分辨度量来讲，H 的后代的隐匿操作的惩罚就是 $|D| \cdot |S|$；对分类度量，其惩罚是 $|S|$。

泛化产生的 penalty　泛化方案的后代更加具体，产生的等价类更多:

观察 5.2　如果一个节点 H，其上的全集是 A，则由 A 产生的等价类是由 H 的后代产生的等价类的子集。

定义分辨度量　定义 $E_{A,t}$ 是含有元素 t 且被全集 A 泛化的等价类。容易看出，E 中的元组 t 最小的惩罚应当时 $|E|$，当 $|E| < k$ 时，其惩罚可取 k。因此，分辨度量定义为:

$$LB_{DM}(H, A) = \sum_{\forall t \in D} \begin{cases} |D| & \text{when t is suppressed by H,} \\ \max(|E_{A,t}|, k) & otherwise \end{cases} \tag{12}$$

定义分类度量 定义为:

$$LB_{CM}(H,A) = \sum_{\forall E \text{induced by A}} \begin{cases} |D| & \text{when E is suppressed by H,} \\ |minority(E)| & otherwise \end{cases} \tag{13}$$

7 Mondrian Multidimensional K-Anonymity[7]

7.1 Introduction

本文贡献 本文的贡献主要是一种新的多维重编码模型和一种贪心划分算法，其优点是:

- 更有效率，其时间复杂度为 $O(nlogn)$;
- 产生的结果质量更高。

一般目的质量度量 最简单一种质量度量方法是基于等价类的数量。

7.2 Multidimensional Global Recoding

全局重编码 通过对 QI 属性的值域进行泛化来实现匿名化，它可进一步分为两类：单维全局重编码和多维全局重编码。多维全局重编码可同时用于离散数据 (categorical data) 和连续数据 (numeric data)，而单维划分模型则用于连续数据。

划分模型 单维划分将一个属性的值域划分为一个互不重叠的集合。单维划分可扩展至多维划分，其定义为一堆 d 维元组 $(p_1, ...p_d), (v_1, ..., v_d) \in D_{X_1} \times ... \times D_{X_d}$ 存在 $\forall i, p_i \leq v_i$。

7.3 Multidimensional Local Recoding

7.4 A Greedy Partitioning Algorithm

7.5 Workload-Driven Quality Measurement

8 l-Diversity: Privacy Beyond k-Anonymity[8]

8.1 Introduction

8.1.1 k-anonymity 面临的两种形式的攻击

k-anonymity 面临两种形式的攻击:

同质化攻击（Homogeneity attack**）** 如同前面所提的的“互补发布攻击”、“时间攻击”，由于数据发布时，并不是所有字段都是 QI，而同一组 QI 的值相同的数据中，会出现敏感字段都相同的情况，而且这种情况并不罕见。在这种情况下，如果数据集仅执行 k-anonymity，则会由于敏感字段缺乏多样性而使得攻击者可以重新识别数据集个体，从而造成信息纰漏。同质化攻击问题的一种解决方案就是本文所讲的 l-diversity 模型。

背景知识攻击 (Background knowledge attack) 就算数据由于多样性的存在而不会受到同质攻击，但也有可能会由于背景知识（例如，可以从日常生活看出某人肯定不会患有某种特定疾病）的存在而是多样性容易排除，从而使得敏感信息泄露。

8.2 定义

数据泛化（Data generalization**）** 将一个数据集进行分组，各个分组之间互不重叠。其形式化定义为：域 $D^* = \{P_1, P_2, ...\}$ 是域 D 的泛化，如果 $\cup P_i = D$ and $P_i \cap P_j = , i \neq j$。$\phi_{D^*}(x)$ 表示 D^* 包含 x 的元素 P。

攻击者（Adversary**）的背景知识** 攻击者可能知道以下几类背景知识:

攻击者可能知道已发布的表 T 的泛化 T^*，也就是说，知道 T 的属性的值域。

攻击者可能知道某些个体存在于表中。

统计（demographic**）背景知识**　攻击者可以知道表中字段的值的分布。

8.3　贝叶斯优化隐私模型（Bayes-Optimal Privacy, BOP）

两个假设　为简化模型，给出两个假设：

- 数据表 T 是全集的一个随机样本集；
- 个体只有一个敏感字段。

另外，还假设一个比较糟糕的情况：攻击者知道准标识符 Q 和敏感字段 S 的联合分布。

度量　我们用两个指标对隐私纰漏进行度量：

先验置信（prior belief**）**　已知目标个体 t 的准标识符的值 q 的情况下，其敏感字段值为 s 的概率：

$$\alpha_{(q,s)} = P_f(t[S] = s | t[Q] = q) \tag{14}$$

后验置信（posterior belief**）**　已知目标个体 t 的准标识符的值 q，以及其泛化 t^* 的情况下，其敏感字段值为 s 的概率：

$$\beta_{(q,s,T^*)} = P_f(t[S] = s | t[Q] = q \wedge \exists t^* \in T^*, t \overset{*}{\rightarrow} t^*) \tag{15}$$

定理 3.1　令 $n_{q^*,s'}$ 是表中出现 (q^*, s') 对的个数。

定理 3.1 **的证明**

8.3.1　隐私原则

数据表发布后可能泄露隐私的两种方式

正纰漏 (Positive Disclosure)　如果发布数据表后，可以提升攻击者正确识别目标的概率，例如，对于一个 q，其对应的 s 的值都是同一个（同质化攻击），那就是正纰漏。

负纰漏 (Negative Disclosure)　如果发布数据表后，攻击者可以从中排除一些关于目标的错误的敏感值，例如，根据背景知识可以排除特定个体的一些敏感值，那就是负纰漏。

不提供信息原则 (Uninformative Principl)　隐私的理想定义应该满足不提供信息原则，即：表的发布并不能为攻击者带来除了背景信息以外的其他信息。此原则有很多实例化的方式，他们都属于贝叶斯优化隐私定义。其中一个是：

$(\rho_1, \rho_2) - Private$　给定表 T^* 和两个常数 ρ_1, ρ_2。如果 $\alpha_{(q,s)} < \rho_1 \wedge \beta_{(q,s,T^*)} > \rho_2$，或者 $\alpha_{(q,s)} < 1 - \rho_1 \wedge \beta_{(q,s,T^*)} > 1 - \rho_2$，我们就说发生了“$(\rho_1, \rho_2)$-privacy breach”。如果 (ρ_1, ρ_2)-privacy breach 没有发生，我们就说表 T^* 满足 $(\rho_1, \rho_2) - Private$。

8.3.2　BOP 的局限

BOP 存在着一些缺陷，使其无法在实践中使用。包括：

不充分知识　数据发布者不可能完全知道数据的分布。

对攻击者的无知　数据发布者不知道攻击者到底知道什么。

实例知识　理论定义不能定义那些无法被概率模型建模的知识。例如，Bob 的儿子可能会告诉 Alice，Bob 没有糖尿病。

多攻击者　可能会有多个不同知识水平的攻击者存在，他们具有的背景知识都不同，数据发布者都必须考虑到。

8.4 l-diversity: 一种实用的隐私定义

l-diversity 可以克服上述的 BOP 实践中存在的问题。

8.4.1 l-diversity 原则

等价类 $(q^* - block)$ 表 T^* 中的一组数据，他们的非敏感值都被泛化为 q^*。

l-diversity **原则** 如果有一个 $q^* - block$，它的隐私属性 S 包含至少 l 个良表示 (well-represented) 值，我们就说这个 $q^* - block$ 是 $l - diverse$ 的，如果表中所有的 $q^* - block$ 都是 $l - diverse$ 的，那么这个表就是 $l - diverse$ 的。

此原则主张确保每一个 $q^* - block$ 的敏感字段具有至少 l 个良表示值，但并没有说明良表示意味着什么。最简单的，我们可以将“l 个良表示值”理解为“l 个不同的值”。另外，定义不同的良表示值也可以得到不同的实例。

8.4.2 l-diversity 的实例

根据不同的良表示定义，可以得到不同的 l-diversity 定义。包括：可区分 l-diversity、熵 l-diversity、递归 l-diversity 等。

Distinct(**可区分**) l-diversity 要求每个等价类中至少有 l 个可区分的值。它存在一个缺陷：如果某一个取值的频率明显高于其他取值，这将使得观察者可以以较高的置信度认为这一等价类中的敏感属性都取这个值。

Entropy l-diversity 设 $p_(q^*, s)$ 为等价类中敏感值为 s 的概率，那么表满足 l-diversity 的条件是，每一个等价类都满足：

$$-\Sigma_{s\in S} p_(q^*, s)\log(p_(q^*, s)) \geq \log(l) \tag{16}$$

这要求整个表的熵至少是 $\log(l)$，也就是要求至少有 l 个可区分的值。这并不容易实现，特别是当某个敏感值非常常见的时候。

Recursive(c,l)-diversity 递归 l-diversity。

8.4.3 多敏感属性的情形

以上讨论都基于一个基本的假设：每条数据只有一个敏感字段。

Multi-Attribute l-diversity 若把 S_i 之外的所有 S 都当做 Q 还满足 l-diversity，那么表 T 就满足 l-diversity。

参考文献

[1] Shui Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data", IEEE access, vol. 4, pp. 2751–2763, 2016. 1

[2] Latanya Sweeney, "k-anonymity: A model for protecting privacy", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 557–570, 2002. 5

[3] Pierangela Samarati, "Protecting respondents identities in microdata release", IEEE transactions on Knowledge and Data Engineering, vol. 13, no. 6, pp. 1010–1027, 2001. 7, 14

[4] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan, "Incognito: Efficient full-domain k-anonymity", in Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, 2005, pp. 49–60. 14

[5] Vijay S Iyengar, "Transforming data to satisfy privacy constraints", in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002, pp. 279–288. 18, 20

[6] Roberto J Bayardo and Rakesh Agrawal, "Data privacy through optimal k-anonymization", in null. IEEE, 2005, pp. 217–228. 19

[7] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan, "Mondrian multidimensional k-anonymity", in null. IEEE, 2006, p. 25. 23

[8] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam, "\ell-diversity: Privacy beyond\kappa-anonymity", in null. IEEE, 2006, p. 24. 24