PRIVACY IN COMPLEX SAMPLE BASED SURVEYS

by

Shawn Merrill

A Dissertation

Submitted to the Faculty of Purdue University
In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Computer Science
West Lafayette, Indiana
December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Clifton Bingham

Department of Computer Sciences

Dr. Ninghui Li

Department of Computer Sciences

Dr. Jeremiah Blocki

Department of Computer Sciences

Dr. Bruno Ribeiro

Department of Computer Sciences

Approved by:

Dr. Kihong Park

To my family, thank you for all of the support over the years.

ACKNOWLEDGMENTS

This work was partially supported by the United States Census Bureau under CRADA CB16ADR0160002. The views and opinions expressed in this writing are those of the authors and not the U.S. Census Bureau.

TABLE OF CONTENTS

IV CONCLUSION 118

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In the last few decades, there has been a dramatic uptick in the issues related to protecting user privacy in released data, both in statistical databases and anonymized records. Privacy-preserving data publishing is a field established to handle these releases while avoiding the problems that plagued many earlier attempts. This issue is of particular importance for governmental data, where both the release and the privacy requirements are frequently governed by legislature (e.g., HIPAA, FERPA, Clery Act). This problem is doubly compounded by the complex survey methods employed to counter problems in data collection. The preeminent definition for privacy is that of differential privacy, which protects users by limiting the impact that any individual can have on the result of any query.

The thesis proposes models for differentially private versions of current survey methodologies and, discusses the evaluation of those models. We focus on the issues of missing data and weighting which are common techniques employed in complex surveys to counter problems with sampling and response rates. First we propose a model for answering queries on datasets with missing data while maintaining differential privacy. Our model uses k-Nearest Neighbor imputation to replicate donor values while protecting the privacy of the donor. Our model provides significantly better bias reduction in realistic experiments using existing data, as well as providing less noise than a naive solution. Our second model proposes a method of performing Iterative Proportional Fitting (IPF) in a differentially private manner, a common technique used to ensure that survey records are weighted consistently with known values. We also focus on the general philosophical need to incorporate privacy when creating new survey methodologies, rather than assuming that privacy can simply be added at a later step.

Part I

INTRODUCTION

1. INTRODUCTION

It was a Sunday in 1996 when then Massachusetts governor William Weld collapsed at a commencement at Bentley College in Waltham, MA. When he was taken to the hospital it was then reported that doctors had found no serious issues with the governor and the incident was chalked up to a likely case of the flu. A short while later, as part of an effort to help researchers, the Massachusetts Group Insurance Commission would provide de-identified health records for every state employee hospital visit. Following standard practice at the time, they removed all directly identifiable information. A few years later a graduate student at MIT named Latanya Sweeney showed exactly how little protection there was in that dataset. Using the freely provided health records and combining that information with the publicly purchasable voter registration information for Cambridge, where the governor lived, she was able identify the hospital records pertaining to that incident [1]. Using that incident as a starting point, she was able to identify all records associated with former Governor.

There are many other examples that are less (in-)famous than the above Weld example but no less consequential for those involved such as AOL search records leading to a senior citizen in Georgia [2], Target identifying and providing ads for a pregnant teenager before she told her parents [3], and the de-identification of a good portion of the Netflix challenge dataset [4]. Typically these attacks involved outside information that was then used to learn about specific individuals from the dataset. These examples tend to be older since many companies have since learned to avoid these obvious situations, but it is important to note that privacy breeches of this kind still occur even now. The US Census Bureau was able to reconstruct exact records for a significant percentage of the 2010 Census by using the large number of summary statistics that are published on the data [5]. While not as flashy, these newer attacks incorporate much more sophisticated analysis techniques to break the privacy of datasets. The attacks highlight the need for strong formal definitions of privacy to better understand the privacy guarantees and risks associated with the different models of privacy as well as what types of attacks data curators are concerned with.

The early work on data releases focused exclusively on PII (Personally Identifiable Information), which is information specific to an individual. For example things like name and social security number are obviously PII. However, the Weld case shows that even removing PII does not make records immune to reconstruction attacks. Later laws such as HIPAA (Health Insurance Portability and Accountability Act) specify mean by which information must be de-identified if they are to be released. Title 13, which governs the US Census Bureau, has a similar statement, if not specific examples, of the requirement to protect the identity of information released. This spurred the creation of a number of more robust privacy frameworks to better specify the amount of privacy achieved.

k-anonymity[6] was one of the first definitions of privacy, informally described as 'hiding in a crowd of at least k people'. This definition seeks to protect from the same reconstruction attack that Sweeney was able to apply to the Weld case, and later research showed that much of the US was vulnerable to the intersection of a small number of *quasi-identifiable attributes*[7], attributes which on their own do not uniquely identify an individual, but the intersection of a small number of them is likely to single out a unique individual. While this protects from direct deterministic reconstruction, many other modifications of k-anonymity were created to avoid other related attacks ([8],[9]). However, these definitions have fallen out of favor recently with the introduction of differential privacy and related definitions due to their ease of application, and stronger mathematical definitions of both privacy and privacy loss. Differential privacy takes a multiple-worlds approach, preventing an attacker from knowing if the dataset includes an individual's results or not [10]. Informally, this is equivalent to saying that your privacy cannot be violated if I could learn something about you from a dataset without your information included. In particular, inferences made from population-level statistics are not considered privacy violations. It should be noted that this definition of privacy, while allowing data curators to quantify an individual's privacy risk, does not always match up with an intuitive definition of privacy. The Target example above would not constitute a "privacy violation" under this definition, since it is likely that the models used to identify pregnant women was created without the 16 year-old's data.

1.1 Problem Statement

Differential privacy generally works on the basis of global sensitivity, the maximum possible change between any two potential datasets, but complex surveys for a number of reasons make this type of analysis hard to impossible. Namely complex surveys tend to contend with practical considerations like budgets and other business and time constraints. Therefore, these surveys typically cannot guarantee that pathological cases which are terrible for privacy can be avoided, for example only a few people answering a question having to stand in for a large group. While these surveys try to avoid such circumstances, they generally cannot be entirely excluded. Further compounding the difficulty of providing privacy is the number of statistical modifications that are done to the results. In this thesis we discuss two such modifications in greater depth - missing data imputation and post stratification weighting. These are examples of ubiquitous modifications which make guaranteeing privacy a difficult task. Notably, these surveys require a fundamental shift in how privacy is considered. The typical view of privacy is that is something that can be added later in the process, essentially that it is a property of the final query system or the final dataset and the previous modifications are not considered. Complex sample based surveys require consideration not only of the privacy but of the modifications that are made to the data along the way, and the impact that each of these modifications has on the privacy guarantee. Commensurate with that is a consideration of what modifications are made and their impact on either the bias or variance of the data fundamentally, in addition to the modifications needed to maintain privacy. Every component of the survey must be evaluated for its impact on both privacy and utility. Privacy cannot simply be a post-processing step.

Part II

BACKGROUND

2. PRIVACY TECHNOLOGY

2.1 k-Anonymity and its extensions

One of the first definitions of sanitization in the computing literature was that of k-anonymity [7], [11]. A dataset is said to be *k-anonymized* if every record in the table is indistinguishable from at least $k-1$ other records. This is generally accomplished by grouping records into blocks of size at least k and then anonymizing the records within a block to make them indistinguishable from one another. These blocks are called *quasi-identifier equivalence classes*, because for all records in a block, each value is the same for all of the quasi-identifiable attributes. Recall that a *quasi-identifier* is one which on their own do not uniquely identify an individual, but the intersection of a small number of them is likely to single out a unique individual. The standard way to anonymize the records within a block is by means of a *domain generalization hierarchy*, which is an ordered set of increasingly general values of the quasi-identifier fields. If a record within a block fails to be indistinguishable from the other records in the block, the value of one or more quasi-identifiers is replaced by a more general value from the hierarchy, until all the records within the block are identical.

Used naively, k-anonymized datasets can still leak information about an individual in the population. For instance, consider a table of medical records, with sensitive attribute Disease, in which every record is indistinguishable from at least k other records. It may happen, that all k records in a given equivalence class have the sensitive value cancer. Although k-anonymity is satisfied, clearly the privacy of these k individuals has been violated. Examples such as this motivated the following extension of k-anonymity known as ℓ-diversity [12], discussed below.

The mechanism whereby a dataset achieves k-anonymity (or ℓ-diversity) has also garnered much attention in the literature, falling largely into two types: *local* and *global recoding* [13]. Local recoding only generalizes an equivalence class when it must to achieve k-anonymity. Global recoding stipulates that whenever a quasi-identifier value is generalized within one class, it is automatically generalized to that level in every class, regardless of whether such generalization is needed to achieve k-anonymity. In particular, global recoding adds an additional stipulation beyond the simple definition k-anonymity. Clearly local recoding results in a sanitized dataset that may have less information loss than a globally recoded dataset, but local recoding has largely fallen out of favor as it introduces additional difficulty to data analysis, and increases susceptibility to various attacks [14], [15].

If we have multiple datasets, $\{\mathcal{A}_{\mathrm{i}}\}_{\mathrm{i}=1}^{n}$, each of the which satisfies k-anonymity on its own, then the union clearly contains at least $k-1$ individuals indistinguishable from other individuals, and hence also satisfies k-anonymity. This also holds for the local recoding variant of k-anonymity. We refer to taking the union of these datasets as *parallel composition*, and we ask whether various privacy definitions, and their popular implementations, satisfy this property.

k-anonymity under global recoding does not satisfy parallel composition. Remember that global recoding requires that if a value is generalized to a certain level (e.g., birth date to birth year), then it must be generalized that way for every data instance. This addresses some attacks that can “undo” the generalization of k-anonymization mechanisms, enabling adversaries to learn more than expected about individuals. Under parallel composition, the different $\mathcal{A}_{\mathrm{i}}$ may find different generalizations appropriate for their D_{i}. The union could then have the same base value generalized to different levels of the hierarchy for different data items, violating the global recoding requirement.

ℓ-diversity [12], [16] places an additional requirement on k-anonymity: within each group of k individuals with the same quasi-identifying information, there must be no greater than a $1/\ell$ proportion of any particular sensitive value. Clearly ℓ-diversity satisfies parallel composition because the union of equivalence classes in which no sensitive value appears with frequency greater than $1/\ell$ still has this property.

t-closeness [9] is a further refinement of ℓ-diversity, requiring that the distribution of sensitive values within every equivalence class approximately match the relative distribution of those attributes across the entire dataset. This prevents learning if the probability of an individual having a particular sensitive value is greater than that of the overall population. This also does not satisfy parallel composition, as different datasets may have different distributions of sensitive values, and thus a group with identical quasi-identifiers that satisfies t-closeness in an individual dataset may not satisfy it with respect to the overall distribution in the union of datasets. For example, suppose that we have two datasets taken from medical institutions, one of which has roughly 5% of individuals with HIV, and the other has no occurrence of HIV at all. Considered separately, both of these datasets can satisfy t-closeness, but when we union them, it may be that any equivalence class which contains HIV-positive individuals overrepresents that attribute with regard to the *new* proportion of HIV-positive individuals in the combined dataset.

2.2 Differential Privacy

2.2.1 Background

Throughout this thesis, we use the notation D to refer to a dataset from some universe $\mathcal{D}$. For two datasets D, D', we denote by $d(D, D')$ the Hamming distance between D and D'. We recall that a randomized algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ is said to satisfy ε-*differential privacy*[17] if for all $D, D' \in \mathcal{D}$ such that $d(D, D') = 1$ and for all (measurable) $S \subset \mathcal{R}$,

Definition 2.2.1 (Differential Privacy)**.**

$$\Pr\left[\mathcal{A}(D) \in S\right] \leq \mathrm{e}^{\varepsilon} \cdot \Pr\left[\mathcal{A}(D') \in S\right]$$

These datasets differ only in the inclusion/exclusion of the single record that is being protected. This form of differential privacy is called *unbounded differential privacy* and is the default version of differential privacy. There is another slightly altered form of differential privacy called *bounded differential privacy* which maintains the same restrictions and guarantees but requires that the size of the datasets are the same across neighbors. This can be thought of as modifying a single record instead of adding/removing a record. This form of differential privacy is much less common but important for our work in Chapter 5.

All forms of differential privacy assume an attacker knows any and all other records and the value of the record being protected. Since we don't specify which record, and we require it to protect every such record, an attacker is unable to know if any individual record is included in the dataset or not. It should be noted that this definition frequently provides unnecessarily strong guarantees of privacy and there are many accepted weakenings of this definition but our work has focused on ε-differential privacy as it is the most accepted definition.

The intuition behind differential privacy is that we want to prevent an attacker from knowing if an individual is in the dataset. As we mentioned earlier, this can lead to some odd assumptions about "privacy" that do not line up with the intuitive notions that many hold, but this definition allows us to make strong guarantees about inferences made on the data. It is a generally held principle in privacy that there is an inverse relationship between the utility of the data and the protections provided to the respondents. One advantage of differential privacy is that we can view the ε as something of a tuning parameter to quantify this trade-off. Smaller values of ε provides a strong guarantee of privacy with the equivalent reduction in utility whereas large values of ε don't protect respondents much. Special cases of ε would be $\varepsilon = 0$, which corresponds to perfect privacy but no practical utility, and $\varepsilon = \infty$, which corresponds to no privacy but the best utility we can provide. Despite the inverse relationship, we can seek to find more utile algorithms for a given ε since the full impact of the noise infusion is not always obvious. It is important to note that different algorithms may be able to leverage the privacy guarantee in different ways providing different results for the utility of a given query.

Differential privacy has a number of useful properties[18] that help with implementation and allow for more complex analyses and uses of the data. We will focus on the three key properties of differential privacy - sequential composition, parallel composition, and post-processing.

2.2.2 Sequential Composition

Sequential composition[18] is an important aspect of the practical uses of differential privacy. The property says that, when run in sequence, any number of individually differentially private mechanisms will still provide differential privacy, albeit with a larger ε (weaker privacy guarantee). This means that not only can multiple queries be run on the same dataset but the results of previous queries can be incorporated into future queries. This motivates the discussion of ε as a *privacy-loss budget* rather than a single privacy guarantee. As sequential calculations add their respective ε's together, it can be useful to discuss the overall ε as a singular value when considering the privacy guarantee. Another helpful metaphor is to view the budget as a gas tank, and every query on the data uses some of the fuel. Once it runs out (i.e., you have depleted the budget), no more queries can be run while maintaining the privacy guarantee.

2.2.3 Parallel Composition

Parallel composition[18] is the fundamental idea that differential privacy run on disjoint datasets can be combined and remain differentially private. An important caveat is that the disjoint datasets must be determined independently of the data. This property is very useful for many queries where results are desired to be run on different populations simultaneously, for example each state can be calculated without impacting the privacy of the other states.

2.2.4 Post-Processing

Post-processing is the final key property of differential privacy that allows for its use. Post-processing means that once the differential privacy has been applied (the noise infusion has happened) any modifications may be done to the data so long as they are data independent. Intuitively it is easy to think of a noise barrier, essentially once the noise has been applied (we are past the barrier) we can do any desired modifications so long as we do not consider the original data (before the barrier). This allows for some statistical modifications to be done without degrading the privacy guarantee, a key example would be the rounding results to the nearest integer or making all values non-negative. These modifications will still impact the utility of the data (the non-negativity will bias the data to higher values) but will not affect the privacy guarantee.

2.2.5 Global Sensitivity

While differentially private mechanisms all work on the requirement of protecting difference between neighbors, there is some variance in the universe of datasets ($\mathcal{D}$) that must be considered. The concept of 'sensitivity' in differential privacy is based on the greatest change to a specific query between two neighboring datasets. It is important to stress that differential privacy is considered a mechanism on the answer , so sensitivity is a query-specific attribute and a given dataset will have different sensitivity depending on the query being run. For example counting queries (how many people have X attribute) always has sensitivity 1 in unweighted datasets, but a query like “average income” will have a sensitivity based both on the actual salary values and the size of the dataset. Since mechanisms must provide noise to cover any discrepancy, a smaller sensitivity results in less noise and therefore a more accurate response. Each of the notions of sensitivity that we will discuss can be thought of as a maximization process over a specific domain of datasets. The simplest and most common notion of sensitivity is referred to as ‘global sensitivity’, which puts no further restrictions on these datasets besides belonging to $\mathcal{D}$.

Definition 2.2.2. Let q be an arbitrary query. Then the *global sensitivity* of q is:

$$GS_q := \max_{D,D':d(D,D')=1} ||q(D) - q(D')||_1,$$

where the maximum is over all datasets $D, D' \in \mathcal{D}$ which are neighbors in the Hamming distance.

Any differentially private mechanism employing this notion of sensitivity will require noise sufficient to cover any possible change between two datasets. This is a strong guarantee but frequently requires protecting pathological cases such as only a single respondent to a survey or other unlikely situations.

A perfect example of such pathological cases occurs with sample based surveys. If we consider no hard restrictions to the universe of datasets $\mathcal{D}$ then it is possible for a sample to have a small number of respondents. We consider this example as either a missing data result where those values for the few respondents will be replicated to those missing values or as a weighting exercise where the respondents will be weighted to some outside value. For specific numbers take the county of Los Angeles, which in 2010 had roughly 9.8 million people, and two respondents to the data. If we consider the situation as missing data then, assuming an even split between the two respondents, each value would be replicated 4.9

million times. In the weighting case we get a weight of 4.9 million for each respondent. This situation, or one substantially similar to it, is one that global sensitivity must consider because frequently sample based surveys will have a set of soft rules and a budget rather than a set of strict cut-offs. We consider this a pathological case since it can be, and would be in practice, avoided but must be considered in the realm of possibilities no matter how unlikely it would occur.

2.2.6 Local Sensitivity

The natural counter to requiring us to maximize over any possible dataset would be to only consider the dataset (and neighbors) that we currently have. This has the added benefit of avoiding the consideration of those pathological cases from our example unless they actually occurred. This is referred to as 'local sensitivity':

Definition 2.2.3. Let q be an arbitrary query and let D be a dataset. The *local sensitivity* of q at D is:

$$LS_q(D) := \max_{D':d(D,D')=1} ||q(D) - q(D')||_1,$$

where the maximum is taken over all $D' \in \mathcal{D}$ which is a neighbor of the given dataset D.

Obviously, the local sensitivity of a query will never be larger than the global sensitivity, and given that it does not require considering extreme datasets, it will frequently be much smaller. However, there are some circumstances where using a mechanism based on local sensitivity will leak information and therefore not satisfy differential privacy (e.g. median, see [19]). This type of potential information leak frequently means that local sensitivity-based mechanisms are discounted out of hand, especially in cases where the set of queries is not known prior to the noise infusion.

2.2.7 Smooth Sensitivity and the Generalized Cauchy Mechanism

Given the two extremes of global sensitivity requiring us to consider ridiculous cases and local sensitivity potentially leaking information, it is important to discuss a third option for sensitivity which aims to provide a compromise between the two extremes. Smooth sensitivity frequently allows us to deal with queries which have (potentially unboundedly) high global sensitivity due to pathological inputs which are unlikely to occur in practice. So instead of resorting to a worst-case scenario for computing the privacy guarantees, we instead consider the local sensitivies of datasets near the actual observed dataset. We first recall the following definitions (repeated for ease of comparison).

Definition 2.2.4. Let $q : \mathcal{D} \to \mathbf{R}$ be a real-valued query.

1. The *global sensitivity* of q is defined as

$$GS_q := \max_{D, D' : d(D, D') = 1} |q(D) - q(D')|.$$

2. The *local sensitivity* of q is the function $LS_q : \mathcal{D} \to \mathbf{R}^+$ given by

$$LS_q(D) := \max_{D' : d(D, D') = 1} |q(D) - q(D')|.$$

3. [19, Definition 2.1] Let $\beta > 0$. A *β-smooth upper bound* of the local sensitivity of q is a function $S : \mathcal{D} \to \mathbf{R}^+$ such that

$$\begin{aligned} \forall D \in \mathcal{D} : \qquad & S(D) \geq LS_q(D) \\ \forall D, D' \in \mathcal{D}, d(D, D') = 1 : \qquad & S(D) \leq \mathrm{e}^{\beta} \cdot S(D'). \end{aligned}$$

The canonical example of a β-smooth upper bound is given by

$$S(D) := \max_{k \geq 0} \max_{D' : d(D, D') = k} \mathrm{e}^{-\beta k} LS_q(D').$$

Note that $k = 0$ corresponds to the local sensitivity at D, and that as one moves away from D, the impact of datasets are exponentially penalized.

The standard mechanism for realising perturbed query results based on smooth sensitivity that satisfies pure differential privacy is the *generalized Cauchy mechanism.* This is based on the generalized Cauchy distribution, whose pdf is given by

$$f_\gamma(x) \propto \frac{1}{1+|x|^\gamma}, \quad -\infty < x < \infty,\ \gamma > 1.$$

Theorem 2.2.5. *[19, Lemma 2.7] Let $\varepsilon > 0$ and $\gamma > 1$. If q is a real-valued query and S is a $\left(\frac{\varepsilon}{2(\gamma-1)}\right)$-smooth upper bound of the local sensitivity of q, then the mechanism*

$$\mathcal{A}(D) = q(D) + \frac{2(\gamma-1)\cdot S(D)}{\varepsilon}\cdot X,$$

where X is sampled from the generalized Cauchy distribution, satisfies ε-differential privacy.

We remark that the generalized Cauchy mechanism has finite expectation if and only if $\gamma > 2$ and finite variance if and only if $\gamma > 3$. We typically take $\gamma = 4$, for which $E[X] = 0$ and $\mathsf{Var}(X) = 1$. We note that, particularly in the case of $\gamma < 4$ that understanding the full impact of the noise infusion will likely take many draws of the data since there are an number of comparatively low probability but high impact draws that can skew the data considerably.

It should be noted that the trade-off for the use of smooth sensitivity is that calculating the sensitivity becomes a data-dependent process and frequently a difficult or computationally expensive one at that, since "neighbors" must be considered in an expanding radius away from the given dataset.

2.2.8 Laplace Mechanism

The Laplace mechanism[17] is the original mechanism for providing differential privacy. This mechanism pulls noise from a Laplace distribution $f(x) \propto \mathrm{e}^{-|x|/\lambda}$ with $\lambda = \frac{\Delta q}{\varepsilon}$ where Δq is the sensitivity of the query. This mechanism, combined with global sensitivity, is considered the default for differential privacy and in many circumstances provides sufficient privacy while not impacting the utility of the result largely. Unfortunately, this mechanism does not work for smooth sensitivity which requires the Cauchy mechanism described above.

2.2.9 Exponential Mechanism

The exponential mechanism, first introduced in [20], is an extremely general class of release mechanisms which satisfy differential privacy. Roughly speaking, the idea is to choose from among a set of candidate releases by assigning to each input-output pair a *score*, determined by a function f. Let $\mathcal{D}$ denote the set of possible inputs to the mechanism (thought of as the set of possible datasets), and $\mathcal{R}$ the set of possible outputs. Suppose further that we have a measure μ on the space of outputs, which may for example be taken to be the uniform measure if the set $\mathcal{R}$ is finite (as is the case in this thesis). Let $f : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$, fix a constant $\alpha > 0$, and assume that for every $D \in \mathcal{D}$,

$$\int_{\mathcal{R}} \mathrm{e}^{\alpha f(D,r)} \mu(r) < \infty. \tag{2.1}$$

Given an input D, the exponential mechanism chooses an output r with probability proportional to the value $\mathrm{e}^{\alpha f(D,r)} \mu(r)$. In the special case where μ is the uniform measure on a finite subset of $\mathcal{R}$, this says that the output which is most probable is the one which maximizes the score function $f(D, r)$.

Theorem 2.2.6. *The exponential mechanism as described above satisfies* $(2\alpha \cdot \Delta f)$*-differential privacy, where* Δf *is the global sensitivity of the score function* f*, i.e.*

$$\Delta f := \max_{r \in R} \max_{d(D,D')=1} |f(D, r) - f(D', r)|.$$

3. COMPLEX MULTI-STAGE SURVEYS

3.1 Random Sampling

The simplest form of sampling is the *simple random sample.* The idea is that we can select a percentage of the population uniformly at random to provide a sufficiently accurate representation of the whole population. A common threshold would be a 5% sample. This could be used for things like presidential polls and other situations where all members of a population are considered interchangeable. This type of sampling has disadvantages in situations where a random sample won't accurately reflect the population of interest, one obvious example would be when a population of interest is not a large percentage of the total population. For example the American Indian/Alaskan Native population of the US is .9% [21] which means that a random sample almost surely will not adequately sample this population. Situations like this can cause random sampling to have a high sampling variance.

3.1.1 Differential Privacy under Random Sampling

The simplicity of the sampling mechanism, a potential hindrance for getting accurate results, is actually a benefit for differential privacy. The simplicity of the sample means we generally avoid the need to weight the data. The lack of weighting means that the impact of one person is limited to their direct impact on the query. Furthermore, if we answer queries via a random sample of our data, we can utilize so called "secrecy of the sample" to greatly improve the differential privacy guarantee we can make about the data. The most direct form of this improvement was shown by Ninghui Li [22] to allow for a near linear improvement in the guarantee provided by differential privacy. This means not only do we avoid the need for weighting and the difficulty that comes with weighting schemes but can potentially also leverage the secrecy of the sample itself to provide either a stronger guarantee (lower ε) or more utility while maintaining the same privacy guarantee.

3.2 Stratified Sampling

In light of the preceding discussion, more sophisticated sampling techniques are generally used for large scale complex surveys, since the likelihood of providing a representative sample of the population with a truly random sample is low. In *stratified sampling* the population is divided among a set of attributes that the domain experts agree are important (for the previous voting example things like ability to vote and whether they have previously voted might be used). The idea is that once these divides are accounted for then we have created equivalence classes for the population where individuals are considered interchangeable with others in their same stratum. How the strata are created and how deeply to sample from each stratum is a situation where there is multiple popular solutions. The most common of these solutions for stratified sampling is proportional allocation. Other allocation considerations that are used in practice are the likelihood of response and the how homogeneous the given stratum is and how accurate information about a specific stratum must be (all of these examples are from the American Community Survey which has to consider this and more in their strata and sampling considerations[23]). While these mechanisms can be used to decrease the sampling variance and bias, they can cause problems for differential privacy mainly due to their continued reliance on the data and the tailored aspects of the solution[24].

3.2.1 Differential Privacy under Complex Sampling

While the idea of differential privacy under complex sampling techniques has only started to be explored[24] the initial work has showed problems with these more complex sampling solutions. The general rule of thumb in differential privacy is that the more data-dependent the solution the greater the sensitivity, the more noise is needed to provide privacy for the data subjects, and therefore the worse the utility of the data. This frequently has the effect of providing a bias for simpler solutions or solutions that are based on outside information.

Stratified sampling also introduces another fundamental problem for differential privacy - that of weighting. The most common solution to stratified sampling is to weight the records to counter the sampling mechanisms to better make the sampled population a representative group for the population as a whole. A record with a weight of 100 is intended to stand in for 100 people, but under differential privacy this also multiplies their impact on any query by 100. The re-weighting allows for more complex sampling strategies such as purposefully oversampling strata that are known to be heterogeneous or less likely to respond but unable to be stratified further, or small populations where sampling error causes problems (e.g. American Natives in the ACS [21]). These types of sampling modifications can be easily countered in weighting, for example if a given stratum is sampled at twice the rate as others, then the individuals in that stratum will have weights that are assumed to be half of those individuals in other stratum. Weighting is a fundamental problem for differential privacy because weighting is done to counter data-specific issues, and this means that any solution will be even more data-dependent and therefore need more noise infusion to provide privacy.

3.3 Weighting to known values

Another common strategy is to force the sample to conform to some external or internal information. Conforming to internal information is done by making sure that multiple ways to calculate the same information provide the same or approximately the same answer. There are many strategies for doing this, and while it can impact the privacy guarantee on the data, this problem has been explored and strategies exist for it, a perfect example is the decennial census [25] which uses such strategies for its calculations. An obvious example of such internal consistency is that state totals are forced to sum to the national total. Conforming to outside information is done to increase confidence in the results of the survey and to re-weight the records to make sure the survey is as representative as possible. This is done with the ACS via the Population Estimates Program[23].

We focus our efforts on the use of Iterative Proportional Fitting (IPF) which allows for a contingency table to be weighted to a set of marginal constraints. We found this technique to be of importance since this is the algorithm that the U.S. Census uses as part of the weighting for the ACS[23] as well as being in practice in other surveys. This situation also precludes the use of a naive solution which weights inversely to the sampling rate since this data is frequently to controlled to different attributes than are used in the stratified sampling.

3.3.1 Differential Privacy under weighting to known values

The need to incorporate external information that is not assumed to be private would normally not provide much of a problem for differential privacy since frequently incorporating this information can be considered a form of post-processing. However, this circumstance is different in that we need to incorporate both information that is assumed to be public as well as the direct values from the data which causes problems for global sensitivity. Our work on this problem (Chapter 7) has led us to focus instead on smooth sensitivity, since one of the possible datasets that must be covered in the global sensitivity case is that of a single respondent (or more generally an arbitrarily small number of respondents). This case is the one that drives the sensitivity to useless levels of noise required.

3.4 Missing Data

Missing data or missing values are a problem that plague all forms of surveys but the issues are particularly apparent in sample based surveys. As the individuals are already standing in for large groups the more missing data there is the worse the utility of the results. This is compounded by the fact that for many attributes there are implications and biases in non-response, for example there is a large bias in socio-economic status with regards to missing data[26].

3.4.1 Differential Privacy under Missing Data

Our work in Chapter 6 allows for some query results from allocated missing data but the practical impacts of the results remain to be seen. Ultimately, the problem of missing data is either very simple, for missing data strategies that focus on other information provided by the record without looking at other records, to complex and potentially irreducible, where the need to allocate and replicate values is included. The more complex strategies for missing data are obviously used to reduce bias in the sample but the need to weigh such modifications against the impact to privacy and where that trade-off exists remains to be seen.

Part III

CONTRIBUTIONS

4. EVALUATION OF FORMAL PRIVACY UTILITY

Most of the current focus of the differential privacy community is on adding privacy after the modifications to the data have occurred. This is easy to understand in the original context in which differential privacy found itself, a way of adding privacy to a query answering system without needing to modify the underlying data. However, this assumption fails in the case of complex surveys. As we've seen from some of the background and as we found in our own research the modifications that we make to the data, and the modifications required to maintain privacy, are choices that must have their respective privacy considerations weighed. In this way differential privacy must influence the data itself if it is to maintain sufficient privacy for the participants while not requiring so much noise as to render any results useless. We highlight that, especially for complex surveys, the choice frequently required by privacy is not simply the application of differential privacy mechanism "A" or differential privacy mechanism "B" but rather the choice of differential privacy mechanism follows directly from modifications made to the data in the hope of improving the statistical results.

With that consideration in mind, we find that focusing on three, seemingly obvious, questions helps to evaluate and highlight the impact of differential privacy, particularly in complex surveys: What are we actually measuring? How are we measuring "it"? What generalities can we make?

We will briefly outline how these questions impact the evaluation of differential privacy and highlight how our experience with this domain can help expose some common miscommunication and limited evaluations.

4.1 What are we measuring?

The common comparison/measure assumed in differential privacy literature is that of improving the ε for a given result or reducing the noise added by changing the method by which noise is introduced. This assumes that the common choice in privacy is that of "privacy" vs "privacy". We make the assumption that some form of privacy is to be applied and as such we want to provide the least amount of noise while making a sufficient guarantee. While this focus is useful for the pure idea of privacy, we find it frequently is the wrong focus for the current state of the domain. We find that much of the current miscommunication is based on the assumption of adding noise to provide privacy where previous users assumed no noise and "no privacy". We can see this in the lawsuit[27] filed against the US Census bureau over the "new" application of privacy measures. That lawsuit, and many of the comments surrounding it, also focused on the definition of privacy and the guarantee of differential privacy. The argument from the state of Alabama is that differential privacy will "skew"[27] the data used for redistricting and that the Census Bureau is violating its mandate when providing the modified data. This argument implicitly ignores that previous results used for redistricting had methods to protect privacy and other alterations to improve data quality, instead the focus is on the "new" use of privacy technology. We believe that not enough focus has been paid to these types of comparisons. Obviously, this type of focus/evaluation is not the simplest but we find that adding context to the numbers helps greatly with this metric. As we explored above, we should contrast the noise added by differential privacy with that added by the statistical modification itself to better provide understanding of the full scope of the impact, both the modification and the privacy required.

In our work on missing data (Chapter 6), we found the direct comparison was essentially that of "no privacy" vs "privacy" because, as far as we know, our work was the first to attempt to solve the problem of missing data in a sampling context. This leads to the situation where noise is being applied where it wouldn't have been earlier. Context was important when considering the results of this work because the total impact was not simply the noise infusion (which increases variance) but also the bias reduction we got from imputing the missing data to better reflect the truth. To illustrate the issues we are highlighting we will dig into this work as a running example in the sections below.

4.2 How are we measuring?

The typical approach of singling out and quantifying the noise required for differential privacy, in isolation, often leaves data users dissatisfied. This data feels noisier in their opinion than the data they are used to, oftentimes because noise was always present (in some form), but it was not explicit.

Ultimately, each modification done to improve the data, since they are almost all data-dependent, needs to be protected by differential privacy. This means that an evaluation that focuses solely on the noise from privacy is missing an important component - the noise (or reduction) from the statistical method that the mechanism is trying to account for. For example, in our work with missing data we saw a significant bias reduction by imputing the data even if the noise required to cover the sensitivity increased. These two modifications are not separate issues in a practical sense.

Most modifications work to either reduce bias or variance in responses, possibly with a potential increase in the other, sometimes referred to as a bias/variance trade-off. We found evaluation of privacy in complex surveys to be much better understood in the broader context of all (or as many as can be modeled) sources of noise. As mentioned above, this is because the important choice in our k-NN smooth sensitivity algorithm is not (currently) our algorithm vs another algorithm, it is our algorithm, with its privacy guarantee, or ignoring the missing data. We can see from other papers[24] that sometimes the better statistical modification can be swamped by the amount of noise needed to protect it.

It is important to model as much of the noise as possible. We found that having a dataset robust enough to provide a nominal “ground truth” and then implementing as many of the pertinent parts of the pipeline to be the best solution.

We will discuss the general process. Note that our work is focused on complex surveys and our discussion assumes some number of statistical modifications that complicate the differential privacy process

4.2.1 Dataset

The goal with determining a dataset is to find something that can provide a ground truth that is a sufficient representation of the real dataset. It should be noted that we assume we are unable to do these experiments on the real dataset since releasing the results would violate privacy, so we focus on discussion of sufficient alternatives. An exact ground truth or an exact match to the real data is not needed. We provide examples for alternate datasets that can be used for privacy experiments. Examples will focus on governmental statistics since these are released regularly and providing for privacy mechanisms for these releases in an ongoing challenge and open area for research.

Data from an equivalent but geographically different locale may provide the data but consideration must be made about subtle differences between geographies in terms of definitions [28]. This problem is well known and it is possible to find sources that have attempted to homogenize the differing definitions to allow for direct comparison. While ideally these differences have no impact they can result in bias in the results and must be considered when finding a representative dataset.

Data from a different time period is also a good target for a dataset. While this type of dataset does have uses, caveats must be made about the changing definitions (similarly to the previous example). It should be noted that the IPUMS project provides for a homogenized version of the data to avoid many of the technical differences between years for Census data[29] this can help to avoid some of the problems with temporally different data. Another potential complication with temporally different datasets is the context for such datasets. A good example of a dataset that is publicly available and relatively robust is the previous Census records for a given country. In the United States the most recent publicly available complete Census as of writing is the 1940[30] Decennial Census. However, when using temporally different datasets we must consider the context of that data and how it compares to the current context. For example experiments about veteran status would not likely find a good dataset in the 1950 or 1920 Census records since the impact of WWII and WWI respectively. There is likely a significant difference in the demographics of veteran status immediately after a draft for the war compared to the current decades long all-volunteer military that the US currently has.

The last popular option for a surrogate data is the use of a partially or purely synthetic dataset. These datasets are gaining popularity for release since ideally they provide little to no privacy implications for the respondents since no real records are released. The strength of these datasets will depend greatly on the process by which they were created, the exact details of which may not be public. These datasets do tend to avoid more of the translation and temporality/context problem, however.

While it seems unnecessary to point out that a surrogate dataset will not match the real dataset it is important to note that the statistical modifications that must be protected frequently aim to reduce some form of real or assumed error in the data.

Continuing with our missing data example, our experiments used the 1940 Disclosure Avoidance Research[30] release provided by the US Census Bureau. Our focus for the experiment was on the attributes: age, employment status, gender, marital status, education, relationship to householder, and salary. We had to make sure that the attributes were all incorporated in the "short form" portion of the Census (answered by everyone) because later Censuses moved some of the attributes, particularly salary, to the randomly sampled "long form" Census that only a sample of the population answers. While these attributes were represented in the 1940 data as well as the current ACS data that we would use for modeling non-response, there was some extra translation that needed to done. We needed to homogenize the data between the two datasets because we knew that we would be using the current ACS data for modeling missing data, discussed more below. The differences in the datasets meant that some of the attributes were more specific in one dataset or the other. Section 6.5.1 provides more detailed discussion of the various modification required for the datasets to match attributes.

4.2.2 Model Error

As the statistical modifications done to the data are done to counter some form of known or expected error, it is important to model as much of the data generation process as possible when creating experiments, particularly focusing on the areas of the process where the errors are generated from. Obviously, in real circumstances modeling all errors is not feasible but the aim to create a reasonable approximation of the real data generation process is important. For example, sampling noise is one of the most common forms of noise for most surveys. The exact impact is not known for the one sample taken for the survey but the overall likely and expected impact on the results is well studied. Incorporating this noise in a realistic manner helps to provide a scale by which the other noise/error can be evaluated, in our missing data paper we found that the sampling noise swamped the noise from differential privacy in most circumstances. This is not to argue that more noise is fine, obviously the ultimate goal is always as little noise as possible while maintaining the privacy guarantee but, especially in "no privacy" vs "privacy" discussions, it can help motivate discussions if the full context of the error is known.

The modeling process itself is not always obvious since the full impact and particulars of the error are frequently assumed from other data. Sometimes the error is directly known and can be accounted for in the generation process without any more steps. The stratified sampling that is done for complex surveys is one of these types of errors. Since the sampling rates are frequently known or an approximate set of sampling rates can be used it is easy to incorporate this in the data generation process. Other types of error are always going to be an approximation, particularly anything related to the responses from sampled participants. Missing data is a key example of one such type of error. In such cases we found it useful to look for surrogate datasets where this type of error is better expressed. The same caveats from above still apply but depending on the setting the risk may be minor. When considering models for this, it is key that the models be stochastic in nature. Stochastic models allow for multiple simulations of the process to be done that will create statistically similar but distinct datasets.

Missing data, or equivalently non-response error, is one of the types of errors that cannot be known without feedback from the survey participants, as it is a direct consequence of their action or inaction. While the 1940 does nominally have some missing data, in that there is a special value for the salary to imply the value was missing, we felt a more modern understanding of missing data was important since our method would be applied to current surveys. For this we used the 2016[31] and the 2017[32] 1-year PUMS data since these datasets have the benefit of having flags for missing values. We learned a number of models for non-response on the 2016 data and then validated them using the 2017 data. These datasets were close enough to have the same definitions for attributes though both were homogenized the same way as the 1940 data to for ease of using the model. We used the best model trained to create a set of data from the 1940 Census with missing data roughly in line with the current expectations. By doing our experiments this way we were able to provide both a ground truth (by ignoring the missing data) as well as the impact of the sampling and non-response error that were our focus for the experiment.

4.2.3 Simulate

This is likely the most obvious step but once a dataset and the data generation process are adequately approximated, then multiple simulations should be run. This is particularly important with smooth sensitivity and other non-global sensitivity schemes (which are likely to be more common in complex surveys given the nature of such surveys). In missing data donor imputation the noise depends on the largest possible donor. Obviously this depends on the data in terms of which records are assigned to be missing an attribute (in the simulation) but also depends on how the donors match up to those incomplete tuples. Since such assignments are deterministic (at least in our case) it is possible to predict the entirety of the impact of the process but we found that simply running the experiment multiple times not only gave a sufficient understanding of both the bias and the variance.

The key to simulation is to gain an understanding of the of the impact both of the errors (e.g., sampling error, nonresponse error, coverage error) and the full impact of the noise infusion from differential privacy. As all of these generative processes are stochastic in nature, multiple runs are needed to give an understanding of the full impact on variance and bias. We can run the experiments many times to provide plausible but distinct datasets with which to run our modifications and privacy mechanisms to provide results.

4.2.4 Presentation

It is important when presenting the data that an understanding of the noise from differential privacy is expressed, since that is the focus of our research. However it is equally important that sufficient context for this noise is expressed as well. It has become obvious that there are other audiences that are watching the differential privacy literature that might not always understand the relative noise infusion from various aspects, instead focusing solely on the noise “added” from differential privacy.

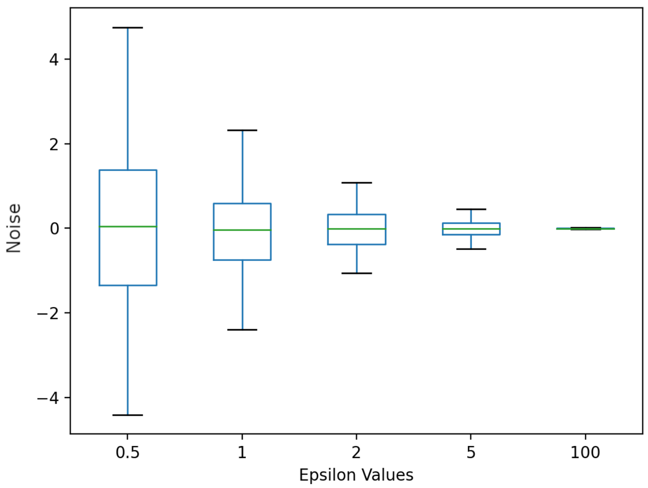

Figure 4.1. The box and whisker plot are one of the more common forms of graphs in the literature. These plots tend to focus solely on the noise infusion from differential privacy. This is the common focus of most graphs in the literature. These graphs gives an idea of the impact on variance for various values of ε. This graph uses the Laplace mechanism as an example but other more complicated mechanisms will frequently feature a graph providing this same type of information.

Figure 4.1 provides the typical information and focus for differential privacy with the direct focus on the mechanism and how the choice of ε effects the results directly. While this figure is one of the simpler forms of this type of graph the information in most papers is focused on discussion about accuracy or effectiveness of the mechanism. As we can see from this graph it is possible to get a narrow focus only on the newly introduced noise from the privacy mechanism. If we instead allow for some context common to sampling surveys, by incorporating some of the common forms of error, we get more information and can produce graphs akin to Figure 4.2 which places the noise from the privacy mechanism embedded in the noise from a 10% sample. Typically a 10% sample is larger than most sample based surveys can guarantee but even then we can see that for Laplace noise provided by our mechanism the sampling noise dwarfs the noise infusion from privacy quickly, anything above $\varepsilon \geq .05$ would not see a noticeable difference in noise from the inclusion of privacy. Such low values of ε can help to assuage fears from users on the hindrance to their results from privacy. The threshold, and relative impact, will vary greatly depending on the mechanism but providing this context can help data curators and data users to understand the likely impact of the privacy mechanism in a more concrete form. If we have chosen our dataset with care we can also incorporate a ground truth like in Figure 4.3 which allows for us to discuss the relative bias reduction that the survey methodologies can provide while still providing sufficient information about the variance introduced by our privacy mechanisms. In each of these variant figures we are still providing the same information about the privacy mechanism but the greater context allows data users to understand the relative impact of the privacy mechanism with regards to errors and situations that statistics has historically dealt with and are more understood by the data users. In this way we can still provide the same information to researchers focused on the differential privacy aspect. This will allow readers from other disciplines to understand the practical impact from differential privacy on their query results.

Our experiments provide us the ability to highlight both the bias and variance of the errors (sampling and non-response) that we were considering. This meant that we could provide the context for the privacy mechanism to show the beneficial impact while still noting the increase in variance. Our ability to establish a ground truth allowed us to determine the

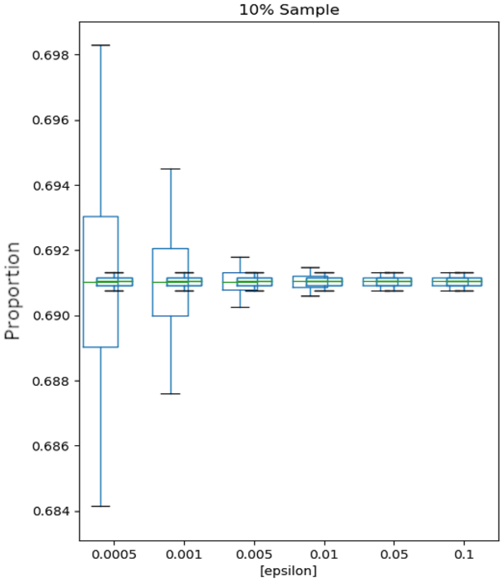

Figure 4.2. This graph, taken from our work in Chapter 5, helps to provide context for the noise infusion from differential privacy. The left offset box (the larger one in most circumstances) incorporates both the sampling noise and the noise from differential privacy. The right offset box is strictly the sampling error from those same samples. As we can see from this graph that even Laplace noise with $\varepsilon = .05$ the sampling noise swamps the noise from the privacy mechanism. It is important to note that the point with which the differential privacy noise is over shadowed will depend on the sampling rate and the privacy mechanism applied but this information is important for other fields to understand the noise in context and for future data curators to understand.

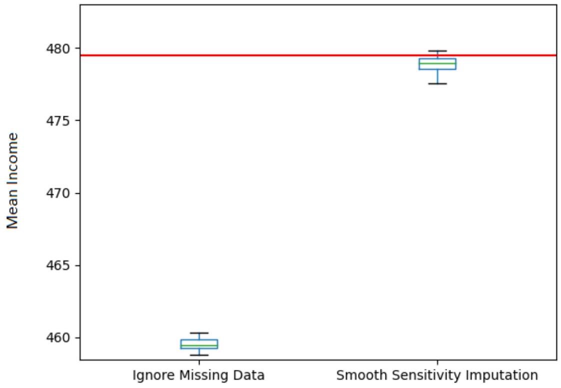

Figure 4.3. This graph, taken from our work in Chapter 6, helps to provide even more context by incorporating a ground truth to show the bias in the results as well as the variance of the mechanism and possible error, though in this circumstance since we were didn't run our experiments on a sample. As many of the most important statistical modifications are done to counter bias, it is important to be able to highlight both variance as well as bias reduction from the modifications.

bias in the results if we were simply to ignore the missing data instead. Our presentation highlights the bias reduction and the noise infusion from differential privacy, though for simplicity the figures in Chapter 6 do not include sampling. Figure 4.3 highlights the ground truth with the singular line across the results while providing the sampling error in the form of the "ignored" category. All of the figures featured in Chapter 6 incorporate the ground truth to highlight any bias that results from ignoring the presence of missing data.

Other forms of measurement are operational considerations. Smooth sensitivity solutions are powerful in theory but frequently difficult to calculate a sufficient upper bound and can at times be computationally infeasible to do so. Many data curators have limited computational power which can influence the choice of statistical modification and privacy infusion. Providing context on this, especially in the case of smooth sensitivity, will also provide needed context to those looking to adopt the strategies being researched.

4.3 What generalities can we learn?

Current work in the field tends to focus on specific questions that are driven from a theoretical side, we solved problem A let's solve A' now. But frequently the very places where differential privacy is going to be applied are those agencies that employ statisticians to do sophisticated modifications of the data to improve the results, the very modifications that have been largely ignored and are only now gaining focus in some parts of the formal privacy literature.

These modifications will need to be considered and privacy solutions are likely to be tailored to be modification. This means that statistical modifications should now be applied with an eye towards the privacy needed to cover for their impact. The work of Bun et al[24] provides a perfect example where the more "optimal" solutions can require noise which outweighs their beneficial impact without modification. It is likely that the simpler statistical modifications are going to be easier to protect until such a time as the field has had time to mature, though it is likely that some of the most complex modifications will irreducibly add so much noise as to render their results unusable.

It is important to note that as differential privacy is applied to more and more spaces that the level of knowledge and sophistication we can assume about the data curator will go down. This can have mixed results on the correct application of these methods but it will result in less and less ability for the data curators to directly defend the results of the privacy modifications. While not an important issue in the short term where formal privacy in complex surveys is still the domain of large institutions/governments with the help of researchers, it is important that many of the issues, particularly with the "introduction" of privacy where none was assumed before (whether that belief was accurate or not), are discussed early and with sufficient context for the users of the data. As the recent legal arguments over the 2020 Decennial Census[27] have shown, there is currently no small amount of animosity about adding strong privacy guarantees to the data.

5. A PARTITIONED RECODING SCHEME FOR PRIVACY PRESERVING DATA PUBLISHING

Material From: 'C. Clifton, E Hanson, K. Merrill, and S. Merrill, A Partitioned Recoding Scheme for Privacy Preserving Data Publishing, Privacy in Statistical Databases, published 2020, Springer International Publishing'

5.1 Introduction

We present a random partitioning approach that can be applied with many existing privacy schemes. In many cases, this provides the same privacy guarantee as treating the entire dataset as a whole, a similar concept to parallel composition definition introduced in Section 2.2.3. We focus in particular on differential privacy. In this setting, depending on how the partitioning is done, there can be subtle differences in how privacy budgets from different partition elements are combined. We explore and clarify these issues, in particular emphasizing the difference between bounded and unbounded neighboring databases. We also show that when the differentially private results computed on each partition element are released to distinct, non-colluding data users, the secrecy of which individuals fall into which partition elements allows us to make use of *amplification*[22].

The layout of this chapter is as follows: In Section 2.2.1, we gave an overview of basic definitions and necessary background. In Section 5.3, we formally define partitioned preprocessing and show that under differential privacy, this provides equivalent privacy to treating the dataset as a whole. We give both analytical and computational examples using counting and proportion queries in Section 5.4. Finally, we extend the results of amplification to bounded differential privacy and discuss their significance to partitioned preprocessing in Section 5.5.

5.2 Definitions

We recall the definition of parallel composition from Section 2.2.3 and provide the definition in more specific detail as our work expand parallel composition to bounded differential privacy.

Definition 5.2.1 (Parallel Composition)**.** *We say that a privacy guarantee* $\mathcal{P}$ *satisfies* parallel composition *if, for all sanitization schemes* $\mathcal{A}_1, ..., \mathcal{A}_n$ *meeting* $\mathcal{P}$ *and disjoint datasets* $D_1, ..., D_n$ *(i.e., any individual appearing in any of the* D_{i} *appears in only that dataset), the union* $\bigcup_{\mathrm{i}=1}^{n} \mathcal{A}_{\mathrm{i}}(D_{\mathrm{i}})$ *satisfies the privacy guarantee* $\mathcal{P}$*.*

ε-differential privacy is known to satisfy parallel composition [10]:

Lemma 5.2.1. Let $D_1, \ldots, D_n$ be disjoint datasets, that is, datasets that can be assumed to have no individual in common. Let $\mathcal{A}$ satisfy ε-DP. Let $\mathbf{D} = (D_1, \ldots, D_n)$, considered as a vector of datasets, and let $\mathbf{D}'$ be any other vector with $\|\mathbf{D} - \mathbf{D}'\|_1 = 1$. Then for any outcome $\mathbf{S} \subseteq \mathrm{Range}(\mathcal{A})^n$,

$$P(\mathcal{A}(\mathbf{D}) \in \mathbf{S}) \leq \mathrm{e}^{\varepsilon} \cdot P(\mathcal{A}(\mathbf{D}') \in \mathbf{S}).$$

There is a subtle caveat to the above parallel composition statement for differential privacy: The datasets must be *given* as disjoint datasets. Specifically, a change to one of the datasets is not allowed to affect the others. This is not quite the same as being able to partition a dataset along the input domain and then apply the anonymization technique. In the latter case, the following shows that the privacy parameter doubles.

Lemma 5.2.2. Let $S_{\mathrm{i}}, \mathrm{i} = 1, \ldots, n$ be a partitioning of the input domain, and let $M_{\mathrm{i}}, \mathrm{i} = 1, ..., n$ satisfy ε-DP. Given a dataset D, the mechanism that returns the sequence $M_{\mathrm{i}}(D \cap S_{\mathrm{i}})$ satisfies 2ε-DP.

It is again important to note that the reason for the 2ε term is that we are using *bounded* differential privacy. This means two of the subdatasets S_{i} can change with a single change to the dataset D. For example, if we partition based on whether each tuple represents a minor, an adult under 65, or a senior citizen, we could impact two partition elements by replacing a minor with a senior citizen. On the other hand, with unbounded differential privacy, the addition or deletion of a tuple only impacts a single partition element, meaning ε-differential privacy is satisfied as in Lemma 5.2.1. This key distinction lead to two of the earlier papers on differential privacy giving two different parallel composition theorems corresponding to Lemmas 5.2.1 [18] and 5.2.2 [10] due to different definitions of neighboring databases.

5.3 Partitioned Preprocessing

The main contribution of this work is to establish a random partitioning scheme (partitioned preprocessing) that leads to ε-differential privacy in the *bounded* case (see Theorem 5.3.1). In other words, we show the result of Lemma 5.2.1 also applies to bounded differential privacy when the distinction between partition elements is not data-dependent.

We note that this is not the first work to look at partitioning in a mechanism for differential privacy. Ebadi, Antignac, and Sands propose partitioning-based approaches to deal with limitations in PINQ [33]. Their work assumes the unbounded case (where Lemma 5.2.1 applies); their approach could be a potential application of our work in the case where we *cannot* assume unbounded sensitivity.

We start with our definition.

Definition 5.3.1 (Partitioned preprocessing)**.** *Let D be a dataset of size n. Choose a random decomposition $\mathbf{n} = \{n_i\}$ of n into positive integers from any distribution, and choose a permutation π on n letters uniformly at random. Let D_π denote the dataset whose ℓ-th entry is the $\pi^{-1}(\ell)$-th entry of D. Denote by $D_{\pi,1}$ the first n_1 elements of D_π, by $D_{\pi,2}$ the next n_2 elements, and so on. We refer to the collection of datasets $\{D_{\pi,i}\}_i$ as the* partitioned preprocessing *of D.*

We remark that this definition is consistent for any distribution on the decompositions of n. For example, the sizes of the partition elements, $\mathbf{n}$, can be fixed ahead of time. This corresponds to taking the distribution that selects $\mathbf{n}$ with probability 1.

The idea behind partitioned preprocessing is to apply a mechanism (or set of mechanisms) $\mathcal{A}$ that satisfies some privacy definition to each partition element separately. When the definition satisfies parallel composition, it is often the case that this preserves the original privacy guarantee.

5.3.1 Differential Privacy under Partitioned Preprocessing

As we have seen in Lemma 5.2.2, under bounded differential privacy, naively partitioning a dataset and ensuring that the use of each partition element satisfies ε-differential privacy only guarantees 2ε-differential privacy. We now show that under partitioned preprocessing, if we satisfy ε-DP for each partition element, we still satisfy ε-DP overall.

Theorem 5.3.1 (Partitioned Preprocessing for ε-DP)**.** *Let D be a dataset with n elements and let j be a positive integer. Choose a decomposition $\mathbf{n}$ of n with j elements based on any distribution and choose a permutation $\boldsymbol{\pi}$ on n elements uniformly at random. Consider the partitioned preprocessing of the dataset $D_{\boldsymbol{\pi}}$ into j pieces $\{D_{\boldsymbol{\pi},\mathrm{i}}\}_{1\le \mathrm{i}\le \mathrm{j}}$. For $1 \le \mathrm{i} \le \mathrm{j}$, let $\mathcal{A}_{\mathrm{i}}$ be a mechanism which satisfies ε_{i}-DP. Apply $\mathcal{A}_{\mathrm{i}}$ to the piece $D_{\boldsymbol{\pi},\mathrm{i}}$, and return the (ordered) list of results. Then the scheme $\mathcal{A}$ returning $(\mathcal{A}_1(D_{\boldsymbol{\pi},1}), \dots, \mathcal{A}_{\mathrm{j}}(D_{\boldsymbol{\pi},\mathrm{j}}))$ satisfies ε-DP, where $\varepsilon := \max_{1\le \mathrm{i}\le \mathrm{j}} \varepsilon_{\mathrm{i}}$.*

We note that when $\mathrm{j} = 1$, this reduces to applying the mechanism $\mathcal{A}_1$ to the dataset as a whole with (total) privacy budget ε_1.

Proof. Let $D = (x_1, \dots, x_n)$ be a dataset with n elements. For convenience, set $t := x_1$ and let t' be another tuple that is not necessarily in D. Let $D' := (t', x_2, ..., x_n)$. For a fixed positive integer j, we denote by $P_{n,\mathrm{j}}$ the set of all decompositions of n into j pieces, i.e., all choices $\{n_1, \dots, n_{\mathrm{j}}\}$ satisfying that $n_{\mathrm{i}} \in \mathbb{Z}_{>0}$ and $\sum_{\mathrm{i}=1}^{\mathrm{j}} n_{\mathrm{i}} = n$. Let μ denote a probability measure on $P_{n,\mathrm{j}}$, and represent an arbitrary element by $\mathbf{n}$. Let S_n denote the collection of all permutations on n elements, so that $|S_n| = n!$. Given $\mathbf{n} \in P_{n,\mathrm{j}}$ and an index $1 \le \ell \le \mathrm{j}$, let $S_{\mathbf{n}}^{\ell}$ denote the set of permutations $\boldsymbol{\pi}$ which place t in the partition elements $D_{\boldsymbol{\pi},\ell}$. We note that $S_{\mathbf{n}}^{\ell}$ is precisely the set of $\boldsymbol{\pi} \in S_n$ for which $n_1+n_2+...+n_{\ell-1} < \boldsymbol{\pi}^{-1}(1) \le n_1+n_2+...+n_{\ell}$, that the collection $\{S_{\mathbf{n}}^{\ell}\}_{1\le \ell\le n}$ gives a disjoint decomposition of S_n, and that $|S_{\mathbf{n}}^{\ell}| = (n-1)!n_{\ell}$.

Now fix intervals $T_1, \dots, T_{\mathrm{j}}$ in $\mathbb{R}$. Then

$$\begin{aligned} P(\mathcal{A}(D) \in T_1 \times \cdots \times T_{\mathrm{j}}) &= \sum_{\mathbf{n}\in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\boldsymbol{\pi}\in S_n} P\left(\bigcap_{k=1}^{\mathrm{j}} \mathcal{A}_k(D_{\boldsymbol{\pi},k}) \in T_k\right) \\ &= \sum_{\mathbf{n}\in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\boldsymbol{\pi}\in S_n} \prod_{k=1}^{\mathrm{j}} P(\mathcal{A}_k(D_{\boldsymbol{\pi},k}) \in T_k) \\ &= \sum_{\mathbf{n}\in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\ell=1}^{\mathrm{j}} \sum_{\boldsymbol{\pi}\in S_{\mathbf{n}}^{\ell}} \prod_{k=1}^{\mathrm{j}} P(\mathcal{A}_k(D_{\boldsymbol{\pi},k}) \in T_k). \end{aligned}$$

Now for a fixed $\mathbf{n} \in P_{n,\mathrm{j}}$, $\boldsymbol{\pi} \in S^{\ell}_{\mathbf{n}}$, and a fixed index k, if $k \neq \ell$ we have $D_{\pi,k} = D'_{\pi,k}$, and hence

$$P(\mathcal{A}_k(D_{\pi,k}) \in T_k) = P(\mathcal{A}_k(D'_{\pi,k}) \in T_k).$$

On the other hand, if $k = \ell$, by the definition of ε-differential privacy, we have

$$P(\mathcal{A}_\ell(D_{\pi,\ell}) \in T_\ell) \leq \mathrm{e}^{\varepsilon_\ell} P(\mathcal{A}_\ell(D'_{\pi,\ell}) \in T_\ell).$$

Therefore, returning to our formula,

$$\begin{aligned}
P(\mathcal{A}(D) \in T_1 \times \cdots \times T_{\mathrm{j}}) &\leq \sum_{\mathbf{n} \in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\ell=1}^{\mathrm{j}} \sum_{\pi \in S^{\ell}_{\mathbf{n}}} \mathrm{e}^{\varepsilon_\ell} \prod_{k=1}^{\mathrm{j}} P(\mathcal{A}_k(D'_{\pi,k}) \in T_k) \\
&\leq \mathrm{e}^{\varepsilon} \sum_{\mathbf{n} \in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\pi \in S_n} \prod_{k=1}^{\mathrm{j}} P(\mathcal{A}_k(D'_{\pi,k}) \in T_k) \\
&= \mathrm{e}^{\varepsilon} \sum_{\mathbf{n} \in P_{n,\mathrm{j}}} \frac{\mu(\mathbf{n})}{n!} \sum_{\pi \in S_n} P\left(\bigcap_{k=1}^{\mathrm{j}} \mathcal{A}_k(D'_{\pi,k}) \in T_k \right) \\
&= \mathrm{e}^{\varepsilon} P(\mathcal{A}(D') \in T_1 \times \cdots \times T_{\mathrm{j}}).
\end{aligned}$$

$\square$

The crucial difference between the above theorem and Lemma 5.2.2 is that the partitioning is done in a *data-independent* manner. This is what allows us to preserve the privacy parameter ε instead getting 2ε. The key is that the partitioning of the data is completely determined by the sizes of the partition elements $\mathbf{n}$ and the permutation $\boldsymbol{\pi}$ used to order the elements; once we condition on those choices, replacing $t \mapsto t'$ therefore only affects a single partition element, and hence introduces only a single factor of e^{ε}.

We note that because the preprocessed extension of any differentially private mechanism is again differentially private, all the usual results about post-processing, sequential and parallel composition, and the like still hold for this extension.

5.4 Detailed Examples

In this section, we provide analytical and computational examples of results obtained from using partitioned preprocessing and differential privacy. This aims to enlighten both the process of our method and the utility of the noisy query results it generates.

For differentially private mechanisms, the notion of utility is frequently measured by the amount of noise the mechanism needs to add in order to satisfy the definition. We will focus here on the variance of the noise, the idea being that the smaller the variance the more faithful the statistics, and hence the more useful for data analysis and data mining.

For specificity, fix a positive integer n. We recall that the (global) *sensitivity* of a query f at size n is

$$\Delta_n f := \max_{d(D,D')=1} |f(D) - f(D')|, \qquad |D| = |D'| = n.$$

In other words, $\Delta_n f$ represents the maximum impact that any one individual can have on the answer to the query f between an arbitrary dataset of size n and its neighbors. We note that for the purposes of computing sensitivities at two different sizes, the set of possible tuples of the datasets are taken to be the same, but the size of such allowed datasets has changed.

We will focus for the remainder of this section on the well-known Laplace mechanism [10], a differentially private mechanism that returns for a query f on a dataset D (with $|D| = n$) the answer $f(D)$ *plus* some noise, drawn from a Laplace (also called a double exponential) distribution with mean 0 and variance $2(\Delta_n f/\varepsilon)^2$, where ε is our privacy budget.

We recall that in differential privacy, one needs to set the number of queries allowed on the dataset ahead of time, since the amount of noise added to each query must compensate for the total number of queries (otherwise by asking the same query repeatedly, one could appeal to the (Weak) Law of Large Numbers to obtain an accurate estimate of $f(D)$, since the noise has mean 0 and would "cancel out" in the long run). Traditionally, if we are going to allow k queries on the dataset, we would add noise corresponding to ε/k.

One way to interpret our expansion is the following: Instead of immediately splitting the privacy budget ε, we first prepartition (at random) the dataset D into j pieces, $D_{\pi,1}, \ldots, D_{\pi,\mathrm{j}}$, for some permutation $\boldsymbol{\pi} \in S_n$ and sizes $\mathbf{n} = (n_1, \ldots, n_{\mathrm{j}})$. We then ask some number of queries on each piece. The motivating idea is that each piece of the dataset can be used to answer the queries of a different data user. For example, suppose a data user wishes to know the answer to $k' < k$ queries. Then on that data user's piece of the data, we add only ε/k' noise to the query responses.

For example, suppose a data user has been 'assigned' the partition element $D_{\pi,\mathrm{i}}$ and that they wish to run only some query f. The mechanism then returns $f(D_{\pi,\mathrm{i}}) + \mathrm{Lap}(\Delta_{n_\mathrm{i}} f/\varepsilon)$ and the data user is free to use this information as they wish. This includes sharing this information with other data users, as the guarantee of Theorem 5.3.1 makes no assumption that a data user is given access to only the results of the queries asked on a single partition element. When this is the case, we can leverage amplification, as detailed in Section 5.5.

We observe that the noise added to satisfy differential privacy is smaller under partitioned preprocessing than when all queries are answered on the entire dataset. This is primarily because partitioning the data is effectively the same as sampling, which introduces noise that may outweigh the benefit of having larger ε. We will demonstrate this in Section 5.4.2. Another potential complication comes from the need to account for query responses being computed on a smaller sample than D. For example, counting queries will need to be scaled by a factor of $|D|/|D_{\pi,\mathrm{i}}|$. This has the impact of scaling the sensitivity by the same factor. Since we are using bounded differential privacy, the value of $|D|$ and $|D'|$ are the same, and under partitioned processing, the size of the partitions $|D_{\pi,\mathrm{i}}|$ do not change. As a result, these values are sensitivity 0 and can be released without compromising privacy.

The remainder of this section is aimed at understanding the variance (both analytically and empirically) of proportion queries answered under partitioned preprocessing.

5.4.1 Variance of Proportion Queries

Suppose that f is a proportion query, that is, $f(D)$ is the *proportion* of records that satisfy some attribute. Then we have $\Delta_m f = 1/m$ for all m. As in the previous section, we suppose $|D| = n$ and we have partitioned D into $D_{\pi,1}, \ldots, D_{\pi,\mathrm{j}}$ with $|D_{\pi,\mathrm{i}}| = n_\mathrm{i}$. For simplicity, we assume we are running a total of j queries.

If we run f on the entire dataset D, we return $f(D) + \mathrm{Lap}(\Delta_n f/(\varepsilon/\mathrm{j}))$, whereas under partitioned preprocessing, we would run f on only $D_{\pi,\mathrm{i}}$, and return $f(D_{\pi,\mathrm{i}}) + \mathrm{Lap}(\Delta_{n_\mathrm{i}} f/\varepsilon)$. Since probabilistically we expect that $f(D_{\pi,\mathrm{i}}) \approx f(D)$, we compare the variances in each of these cases.

In the former, we have variance equal to $2(\Delta_n f/(\varepsilon/\mathrm{j}))^2 = 2\frac{\mathrm{j}^2}{|D|^2\varepsilon^2}$. In the latter case, we recall that we have two independent sources of noise: that coming from the partitioning (hypergeometric) and that coming from differential privacy (Laplacian). The total variance is the sum of the two:

$$\begin{aligned}\sigma^2 &= \frac{p(1-p)}{|D_{\pi,\mathrm{i}}|}\frac{|D|-|D_{\pi,\mathrm{i}}|}{|D|-1} + 2(\Delta_{n_\mathrm{i}} f/\varepsilon)^2 \\ &= \frac{p(1-p)}{|D_{\pi,\mathrm{i}}|}\frac{|D|-|D_{\pi,\mathrm{i}}|}{|D|-1} + \frac{2}{|D_{\pi,\mathrm{i}}|^2\varepsilon^2} \approx \frac{p(1-p)}{|D_{\pi,\mathrm{i}}|}\frac{|D|-|D_{\pi,\mathrm{i}}|}{|D|-1} + \frac{2\mathrm{j}^2}{|D|^2\varepsilon^2}.\end{aligned}$$

We see that in this case, the variance under partitioned preprocessing is always slightly larger than in the traditional scheme by an additional factor of

$$\begin{aligned}\frac{p(1-p)}{|D_{\pi,\mathrm{i}}|}\frac{|D|-|D_{\pi,\mathrm{i}}|}{|D|-1} &\approx \frac{p(1-p)}{|D_{\pi,\mathrm{i}}|}\frac{(\mathrm{j}-1)|D_{\pi,\mathrm{i}}|}{|D|-1} \\ &= \frac{p(1-p)(\mathrm{j}-1)}{|D|-1} \leq \frac{\mathrm{j}-1}{4(n-1)}.\end{aligned}$$

Even though the partitioned preprocessing has greater variance than the original mechanism in the proportion case, we note that the noise coming from partitioning (sampling) is identical for our method and the traditional method of partitioning. Thus when a data curator has opted for the use of partitioning and bounded differential privacy, using partitioned preprocessing rather than the traditional method effectively doubles the privacy budget without changing the noise coming from partitioning.

5.4.2 Empirical Demonstration

Perhaps an easier way to understand the utility of random partitioning is through the impact on real queries. We give an example of a proportion query on a 1940 U.S. Census dataset released for disclosure avoidance tests [30]. This dataset consists of over 100M individuals, and avoids the complex use methods (and consequent difficulty of determining sensitivity) of many other public use microdata sets.

We use a typical proportion query, the proportion of adult males to all males. This is run on a 10% sample of the data, and then on an element of a partition of that sample into 10 pieces (essentially a 1% sample) with a correspondingly higher privacy budget (Figure 5.1); the box plots show the distribution of results over 1000 runs.

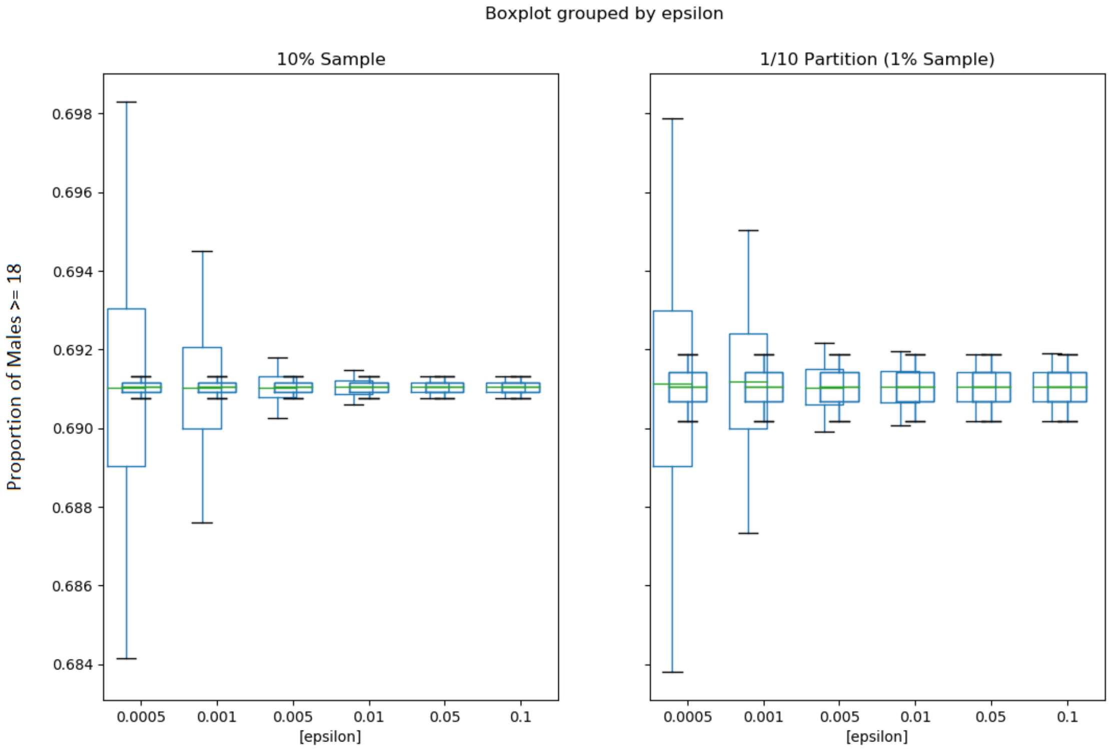

Figure 5.1. Distribution of query results with and without differential privacy on a 10% sample database (left) vs. partition of the 10% sample into 10 pieces with correspondingly higher budget (right): Proportion of Adult Males to All Males in the 1940 Census. The outer/left box at each value of ε represents the result with differential privacy and the inner/right box at each value of ε represents the result without differential privacy (sampling error only).

The idea is that if we had other queries, they could be run on other partition elements without affecting this query – but if run on the full dataset, they would need shares of the total privacy budget to achieve the same privacy level, thus requiring a smaller value of epsilon.

We see that as expected, for small values of ε, the impact of partitioning is small relative to the noise added to protect privacy. The distribution of results for the partitioned data and the full 10% sample is basically the same for $\varepsilon \leq 0.001$. (Note that even at $\varepsilon = 0.0005$, the majority of the time the error is less than 0.5%.) For larger values of ε, little noise is required for differential privacy, and the sampling error dominates – but even so, at a 90% confidence interval, the error is well under 0.5% for the partitioned approach.

5.5 Amplification

We now give an overview of the amplification technique of [22] and discuss its relationship with partitioned preprocessing. We prove a version of amplification for *bounded* differential privacy; the original paper is for unbounded differential privacy and the proof does not easily generalize.

The idea behind amplification is that sampling with known probability before answering queries greatly increases the privacy budget for queries answered on the sample. More precisely, we have the following, which we prove in the appendix.

Theorem 5.5.1 (Amplification for bounded ε-DP)**.** *Let $\mathcal{A}$ satisfy ε-DP. Let D be a dataset with $|D| = n$ and choose an integer $n' < n$. We denote $\beta = n'/n$. Choose a subdataset $D' \subset D$ with $|D'| = n'$ uniformly at random. Then the mechanism which returns $\mathcal{A}(D')$ satisfies ε'-DP, where*

$$\varepsilon' = \ln\left(\frac{\mathrm{e}^{\varepsilon}\beta + 1 - \beta}{1 - \beta}\right).$$

Our statement differs from that in [22] by the factor of $1 - \beta$ in the denominator, which comes from our use of bounded differential privacy. It also adds the assumption that the size n' of the subdataset is fixed for a given β and n, which will be subtly used in the proof. Moreover, it is possible that $\varepsilon' > \varepsilon$, in which case the mechanism still satisfies the original ε-DP guarantee. Table 5.1 shows some examples of this as well as examples where amplification significantly increases the privacy budget.

As an immediate consequence of Theorem 5.5.1, we have the following.

Corollary 5.5.1. Let $\mathcal{A}$ satisfy ε-DP. Let D be a dataset with n elements and let j be a positive integer. Fix a decomposition $\mathbf{n}$ of n with j elements and choose a permutation $\boldsymbol{\pi}$ on

Table 5.1. Examples of maximum values of β for amplification to provide savings and values of ε' when $\beta = 1/10$ for various values of ε.

ε	.1	.5	1	2	5	10
max. β for $\epsilon' < \epsilon$	.087	.282	.387	.464	.498	.499
ε' at $\beta = 1/10$	N/A	.17	.26	.60	2.86	7.80

n elements uniformly at random. Choose an index $1 \leq \mathrm{i} \leq \mathrm{j}$ and let $D_{\pi,\mathrm{i}}$ be the i-th partition element of the partitioned preprocessing of D. Then the mechanism that returns $\mathcal{A}(D_{\pi,\mathrm{i}})$ satisfies ε'-DP, where ε' is as defined in Theorem 5.5.1 with $\beta = n_{\mathrm{i}}/n$.

The caveat to this result is that the mechanism is *only* returning the result computed on $D_{\pi,\mathrm{i}}$. Indeed, if we return $(\mathcal{A}(D_{\pi,\mathrm{i}_1}), \ldots, \mathcal{A}(D_{\pi,\mathrm{i}_k}))$, we have effectively changed our parameter β to $\frac{1}{n}(n_{\mathrm{i}_1} + \cdots + n_{\mathrm{i}_k})$. Amplification could still provide a benefit in this case, but only if this new sampling rate is still much smaller than 1. Overall, amplification with partitioned preprocessing is most appropriate when the results of queries run on different partition elements are released to distinct data users who do not share their results.

5.6 Proof of Amplification for Bounded Differential Privacy

We first fix some notation. Let $\mathcal{A}$ be a ε-differentially private mechanism. Let D be a dataset with $|D| = n$ and choose an integer $n' < n$. Fix some tuple $t \in D$. We denote by Y_t the set of all subdatasets $D_s \subset D$ with $|D_s| = n'$ and $t \in D_s$ and by N_t the set of all subdatasets $D_s \subset D$ with $|D_s| = n'$ and $t \notin D_s$. We observe

$$|Y_t| = \binom{n-1}{n'-1} \qquad |N_t| = \binom{n-1}{n'}.$$

For $D' = D \setminus \{t\} \cup \{t'\}$ a neighbor of D, we define Y'_t and N'_t analogously. We observe that $N_t = N'_t$.

We will need the following lemma.

Lemma 5.6.1. Let $t \in D$ and $S \subset \mathrm{range}(\mathcal{A})$. Then

$$\sum_{D_s \in Y_t} \frac{P(\mathcal{A}(D_s) \in S)}{|Y_t|} \leq \mathrm{e}^{\varepsilon} \sum_{D_s \in N_t} \frac{P(\mathcal{A}(D_s) \in S)}{|N_t|}.$$

Proof. For each $D_s \in Y_t$, we can replace the tuple t by any of the $n - n'$ tuples in $D \setminus D_s$ to create a dataset in N_t that is a neighbor of D_s. Similarly, given any $D_s \in N_t$, we can replace any of the n' tuples in D_s with t to create a dataset in Y_t that is a neighbor of D_s.

Now consider

$$(n - n') \sum_{D_s \in Y_t} P(\mathcal{A}(D_s) \in S)$$

as counting each $D_s \in Y_t$ with multiplicity $n - n'$. Thus we replace the $n - n'$ copies of $D_s \in Y_t$ in this sum with its $n - n'$ neighbors in N_t. By differential privacy, each such change causes the probability to grow by no more than e^{ε}. Moreover, each dataset in N_t will occur n' times in the new sum. Thus

$$(n - n') \sum_{D_t \in Y_t} P(\mathcal{A}(D_t) \in S) \leq \mathrm{e}^{\varepsilon} n' \sum_{D_t \in N_t} P(\mathcal{A}(D_t) \in S).$$

The result now follows from the observation that $\frac{n'}{n-n'} = \frac{|Y_t|}{|N_t|}$. □

This lemma captures the reason we have assumed the size of the subdataset to be fixed. In the unbounded case, if we delete a tuple t to pass from dataset D to D', then for each $D_s \subseteq D$ with $t \in D_s$, there is a unique $D'_s \subseteq D'$ with $d(D_s, D'_s) = 1$. Lemma 5.6.1 is our generalization of this fact to the unbounded case.

We are now ready to prove Theorem 5.5.1, which we restate here for convenience.

Theorem 5.5.1. Let $\mathcal{A}$ satisfy ε-DP. Let D be a dataset with $|D| = n$ and choose an integer $n' < n$. We denote $\beta = n'/n$. Choose a subdataset $D' \subset D$ with $|D'| = n'$ uniformly at random. Then the mechanism which returns $\mathcal{A}(D')$ satisfies ε'-DP, where

$$\varepsilon' = \ln\left(\frac{\mathrm{e}^{\varepsilon}\beta + 1 - \beta}{1 - \beta}\right).$$

Proof. Let $S \subset \text{range}(\mathcal{A})$ and let $D' = D \setminus \{t\} \cup \{t'\}$ be a neighbor of D. We will use the law of total probability twice, conditioning first on whether $t \in D_s$ (i.e. on whether $D_s \in Y_t$ or $D_s \in N_t$), then on the specific subdataset chosen as D_s. This gives

$$\begin{aligned}
P(\mathcal{A}(D_s) \in S) &= \beta \sum_{D_t \in Y_t} \frac{P(\mathcal{A}(D_t) \in S)}{|Y_t|} + (1-\beta) \sum_{D_t \in N_t} \frac{P(\mathcal{A}(D_t) \in S)}{|N_t|} \\
&\leq \beta \mathrm{e}^{\varepsilon} \sum_{D_t \in N_t} \frac{P(\mathcal{A}(D_t) \in S)}{|N_t|} + (1-\beta) \sum_{D_t \in N_t} \frac{P(\mathcal{A}(D_t) \in S)}{|N_t|} \\
&= (\beta \mathrm{e}^{\varepsilon} + 1 - \beta) \sum_{D_t \in N_t} \frac{P(\mathcal{A}(D_t) \in S)}{|N_t|} \\
&= (\beta \mathrm{e}^{\varepsilon} + 1 - \beta) \sum_{D'_s \in N'_t} \frac{P(\mathcal{A}(D'_s) \in S)}{|N'_t|}.
\end{aligned}$$

by the lemma and the fact that $N_t = N'_t$. By analogous reasoning, we have

$$\begin{aligned}
P(\mathcal{A}(D'_s) \in S) &= \beta \sum_{D'_t \in Y'_t} \frac{P(\mathcal{A}(D'_t) \in S)}{|Y'_t|} + (1-\beta) \sum_{D'_t \in N'_t} \frac{P(\mathcal{A}(D'_t) \in S)}{|N'_t|} \\
&\geq (1-\beta) \sum_{D'_t \in N'_t} \frac{P(\mathcal{A}(D'_t) \in S)}{|N'_t|}.
\end{aligned}$$

Combining these two inequalities yields

$$P(\mathcal{A}(D_s) \in S) \leq \frac{\beta \mathrm{e}^{\varepsilon} + 1 - \beta}{1-\beta} P(\mathcal{A}(D'_s) \in S).$$

□

5.7 Conclusion

Independently sanitizing subsets of a dataset raises a number of challenges. By introducing partitioned preprocessing, we have provided an expansion of the tighter bound on the differential privacy parallel composition theorem for unbounded neighboring datasets to support bounded differential privacy in certain cases. Moreover, when the sizes of the partition elements are known and the results of the mechanisms run on different elements are given to distinct, non-colluding data users, we can leverage amplification to increase the privacy budget.

This does come at some cost in result quality, although not significant for small values of ε or high sensitivity queries. Even so, there are still logistical advantages to using random partitioning with differential privacy. One example is if we are only collecting a subset of the data to begin with (sampling). Randomly partitioning the population before sampling gives us the ability to collect further samples and treat the privacy budget independently. More structured partitioning (e.g., stratified sampling based on individual characteristics) can put us into the 2ε situation, where allowing queries to two partition elements requires they share a privacy budget. Another example is when a user would like to employ a differentially private mechanism that is superlinear in runtime; in these cases, partitioning off a small portion of the data can enable a differentially private answer with substantial savings on time investment. One place such mechanisms show up is in the context of smooth sensitivity [19]. This is a concept that replaces global sensitivity with a weighted maximum of the impact of a single individual in datasets sufficiently close to the real dataset, which can often be substantially smaller. As a concrete example, the mechanism used in [34] to build a Naïve Bayes classifier using smooth sensitivity has a runtime of $\mathcal{O}(n^2)$.

The idea of random partitioning also applies to techniques based on generalization. For example, assuring k-anonymity, ℓ-diversity [12], or t-closeness [9] on each partition element provides the same guarantee (with the same parameter) on the dataset as a whole.[1] With techniques based on generalization, the possibility exists that queries may be answerable on some partition element, whereas the necessarily granularity could be generalized away globally. Further discussion is left for future work.

---

[1] ↑There is some subtlety here, as k-anonymity under global recoding is not assured, even if each partition element satisfies it.

6. DIFFERENTIALLY PRIVACY K-NEAREST NEIGHBOR MISSING DATA IMPUTATION

6.1 Introduction

Missing data poses substantial challenges for data analysis and machine learning. In practice, as with survey data, missing data tends to be biased [35], [36] and as a result, analysis based on only collected data gives poor results.

A common solution to this problem is *missing data imputation*. The usual first step is to use known data about an individual to impute any missing values (for example, if an individual leaves their employment status unanswered but reports income from a job, their employment status can be deduced). When this fails, it is common to instead use known values from similar individuals (an approach also referred to as *allocation* [36], [37]). This poses a privacy challenge, as a single individual's value is reflected in multiple records, making it more difficult to keep that value private.

If an individual is a "donor" for several individuals with a missing value, then their impact on the result for many queries (how much the result would change if that individual were removed) increases with the number of individuals who take their value, requiring substantially more noise to cover their impact.

A basic approach to differential privacy requires a worst-case view: the amount of noise added is based on the worst possible scenario that can be constructed. In the case of missing data imputation, we can imagine a scenario like Figure 6.1: If we assume the value of the attribute `color` is imputed from the nearest neighbor (in two-dimensions) with a known value, deleting the black circle will change the imputed color of every square. In particular, the number of gray individuals goes from 1 to the size of the dataset. This essentially requires sufficient noise to hide the entire range of possible answers.

Such pathological cases seem unlikely to occur in practice. As we've discussed previously smooth sensitivity[19] allows us to base our noise not on a global worst case, but on worst cases bearing resemblance to the actual dataset. As more changes to the data are needed to get to a bad case, the impact of the bad case on the noise needed goes down. We now note the key challenge: For a given query, we need to compute the maximum possible change

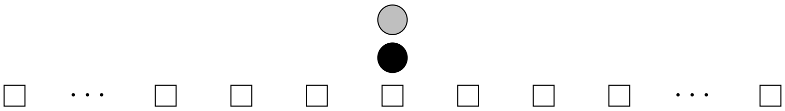

Figure 6.1. Sample dataset with high global sensitivity. Each individual has attributes SHAPE and COLOR. The squares are missing COLOR, so the color of every square is initially imputed as black. After deleting the black circle, the color of every square is imputed as gray.

in that query result after a given number of changes to the dataset. Simply computing all possible changes to determine the maximum is $|T|^s$, where $|T|$ is the number of possible individuals in the universe, and s is the number of steps we are checking; typically this is not a feasible computation. This is difficult, and there have been few examples of problems that can be addressed by smooth sensitivity beyond those worked out in the original paper, most quite recent: see for example synthetic graph generation[38], outlier detection[39], random forests[40], naive bayes[41], and PCA[42].

6.1.1 Problem Statement and Summary of Results

In this chapter, we establish a *smooth upper bound* (in the sense of [19]) for k-nearest-neighbor imputation, regardless of how the data of the neighbors translates into the imputed value (e.g., averaging, majority vote, etc.). Our examples and mechanisms focus on frequencies (often referred to as counts in the differential privacy literature), proportions, means, and variances. The results easily extend to any queries where the impact of an individual is at most multiplicative in the number of times they are a “donor” (sums and correlations are examples).

Our use of smooth sensitivity to achieve differentially private k-nearest neighbor data imputation gives a dramatic reduction in variance compared to using the global sensitivity. While missing data imputation does give substantially higher variance than simply throwing out individuals with missing data, we do get the reduction in bias that is the primary benefit of doing missing data imputation. While this bias/variance trade-off is difficult to quantify analytically (it is heavily dependent on the distribution of the missing data), we do give a synthetic, but realistic, example in Section 6.5 that is reflective of an actual large-scale, high-dimensional survey.

The contents this paper are as follows. In Section 6.2 we discuss related work on allocation, private data cleaning, and low rank estimation. The main technical material of this paper is found in Section 6.3, where we develop the theoretical framework necessary for combining imputation with differential privacy. In Section 6.4, we use this theoretical framework to describe differentially private mechanisms for releasing frequencies, proportions, sample means, and sample variances from imputed data. Finally, we provide empirical demonstration of these mechanisms in Section 6.5

6.2 Related Work

Imputation of missing values for a *data incomplete* tuple is generally done by computing a value based on known values for that individual (edit rules), modeling the value based on known values for other individuals (e.g., using the mean), or inserting a known value from a donor *data complete* tuple (allocation) [36]. We first discuss allocation approaches and why they pose challenges for differential privacy. We also overview existing methods for private data imputation.

6.2.1 Allocation for Missing Data Imputation

Using a value from a donor is more likely to give a legal value (e.g., avoiding the family with 2.4 children). Methods such as *hot deck imputation* have a long history. Although more complex missing data imputation methods exist, such as those using models, hot deck and the associated k-Nearest Neighbors are still frequently used and have some benefits. They are simpler to implement and explain to non-technical users, they avoid invalid answers (as stated above), and they avoid many of the implicit assumptions about the underlying data that model based solutions assume that may or may not be accurate. We should also note that our focus on hot-deck and related solutions was mainly governed by their continued use in governmental data. In its basic form, hot deck uses the last seen value for a missing data item [43]. This works if data is missing at random, but often missing data is biased towards certain classes of individuals. To address this concern, donors are chosen to be similar to the incomplete data item with regards to certain known attributes, under the assumption that individuals similar on those attributes have similar values for the missing data. Unfortunately, this can also lead to biased results, if an individual with an unusual value ends up being the donor to many missing individuals. This has given rise to complicated procedures, such as sorting hot deck data to try to get donors who are similar to the individuals with missing data [43], or the mechanism used in the U.S. Census Bureau's

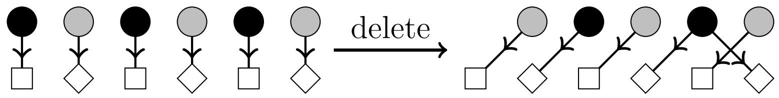

Figure 6.2. High sensitivity of hot deck imputation. Each individual has attributes shape and color. The distance metric is Euclidean distance in $\mathbb{R}^2$, and the missing values are imputed in order from left to right. The arrows represent values being imputed. Deleting the left black circle from the dataset on the left changes the imputed value of color for every data incomplete tuple (shown on the right).

American Community Survey [37]. The latter identifies a similar individual as a donor, but then discards that individual for a period of time before allowing it to be reused as a donor.

Methods that use such a donor have obvious implications for differential privacy. The sensitivity of a query must account for the fact that queries covering imputed missing data may actually be dependent multiple times on the value from the donor. As shown in the example in the introduction (Figure 6.1), this can give arbitrarily high sensitivity.

Methods such as hot deck imputation (or that use some of its concepts, such as [37]) would intuitively seem to be well-suited for differential privacy, since an individual donor's contributions are limited. Unfortunately, such techniques are sensitive to the order in which tuples are processed [43]. The following example shows that this makes global sensitivity arbitrarily large.

In Figure 6.2 we again consider individuals with attributes shape and color and impute values of color based on distance in two dimensions. With hot deck, a potential donor cannot impute on two consecutive individuals. Thus if we delete (from the left dataset) the leftmost black circle, the imputed value of color of each of the data incomplete individuals is swapped (as shown in the right dataset). The (imputed) number of light-gray squares is changed from 0 to the total number of squares with this single deletion. As a result, it is impractical to satisfy the differential privacy guarantee with such approaches.

6.2.2 Private Data Cleaning Methods

There have been some methods proposed for privatized data cleaning, including differentially private mechanisms. The privacy definition used in [44] allows multiple parties to compare their information without disclosing it to each other, allowing the imputation of missing data. This addresses disclosure during the process, but unlike differential privacy, does not address the potential disclosure from the result of the protocol. To our knowledge, other privacy cleaning methods address problems other than missing data, such as correcting values that are presumed to be erroneous.

InfoClean [45] is an automated cleaning mechanism that satisfies a form of information theoretic privacy, but is not shown to satisfy differential privacy. It also addresses a somewhat different problem. The assumption is that a given tuple is known to be erroneous, and is fixed by comparing with a similar tuple retrieved from a master database; the only privacy considered is that of data in the Master database. PACAS [46] addresses a similar problem, using a k-anonymity based privacy metric.

The differentially private data cleaning methods PrivateClean [47] and PrivClean [48] support human-in-the-loop cleaning. Both enable an expert to specify rules for data cleaning, and ensure that the result of a query is differentially private, which may include the impact of the expert looking at data to generate the rules. As such, this is really not comparable with our approach.

6.2.3 Low Rank Estimation

Another common solution to missing data imputation in the differentially private setting is to view the dataset as a matrix with missing entries, and produce a differentially private matrix that is similar in to the original with respect to some norm. Examples of such approaches include [49]–[51]. This is a powerful method, and unlike the present paper, releases a synthetic dataset on which an arbitrary number of queries can then be run. However, these recent efforts have predominantly focused on *recommender systems*, which are homogeneous in their variables and do not exhibit the structure and relationships between variables that allocation maintains. This stands in stark contrast with many surveys, for instance the U.S. Census Bureau's American Community Survey (ACS) which we will focus on below. Some of these attempts have also used a weaker privacy guarantee than that of (pure) differential privacy, which we satisfy in this paper.

6.3 Deterministic Data Imputation

The goal of this section is to establish the necessary theoretical framework to apply differential privacy to a dataset with (k-nearest neighbor) imputation. We do so by examining the relationship between the local sensitivity of a query and the ability to change the donor(s) of a data incomplete tuple.

6.3.1 Theoretical Framework

We start with a few definitions. While these definitions are quite abstract, several well-known motivating examples are listed in Example 6.3.2. Let T denote the set of all possible tuples, in which missing values are allowed. Throughout this section, we fix a set of attributes A. We let T_c denote the subset of T consisting of tuples with no missing values and T_i the subset of T consisting of tuples missing responses to A. We refer to T_c as the set of *complete tuples* and T_i as the set of *incomplete tuples*. We assume that these are the only two possibilities (although the results still hold if not all complete tuples are donors, or all incomplete tuples must have missing values provided).

From now on, we assume that a dataset cannot contain two identical tuples. (This assumption, which is necessary for the proofs that follow, is not difficult to achieve in reality. For example, multiple individuals can be identical up to some unique record label.) This allows us to define $D_c := D \cap T_c$ and $D_\mathrm{i} = D \cap T_\mathrm{i}$ so that $D = D_c \cup D_\mathrm{i}$ and $D_c \cap D_\mathrm{i} = \emptyset$.

For any positive integer k, we define $\binom{T_c}{k} := \{S \subset T_c : |S| = k\}$ and denote by $\mathsf{ord}(T_c)$ the set of total orders on T_c.

For A a set of attributes, we refer to the set of possible responses to A as $\mathsf{resp}(A)$ and the responses given by $y \in T_c$ as $\mathsf{resp}_y(A)$. For example, if the attributes in A are age and voter registration status, then

$$\mathsf{resp}(A) = \{(\mathrm{i}, \text{not registered}) : \mathrm{i} = 0, 1, \ldots, 122\} \cup \{(\mathrm{j}, \text{registered}) : \mathrm{j} = 18, 19, \ldots, 122\}.$$

Definition 6.3.1. Let k be a positive integer.

1. A *deterministic k-nearest neighbor imputation scheme* for the set of attributes A is a pair (f, g) where $f : T_{\mathrm{i}} \to \mathsf{ord}(T_c)$ and $g : \binom{T_c}{k} \to \mathsf{resp}(A)$.

2. Fix a pair (f, g) as above and let $D = D_c \cup D_{\mathrm{i}}$ be a dataset. Then for $x \in D_{\mathrm{i}}$, we denote by $\mathsf{don}_D(x)$ the set of k elements of D_c which are smallest with respect to the ordering $f(x)$. We call $\mathsf{don}_D(x)$ the *donor set* (or just *donor* if $k = 1$) of x. The value imputed to x for the attributes in A is then $g(\mathsf{don}_D(x))$.

The following are some common examples that fit into this framework.

Example 6.3.2.

1. *Nearest Neighbor:* Let $k = 1$. For $x \in T_{\mathrm{i}}$, define $f(x)$ to be the order (with deterministic tie-breakers if necessary) on T_c given by some distance metric (e.g., Hamming distance on a certain set of attributes). For $y \in T_c$, define $g(y) = \mathsf{resp}_y(A)$. Then (f, g) is the imputation scheme that copies the responses to A from the nearest data complete tuple in the dataset.

2. *Mean:* For $x \in T_{\mathrm{i}}$, define $f(x)$ to be the order (with deterministic tie-breakers if necessary) on T_c given by some distance metric (e.g., Hamming distance on a certain set of attributes). For $Y \in \binom{T_c}{k}$, define

$$g(Y) = \frac{1}{k} \sum_{y \in Y} \mathsf{resp}_y(A).$$

   Then (f, g) is the imputation scheme that imputes the average of the responses to A from the k nearest data complete tuples in the dataset.

3. *Majority:* For $x \in T_{\mathrm{i}}$, define $f(x)$ to be the order (with deterministic tie-breakers if necessary) on T_c given by some distance metric (e.g., Hamming distance on a certain set of attributes). For $Y \in \binom{T_c}{k}$, define $\mathrm{i}(Y)$ to be the most common value of $\mathsf{resp}_y(A)$ over $y \in Y$. Then (f, g) is the imputation scheme that imputes the most common response to A from the k nearest data complete tuples in the dataset.

Remark 6.3.3.

1. *Given a dataset* $D = D_c \cup D_{\mathrm{i}}$ *and a data incomplete tuple* $x \in D_{\mathrm{i}}$*, it is necessary for the order* $f(x)$ *to be on* T_c *(the set of* all possible *data complete tuples) rather than* D_c *(the set of data complete tuples actually in* D*). Otherwise, if we make a change to the dataset by adding a new data complete tuple* $y \in T_c \setminus D_c$*, we would have no way to determine whether* y *should replace (one of) the donor(s) of* x*.*

2. *As in the above examples, it is often not necessary to specify or compute the entire function* $f : T_{\mathrm{i}} \to \mathsf{ord}(T_c)$*. Indeed, specifying a metric to determine the nearest neighbor (for example Hamming distance on a certain set of attributes with the difference in record labels as a tiebreaker) is enough that one could construct the function* f *if desired. This will be the approach used in our empirical analysis.*

3. *Definition 6.3.1 is made so that a single change to the dataset can only change at most one donor of a data incomplete tuple. That is, if* $d(D, D') = 1$ *and* $x \in D_{\mathrm{i}} \cap D'_{\mathrm{i}}$*, then* $|\mathsf{don}_D(x) \cap \mathsf{don}_{D'}(x)| \geq k - 1$*.*

For the remainder of this section, we fix a positive integer k and deterministic k-nearest neighbor imputation scheme (f, g). We remark that the definitions and results below do not make explicit reference to the functions f and g, but rather to the sets $\mathsf{don}_D(x)$.

Definition 6.3.4.

1. Let D be a dataset. Given $y \in D_c$, its *set of donees* is

$$\mathsf{don}_D^{-1}(y) := \{x \in D_{\mathrm{i}} | y \in \mathsf{don}_D(x)\}.$$

   That is, $\mathsf{don}_D^{-1}(y)$ consists of those incomplete tuples in D whose values are imputed based on y. For convenience, we will define $\mathsf{don}_D^{-1}(y) = \emptyset$ for $y \in T_c \setminus D_c$.

2. Let D and D' be datasets. Then we denote

$$\mathsf{don}_D^{-1}(D') := \bigcup_{y \in D'_c} \mathsf{don}_D^{-1}(y).$$

3. Let D, D' be datasets. The *donee change* between D and D' is

$$c(D, D') := [D_{\mathrm{i}} \Delta (D')_{\mathrm{i}}] \cup \left[\mathsf{don}_D^{-1}(D \cup D') \Delta \mathsf{don}_{D'}^{-1}(D \cup D') \right].$$

   The first term represents the data incomplete tuples added or deleted as we transform D into D'. The second term represents those incomplete tuples that have their donor set change as we transform D into D'. We note that $c(D, D') = c(D', D)$.

6.3.2 Bounding Donee Changes

We now turn our attention to studying how the function $c(D, D')$ behaves as $d(D, D')$ increases. More precisely, we study the function

$$L_\ell(D) := \max_{\{D' \colon \, d(D, D') = \ell\}} |c(D, D')|.$$

Remark 6.3.5.

1. *Our motivation for studying this function is to construct a smooth upper bound for the local sensitivity of several queries based upon these values. Our definition of $L_\ell(D)$ enables us to use Theorem 6.3.6 later in Section 6.4 to create mechanisms requiring only the computation of $L_1(D)$ for a given dataset.*

2. *The quantity $L_1(D)$ will play a particularly important role in our analysis. Recall that from the unbounded definition of differential privacy, the only changes we allow to the dataset are the addition or deletion of a tuple. $L_1(D)$ can therefore be seen as a maximum over the impacts of such a change.*

The following result describes how the function $L_\ell(D)$ changes as the distance ℓ and dataset D change, and forms the basis of our mechanisms.

Theorem 6.3.6. *Let D be any dataset. Then:*

1. *For all $\ell \in \mathbb{N}$, $L_\ell(D) < L_{\ell+1}(D)$.*

2. *For all $\ell \in \mathbb{N}$, $L_\ell(D) \leq \ell L_1(D)$.*

3. For any dataset D', $L_1(D') \leq L_{d(D,D')+1}(D)$.

Proof. (a) Let D' be a dataset realizing $|c(D,D')| = L_\ell(D)$, and let $x \in T_{\mathrm{i}}$ such that $x \notin D \cup D'$. Then $D' \cup \{x\}$ is a dataset satisfying $d(D, D' \cup \{x\}) = \ell + 1$ and

$$L_{\ell+1}(D) \geq c(D, D' \cup \{x\}) = c(D,D') + 1 > c(D,D') = L_\ell(D).$$

(b) Let D' be a dataset realizing $L_\ell(D) = |c(D,D')|$, and let $D \Delta D' = \{t_1, ..., t_\ell\}$. We will prove this statement by induction on ℓ. For $\ell = 1$, there is nothing to show, so assume the statement holds for $\ell - 1$. We then have

$$c(D,D') = c(D, D_{\ell-1}) \cup (c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})),$$

where $D_{\ell-1}$ is the dataset obtained from D by adding/removing the tuples $t_1, \ldots, t_{\ell-1}$. Hence by the induction hypothesis

$$\begin{aligned} L_\ell(D) &= |c(D,D')| \\ &= |c(D, D_{\ell-1})| + |c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})| \\ &\leq (\ell-1)L_1(D) + |c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})|. \end{aligned}$$

Therefore it suffices to show that

$$|c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})| \leq L_1(D).$$

First observe that if $t_\ell \in T_{\mathrm{i}}$, then $|c(D_{\ell-1}, D')| = 1$, and we are done. Thus we can assume that $t_\ell \in T_c$.

As a set, $c(D', D_{\ell-1})$ represents those incomplete tuples whose donor set changes in the move from $D_{\ell-1}$ to D' and had not changed previously. To simplify notation, denote

$$D \pm \{t_\ell\} = \begin{cases} D \cup \{t_\ell\}, & t_\ell \notin D \\ D \setminus \{t_\ell\}, & t_\ell \in D \end{cases}.$$

We claim

$$c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1}) \subseteq \Big\{x \in D_{\mathrm{i}} : t_\ell \in (\mathsf{don}_D(x))\Delta(\mathsf{don}_{D\pm\{t_\ell\}}(x))\Big\}\,.$$

To see this, let $x \in c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})$. We first note that since t_ℓ is data complete, we have $(D_{\ell-1})_{\mathrm{i}} = (D')_{\mathrm{i}}$. Next, observe that $x \in D_{\mathrm{i}}$. Indeed, if $x \notin D_{\mathrm{i}}$, then x must be equal to some t_{j} and therefore $x \in c(D, D_{\ell-1})$, a contradiction. Since $x \in D_{\mathrm{i}}$ and it does not change donors in the first $\ell - 1$ steps, we have $\mathsf{don}_D(x) = \mathsf{don}_{D_{\ell-1}}(x)$.

Now if $t_\ell \in D$, then $t_\ell \in \mathsf{don}_{D_{\ell-1}}(x) = \mathsf{don}_D(x)$ and we are done. Thus assume $t_\ell \in \mathsf{don}_{D'}(x)$. If $\mathsf{don}_{D'}(x)\setminus\{t_\ell\} \not\subseteq D$ then x changes donors between D and $D_{\ell-1}$, a contradiction. Therefore $t_\ell \in \mathsf{don}_{D\pm\{t_\ell\}}(x)$. This proves the claim.

We conclude that $|c(D', D_{\ell-1}) \setminus c(D, D_{\ell-1})| \leq |c(D, D \pm \{t_\ell\})| \leq L_1(D)$, as needed.

(c) This proof is similar to that of part (2), with one change to be explained later. Let D' be another dataset, and set $\ell := d(D, D')$. Let D'' be the neighboring dataset of D' which realizes $|c(D', D'')| = L_1(D')$. As before, set

$$D\Delta D' = \{t_1, ..., t_\ell\}, \quad D'\Delta D'' = \{t\}.$$

We now break the proof into cases, some of which are trivial. If $t \in T_{\mathrm{i}}$, then $L_1(D') = |c(D', D'')| = 1$, and there is nothing to show. Thus assume that $t \in T_c$. We now have two cases to consider.

(c1) If $t \neq t_{\mathrm{j}}$ for all j, then as in case (2), we have

$$c(D', D'') \subseteq c(D, D''),$$

and hence

$$L_1(D') = |c(D', D'')| \leq |c(D, D'')| \leq L_{d(D,D'')}(D) \overset{(a)}{\leq} L_{\ell+1}(D),$$

where the last inequality holds because $d(D, D'') \leq d(D, D') + 1$ by the triangle inequality and the function $s \mapsto L_s(D)$ is strictly increasing.

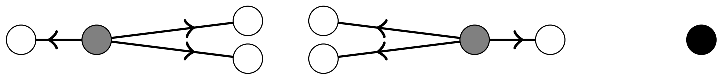

Figure 6.3. A greedy algorithm cannot be used to compute $L_\ell(D)$. See Example 6.3.7 below.

(c2) If $s = t_{\mathrm{j}}$, since the order of those moves does not matter, we can assume without loss of generality that $t = t_\ell$. In other words, the biggest change one can make to the dataset D' is to add (resp. remove) the data complete item we just removed (resp. added) in the previous step. Therefore, $c(D', D'') = c(D_{\ell-1}, D')$, and so

$$L_1(D') = |c(D', D'')| = |c(D_{k-1}, D')| \leq |c(D, D')| \leq L_\ell(D) \overset{(a)}{\leq} L_{\ell+1}(D).$$

□

We now provide an example that shows we can not use a "greedy algorithm" to compute $L_\ell(D)$. More precisely, consider a pair of datasets D, D' with $d(D, D') = 1$ and $L_1(D) = |c(D, D')|$. We would like to be able to say that

$$L_\ell(D) \leq L_1(D) + L_{\ell-1}(D' \setminus c(D, D')).$$

That is, we would like to be able to delete all individuals whose donor has changed in transitioning from D to D', compute $L_{\ell-1}$ on this new dataset, and recover $L_\ell(D)$ from this value. The following example shows that this is not the case.

Example 6.3.7. We consider the dataset shown in Figure 6.3. As in previous examples, individuals have attributes `shape` and `color`. Values of `color` are imputed based on distance in two dimensions and arrows represent donor/donee relationships.

We observe that $L_1(D) = 4$, obtained by adding a black tuple between the four white tuples. Likewise, $L_1(D' \setminus c(D, D')) = 1$, obtained by deleting either gray circle.

On the other hand, we have that $L_2(D) = 6$. This can be realized by deleting both of the gray circles, resulting in all imputing from the right black circle.

This example suggests it is typically not computationally feasible to compute $L_\ell(D)$ for all $\ell \in \mathbb{N}$. We instead use Theorem 6.3.6 to design differentially private mechanisms based only on the value of $L_1(D)$. We discuss computing this value in Section 6.5.2.

6.4 Differentially Private Imputation-based Mechanisms

In this section we describe mechanisms satisfying differential privacy which release query results on imputed datasets. Explicitly, we will use Definition 2.2.4 and 6.3.6 to bound the smooth sensitivity of several queries in terms of the quantity $L_1(D)$. We then use the mechanism in Section 2.2.7 to achieve differential privacy.

Remark 6.4.1. *We recall that the quantity $L_1(D)$ comes from our choice of a completely deterministic k-nearest neighbor imputation scheme; however, the mechanisms make no further reference to this choice. Thus, for simplicity, we shall treat the quantity $L_1(D)$ (or perhaps more precisely the association $D \mapsto L_1(D)$) as fixed for the remainder of this section.*

We will assume from now on that all queries are answered on the imputed dataset. We also remark that these mechanisms only provide benefit when used for queries which are impacted by imputation. In other words, as we have fixed a set of attributes A in our definition of data-complete and data-incomplete tuples, any query which does not involve the attributes in A can and should be answered using standard techniques.

6.4.1 Frequencies and Proportions

We first consider frequency queries (also called counting queries) and proportion queries.

Theorem 6.4.2. *Let q be a frequency query. Then for $\beta \geq \ln 2$,*

$$S(D) = 1 + L_1(D)$$

is a β-smooth upper bound on $LS_q(D)$.

Proof. Let D be a dataset. Let $D \pm \{t\}$ be a neighboring dataset where we have either added or deleted the tuple t. Then t is a donor for at most $L_1(D)$ tuples (in whichever of

D and $D \pm \{t\}$ contains t). This means the largest change between $q(D)$ and $q(D \pm \{t\})$ is $1 + L_1(D)$. That is, $LS_q(D) \leq 1 + L_1(D)$.

Now let

$$U_k(D) = 1 + 2^k L_1(D).$$

Then for any dataset D' with $d(D, D') = 1$ and any $k \in \mathbb{N}$, we have

$$U_k(D) = 1 + 2^k L_1(D) \leq 1 + 2^{k+1} L_1(D') = U_{k+1}(D')$$

by Theorem 6.3.6. Thus by Definition 2.2.4,

$$\max_k \mathrm{e}^{-\beta k} \left[1 + 2^k L_1(D)\right] \tag{6.1}$$

is a β-smooth upper bound on $LS_q(D)$. This converges if and only if $\beta \geq \ln 2$, in which case it converges to $1 + L_1(D)$. □

We emphasize that as a consequence of this theorem, $1 + L_1(D)$ is a $(\ln 2)$-smooth upper bound on $LS_q(D)$, not just an upper bound. The generalized Cauchy distribution from Section 2.2.7 then implies the following (taking $\beta = \ln 2$ and $\gamma = 1 + \frac{\varepsilon}{2\beta}$).

Corollary 6.4.3. *Let q be a frequency query and $\varepsilon > 0$. Then the mechanism which returns*

$$q(D) + \frac{1 + L_1(D)}{\ln 2} \cdot X,$$

where X is sampled from the generalized Cauchy distribution with parameter $\gamma = 1 + \frac{\varepsilon}{2 \ln 2}$, satisfies ε-differential privacy.

Remark 6.4.4.

1. *All of the mechanisms in this section can also be made to satisfy the weaker notion of (ε, δ)-differential privacy by drawing the noise from a Gaussian distribution instead [19].*

2. *The noise added in Corollary 6.4.3 above depends only on ε and $L_1(D)$. Moreover, it is quite possible that $GS_q \neq LS_q(D) = L_1(D) + 1$, in which case the local sensitivity*

and the smooth sensitivity coincide. In such a case, the noise added is proportional to the local *sensitivity of q. Such a phenomenon is also possible for the median query discussed in [19] for certain values of β and certain non-pathological datasets, even if no differentially private mechanism can be based* solely *on the local sensitivity. We also emphasize that the reason β does not appear in our formula is that increasing β beyond* $\ln 2$ *has no impact on the computation of our smooth sensitivity. For $\beta < \ln(2)$, Equation* (6.1) *does not converge.*

We can extend this result to proportion queries since they are quotients of frequency queries. Thus a data-user interested in a proportion could query the numerator and denominator separately and compute the answer as post-processing. We emphasize that in the case that inclusion in the subpopulation of interest is separate from the attributes in A (for example, if the proportion is computed over the whole dataset), then the global sensitivity of the denominator is 1 and the standard Laplace mechanism can be used.

6.4.2 Means

We now address the computation of the mean of an attribute computed over a subpopulation of the dataset. As we are using unbounded differential privacy, releasing a privatized version of the mean is non-trivial, even if computed over the whole dataset, as the size of subpopulation is not public. Our solution is to first query the size of the subpopulation, and use the result as a parameter in the second mechanism (leveraging the sequential composition property of differential privacy [10]).

We consider an attribute Y taking values in some range $[a, b]$. We assume $a \geq 0$, but all arguments generalize to allow $a < 0$ in a straightforward way. The values a and b can be thought of as bottom- and top-coded values; without these, the local sensitivity of the mean is unbounded (regardless of whether there is data imputation). For any tuple x, we denote by $Y_D(x)$ the (possibly imputed) value of the attribute Y for the tuple x.

For a dataset D, we denote by $S \cap D$ the subpopulation over which we wish to compute the mean. We remark that if $d(D, D') = 1$, it is possible that $|(S \cap D)\Delta(S \cap D')| > 1$ if inclusion in the subpopulation is partially determined by attributes in A. We will give two different versions of each result depending on whether or not this is the case.

Now consider two datasets D and D' with $d(D, D') = 1$. Then $D\Delta D'$ contains a single element t. If $t \in D$, then $Y_D(t)$ is well-defined, while $Y_{D'}(t)$ is not (since this means $t \notin D'$). Furthermore, if $t \in D$, then $\mathsf{don}_{D'}^{-1}(t)$ is empty by convention, while $\mathsf{don}_D^{-1}(t)$ may be nonempty. On the other hand, if $t \in D'$, then it is $Y_D(t)$ which is not well-defined and $\mathsf{don}_D^{-1}(t)$ which is defined to be empty. To unify these two cases and increase readability, we therefore denote

$$Y(t) = \begin{cases} Y_D(t) & t \in D \\ Y_{D'}(t) & t \notin D \end{cases} \qquad \mathsf{don}^{-1}(t) = \begin{cases} \mathsf{don}_D^{-1}(t) & t \in D \\ \mathsf{don}_{D'}^{-1}(t) & t \notin D. \end{cases}$$

We note that if $t \in T_{\mathrm{i}}$ (i.e., it is data-incomplete), then $\mathsf{don}^{-1}(t) = \emptyset$. Also observe $|\mathsf{don}^{-1}(t)| \leq L_1(D)$.

We now present the main theorem of this section. In order to interpret the query in Corollary 6.4.5 below as returning a mean, one can consider s as a (noisy) estimate $|S \cap D|$. We emphasize, however, that s is treated as constant (in that it does not depend on the actual dataset D) in the statement of Theorem 6.4.5. See Remark 6.4.7 for additional explanation.

Theorem 6.4.5. *Let s be a positive real number and let $q_s(D) = \frac{1}{s}\sum_{x\in S\cap D} Y_D(x)$.*

1. *If the subpopulation $S \cap D$ does not depend on the (possibly imputed) values of the attributes in A, then for $\beta \geq \ln 2$, a β-smooth upper bound on $LS_{q_s}(D)$ is given by*

$$S(D) = \frac{b + L_1(D)(b-a)}{s}.$$

2. *If the subpopulation $S \cap D$ is determined by the attributes in A, then for $\beta \geq \ln 2$, there is a β-smooth upper bound on $LS_{q_s}(D)$ given by*

$$S(D) = \frac{b\,[1 + L_1(D)]}{s}.$$

Proof. (1) Let D, D' be neighboring datasets with $D \Delta D' = \{t\}$. Suppose first that t is in the subpopulation of interest. Then $S \cap D$ and $S \cap D'$ differ by precisely 1 element. We then have

$$\begin{aligned}
|q_s(D) - q_s(D')| &= \frac{1}{s} \cdot \left| \sum_{x \in S \cap D} Y_D(x) - \sum_{x \in D'} Y_{D'}(x) \right| \\
&= \frac{1}{s} \cdot \left| \pm Y(t) + \sum_{x \in S \cap \mathsf{don}^{-1}(t)} (Y_D(x) - Y_{D'}(x)) \right| \\
&\leq \frac{1}{s} \cdot \left[Y(t) + \sum_{x \in S \cap \mathsf{don}^{-1}(t)} |Y_D(x) - Y_{D'}(x)| \right] \\
&\leq \frac{b + L_1(D) \cdot (b-a)}{s}.
\end{aligned}$$

We note that this bound is larger than the one we would obtain in the case that t is not in the subpopulation of interest. Indeed, in the latter case, $Y(t)$ would not be included in the computation of $|q_s(D) - q_s(D')|$, resulting in a smaller bound. Now let

$$U_k(D) = \frac{b + 2^k L_1(D) \cdot (b-a)}{s}.$$

Then for any $k \in \mathbb{N}$, we have

$$U_k(D) = \frac{b + 2^k L_1(D) \cdot (b-a)}{s} \leq \frac{b + 2^{k+1} L_1(D') \cdot (b-a)}{s} = U_{k+1}(D')$$

by Theorem 6.3.6. Thus by Definition 2.2.4,

$$\max_k \mathrm{e}^{-\beta k} \left[\frac{b + 2^k L_1(D) \cdot (b-a)}{s} \right]$$

is a β-smooth upper bound on $LS_{q_s}(D)$. This converges if and only if $\beta \geq \ln 2$, in which case it converges to $\frac{b+L_1(D)\cdot(b-a)}{s}$.

The proof of (2) is similar. The key difference is that now any tuple in $\mathsf{don}^{-1}(t)$ can move into or out of the subpopulation of interest with the addition or deletion of t. Thus their contributions to the sum can each change by b, rather than $b - a$ as before.

□

We observe that if Y is a binary attribute, then $b = 1$ and $a = 0$, so we recover the frequency query result from the previous section.

By Corollary 2.2.5, we then have the following (taking $\beta = \ln 2$ and $\gamma = 1 + \frac{\varepsilon}{2\beta}$)

Corollary 6.4.6. *Let s be a positive real number and $q_s(D) = \frac{1}{s}\sum_{x \in S\cap D} Y_D(x)$ and $\varepsilon > 0$.*

1. *If the subpopulation $S \cap D$ does not depend on the attributes of A, then the mechanism that returns*
$$q_s(D) + \frac{b + L_1(D) \cdot (b-a)}{s \ln 2} \cdot X,$$
where X is sampled from the generalized Cauchy distribution with parameter $\gamma = 1 + \frac{\varepsilon}{2 \ln 2}$, satisfies ε-differential privacy.

2. *If the subpopulation $S \cap D$ depends on the attributes of A, then the mechanism that returns*
$$q_s(D) + \frac{[1 + L_1(D)] \cdot b}{s \ln 2} \cdot X,$$
where X is sampled from the generalized Cauchy distribution with parameter $\gamma = 1 + \frac{\varepsilon}{2 \ln 2}$, satisfies ε-differential privacy.

As in the previous section, we observe that for some datasets, the noise may be proportional to the local sensitivity.

Remark 6.4.7. *If the value $s := |S \cap D|$ is known, the mechanism in Corollary 6.4.6 can be used directly to return a noisy estimate of the mean $\frac{1}{|S\cap D|}\sum_{x \in S\cap D} Y_D(x)$ which satisfies ε-differential privacy. (This is the assumption and approach we take in the experiments discussed in Section 6.5.) If, on the other hand, this value is not known, we can still release a noisy estimate of the mean which satisfies ε-differential privacy by proceeding as follows. First, we use the mechanism in Corollary 6.4.3 with a privacy budget of $\varepsilon/2$ to release a noisy estimate of (the frequency query) $|S \cap D|$. We denote the output of this mechanism by s', which we may now treat as a fixed constant. We then set $s = \max\{1, s'\}$. (This step is considered post processing and does not impact the privacy guarantee.) Finally, we* *use the mechanism in Corollary 6.4.6 to return a noisy estimate of $q_s(D)$. The result is a noisy estimate for the desired mean which, by the sequential composition theorem, satisfies ε-differential privacy.*

6.4.3 Variances

Recall the sample variance of the attribute Y computed over the subpopulation $S \cap D$ is

$$S_Y(D) = \frac{1}{|S \cap D| - 1} \sum_{x \in S \cap D} (Y_D(x) - \overline{Y}(S \cap D))^2.$$

The goal of this section is to describe a mechanism for returning a differentially private estimate of this value. We do so using a strategy similar to that of the previous section. That is, we replace the data-dependent values $|S \cap D|$ and $\overline{Y}(S \cap D)$ with fixed constants s and y. As with means in Remark 6.4.7, to interpret the output of the resulting mechanism as the variance of Y, one can either treat these constants as true known values or as noisy estimates obtained from the mechanisms in the previous two sections.

As in the previous section, we make the technical assumption that the attribute Y takes values in some interval $[a, b]$ with $a \geq 0$.

Theorem 6.4.8. *Let s and y be real numbers which satisfy $s > 1$ and $a \leq y \leq b$. Let*

$$q(D) = \frac{1}{s-1} \sum_{x \in S \cap D} (Y_D(x) - y)^2 \qquad m = \max\left\{(a-y)^2, (b-y)^2\right\}.$$

Then for any $\beta \geq \ln 2$, there is a β-smooth upper bound on $LS_q(D)$ given by

$$S(D) = \frac{m}{s-1}[1 + L_1(D)].$$

We note that the upper bound is the same regardless of whether inclusion in $S \cap D$ is independent of the attributes in A.

Proof. Let D, D' be neighboring datasets with $D \Delta D' = \{t\}$. As in the proof of Theorem 6.4.5, we assume that t is in the population of interest since this will result in a larger upper bound than if t is not in the subpopulation. Now let $x \in \mathsf{don}^{-1}(t)$. We first suppose that, with the addition or deletion of t, the tuple x moves into (respectively out of) the population of interest. Then x contributes $(Y_D(x) - y)^2$ to the sum in $q(D)$ (resp. $q(D'))$ and 0 to the sum in $q(D')$ (respectively $q(D)$). Thus the change in its contribution is bounded above by m. If, on the other hand, t does not move into (respectively out of) the population of interest (as is the case when inclusion in $S \cap D$ is independent of the attributes in A), then its contribution to the sum changes from $(Y_D(x) - y)^2$ to $(Y_{D'}(x) - y)^2$. This change is readily bounded above by m.

Thus, as our proof of Theorem 6.4.5, we have that

$$|q(D) - q(D')| \leq \frac{m}{s-1}\left[1 + L_1(D)\right].$$

Now let

$$U_k(D) = \frac{m}{s-1} \cdot [1 + 2^k L_1(D)].$$

Then for any $k \in \mathbb{N}$, we have

$$U_k(D) = \frac{m}{s-1} \cdot [1 + 2^k L_1(D)] \leq \frac{m}{s-1} \cdot [1 + 2^{k+1} L_1(D')] = U_{k+1}(D')$$

by Theorem 6.3.6. Thus by Definition 2.2.4,

$$\max_k \mathrm{e}^{-\beta k}\left[\frac{m}{s-1}\left[1 + 2^k L_1(D)\right]\right]$$

is a β-smooth upper bound on $LS_q(D)$. If and only if $\beta \geq \ln 2$ this converges to

$$\frac{m}{s-1}\left[1 + L_1(D)\right].$$

$\square$

By Corollary 2.2.5, we then have the following (taking $\beta = \ln 2$ and $\gamma = 1 + \frac{\varepsilon}{2\beta}$).

Corollary 6.4.9. *Let s and y be real numbers which satisfy $s > 1$ and $a \leq y \leq b$, and let $\varepsilon > 0$. The the mechanism that returns*

$$q(D) + \frac{m \cdot [1 + L_1(D)]}{(s-1)\ln 2} \cdot X,$$

where X is sampled from the generalized Cauchy distribution with parameter $\gamma = 1 + \frac{\varepsilon}{2\ln 2}$, satisfies ε-differential privacy.

As in the previous section, we observe that for some datasets, the noise may be proportional to the local sensitivity. This type of construction can be extended to correlation coefficients, taking means and variance as inputs to the query.

6.5 Empirical Demonstration

We show concrete examples of these results using a 1940 U.S. Census dataset released for testing disclosure avoidance methodologies[30]. We test the impact of imputation on proportion and mean queries, showing the bias/variance trade-off that imputation is designed to improve. We used the 1940 Census data as a ground truth. For simplicity and efficiency we show results for the state of Minnesota; as U.S. Census Bureau imputation is done at a state level or finer, this reflects real-world use.

Our experiments impute wage income of individuals, as this and the variables used for imputation in modern counterparts were present in 1940 Census data. This data is of practical importance and therefore it is valuable to elucidate the impact that imputation and this form of differential privacy would have on the resulting data.

6.5.1 Data Creation

While the 1940 Census data does contain missing data, we want to compare against a known ground truth. We instead ignore actual missing data and instead model missing values from data complete tuples to give a ground truth. We leverage the similarity of the variables between the 1940 Census and current American Community Survey (ACS) Public-Use Microdata Samples to simulate missing values.

We first mapped schemas of the 2016 and 1940 data due to differences in specificity between the two surveys. For example, the relationship to householder differs significantly between the 2016 data and the 1940 data, with the ACS providing 18 different responses for “RELP” while the 1940s data “RESPOND” provides only 8 options. This mapping was a joint refinement of the possible responses to various attributes, and was done strictly on the domain of the variables without reference to the data. For example, in homogenizing the relationship to householder attribute, we mapped the responses “boarder”, “roomate”, “other non-related adult” to the single value of “non-related adult” in the 1940 data. In some cases the 1940 data provided more granularity. E.g., the “EMPSTATD” attribute for employment status responses for “not in labor force” in 1940 provide reasons - “housework”, “unable to work”, “schooling”, “other”. Homogenization on both sides merged similar groups to the greatest extent possible.

We trained a model to learn the probability that the income value was missing in the 2016 data based on the homogenized attributes, validating the model on the 2017 1-year PUMS data. We applied the trained model to each data complete record in the 1940 dataset, predicting the likelihood that the data would be missing. Each run of the experiment generated both a new random sample of the data (simulating a sample-based survey), and flagged a new set of tuples as missing income based on the modeled likelihoods. The differentially private results for each approach are computed once per sample, on the same samples. Thus our experiments capture variance based on sampling, randomness in missing data, and the noise required to satisfy differential privacy.

6.5.2 Imputation

The imputation uses the same variables that ACS currently uses to impute wage per [52]. We use 1-Nearest Neighbor, which requires a definition of distance to determine the “closest” neighbor. As our data has a mix of categorical and ordinal attributes, we had to modify the attributes again to allow for a more meaningful idea of distance. We categorized all of the attributes as either ordinal or categorical. The categorical variables were split into a number of binary variables equal to the number of options, e.g., relationship to householder was split into 7 binary values. AGE was modified to use the tens digit as the value, giving results between 0 and 9. We then used Euclidean distance for calculating distances between records. This has the effect of allowing the ordinal variables to preserve their distance while treating categorical attributes as edit distance, albeit with a penalty that the resulting distance is 2 instead of 1. This limits the scope of the attribute and facilitates efficient calculation of $L_1(D)$, the crucial value required to determine the impact of the imputation on privacy.

Our imputation first creates groups of complete tuples and incomplete tuples and then uses a deterministic tie-breaker to create an ordering for all records in the group. This assures that the resulting algorithm satisfies our definition of a deterministic 1-nearest neighbor imputation scheme. In our case the tie-breaker used was the row number in the original database: incomplete records were assigned to the lowest complete record with a higher row number.

To better express the $L_1(D)$ calculation and the complexity of that calculation, we will first discuss our use of equivalence classes in the experiments. Given the known attributes that are used in the imputation, it is possible to create equivalence classes of all complete tuples and incomplete tuples that match on those attributes, and our discussion of "groups" in this section will rely on that definition. Once we have divided the incomplete tuples into groups, we can identify whether a given group has at least one complete tuple or not.

We identify four cases below; $L_1(D)$ is the maximum of these. Note that in Remark 6.3.5 we interpreted $L_1(D)$ as a maximum over two possibilities; it will be convenient here to divide addition into three cases.

The four possible options are outlined below:

1. Remove an existing complete tuple
2. Add a new complete tuple to a group with an existing complete tuple
3. Add a new complete tuple to a group without an existing complete tuple (but containing incomplete tuples)
4. Add a new complete tuple in a unique location

The impact of case 1 is maximized by removing the complete tuple that imputes onto the most incomplete tuples in the existing dataset. Adding a record to a group with an existing complete tuple cannot change more records than deleting this "maximal donor". The tie-breaker described above is easily calculated and ensures that the impact of case 2 never exceeds the impact of case 1.

Option 3 requires a more nuanced calculation. If a complete tuple is added to an existing group that lacks a complete tuple in the original dataset, then that new complete tuple will donate to the entire group in the modified dataset. The rest of the calculation comes from the impact this complete tuple can have on other groups that don't already have an existing data complete tuple. For each such group, we calculate the distance to this new complete tuple and then determine which, if any, records would would be imputed on by the new tuple.

The calculation for a move of type 4 requires exploring every possible equivalence class not currently represented in the data, and determining the impact of a new complete tuple placed there. Even for the relatively limited scope of our attributes, the brute-force calculation required for determining this value exactly was too expensive. Thankfully, it sufficed to bound the impact of a move of type 4 from above by a number which, *for the dataset we were working with*, was still smaller than the impact of moves 1-3. We first find pairs of groups without existing complete tuples that are pairwise close to each other. The sizes of the unions of such groups provides an upper-bound for the maximum impact of case 4. As discussed previously, a large part of the motivation in grouping age by decade was to guarantee that this upper bound for case 4 was small. This ensures the value of $L_1(D)$ we computed is exact.

6.5.3 Experiment

Given the lack of viable differentially private approaches to dealing with missing data, we compare against ignoring missing data and implementing our Nearest Neighbor scheme using global sensitivity with the Laplace mechanism. We provide two example queries: what proportion of the records had an income that was below the "poverty line," (essentially two frequency queries), and the mean income. As there is no "official" poverty line for the 1940 data we used \$658 as determined by [53] as the necessary income to support a family of four above the poverty level in 1940. For simplicity we assume that the size of all of the datasets is a known quantity. Given our queries, this type of information would be publicly available for Census data.

It is known that ignoring missing data for income leads to biased results [35]; this example is useful because it provides a comparison of the bias improvement from imputation vs. the increased variance in providing differential privacy. Our comparison to a naive global sensitivity differential privacy scheme requires some explanation. It is easy to see that global sensitivity for an imputation scheme will provide unusable amount of noise (see Figure 6.1). Our version of global sensitivity makes the optimistic assumption that the maximum number of records a complete tuple can donate to is the number of missing records; i.e., we limit the impact from a single person donating to the entire state to a single individual donating to every value that was missing from the state. While this is an underestimate of global sensitivity, we will see that even this optimistic estimate makes global sensitivity untenable. We include it to highlight that imputation creates significant challenges for differential privacy.

As we see in Figure 6.4, the global sensitivity of imputation requires unreasonable noise; better results are obtained by ignoring missing values. Our global sensitivity scheme has a Mean Squared Error of 3.9×10^5 compared to simply ignoring the missing data having a MSE with 397.25; our nearest neighbor scheme has a MSE of 1.3. Ignoring missing data results in a value substantially lower than the true result. Our smooth sensitivity imputation method gives results close to the true mean, with only slightly higher variance than that induced by the random variation in which individuals have missing values. Global sensitivity results were similar on other queries, and are omitted to highlight comparison of our method with ignoring missing data.

Figure 6.5 shows the same query, but just for individuals in their 20s or 40s. For the 20 year olds, ignoring missing data provides a MSE of 116.9 compared to our imputation strategy providing a MSE of 7.3. For the 40 year olds ignoring provides a MSE of 1180.3

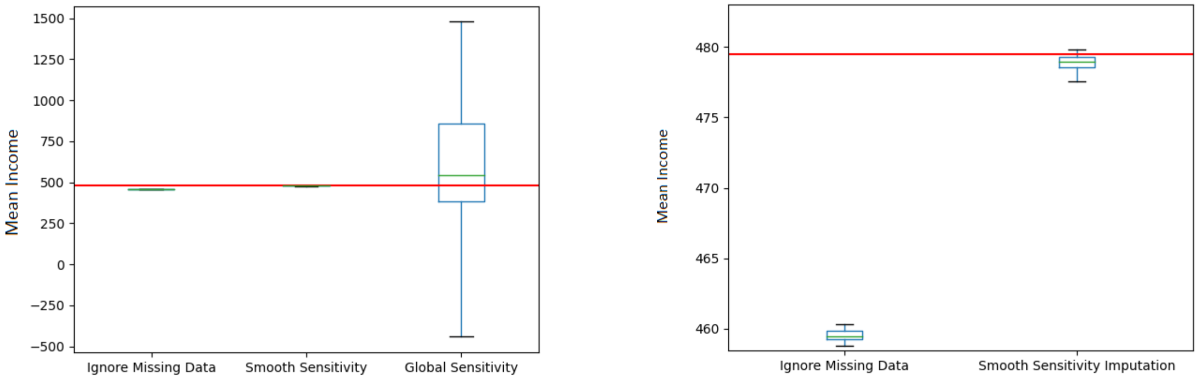

Figure 6.4. Mean individual income, ages 20-59, with $(6 \ln 2)$-differential privacy. Box reflects inter quartile range, whisker is 5-th and 95-th percentiles. Full width line is the true value. “Ignore Missing Data” means missing data was discarded and the Laplace mechanism was used, “Smooth Sensitivity” (resp. “Global Sensitivity”) means missing data was imputed and our smooth sensitivity-based mechanism (resp. the Laplace mechanism) was used.

and our mechanism provides a MSE of 11.1. We see that missing data has a larger impact on some subgroups than others, but imputation largely removes this impact.

Figure 6.6 shows a different query on the same data; the proportion of individuals who make enough to support a family of four above the poverty line. We again see significantly better results for smooth sensitivity imputation. Ignoring the missing data for 20-year olds provides a MSE of 3.3×10^{-5} while our imputation strategy has a MSE of 1.4×10^{-6}. For the 40 year olds, ignoring the missing data provides a MSE of 4.6×10^{-4} and our imputation strategy provides a MSE of 7.5×10^{-7}.

6.6 Conclusions and Future Work

The problem of missing survey data presents an interesting privacy challenge for data curators: ignoring the missing values tends to yield biased results, but imputation methods can dramatically increase an individual’s impact on the dataset, thus increasing the likelihood of reidentification. Global sensitivity-based mechanisms require an untenable amount of noise, even under unreasonable assumptions. We have advocated for an approach based on smooth sensitivity to mitigate these issues. To do this, we developed a smooth upper

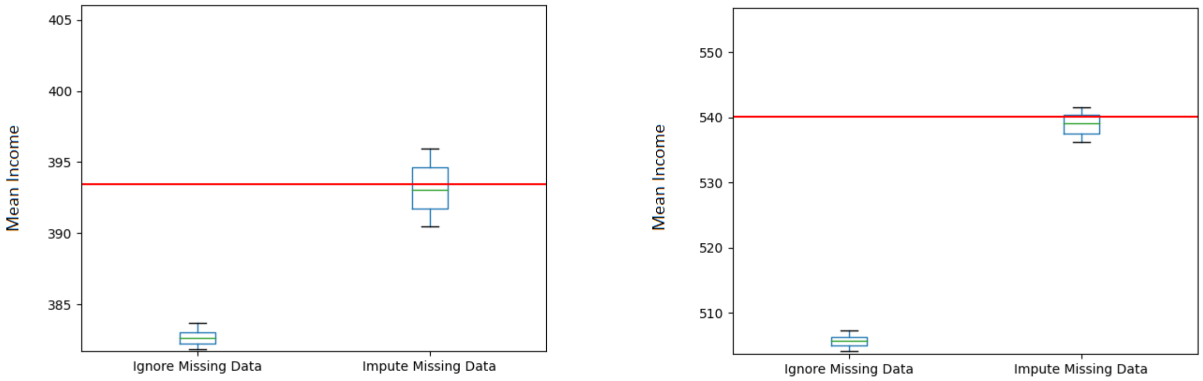

Figure 6.5. Mean individual incomes for persons 20-29 years old (left) and 40-49 years old (right), with $(6\ln 2)$-differential privacy.

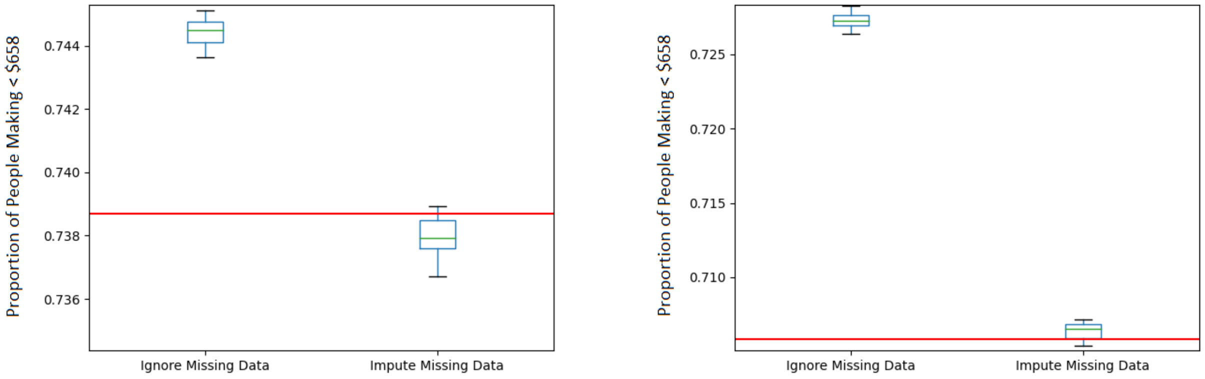

Figure 6.6. Proportions of adults age 20-29 (left) and 40-49 (right) who make enough to support a family of 4 above the poverty level, with $(6\ln 2)$-differential privacy.

bound that in many cases can be computed without computing the local sensitivity an arbitrary number of steps away (potentially exponential in the number of steps); a technical contribution to the differential privacy literature in its own right.

The extent to which this approach can be pushed requires analysis of other queries where changes give a bounded $L_1(D)$. One could also look for similar quantities dependent only on the present dataset D from which similar bounds could be derived.

Each of our mechanisms answers a single query privately on a dataset with missing data. For circumstances where a data curator seeks to answer a large quantity of queries on such data, it would be interesting to investigate the compatibility of our mechanism with approaches like the high dimensional matrix mechanism [54], PriView [55], or differentially private synthetic data releases.

7. DIFFERENTIALLY PRIVATE POST-STRATIFICATION

Many surveys feature weighting for bias reduction purposes. Many surveys will over-sample potentially heterogeneous populations to better represent their differences and weighting can counter any potential differences in survey response, intended or not. This type of weighting is doubly difficult for differential privacy since not only do weights increase the sensitivity of a query (as weight implies the number of people they represent which requires a multiplier on the unweighted sensitivity of their record) but since the weights themselves are based on the seen dataset this causes problems with global sensitivity (akin to the above pathological example). My work has focused on using both the exponential mechanism and a smooth sensitivity mechanism with the generalized Cauchy distribution to provide results on queries with a differentially private weight.

7.1 DP Bin Selection

The goal of this section is to define a mechanism that chooses a set of bins from a binning strategy which avoids sets of bins producing high weights and satisfied differential privacy. As in the previous section, we assume only a single set of attributes A. We fix our set of possible binning strategies $\mathcal{P}_s$. A binning strategy $\mathcal{S}$ for age might be adult vs non-adult or age rounded to the decade. By $\mathcal{P}_s$ we mean all binning strategies that we are choosing among.

Recall from the introduction that to use the exponential mechanism, we need to define a *score function*, or a map

$$f_{\mathcal{P}_s} : \mathcal{D} \times \mathcal{P}_s \to \mathbf{R},$$

where high scores are assigned to sets of bins we want to have a high probability of being selected. As $\mathcal{P}_s$ is a finite set, we can take μ to be the uniform measure so that $\mu(\mathcal{S}) = 1$ for all $\mathcal{S} \in \mathcal{P}_s$.

We propose setting

$$f_{\mathcal{P}_s}(D, \mathcal{S}) = \frac{|\mathcal{S}|}{W_0^{\mathcal{S}}(D)}, \tag{7.1}$$

Set of bins $\mathcal{S}$	separate	minors	adults	minors-adults	all
$f_{\mathcal{P}_s}(A, \mathcal{S})$	5/1000	4/143	3/1000	2/143	1/100
$f_{\mathcal{P}_s}(B, \mathcal{S})$	5/1000	4/1000	3/1000	2/118	1/100

Figure 7.1. Running example: The value of $W_0^{\mathcal{S}}(D)$ is the maximum weight within a bin in $\mathcal{S}$. These values can be read off of the table in Figure 7.3.

that is, $f_{\mathcal{P}_s}(D, \mathcal{S})$ is equal to the total number of bins in $\mathcal{S}$ divided by the max potential weight of D under $\mathcal{S}$. This rewards bins which have a low maximum weight and a large number of bins. The idea behind this choice is that we wish to minimize the total variance. $|\mathcal{S}|$ is meant to stand in as an approximation of the diversity of each bin. Naively, the presence of more bins should mean there was less collapsing across attributes. $W_0(\mathcal{S})$ is meant to stand in for the (smooth) sensitivity of the second mechanism (see Section 7.4). Although it is possible for the smooth sensitivity to increase while the maximum weight decreases (this is a subtle point that depends on the parameter γ in the generalized Cauchy mechanism as well as the privacy parameter), this is a close and easily computable approximation that yields an expression for the sensitivity of the exponential mechanism.

Example 7.1.1. In our running example, Figure 7.1 shows the values of $f_{\mathcal{P}_s}(D, \mathcal{S})$ for each of our five sets of bins. We observe that in Sample A, the set of bins with the highest score is the one which combines all minors. On the other hand, in Sample B, the set of bins with the highest score is the one which combines all minors and combines all adults.

We now define our mechanism.

Definition 7.1.2. For $\alpha > 0$, let $\mathcal{A}^{bin}_{\mathcal{P}_s} : \mathcal{D} \to \mathcal{P}_s$ be the mechanism that returns $\mathcal{S} \in \mathcal{P}_s$ with probability proportional to $\exp(\alpha \cdot f_{\mathcal{P}_s}(D, \mathcal{S}))$.

To determine the privacy parameter of our mechanism, we need to compute the global sensitivity of $f_{\mathcal{P}_s}$. Let G be a tuple of highest possible initial weight and let G' be a tuple of lowest possible initial weight. We note that $\mathsf{bwt}(X)$ is determined by the sampling strategy, independent of the sample itself, and will typically be the inverse of the sampling rate but can be more complex as needed. Let

$$\Delta_{\mathcal{P}_s} = \max_{\mathcal{S}\in\mathcal{P}_s} \frac{|\mathcal{S}| \cdot \mathsf{bwt}(G)}{\mathsf{bwt}(G') \cdot \min_{S\in\mathcal{S}} \hat{N}(S)}. \tag{7.2}$$

Theorem 7.1.3. *$\Delta_{\mathcal{P}_s}$ is an upper bound for the global sensitivity of $f_{\mathcal{P}_s}$. In particular, the mechanism $\mathcal{A}^{bin}_{\mathcal{P}_s}$ satisfies $(2\alpha \cdot \Delta_{\mathcal{P}_s})$-differential privacy by Theorem 2.2.6.*

Proof. We need to show that

$$\Delta_{\mathcal{P}_s} \geq \max_{\mathcal{S}\in\mathcal{P}_s} \max_{d(D,D')=1} \left| f_{\mathcal{P}_s}(D,\mathcal{S}) - f_{\mathcal{P}_s}(D',\mathcal{S}) \right|.$$

For $\mathcal{S} \in \mathcal{P}_s$, define

$$\begin{aligned} \Delta_{\mathcal{S}} &= \max_{d(D,D')=1} \left| f_{\mathcal{P}_s}(D,\mathcal{S}) - f_{\mathcal{P}_s}(D',\mathcal{S}) \right| \\ &= |\mathcal{S}| \max_{d(D,D')=1} \left| \frac{1}{W_0^{\mathcal{S}}(D)} - \frac{1}{W_0^{\mathcal{S}}(D')} \right| \end{aligned}$$

Fix a pair D, D' of neighboring datasets. We can assume without loss of generality that an individual is deleted to move from D to D'. Thus in passing from D to D', the weight of every individual either increases or remains the same. This implies that

$$W_0^{\mathcal{S}}(D) \leq W_0^{\mathcal{S}}(D') \leq W_1^{\mathcal{S}}(D).$$

In particular, we can drop the absolute value from our equation for $\Delta_{\mathcal{S}}$.

We'll assume that both $W_0^{\mathcal{S}}(D)$ and $W_0^{\mathcal{S}}(D')$ are realized by individuals within non-empty bins that are in the dataset D. The cases where one (or both) of these are realized by the known population total of an empty bin or from an addition give the same bound with similar arguments. Thus there individuals $x \in D$ and $x' \in D'$ so that

$$\begin{aligned} W_0^{\mathcal{S}}(D) &= \mathsf{wt}_D(x) = \mathsf{bwt}(x) \cdot \frac{\hat{N}(S(x))}{\sum_{y \in S(x) \cap D} \mathsf{bwt}(y)} \\ W_0^{\mathcal{S}}(D') &= \mathsf{wt}_{D'}(x') = \mathsf{bwt}(x') \cdot \frac{\hat{N}(S(x'))}{\sum_{y \in S(x') \cap D'} \mathsf{bwt}(y)}. \end{aligned}$$

In particular, this means x' is of highest initial weight within $D \cap S$. Moreover, by assumption, we have $\mathsf{wt}_D(x') \leq \mathsf{wt}_D(x)$. This means

$$\begin{aligned} \frac{1}{W_0^{\mathcal{S}}(D)} - \frac{1}{W_0^{\mathcal{S}}(D')} &\leq \frac{1}{\mathsf{wt}_D(x')} - \frac{1}{\mathsf{wt}_{D'}(x')} \\ &= \frac{\sum_{y \in S(x') \cap D} \mathsf{bwt}(y)}{\mathsf{bwt}(x') \cdot \hat{N}(S(x'))} - \frac{\sum_{y \in S(x') \cap D'} \mathsf{bwt}(y)}{\mathsf{bwt}(x') \cdot \hat{N}(S(x'))} \\ &= \frac{\mathsf{bwt}(t)}{\mathsf{bwt}(x') \cdot \hat{N}(S')} \\ &\leq \frac{\mathsf{bwt}(G)}{\mathsf{bwt}(G') \cdot \hat{N}(S')}, \end{aligned}$$

where t is the individual deleted to pass from D to D'. We conclude

$$\max_{\mathcal{S} \in \mathcal{P}_s} \max_{d(D,D')=1} |f_{\mathcal{P}_s}(D, \mathcal{S}) - f_{\mathcal{P}_s}(D, \mathcal{S})| \leq \max_{\mathcal{S} \in \mathcal{P}_s} \Delta_{\mathcal{S}} = \Delta_{\mathcal{P}_s},$$

as desired. □

Example 7.1.4. In our running example, this gives a sensitivity of $\Delta_{\mathcal{P}_s} = 5/100000 = 5 \cdot 10^{-5}$. Thus if we take a privacy parameter of $\varepsilon = 0.01$, then $\alpha = 100$ and the mechanism $f_{\mathcal{P}_s}(D, \mathcal{S})$ returns the set of bins $\mathcal{S}$ with probability proportional to $f_{\mathcal{P}_s}(D, \mathcal{S})$. These probabilities for can be found in Figure 7.2. Recall that the values of $f_{\mathcal{P}_s}(D, \mathcal{S})$ were computed in Figure 7.1.

We conclude this section with a discussion of two ways to choose a binning strategy to use with the mechanism $\mathcal{A}^{bin}_{\mathcal{P}_s}$. The first is to choose a binning strategy before making any decisions about the privacy parameter or the probability the mechanism will return a low score bin. Once the binning strategy has been fixed, the sensitivity $\Delta_{\mathcal{P}_s}$ can be computed. At

Set of bins $\mathcal{S}$	separate	minors	adults	minors-adults	all
$\Pr\left[\mathcal{A}^{bin}_{\mathcal{P}_s}(A) = \mathcal{S}\right]$	.063	.627	.052	.155	.104
$\Pr\left[\mathcal{A}^{bin}_{\mathcal{P}_s}(B) = \mathcal{S}\right]$	.130	.118	.107	.430	.215

Figure 7.2. Running example: The probability that the mechanism $\mathcal{A}^{bin}_{\mathcal{P}_s}$ returns each set of bins $\mathcal{S}$ at $\varepsilon = 0.01$. The scores for each set of bins can be found in Figure 7.1. In both examples, at $\varepsilon = 0.1$, the mechanism essentially always returns the highest score bin ($\Pr > .999$).

this point, one will need to choose a trade-off between a higher privacy budget (the product $\varepsilon = 2\alpha\Delta_{\mathcal{P}_s}$) and a higher probability of the mechanism returning a low score set of bins (roughly inversely proportional to α). When the privacy budget is fixed, this could lead to less accurate results from the second mechanism.

The second option is to first fix a privacy parameter ε and the parameter α in order to bound the probability the mechanism returns a low score bin. From this, one can construct the constant $\Delta_{\mathcal{P}_s} = \frac{\varepsilon}{2\alpha}$ and choose a binning strategy so that the sensitivity is bounded by $\Delta_{\mathcal{P}_s}$. This will lead to more accurate results when releasing weighted counting queries, but at the cost of limiting the possible candidate sets of bins.

7.2 Determining Highest Weights

Two of our mechanisms, the binning selection and the weighted counting query protection, require an understanding not only of the highest weight in a dataset D, but the highest possible weight in datasets near D. This will appear in the score function of the mechanism for DP bin selection and is a prerequisite for using smooth sensitivity to answer weighted counting queries. Calculating a smooth upper bound on the weights (sensitivity) is the primary technical challenge addressed by this thesis.

In this section, we consider only a single set of attributes A. When we combine our methods with IPF, the ideas presented here will be used both on the individual sets of attributes $A^1, \ldots, A^n$ and their join. Throughout this section, fix a dataset D and a set of bins $\mathcal{S}$.

We first define

$$W_0^{\mathcal{S}}(D) = \max\left\{\max_{x \in D} \mathsf{wt}_D(x), \max_{y \notin D} \mathsf{wt}_{D \cup y}(y), \max_{S \in \mathcal{S}: S \cap D = \emptyset} \hat{N}(S)\right\} \tag{7.3}$$

That is, $W_0^P(D)$ is the highest weight of any tuple in D (under $\mathcal{S}$), the highest weight of any tuple which could be added to D, or the highest population control of an empty

Set of bins $\mathcal{S}$	separate	minors	adults	minors-adults	all
$W_0^P(A)$	1000	143	1000	143	100
$W_0^P(B)$	1000	1000	1000	118	100

Figure 7.3. Running example: The value of $W_0^{\mathcal{S}}(D)$ is the maximum weight within a bin in P. These values can be read off of the table in Figure 7.6. The names are defined in Example 7.5.5.

bin, whichever of the three is larger[1]. We observe that the second maximum automatically bounds the third, but we retain this redundant case for clarity. We call $W_0^{\mathcal{S}}(D)$ the *max potential weight* of D (under $\mathcal{S}$).

Example 7.2.1. In our running example, Figure 7.3 shows that values of $W_0^{\mathcal{S}}(D)$ for each of our five sets of bins. As there are no empty bins and all individuals have the same initial weight, these values are simply the highest weights coming from each set of bins.

For an integer $k \geq 0$, we likewise define

$$W_k^{\mathcal{S}}(D) = \max_{D' : d(D, D') = k} W_0^{\mathcal{S}}(D').$$

This is the largest max potential weight of any dataset k steps away from D. The mechanisms presented in Sections 7.1 and 7.4 make reference to the numbers $W_k^{\mathcal{S}}(D)$, but not to the feasibility of their computation. The remainder of this section focuses on addressing this computability issue by reducing the definition of $W_k^{\mathcal{S}}(D)$ to a simpler formula.

Example 7.2.2. In our running example, let $\mathcal{S}$ be the set of bins with no combining (what we have called 'separate'). We then have

$$W_k^{\mathcal{S}}(A) = \begin{cases} 100,000/(100-k) & k < 100 \\ 100,000 & k \geq 100 \end{cases}$$

---

[1] ↑A binning strategy should eliminate *structural zeros*, such as children who are householders. However, even when the actual population of a bin is non-zero, it is possible that none are sampled, so the bin is empty.

To simplify the presentation of Proposition 7.2.3 below, we now restrict to the case where the initial weights of every individual are the same. We generalize this to varied initial weights (e.g., stratified sampling) in the appendix (see Proposition 7.3.8).

Proposition 7.2.3. *Let D be a dataset and $k \geq 0$ be an integer. Suppose all initial weights are equal to 1. Then*

$$W_k^{\mathcal{S}}(D) = \max_{S \in \mathcal{S}} \begin{cases} \hat{N}(S), & |S \cap D| \leq k+1 \\ \frac{\hat{N}(S)}{|S \cap D| - k}, & |S \cap D| \geq k+1 \end{cases}$$

Proof. Let D be a dataset and let S be a bin. We wish to find the highest potential weight of a tuple in S in a dataset k steps away from D.

We first observe that, since all initial weights are the same, adding a tuple to S can only lower the weight of the other individuals in that bin. The exception is when S is empty, in which case the potential weight is $\widehat{N}(S)$. In this case, adding a tuple to S has no effect on the potential weight, as the new tuple would just be given a weight of $\widehat{N}(S)$.

Based on this reasoning, the potential weight in S is maximized by deleting as many tuples from S as possible. We now separate into two cases.

If $|S \cap D| \leq k$, then it is possible to completely empty the bin S in (at most) k steps. At this point, the potential weight within S is $\widehat{N}(S)$, which is as high as possible.

If $|S \cap D| \geq k+1$, then the highest potential weight in S occurs when k individuals are deleted. These weights are then exactly $\frac{\hat{N}(S)}{|S \cap D| - k}$. □

The value of $W_k^{\mathcal{S}}(D)$, which is efficiently computable by the proposition, will be used in defining both of our proposed mechanisms.

Remark 7.2.4. The statement and proof of Proposition 7.2.3 are given only for the simple random sample case (this is generalized to the general case in Proposition 7.3.8 in the appendix), but we now *drop* that assumption for the rest of the section.

7.3 DP Iterative Proportional Fitting

In this section, we use the mechanism from the previous section and the Laplace mechanism to release an n-way contingency table. We then perform IPF on this table as post-processing to produce population estimates for the join of the n sets of attributes.

As in Section 2.2.1, we fix sets of attributes $A^1, \ldots, A^n$ and binning strategies $\mathcal{P}_s^1, \ldots, \mathcal{P}_s^n$. Our first step is to apply the mechanisms $\mathcal{A}^{bin}_{\mathcal{P}_s^{\mathrm{i}}}$.

Theorem 7.3.1. *Choose positive real numbers $\alpha_1, \ldots, \alpha_n$. Then the algorithm $\mathcal{A}^{bin}_{\widetilde{\mathcal{P}_s}} : \mathcal{D} \to \widetilde{\mathcal{P}_s}$ which returns $(\mathcal{A}^{bin}_{\mathcal{P}_s^1}(D), \ldots, \mathcal{A}^{bin}_{\mathcal{P}_s^n}(D))$, with each $\mathcal{A}^{bin}_{\mathcal{P}_s^{\mathrm{i}}}$ computed with parameter α_{i}, satisfies ε-DP for $\varepsilon = 2\sum_{\mathrm{i}=1}^n \alpha_{\mathrm{i}} \Delta_{\mathcal{P}_s^{\mathrm{i}}}$.*

Proof. The result follows immediately from Theorem 7.1.3 and the sequential composition theorem. □

Now given an output $(\mathcal{S}^1, \ldots, \mathcal{S}^n)$ of $\mathcal{A}^{bin}_{\widetilde{\mathcal{P}_s}}$, we proceed by querying the associated contingency table.

Theorem 7.3.2. *Let $\varepsilon > 0$ and let $\mathcal{S} = (\mathcal{S}^1, \ldots, \mathcal{S}^n) \in \widetilde{\mathcal{P}}_s$ be a tuple of bins. Let $N = \prod_{\mathrm{i}=1}^n |\mathcal{S}^{\mathrm{i}}|$. Then the mechanism $\mathcal{A}^{\mathrm{init}}_{\mathcal{S}} : \mathcal{D} \to \mathbb{R}^N$ which returns the values $|S \cap D|$ for all $S = (S^1, \ldots, S^n) \in \mathcal{S}^1 \times \cdots \times \mathcal{S}^n$ plus iid noise sampled from a Laplace distribution with parameter $1/\varepsilon$ satisfies ε-DP.*

Proof. todo: find reference □

Remark 7.3.3. The contingency table queried in Theorem 7.3.2 may contain *structural* zeroes. There is no privacy loss associated with returning the values of these cells as zero (as opposed to a "noisy zero") as long as it is public knowledge that these cells are structural zeroes.

Remark 7.3.4. Using the sequential composition theorem (see [56, Theorem B.1]), if we first release a tuple of sets of bins $\mathcal{S}$ using $\mathcal{A}^{bin}_{\widetilde{\mathcal{P}_s}}$ with privacy parameter ε_1 and then release the contingency table using $\mathcal{A}^{init}_{\mathcal{S}}$ with privacy parameter ε_2, the combined mechanism satisfies $(\varepsilon_1 + \varepsilon_2)$-DP.

For the remainder of this section, consider a fixed tuple of sets of bins $\mathcal{S}_{init} = (\mathcal{S}^1_{\text{init}}, \ldots, \mathcal{S}^n_{\text{init}})$ and a fixed output T_{init} of the mechanism $\mathcal{A}^{init}_{\mathcal{S}_p}$. We would like to perform IPF on the table T_{init}; however, there may be cells in T_{init} which contain negative counts due to the symmetry of the Laplace distribution. Moreover, any value of 0 will remain 0 following iterative proportional fitting. Thus, we proceed by merging cells with non-structural zeroes and negative counts with neighboring cells so that the only non-positive cells remaining are structural zeroes. As this is a post processing step, this can be done in any way a data curator desires.

The result is a new tuple of sets of bins $\mathcal{S} = (\mathcal{S}^1, \ldots, \mathcal{S}^n)$ such that $\mathcal{S}^{\text{i}}_{init}$ is a refinement of $\mathcal{S}^{\text{i}}$ for all i, together with a non-negative contingency table T_0 whose cells are the join bins of $\mathcal{S}$. That is, for any bin $S^{\text{i}}_{init} \in \mathcal{S}^{\text{i}}_{\text{init}}$, there exists a bin $S^{\text{i}} \in S^{\text{i}}$ with $S^{\text{i}}_{\text{init}} \subseteq S^{\text{i}}$. Moreover, for any $S \in \mathcal{S}$, let $\mathcal{U}$ be the set tuples of bins $U \in \mathcal{S}$ for which $S \subseteq U$. Then $T_0(S) = \sum_{U \in \mathcal{U}} T_{\text{init}}(U)$. We will call T_0 a *non-negative collapse* of T_{init}.

Remark 7.3.5. The post-processing collapsing of T_{init} could in theory also be used to merge cells for which IPF produces high weights. The probability that this is necessary is low, however, when the mechanism $\mathcal{A}^{init}_{\mathcal{S}}$ is used to select the tuple $\mathcal{S}$.

Remark 7.3.6. It is also possible to choose each $\mathcal{S}^{\text{i}} = \{\{B : B \in \mathcal{B}(A_{\text{i}})\}\}$ and perform all collapsing as post-processing. We have chosen not to do so since combining two cells necessarily increases the variance of the resulting cell.

The purpose of this Appendix is to establish the same results but in the general case where all base weights are *not* assumed equal.

Recall that the max potential weight is defined as.

$$W_0^{\mathcal{S}}(D) = \max\left\{\max_{x \in D} \ \mathsf{wt}_D(x), \ \max_{S \in P : S \cap D = \emptyset} \ \hat{N}(S), \ \max_{y \notin D} \ \mathsf{wt}_{D \cup y}(y)\right\}.$$

That is, $W_0^{\mathcal{S}}(D)$ is the highest weight of any tuple in D (under $\mathcal{S}$), or the highest population control of an empty bin, or the highest possible weight of an added tuple, whichever of the three is larger.

We first observe the following.

Remark 7.3.7. The third term in this maximum can be computed without making reference to every possible addition. Namely, let G be a tuple of largest possible initial weight. Then

$$\max_{y \notin D} \mathsf{wt}_{D \cup y}(y) = \max_{S \in \mathcal{S}} \frac{\hat{N}(S) \cdot \mathsf{bwt}(G)}{\mathsf{bwt}(G) + \sum_{x \in D \cap S} \mathsf{bwt}(x)}.$$

In words, this says that the highest weight of an added individual is realized when the initial weight of that individual is as high as possible. Thus we need only consider such individuals when computing this maximum.

Recall from the proof of Proposition 7.2.3, that the largest max potential weight k steps away was realized either by emptying a bin or by deleting k individuals from a larger bin. The definitions of $EW_k^{\mathcal{S}}$ and $DW_k^{\mathcal{S}}$ below are meant to generalize these cases. In the deletion case, we want to create an individual of as high a weight as possible. This is the reason we keep the individual of highest initial weight and delete the next k highest initially weighted individuals in the definition of D_S^1.

The presence of variable initial weights also adds a third case not present in the simple case. If every individual in a bin is of relatively low initial weight, then adding an individual of high initial weight can represent the max potential weight. This is the case dealt with in the definitions of D_S^2 and $AW_k^{\mathcal{S}}$ below, where the idea is to make the initial weights in S as low as possible in the first k steps so that adding an individual with high initial weight has the largest possible impact.

The proposition below essentially states that these three cases are the only cases that need to be considered.

Proposition 7.3.8. *Let D be any dataset and $k \geq 0$. Let $S \in P$ be a bin. If $|D \cap S| > k$, define D_S^1 to be the dataset D with the 2nd - $(k+1)$th highest initial weight individuals in*

S deleted. Let D_S^2 be the dataset with the 1st - kth highest initial weight individuals in S deleted. Define

$$\begin{aligned} EW_k^{\mathcal{S}}(D) &= \max_{S \in P : |D \cap S| \le k} \hat{N}(S) \\ DW_k^{\mathcal{S}}(D) &= \max_{S \in P : |D \cap S| > k} \max_{x \in D_S^1 \cap S} \mathsf{wt}_{D_S^1}(x) \\ AW_k^{\mathcal{S}}(D) &= \max_{S \in P : |D \cap S| > k} \frac{\hat{N}(S) \cdot \mathsf{bwt}(G)}{\mathsf{bwt}(G) + \sum_{x \in D_S^2 \cap S} \mathsf{bwt}(x)} \end{aligned}$$

where G is a tuple with the highest possible initial weight. Then

$$W_k^{\mathcal{S}} = \max \left\{ EW_k^{\mathcal{S}}, DW_k^{\mathcal{S}}, AW_k^{\mathcal{S}} \right\}.$$

Proof. We first observe that for $S \in P$, we have $d(D, D_S^1) = k = d(D, D_S^2)$. Moreover, $DW_k(P)$ is (a maximum of) the maximum weight of any individual in D_S^1. Likewise, $AW_k^{\mathcal{S}}$ is (a maximum of) the maximum weight of an individual added to D_S^2. Likewise, there exists a dataset D' for which $D' \cap S = \emptyset$ and $d(D, D') = k$ if and only if $|D \cap S| \le k$. This shows that $W_k^{\mathcal{S}}(D) \ge \max\{EW_k^{\mathcal{S}}(D), DW_k^{\mathcal{S}}(D), AW_k^{\mathcal{S}}(D)\}$. Thus we need only show that $W_k^{\mathcal{S}}(D)$ is no larger than the maximum of these three quantities.

Let D' realize $W_k^{\mathcal{S}}(D)$, that is $W_k^{\mathcal{S}}(D) = W_0^{\mathcal{S}}(D')$. Assume first that $W_0^{\mathcal{S}}(D') = \max_{S \in P : D' \cap S = \emptyset} \hat{N}(S)$. That is, the max potential weight in D' comes from an empty bin. As previously remarked, this implies that $|S \cap D| \le k$. In particular, in this case we have $W_0^{\mathcal{S}}(D') = EW_k^{\mathcal{S}}(D)$.

Next assume that $W_0^{\mathcal{S}}(D') = \max_{x \in D'} \mathsf{wt}_{D'}(x)$. That is, the highest potential weight in D' is the weight of an individual in some bin. Suppose this maximum is obtained at the tuple $t \in D'$ and let $S = S(t)$ be the bin containing t. By definition,

$$\mathsf{wt}_{D'}(t) = \frac{\hat{N}(S) \cdot \mathsf{bwt}(t)}{\mathsf{bwt}(t) + \sum_{y \in D' \setminus \{t\} \cap S} \mathsf{bwt}(y)}.$$

We assume without loss of generality that $|D \cap S| > k$. Indeed, if this is not the case, then $\mathsf{wt}_{D'}(t) \le \hat{N}(S) \le EW_k^{\mathcal{S}}(D)$ and we are in the first case.

We now observe that adding or removing individuals from a bin other than S has no impact on computing the maximum weight within $D' \cap S$. This means we can assume without loss of generality that all individuals which were added to or deleted from D to reach D' are in the bin S. Moreover, we can assume we did not add and delete the same individual, as then $d(D, D') < k$.

Assume we added some individual t''. In particular, this means $|D' \cap S| > 1$. If $t \neq t''$, then $\mathsf{bwt}(t'')$ contributes only to the denominator of the expression for $\mathsf{wt}_{D'}(t)$. Thus, the weight of t would be larger had we instead deleted an individual, a contradiction. This means $t = t''$. In particular, this implies that only one tuple was added, and the other $k-1$ changes were deletions. However, let $x \in D' \setminus \{t\}$ be of highest initial weight and let $D'' = D' \setminus \{x\}$. Then

$$\mathsf{wt}_{D'}(t) < \mathsf{wt}_{D''}(t) \leq W_k^{\mathcal{S}}(D)$$

where the last inequality follows from the fact that D'' can be obtained from D by first deleting k individuals then adding t. This is a contradiction. We conclude that $D' \subset D$; that is, we only made deletions. Moreover, since $\mathsf{wt}_{D'}(t)$ is maximized, we know the tuples deleted are the highest initial weight individuals of $D \setminus \{t\} \cap S$.

Now $t' \in D_S^1$ realize $\max_{x \in D_S^1} \mathsf{wt}_{D_S^1}(x)$. We claim that $\mathsf{wt}_{D_S^1}(t') = \mathsf{wt}_{D'}(t)$; that is, $W_k^{\mathcal{S}}(D)$ is equal to $DW_k^{\mathcal{S}}(D)$.

By assumption, we have $\mathsf{wt}_{D_S^1}(t') \leq \mathsf{wt}_{D'}(t)$. Moreover, we observe that $\sum_{y' \in D_S^1 \setminus \{t'\}} \mathsf{bwt}(y') \leq \sum_{y' \in D' \setminus \{t\}} \mathsf{bwt}(y')$ since $D_S^1 \setminus \{t'\}$ is obtained from D by deleting the $k+1$ individuals of highest initial weight. We then have

$$\begin{aligned}
\mathsf{wt}_{D_1^S}(t') - \mathsf{wt}_{D'}(t) &= \frac{\mathsf{bwt}(t')}{\mathsf{bwt}(t') + \sum_{y' \in D_S^1 \setminus \{t'\}} \mathsf{bwt}(y')} - \frac{\mathsf{bwt}(t)}{\mathsf{bwt}(t) + \sum_{y \in D' \setminus \{t'\}} \mathsf{bwt}(y)} \\
&= \frac{\mathsf{bwt}(t') \sum_{y \in D' \setminus \{t\} \cap S} \mathsf{bwt}(y) - \mathsf{bwt}(t) \sum_{y' \in D_S^1 \setminus \{t'\} \cap S} \mathsf{bwt}(y')}{(\mathsf{bwt}(t') + \sum_{y' \in D_S^1 \setminus \{t'\}} \mathsf{bwt}(y')) \cdot (\mathsf{bwt}(t) + \sum_{y \in D' \setminus \{t'\}} \mathsf{bwt}(y))} \\
&\geq 0
\end{aligned}$$

This shows that in this case, $W_k^{\mathcal{S}}(D) = DW_k^{\mathcal{S}}(D)$.

The final case to consider is when $W_0^{\mathcal{S}}(D') = \dfrac{\hat{N}(S) \cdot \mathsf{bwt}(G)}{\mathsf{bwt}(G) + \sum_{x \in D' \cap S} \mathsf{bwt}(x)}$. That is, the highest potential weight in D' comes from adding an individual. As before, we can assume without loss of generality that $|D \cap S| > k$ and that all additions and deletions were of individuals in S. We observe that the expression for $W_0^{\mathcal{S}}(D')$ in this case is decreasing in $\sum_{x \in D' \cap S} \mathsf{bwt}(x)$. Thus, this expression is maximized precisely when D' is obtained by deleting the k highest initial weight individuals from $D \cap S$. We conclude that $D' = D_S^2$ and in this case $W_k^{\mathcal{S}}(D) = AW_k^{\mathcal{S}}(D)$. $\square$

7.4 Weighted Counting Queries

In this section, we use the generalized Cauchy mechanism to answer (weighted) counting queries on weighted data. We will assume only a single set of attributes A and fix a set of bins $\mathcal{S}$.

Remark 7.4.1. In the case we have n sets of attributes $A^1, \ldots, A^n$, we consider an output T_{init} of $\mathcal{A}_{\mathcal{S}'}^{init}$. We then choose a non-negative collapse T_0 of T_{init} with sets of bins $\mathcal{S}$. We identify $\mathcal{S}$ with the set of join-bins of tuples $S \in \mathcal{S}$, making $\mathcal{S}$ into a set of bins for the attributes A. Finally, we let T be the IPF contingency table induced by T_0 and for $S \in \mathcal{S}$ define $PEP(S) := T(S)$. In this way, we have used IPF to estimate the population totals for the attributes in A.

Let q be a counting query. For each tuple x, we denote $q(x)$ the result of q on the unweighted dataset containing only x. We wish give a noisy answer to the query that returns

$$q(D, \mathcal{S}) = \sum_{x \in D} \mathsf{wt}_D(x) \cdot q(x)$$

where the weights are computed with respect to the set of bins $\mathcal{S}$.

We observe that the global sensitivity this query is $\max_{S \in \mathcal{S}} \hat{N}(S)$. This is realized by starting with a dataset where the largest bin (by known population) contains only one tuple (which would have weight $\hat{N}(S)$, giving a query answer of $\hat{N}(S)$) and deleting that tuple (giving an answer of 0). Fortunately, we can get a much better answer using smooth sensitivity.

Proposition 7.4.2. *Fix a set of bins $\mathcal{S}$ and a constant $\beta > 0$. Define*

$$SS_\beta(D, \mathcal{S}) = \max_{k \geq 0} \mathrm{e}^{-\beta k} W_k^{\mathcal{S}}(D).$$

Then $SS_\beta(D, \mathcal{S})$ is a β-smooth upper bound on the local sensitivity of q.

Proof. Let D be a dataset. We must first show that $LS_q(D, \mathcal{S}) \leq SS_\beta(D, \mathcal{S})$.

By definition, we have

$$LS_q(D, \mathcal{S}) = \max_{d(D,D')=1} \left| \sum_{x \in D} \mathsf{wt}_D(x) \cdot q(x) - \sum_{x \in D'} \mathsf{wt}_{D'}(x) \cdot q(x) \right|$$

Assume this is maximum is realized at D'. Let t be the tuple which is not in both D and D' and let $S(t) \in \mathcal{S}$ be its corresponding bin. It follows that the weights of all individuals not in $S(t)$ are the same in both D and D'. Thus we have

$$LS_q(D, \mathcal{S}) = \left| \sum_{x \in S(t) \cap D} \mathsf{wt}_D(x) \cdot q(x) - \sum_{x \in S(t) \cap D'} \mathsf{wt}_{D'}(x) \cdot q(x) \right|$$

We observe that these two sums differ by at most $\mathsf{wt}(t)$, the weight of t (in whichever of D, D' it is in). This is realized when t is in the support of q and every other tuple in $S \cap D$ is not in the support of q (or vice versa). Moreover, we have $\mathsf{wt}(t) \leq W_0^{\mathcal{S}}(t)$ by the definition of $W_0^{\mathcal{S}}(D)$. We conclude that $LS_q(D, \mathcal{S}) \leq W_0^{\mathcal{S}}(D) \leq SS_\beta(D, \mathcal{S})$.

We now must show that $SS_\beta(D, \mathcal{S}) \leq \mathrm{e}^\beta SS_\beta(D', \mathcal{S})$ for all neighboring D, D'. To see this, we first observe that $W_k(D) \leq W_{k+1}(D')$. Indeed, any dataset within k steps of D is automatically within $k+1$ steps of D'. Thus

$$\begin{aligned}
SS(D, P) &= \max_{k \geq 0} \mathrm{e}^{-\beta k} W_k^{\mathcal{S}}(D) \\
&= \mathrm{e}^\beta \max_{k \geq 0} \mathrm{e}^{-\beta(k+1)} W_k^{\mathcal{S}}(D) \\
&\leq \mathrm{e}^\beta \max_{k \geq 0} \mathrm{e}^{-\beta(k+1)} W_{k+1}^{\mathcal{S}}(D') \\
&\leq \mathrm{e}^\beta \max_{k \geq 0} \mathrm{e}^{-\beta(k)} W_k^{\mathcal{S}}(D') \\
&= \mathrm{e}^\beta SS_\beta(D', \mathcal{S})
\end{aligned}$$

as desired. □

Given this smooth upper bound, we use the generalized Cauchy mechanism to output a noisy query result given a binning strategy.

Corollary 7.4.3. *Let $\mathcal{S}$ be a set of bins and choose $\varepsilon > 0$ and $\gamma > 1$. Let $\beta = \frac{\varepsilon}{2(\gamma-1)}$. Then the mechanism*

$$\mathcal{A}_{wc} : \mathcal{D} \times \mathcal{S} \to \mathbf{R}$$

which returns

$$q(D, \mathcal{S}) + \frac{2(\gamma - 1) \cdot SS_\beta(D, \mathcal{S})}{\varepsilon} \cdot X,$$

where X is sampled from the generalized Cauchy distribution with parameter γ, satisfies ε-differential privacy.

Example 7.4.4. Returning to our running example, we will compute the smooth sensitivities and standard deviations for each sample under each set of bins. We take $\varepsilon = 1$ and $\gamma = 4$, meaning $\beta = 1/6$. Straightforward computation shows that in this example, we have

$$SS_\beta(D) = W_0^{\mathcal{S}}(D)$$

for $D \in \{A, B\}$ and all S.

As the generalized Cauchy mechanism with parameter $\gamma = 4$ has variance 1, the standard deviation of the output of $\mathcal{A}_{wc}(D, \mathcal{S})$ is

$$\sigma_{\mathcal{A}_{wc}(D,\mathcal{S})} = \frac{2(\gamma - 1)SS_\beta(D, \mathcal{S})}{\varepsilon} = 6W_0^{\mathcal{S}}(D)$$

for $D \in \{A, B\}$ and all S. The values of these standard deviations can be found in Figure 7.4.

Set of bins S	separate	minors	adults	minors-adults	all
$SS_\beta(A) = W_0^{\mathcal{S}}(A)$	1000	143	1000	143	100
$\sigma_{\mathcal{A}_{wc}(A,\mathcal{S})}$	6000	858	6000	858	600
$SS_\beta(B) = W_0^{\mathcal{S}}(B)$	1000	1000	1000	118	100
$\sigma_{\mathcal{B}_2(B,\mathcal{S})}$	6000	6000	6000	708	600

Figure 7.4. Running example: The smooth sensitivities and standard deviations of the mechanism $\mathcal{A}_{wc}$ and $\varepsilon = 1$ and $\gamma = 4$. The values of $W_0^{\mathcal{S}}$ can be found in Figure 7.3.

By weighting the standard deviations in Figure 7.4 by the probabilities in Figure 7.2, we end our example by computing the expected standard deviation of the mechanism $\mathcal{A}_{wc}(D, \mathcal{A}_{bin}(D))$, which has privacy parameter $\varepsilon = 1 + .01 \approx 1$. This yields

$$\begin{aligned}\mathbb{E}_{P\sim\mathcal{A}_{bin}(A)}\left[\sigma_{\mathcal{A}_{wc}(A,\mathcal{S})}\right] &= 1000 \\ \mathbb{E}_{P\sim\mathcal{A}_{bin}(B)}\left[\sigma_{\mathcal{A}_{wc}(B,\mathcal{S})}\right] &= 2563.\end{aligned}$$

7.5 Setup

The goal of our proposed mechanisms is to conform to known population totals, reducing error in results due to under- or over- response for particular subgroups.

Given a set of attributes A - for example {age, sex, race} - let $\mathcal{B}(A)$ be the set of all unique values of these attributes . Intuitively, we think of $\mathcal{B}(A)$ as a set of base bins, containing the individuals with the specified values of the attributes in A.

We denote by $\mathcal{P}(\mathcal{B}(A))$ the set of partitions of the elements of $\mathcal{B}(A)$. We call an element $\mathcal{S} \in \mathcal{P}(\mathcal{B}(A))$ a *set of bins* an element $S \in \mathcal{S}$ a *bin*. Now we suppose there is a subset $\mathcal{P}_E(\mathcal{B}(A)) \subseteq \mathcal{P}(\mathcal{B}(A))$ so that for all sets of bins $\mathcal{S} \in \mathcal{P}_E(\mathcal{B}(A))$ and for all bins $S \in \mathcal{S}$ we have a known population total $\hat{N}(S)$ for the number of individuals in S.

Remark 7.5.1. Clearly for any bin S as above and any partition $\mathbf{S}$ of S, we have $\hat{N}(S) = \sum_{S' \in \mathbf{S}} \hat{N}(S')$. Thus, if they are known, the population totals $\hat{N}(\{B\})$ for all $B \in \mathcal{B}(A)$ are all that need to be specified. In this case, we have $\mathcal{P}_E(\mathcal{B}(A)) = \mathcal{P}(\mathcal{B}(A))$. We will, however, need the more general setup in Section 7.3 when we interpret iterative proportional fitting as a way of estimating known population totals.

Example 7.5.2. To help clarify these ideas, we will use a running example based on age-based binning. We assume five bottom-level bins: Children (0-12), teens (13-18), young adults (19-39), older adults (40-64), and seniors (65+). Figure 7.5 shows the total known population ($\hat{N}$) for each bin, and the number of samples in each bin for two possible response examples, A and B. Each response example represents a 1% sample of the population, meaning all weights would be equal to 100 if sampling was uniform across the bottom-level bins. Sample A has a low response for the teens bin (resulting in a high weight for just that bin);

Age Block	0-12	13-18	19-39	40-64	65+
Population Total	100,000	100,000	100,000	100,000	100,000
In Sample A	1300	100	1200	1200	1200
In Sample B	1600	100	1600	1600	100

Figure 7.5. Running example: Age-based binning strategy with two example samplings from a know population total.

Sample B has a low response for teens and seniors (resulting in a high weight for both those bins.) We will see how this leads (with high probability) to choosing different ways to combine the bins and examine the impact on any query that counts (with weights) some subset of the population (e.g., “number of seniors living alone”).

Let us suppose each possible tuple x has an initial weight $\mathsf{bwt}(x)$. This may be a base weight coming from stratified sampling or something more complicated. The only assumption made about $\mathsf{bwt}(x)$ is that it depends only on x, not on the dataset containing x; i.e., it is based on the sampling design, not the responses. For the case of a simple random sample where the only adjustment made is conforming to population totals, we can use $\mathsf{bwt}(x) = 1$ for all x. Given a dataset D and a bin S, we denote by $S \cap D$ the set of tuples in D which are in the bin S.

Definition 7.5.3. Let $\mathcal{S}$ be a set of bins and D a dataset. For $x \in D$, the *weight of x under $\mathcal{S}$* is

$$\mathsf{wt}_{\mathcal{S},D}(x) = \mathsf{bwt}(x) \cdot \frac{\hat{N}(S(x))}{\sum_{y \in S(x) \cap D} \mathsf{bwt}(y)},$$

where $S(x)$ is the bin in $\mathcal{S}$ containing x.

Remark 7.5.4. We observe that $\mathsf{wt}_{\mathcal{S},D}(x)$ depends on both $\mathcal{S}$ (and hence A) and D, even though $\mathsf{bwt}(x)$ does not. Moreover, within a bin, individuals of higher initial weights are of higher weight. In the case all initial weights are the same, this means all individuals within the same bin are given the same weight.

Bin	0-12	13-18	19-39	40-64	65+	0-18	19+	0+
Population Total	100k	100k	100k	100k	100k	200k	300k	500k
In Sample A	1300	100	1200	1200	1200	1400	3600	5000
Weights for A	77	1000	83	83	83	143	83	100
In Sample B	1600	100	1600	1600	100	1700	3300	5000
Weights for B	63	1000	63	63	1000	118	91	100

Figure 7.6. Running example: Weights and in-sample totals in all possible bins for Samples A and B. See Example 7.5.5 for binning definitions.

7.5.1 Binning Strategies

In practice, the set of bins $\mathcal{S}$ is often adjusted if the value of $\mathsf{wt}_{\mathcal{S},D}(x)$ is too large for some $x \in D$. This happens if, for example, there is some $S \in \mathcal{S}$ such that $|S \cap D|$ is too small (the American Community Survey uses 10 as its cutoff for "small") and/or $|S \cap D|/\hat{N}(S)$ is too small (the American Community Survey uses 1/3.5 as its cutoff for "small"). Since this decision depends on the dataset D, we instead propose choosing a set of "candidate sets of bins" $\mathcal{P}_s \subseteq \mathcal{P}_E(\mathcal{B}(A))$ independently of the data. We call such a set $\mathcal{P}_s$ a *binning strategy*. We remark that even though a binning strategy must be chosen without accessing the actual data, it may be based on the known population totals as well as semantics of the problem (e.g., the minors vs. adults distinction in Figure 7.5.)

Example 7.5.5. In our running example, we will choose $\mathcal{P}_s$ to have 5 sets of bins: separate (0-12, 13-18, 19-39, 40-64, 65+), minors (0-18, 19-39, 40-64, 65+), adults (0-12, 13-18, 19+), minors-adults (0-18, 19+), and all (0+). For ease of exposition, in this example all initial weights are equal. The weights assigned to the individuals in each of these possible bins are shown in Figure 7.6. For clarity of exposition, weights in all figures are rounded to the nearest integer.

Remark 7.5.6. In our running example, the binning strategy is effectively a collapsing order on the set of base-level bins, but our results hold more generally. As a simple example, a binning strategy could consist of a single set of bins. In this case our mechanism for differentially-private bin selection would not be necessary.

7.5.2 Iterative Proportional Fitting

So far in our setup, we have not made any assumption about the set of partitions $\mathcal{P}_E(\mathcal{B}(A))$ for which we have known population estimates. In practice, it is often the case that one knows marginal population totals for multiple attributes, but the totals for the join of these attributes are unknown. For example, the American Community Survey uses totals from the Population Estimates Program for two different sets of attributes: age/sex/race and household type.

When this is the case, combining these into a single set of attributes A within our framework will not produce interesting sets of bins. The common approach to this problem is to treat the sets of attributes separately and use *iterative proportional fitting* (IPF) to compute weights. In this section, we give a brief overview of IPF and explain how it fits within our setup.

Let $A^1, \ldots, A^n$ be sets of attributes. We assume the sets A^{i} are mutually disjoint from one another. In this section, we will assume that $\mathcal{P}_E(\mathcal{B}(A^{\mathrm{i}})) = \mathcal{P}(\mathcal{B}(A^{\mathrm{i}}))$ for each i. We now choose n binning strategies $\mathcal{P}_s^1, \ldots, \mathcal{P}_s^n$ with each $\mathcal{P}_s^{\mathrm{i}} \subseteq \mathcal{P}_(\mathcal{B}(A^{\mathrm{i}}))$.

For simplicity of notation, we denote

$$\widetilde{\mathcal{P}}_s := \mathcal{P}_s^1 \times \cdots \times \mathcal{P}_s^n.$$

Definition 7.5.7. Given sets of bins $(\mathcal{S}^1, \ldots, \mathcal{S}^n) \in \widetilde{\mathcal{P}}_s$, let T_0 be the data of a non-negative real number $T_0(S)$ for all tuples of bins $S = (S^1, \ldots, S^n) \in (\mathcal{S}^1, \ldots, \mathcal{S}^n)$. We call T_0 an *initial contingency table* for $(\mathcal{S}^1, \ldots, \mathcal{S}^n)$.

For any integer j, we denote [j] the unique integer in $\{1, \ldots, n\}$ which is congruent to j modulo n.

Definition 7.5.8 (Iterative Proportional Fitting)**.** Let $(\mathcal{S}^1, \ldots, \mathcal{S}^n) \in \widetilde{\mathcal{P}}_s$ be a tuple of sets of bins and let T_0 be an initial contingency table for $(\mathcal{S}^1, \ldots, \mathcal{S}^n)$. Given $S = (S^1, \ldots, S^n) \in \mathcal{S}^1 \times \cdots \times \mathcal{S}^n$, we denote $\mathcal{U}[\mathrm{j}]$ the set of $S' \in (\mathcal{S}^1, \ldots, \mathcal{S}^n)$ which have $S^{[\mathrm{j}]}$ as their j-th coordinate.

For j > 0, iteratively define

$$T_{\mathrm{j}}(S) := T_{\mathrm{j}-1}(S) \cdot \frac{\widehat{N}(S^{[\mathrm{j}]})}{\sum_{S' \in \mathcal{U}[\mathrm{j}]} T_{\mathrm{j}-1}(S')}.$$

Choose some threshold $\delta > 0$ and let $\mathrm{j} > 1$ be the lowest integer satisfying

$$|T_{\mathrm{j}n-k}(S) - T_{(\mathrm{j}-1)n-k}(S)| < \delta$$

for all $k \in \{0, \ldots, n-1\}$ and $S \in \mathcal{S}^1 \times \cdots \mathcal{S}^n$. Then the values $T(S) := T_{\mathrm{j}n}(S)$ are the *IPF contingency table* for $(\mathcal{S}^1, \ldots, \mathcal{S}^n)$ induced by T_0.

Definition 7.5.9. Let $(\mathcal{S}^1, \ldots, \mathcal{S}^n) \in \widetilde{\mathcal{P}}_s$ be sets of bins and let $(S^1, \ldots, S^n) \in \mathcal{S}^1 \times \cdots \times \mathcal{S}^n$ be a tuple of bins. Then the *join-bin* of $(S^1, \ldots, S^n)$ is the set of $B \in \mathcal{B}(A^1 \cup \cdots \cup A^n)$ which, for all i, lie in $\mathcal{S}_{\mathrm{i}}$ when restricted to the attributes in A_{i}.

Remark 7.5.10. Let $(S_1, \ldots, S_n)$ be bins as above and let S be their join bin. Then for any dataset D, we have

$$D \cap S = \bigcap_{\mathrm{i}=1}^{n} D \cap S_{\mathrm{i}}.$$

Definition 7.5.11. Let T_0 be an initial contingency table for $(\mathcal{S}^1, \ldots, \mathcal{S}^n)$ and let T be the induced IPF contingency table. Let $(S^1, \ldots, S^n) \in \mathcal{S}^1 \times \cdots \mathcal{S}^n$ and let S be their join bin. Then $T(S) := T(S_1, \ldots, S_n)$ is called the *IPF population total* of S induced by T.

Moving forward, we will abuse notation by writing $S = (S^1 \ldots, S^n)$ to refer to both the tuple of bins and their join bin.

7.6 Evaluation

Our experiments were created using the 1940 U.S. Census dataset released for testing disclosure avoidance methodologies[25], specifically we assume that the population that we are sampling from is the state of Delaware. Our focus is on counting queries as the post stratification weighting allows for those results to be calculated directly and the data produced by the U.S. Census Bureau includes a large number of counts. We base most of our experiments on a situation similar to that of the American Community Survey where we assume that an outside source of information is available for a limited set of population controls[23], though for simplicity we assume that information is only known about univariate statistics. As such, we assume that information about the contingencies between two attributes is not known and Population Estimates Program can provide for an approximation of this information. Similar to our other experiments, we focus on income (‘INCWAGE’) and focus on similar attributes to our previous experiments, in this case we focused on age, race, sex, educational attainment, and employment status.

Our mechanism makes use of the exponential mechanism to choose binning strategies for three of the attributes - age, race and educational attainment. We highlight that the privacy guarantee of the exponential mechanism is based on the Δf, which is the possible change to the score function, and the α value, for which higher values provide more of a bias toward the higher scoring binning strategies. We specifically focused on providing a larger α for the educational attainment and the age attributes to give them a stronger bias to higher scoring binning strategies, while race was provided a smaller α-value. While this was done based on some experiments with the data, it is possible to do a proxy calculation for this information using the full dataset and the binning strategies in practice would be determined by subject matter experts who are much more knowledgeable about the particulars of the data than we are.

7.7 Empirical Results

Our queries focused on counting the number of individuals below “poverty”. As there is no “official” poverty line for the 1940 data we used \$658 as determined by [53] as the needed income to support a family of four. As stated above, we assume that univariate statistics are known but cross-attribute contingencies are not assumed to be known.

We compare our iterative proportional fitting mechanism to two versions of a naive solution which differ based on the amount of knowledge they require. The “Naive [Pop]” mechanism, the stronger of the two naive mechanisms discussed, takes the proportion of those under the threshold in sample and scales it up to the whole population by multiplying the proportion by the *known* population total. Population in this context is the population of interest for the query. So for example, when querying the number of females ages 30-39 making less than \$658, the use of "Naive [Pop]" would require knowledge of the total count of females ages 30-39, which for the purposes of this experiment we assume are not known. Similarly for the query discussed in Figure 7.9 we require knowledge of the number of females ages 20-29 which we assume is not known. The "Naive [Samp]" mechanism takes the count in sample and multiplies it by the inverse of the base sampling rate. Since we assume our base sampling rate in all experiments to be 10%, this mechanism multiplies the count it gets by $\frac{1}{.1} = 10$ to scale up the results to the larger population. This mechanism assumes that the size of the sub-population is not known. Due to our assumptions, the stronger background knowledge assumption is true for some of the experiments we run and not true for others. We highlight which of the naive solutions is the most appropriate comparison when discussing results.

We divide our results into two sections - a set of biased experiments and a set of unbiased experiments. One of the main practical uses of post stratification weighting, of which iterative proportional fitting is one possible mechanism, is to counter bias in the sampling frame. This bias most commonly comes from either explicit under-sampling or low response rates in certain groups. As such, we wanted to highlight the impact to bias that our mechanism can provide. Ultimately, the mechanism we explore in this chapter comes with a higher variance than either of the naive solutions, so the counter to bias is key to practical use.

All experiments use $\varepsilon = 5$. This value was chosen to be give sufficient ε to each of the queries in the experimental mechanism, it is possible to choose a smaller ε but this will result in greater noise. As some of the ε goes to the exponential mechanism choosing a binning strategy it is possible for a subject matter expert to limit the choices of binning strategies and such actions could help to reduce the privacy budget.

7.7.1 Unbiased Experiments

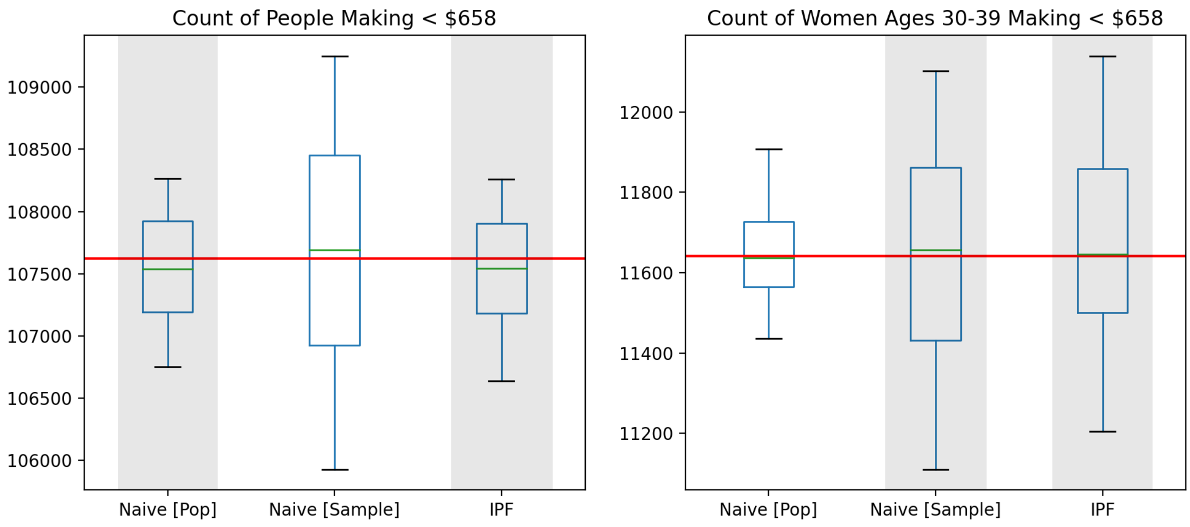

Figure 7.7. The count of those people who make under \$658 in a given population. The left graph is the count for the entire population. When comparing our experimental results to the naive solutions, we believe "Naive [Pop]" to be the more accurate comparison since we assume that the total population is known (as it is the population of the state of Delaware). We are slightly better on variance than the known population naive solution in this experiment. The right graph is focused on the sub-population of females ages 30-39, and as mentioned above we don't expect the population total for this specific group to be available. The accurate comparison, with respect to available background knowledge, would then be the "Naive [Samp]" solution which we can see has much higher variance than our solution even for this relatively small population. Our mechanism has higher variance than the known population naive solution but that mechanism requires strong background knowledge for this sub-population.

The experiments in Figure 7.7 show the impact of our mechanism on an unbiased 10% sample. We note that frequently the use of iterative proportional fitting is done to counter bias, so the use of the mechanism for the simple random sample that we used here was likely overkill and not of much practical use. We can see from the left graph that we provide roughly equivalent results to the "Naive [Pop]" mechanism in both bias and variance, though we win ever so slightly in variance. This result is expected since the fitting mechanism isn't doing much more than scaling the results to the larger population in the same way that the naive solution is. We can see the impact of the fitting mechanism more in the right

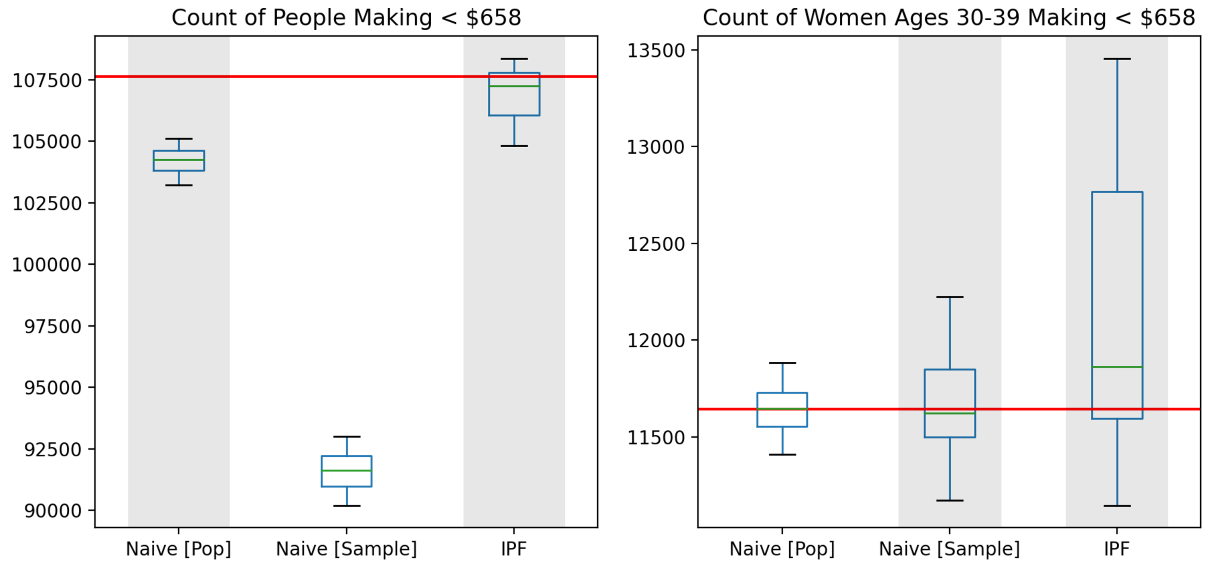

Figure 7.8. The count of those people who make under $658 in a given population. The left graph is the count for the entire population. Here we see the reduction in bias that is possible with our iterative proportional fitting mechanism. We have lower bias than either of the naive solutions. The right graph shows the inverse problem as we have a small amount more bias in our results on women ages 30-39 even though they were sampled without bias. The results are impacted by the fitting mechanism and the resulting weights. The full impact of the fitting mechanism on other groups is not something we were able to meaningfully express so it is possible that the impact can be mitigated by a different set of binning strategies.

hand graph where we see that the iterative proportional fitting mechanism results in higher variance than the "Naive [Pop]" mechanism but has lower variance than the variance than the "Naive [Samp]" due to being a bit more targeted and able to counter the sampling error a bit better than simply inverting the sampling proportion.

7.7.2 Biased Experiments

These experiments introduced a relatively minor bias to a subset of the population. The base sampling rate was 10% while those individuals under 25 were sampled at half the rate

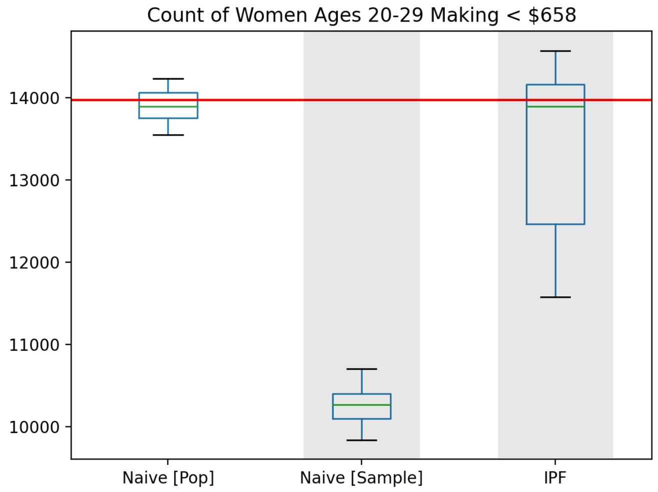

Figure 7.9. The count of those people who make under $658 in a given population. This group is partially biased since we have a lower response for all ages < 25. This graph shows the increased variance from our mechanism, particularly on smaller populations but also shows the bias reduction over the naive solution that matches our background knowledge (Naive [Samp]). We do however, lose to the naive solution if this population total is known.

(5%). As this bias was artificially introduced the impact is not particularly drastic in the bias reduction, but we felt this was a reasonable amount of bias since in practice this type of sampling difference could realistically be by design, as in the case of ACS, or possible to occur in practice, as in the case of non-response. We assume in our case that this bias is unknown by the data curator.

Figures 7.8 and 7.9 show the impact of this bias on various queries on the data. As we are under-sampling this group, which has a higher rate of low incomes (below the threshold) we would assume that under-sampling this group would result in lower counts for these queries. We can see this most obviously in the left graph of Figure 7.8 where our iterative proportional fitting mechanism has less bias than both of the naive solutions, though this comes at the cost of greater variance. We see though in the right hand graph of Figure 7.8 that the fitting mechanism causes some changes to unaffected groups. In that query we had higher variance as well as slightly more bias due to the fitting mechanism. Countering this impact is likely a domain and query-specific modification so ideally a knowledgeable data curator could mitigate this risk. We see from Figure 7.9 that on the population of interest, the population that is partially biased, we have a significant reduction in bias compared to the “Naive [Samp]” mechanism, the mechanism that matches our background knowledge for this query.

Part IV

CONCLUSION

8. CONCLUSIONS AND FUTURE WORK

The main focus of future work will be similar to the existing work as our existing work doesn't adequately solve the underlying problem of meaningful (and hopefully non-technical) privacy in complex surveys. This area in general has been gaining some focus with the US Census Bureau working on multiple aspects of privacy in complex surveys. I think that as more meaningful protections can be guaranteed for governmental data more and more of that data will attempt to be formally private. My particular interest in the evaluation of these mechanisms will be of particular importance since many of the comments from Ch. 4 were the direct result of seeing the current situation with the US Census Bureau's attempts at formal privacy in their data products.

The natural extensions of our work would be focused on the missing data imputation and weighting problems we did initial research in. Missing data imputation is of particular interest since the issue of missing data tends to be universal to basically all surveys. While our initial work was focused on k-Nearest Neighbor imputation, I am currently working on a project attempting to provide more meaningful imputation mechanisms while maintaining privacy with a focus on model-based solution and multiple inference. We are also discussing how this work would expand to fit with the current work on fully synthetic data (as many of the models are similar) and small area estimation (a middle ground between fully synthetic and missing data).

I am also interested in some of the relaxations of ε-differential privacy. While ε, δ-differential privacy is the main relaxation of choice in many circumstances, I find the definition itself to be lacking in many ways. I am also interested in smooth sensitivity since I do not personally feel that complex surveys can possibly consider any real global sensitivity situation due to the possibility of an arbitrarily small number of people standing in for the whole, however smooth sensitivity is notoriously difficult to deal with and find solutions do. I am interested if another version of sensitivity or modification differential privacy might be better able to capture the inherent privacy risks while not requiring significant research and implementation time tailored to solving each problem.

As mentioned before my evaluation work is also focused on simplifying as much of the work of providing privacy as possible and working to help those fields with the actual questions (sociology and economics being the most prominent) to understand the full implications of providing differential privacy and provide the best version of privacy while maintaining a sufficiently strong privacy definition.

REFERENCES

[1] P. Ohm, “Broken promises of privacy: Responding to the surprising failure of anonymization,” *UCLA Law Review*, vol. 57, pp. 1701–1777, 2010. [Online]. Available: https://ssrn.com/abstract=1450006.

[2] M. Barbaro and T. Zeller, *A Face Is Exposed for AOL Searcher No. 4417749*, Apr. 2006. [Online]. Available: https://www.nytimes.com/2006/08/09/technology/09aol.html.

[3] K. Hill, “How target figured out a teen girl was pregnant before her father did,” *Forbes*, Feb. 2012. [Online]. Available: https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/.

[4] A. Narayanan and V. Shmatikov, “How to break anonymity of the netflix prize dataset,” CoRR, Tech. Rep. arXiv:cs/0610105v2 [cs.CR], Nov. 2007. [Online]. Available: http://arxiv.org/abs/cs/0610105.

[5] S. Garfinkel, J. M. Abowd, and C. Martindale, “Understanding database reconstruction attacks on public data: These attacks on statistical databases are no longer a theoretical danger.,” *Queue*, vol. 16, no. 5, pp. 28–53, Oct. 2018, ISSN: 1542-7730. DOI: 10.1145/3291276.3295691. [Online]. Available: https://doi.org/10.1145/3291276.3295691.

[6] L. Sweeney, “Guaranteeing anonymity when sharing medical data, the datafly system,” in *Proceedings, Journal of the American Medical Informatics Association*, Washington, DC: Hanley & Belfus, Inc., 1997.

[7] L. Sweeney, “Uniqueness of simple demographics in the U.S. population,” Laboratory for international Data Privacy (LIDAP-WP4), Tech. Rep., Mar. 2000.

[8] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” in *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE 2006)*, Atlanta Georgia, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1109/ICDE.2006.1.

[9] N. Li and T. Li, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in *Proceedings of the 23nd International Conference on Data Engineering (ICDE ’07)*, Istanbul, Turkey, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1109/ICDE.2007.367856.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proc. of the 3rd Theory of Cryptography Conf.*, 2006, pp. 265–284.

[11] P. Samarati, “Protecting respondent’s privacy in microdata release,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, Nov. 2001. [Online]. Available: http://dx.doi.org/10.1109/69.971193.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1217299.1217302.

[13] L. Willenborg and T. deWaal, *Elements of Statistical Disclosure Control*. Springer Verlag Lecture Notes in Statistics, 2000.

[14] D. Kifer, “Attacks on privacy and definetti’s theorem,” in *SIGMOD Conference*, 2009, pp. 127–138.

[15] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei, “Minimality attack in privacy preserving data publishing,” in *VLDB*, 2007, pp. 543–554.

[16] A. Øhrn and L. Ohno-Machado, “Using boolean reasoning to anonymize databases,” *Artificial Intelligence in Medicine*, vol. 15, no. 3, pp. 235–254, Mar. 1999. [Online]. Available: http://dx.doi.org/10.1016/S0933-3657(98)00056-6.

[17] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, “Toward privacy in public databases,” in *Theory of Cryptography Conference*, Cambridge, MA: IACR, Feb. 2005, pp. 363–385. [Online]. Available: http://www.iacr.org/cryptodb/archive/2005/TCC/3614/3614.pdf.

[18] F. McSherry, “Privacy integrated queries: An extensible platform for privacy-preserving data analysis,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Providence, Rhode Island, Jun. 2009, pp. 19–30. DOI: 10.1145/1559845.1559850. [Online]. Available: https://doi.org/10.1145/1559845.1559850.

[19] K. Nissim, S. Raskhodnikova, and A. Smith, “Smooth sensitivity and sampling in private data analysis,” in *STOC*, 2007, pp. 75–84.

[20] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*, Providence, Rhode Island, Oct. 2007, pp. 94–103. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FOCS.2007.66.

[21] T. Norris, P. Vines, and E. Hoeffel, “The american indian and alaska native population: 2010,” U.S. Census Bureau, Tech. Rep., 2012.

[22] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy: Or, k-anonymization meets differential privacy," in *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'2012)*, Seoul, Korea, May 2012, pp. 32–33. [Online]. Available: http://dx.doi.org/10.1145/2414456.2414474.

[23] "American community survey design and methodology (January 2014)," United States Census Bureau, Tech. Rep. Version 2.0, Jan. 2014. [Online]. Available: https://www.census.gov/programs-surveys/acs/methodology/design-and-methodology.html.

[24] M. Bun, J. Drechsler, M. Gaboardi, and A. McMillan, *Controlling privacy loss in survey sampling (working paper)*, 2020. arXiv: 2007.12674 `[stat.ME]`.

[25] U. C. Bureau, "Disclosure avoidance for the 2020 census: An introduction," U.S. Government Publishing Office, Tech. Rep., Nov. 2021.

[26] K. Warriner, J. C. Goyder, and S. Miller, "Evaluating socio-economic status (ses) bias in survey nonresponse," *Journal of Official Statistics*, vol. 18, pp. 1–12, 2002.

[27] Mar. 2021. [Online]. Available: https://www.brennancenter.org/our-work/court-cases/alabama-v-us-dept-commerce.

[28] C. Kalish, "International crime rates," Bureau of Justice Statistics, Tech. Rep., 1988.

[29] S. Ruggles, S. Flood, R. Goeken, J. Pacas, M. Schouweiler, and M. Sobek, *IPUMS USA: Version 11.0 [dataset]*, Minneapolis, MN, 2021. [Online]. Available: https://doi.org/10.18128/D010.V11.0.

[30] S. Ruggles, S. Flood, R. Goeken, J. Grover, E. Meyer, J. Pacas, and M. Sobek, *IPUMS USA: Version 8.0 extract of 1940 Census for U.S. Census Bureau disclosure avoidance research*, Minneapolis, MN, 2018. [Online]. Available: https://doi.org/10.18128/D010.V8.0.EXT1940USCB.

[31] U. C. Bureau, *2016 american community survey 1-year public use microdata sample*, 2016. [Online]. Available: https://data.census.gov/mdat/#/search?ds=ACSPUMS1Y2016.

[32] U. C. Bureau, *2017 american community survey 1-year public use microdata sample*, 2017. [Online]. Available: https://data.census.gov/mdat/#/search?ds=ACSPUMS1Y2017.

[33] H. Ebadi, T. Antignac, and D. Sands, "Sampling and partitioning for differential privacy," in *14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, NZ, Dec. 2016, pp. 664–673. [Online]. Available: https://doi.org/10.1109/PST.2016.7906954.

[34] F. Zafarani and C. Clifton, *Differentially private naive bayes classifier using smooth sensitivity*, Under review by The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent, Belgium, Sep. 2020.

[35] C. Nicoletti and F. Peracchi, "The effects of income imputation on microanalyses: Evidence from the european community household panel," *Journal of the Royal Statistical Society Series A*, vol. 169, no. 3, pp. 625–646, 16 5 2006. DOI: 10.1111/j.1467-985X.2006.00421.x. [Online]. Available: https://doi.org/10.1111/j.1467-985X.2006.00421.x.

[36] G. Kalton and D. Kasprzyk, "Imputing for missing survey responses," in *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1982, pp. 22–33. [Online]. Available: http://www.asasrms.org/Proceedings.

[37] "American Community Survey design and methodology (January 2014)," United States Census Bureau, Tech. Rep. Version 2.0, Jan. 2014, ch. 10. [Online]. Available: https://www.census.gov/programs-surveys/acs/methodology/design-and-methodology.html.

[38] Y. Wang and X. Wu, "Preserving differential privacy in degree-correlation based graph generation," *Transactions on Data Privacy*, vol. 6, no. 2, pp. 127–145, Aug. 2013. [Online]. Available: http://www.tdp.cat/issues11/abs.a113a12.php.

[39] R. Okada, K. Fukuchi, K. Kakizaki, and J. Sakuma, "Differentially private analysis of outliers," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2015)*, Porto, Portugal, Sep. 2015, pp. 458–473. [Online]. Available: .

[40] S. Fletcher and M. Z. Islam, "Differentially private random decision forests using smooth sensitivity," *Expert Systems with Applications*, vol. 78, pp. 16–31, Jul. 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.01.034.

[41] F. Zafarani and C. Clifton, "Differentially private naive bayes classifier using smooth sensitivity," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 4, to appear, 2021.

[42] A. Gonem and R. Gilad-Bachrach, "Smooth sensitivity based approach for differentially private pca," in *Proceedings of Algorithmic Learning Theory*, vol. 83, Apr. 2018, pp. 438–450. [Online]. Available: http://proceedings.mlr.press/v83/gonem18a.html.

[43] J. C. Bailar III and B. A. Bailar, "Comparison of two procedures for imputing missing survey values," in *Proceedings of the Survey Research Methods Section*, American Statistical Association, 1978, pp. 462–467. [Online]. Available: http://www.asasrms.org/Proceedings.

[44] G. Jagannathan and R. N. Wright, "Privacy-preserving imputation of missing data," *Data & Knowledge Engineering*, vol. 65, no. 1, pp. 40–56, 2007. DOI: 10.1016/j.datak.2007.06.013. [Online]. Available: https://doi.org/10.1016/j.datak.2007.06.013.

[45] F. Chiang and D. Gairola, "Infoclean: Protecting sensitive information in data cleaning," *Journal of Data and Information Quality (JDIQ)*, vol. 9, no. 4, p. 22, May 2018. [Online]. Available: https://doi.org/10.1145/3190577.

[46] Y. Huang, M. Milani, and F. Chiang, "PACAS: Privacy-aware, data cleaning-as-a-service," in *2018 IEEE International Conference on Big Data (Big Data)*, Oct. 2018, pp. 1023–1030. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622249.

[47] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, and T. Kraska, "Privateclean: Data cleaning and differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, San Francisco, California: ACM, Jun. 2016, pp. 937–951. DOI: 10.1145/2882903.2915248. [Online]. Available: https://doi.org/10.1145/2882903.2915248.

[48] C. Ge, I. F. Ilyas, X. He, and A. Machanavajjhala, "Private exploration primitives for data cleaning," arXiv, Tech. Rep. 1712.10266, Aug. 2018. [Online]. Available: arxiv.org/abs/1712.10266.

[49] F. McSherry and I. Mironov, "Differentially-private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, Jun. 2009, pp. 627–636.

[50] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, New Orleans, Jan. 2013, pp. 1395–1414. DOI: 10.1137/1.9781611973105.101. [Online]. Available: https://doi.org/10.1137/1.9781611973105.101.

[51] P. Jain, O. D. Thakkar, and A. Thakurta, "Differentially private matrix completion revisited," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, J. G. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 2220–2229. [Online]. Available: http://proceedings.mlr.press/v80/jain18b.html.

[52] Minnesota Population Center, *Introduction to data editing and allocation*, Aug. 2018. [Online]. Available: https://usa.ipums.org/usa/flags.shtml.

[53] L. Barrington, “Estimating earnings poverty in 1939: A comparison of orshansky-method and price-indexed definitions of poverty,” *The Review of Economics and Statistics*, vol. 79, no. 3, pp. 406–414, 1997, ISSN: 00346535, 15309142. [Online]. Available: http://www.jstor.org/stable/2951387.

[54] R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala, “Optimizing error of high-dimensional statistical queries under differential privacy,” *Proceedings of the VLDB Endowment*, vol. 11, no. 10, pp. 1206–1219, 2018. [Online]. Available: https://doi.org/10.14778/3231751.3231769.

[55] W. H. Qardaji, W. Yang, and N. Li, “PriView: Practical differentially private release of marginal contingency tables,” in *ACM SIGMOD International Conference on Management of Data*, Snowbird, Utah, Jun. 2014, pp. 1435–1446. DOI: 10.1145/2588555.2588575. [Online]. Available: http://doi.acm.org/10.1145/2588555.2588575.

[56] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, ser. Foundations and Trends in Theoretical Computer Science 3-4. Aug. 2014, vol. 9, pp. 211–407, ISBN: 978-1-60198-818-8. DOI: 10.1561/0400000042. [Online]. Available: http://dx.doi.org/10.1561/0400000042.

VITA

Shawn Merrill received his BS in computer science at University of California, Irvine in 2012. His main area of research is privacy preserving data publishing. His interest lies in the interaction between sample methodologies and privacy with a focus on the practical aspects of providing privacy for existing products. He has worked with the US Census Bureau on applying privacy to their products. He received his PhD in December 2021 from the Department of Computer Science at Purdue University.