TRANSACTIONS ON DATA PRIVACY 13 (2020) 171–200

The Impact of Range Constraints on Utility in the Design of Differentially Private Mechanisms

William Lee Croft*, **Jörg-Rüdiger Sack***, **Wei Shi****

*Carleton University, School of Computer Science.

**Carleton University, School of Information Technology.

E-mail: leecroft@cmail.carleton.ca

Received 8 November 2019; received in revised form 14 August 2020; 14 December 2020

Abstract. For the design of differentially private mechanisms, knowledge of constraints on query responses can often be leveraged to improve the utility of a mechanism. This is typically considered in the non-interactive setting, where constraints over batches of query responses can be exploited. Comparatively, little attention is given to the design of constraint-adherent mechanisms in the interactive setting, where queries are posed on an individual basis. The absence of relationships between batched responses in the interactive setting removes much of the structure that mechanisms in the non-interactive setting rely on. Yet, if the valid range of a query is known, the design of range-adherent mechanisms remains a possibility. The generation of noisy responses strictly within the valid range of the query can serve to improve the utility of the mechanism. Furthermore, adherence to range constraints is often beneficial for compatibility with downstream software used to analyze the noisy responses.

In this paper, we consider the design of range-adherent mechanisms for general numeric queries in the interactive setting. We first study requirements and desirable properties for range-adherent mechanisms using a matrix representation of discretized probability density functions. We then propose a linear programming approach for the design of range-adherent mechanisms using a user-independent criterion for optimal utility. We run experiments to compare our linear programming mechanisms to other range-adherent alternatives. In these experiments, we measure utility both in terms of the usability of the noisy query responses as well as the information preservation of the mechanisms. The results demonstrate that our linear programming mechanisms achieve higher utility when compared to existing mechanisms. The improvements in utility are most pronounced when there is a high ratio of query sensitivity to query range. Our mechanisms are thus particularly useful for queries posed on small-sized databases which are more vulnerable to privacy breaches.

Keywords. Differential privacy, Mechanism design, Query range constraints

1 Introduction

It is well-known that when performing analysis on sensitive data, a trade-off between privacy and utility exists. The nature of this trade-off has been studied in the context of a

wide variety of disclosure control methods [8, 11, 30]. Among these methods, differential privacy [12] has garnered a great deal of attention [13, 34] due to the robust, formal privacy guarantee that it provides. Informally, differential privacy offers a guarantee on the level of distinguishability between potential configurations of a sensitive database by adding controlled noise via a randomization mechanism to responses of queries posed on the database. Key properties such as resistance to composition attacks [20] and abstraction from most forms of attacker background knowledge [3, 15] have been important factors in establishing the strength of the privacy guarantee. However, the design of mechanisms that maintain acceptable levels of utility is also an important challenge [6, 10, 14, 24].

In cases where constraints are publicly known about the posed queries, these constraints may be exploited for the design of mechanisms better suited for the preservation of data utility. Particularly in the non-interactive setting, where batches of queries are posed, known constraints over the set of query responses can be used to enforce consistency on the noisy responses in order to improve utility [2, 19]. For example, the sum of the responses for counting queries over disjoint subsets of the database should be equal to the response to a counting query over the union of the same subsets. In the interactive setting, where potentially unrelated queries are answered individually, such constraints are not present since there are no other query responses with which relationships must be preserved. However, knowledge about the range of the query can still be exploited to ensure that noisy query responses fall within the valid range [7, 32].

When the range is known for a database attribute on which a query is posed, it is often possible to infer the valid range of the query response. For example, responses to queries such as min, median, mean and max all adhere to the same range as the attribute itself. Knowledge on attribute ranges can be obtained in a number of ways. In some cases, details about the collection of data may be shared (e.g., predefined attribute value discretizations, use of signal capturing devices with known ranges, etc.) or ranges may be common sense (e.g., percentages are bounded to [0,100], counts over n records are bounded to [0,n], etc.). In practice, most mechanisms in the interactive setting ignore range constraints, allowing for noisy responses to potentially be generated outside the valid range of the query. Since range constraints are quite natural in many cases, this information should be leveraged. A number of reasons motivate the design of range-adherent mechanisms; these include downstream compatibility with other software and improved utility of the noisy data.

1.1 Motivating Example

Consider an example in which a survey is conducted using questions answered with a 5-point Likert scale (i.e., answers are integers in the range [1,5]). This is a common format in scientific studies which involve participant/subject groups and in customer feedback surveys. While it is desirable to analyze and share the data, it is necessary to protect the privacy of the individuals who participate in such studies. Common queries that might be posed on such data include min, median and max queries. These queries, however, are highly sensitive to changes in the underlying data and thus require a high degree of noise to achieve differential privacy. As a result, when such queries are posed via a differentially private mechanism, a large amount of the probability mass assigned by most mechanisms falls outside the valid range of the query. For instance, when the true query response is 3, use of the Laplace mechanism [12] configured for $\epsilon = 1$ results in 60% of the probability mass being assigned to values outside of the valid range of the query (i.e., less than 1 or greater than 5). Such a wide dispersion of probability mass in contrast to such a small valid range for the query is clearly undesirable from the standpoint of utility.

Adherence to range constraints can trivially be achieved by snapping out-of-bounds responses to the nearest valid value. While this avoids the issue of large amounts of out-of-bounds probability mass, this approach may not be conducive to achieving high levels of utility. In Table 1, we show the probability distribution that would be used by the Laplace mechanism configured for $\epsilon = 1$ when discretized to the valid responses using boundary-snapping. Each column represents the probability distribution used by the mechanism given a particular true query response while each row corresponds to one of the potential noisy responses. A cell entry indicates the conditional probability of a noisy response given a true response. The first and last rows of the table show large spikes in probability due to the out-of-bounds probability mass that has been pooled on the boundary response categories. This pooling of large amounts of probability mass leads to undesirable probability distributions. For example, when the true response is 2, the mechanism is over two times more likely to report 5 than to report 2 as the noisy response. When the true response is 3, the mechanism assigns the highest probabilities to reporting 1 or 5, each of which are nearly three times more likely to be reported than the true response.

	$f(D) = 1$	$f(D) = 2$	$f(D) = 3$	$f(D) = 4$	$f(D) = 5$
$K(f(D)) = 1$	0.559	0.441	0.344	0.268	0.208
$K(f(D)) = 2$	0.098	0.118	0.098	0.076	0.059
$K(f(D)) = 3$	0.076	0.098	0.118	0.098	0.076
$K(f(D)) = 4$	0.059	0.076	0.098	0.118	0.098
$K(f(D)) = 5$	0.208	0.268	0.344	0.441	0.559

Table 1: Conditional probability distribution used by a Laplace mechanism K configured with $\epsilon = 1$ for a query f with range [1, ..., 5] posed on a database D. Columns correspond to true query responses while rows correspond to noisy responses.

Alternatively, mechanism design can integrate information about the valid range of the query in order to adhere to the range constraints while also taking the utility of the noisy responses into consideration. In Table 2, we show the probability distribution used by a mechanism generated from a linear program we propose in this paper. Here, the probability mass is more focused around the true query responses. Throughout this paper, we discuss properties that are desirable in range-adherent mechanisms and propose linear programs to generate mechanisms that exhibit these properties.

	$f(D) = 1$	$f(D) = 2$	$f(D) = 3$	$f(D) = 4$	$f(D) = 5$
$K(f(D)) = 1$	0.322	0.322	0.119	0.119	0.119
$K(f(D)) = 2$	0.322	0.322	0.220	0.119	0.119
$K(f(D)) = 3$	0.119	0.119	0.322	0.119	0.119
$K(f(D)) = 4$	0.119	0.119	0.220	0.322	0.322
$K(f(D)) = 5$	0.119	0.119	0.119	0.322	0.322

Table 2: Conditional probability distribution used by a linear programming mechanism K configured with $\epsilon = 1$ for a query f with range [1, ..., 5] posed on a database D. Columns correspond to true query responses while rows correspond to noisy responses.

1.2 Contributions and Paper Outline

In this paper, we study the impact of adherence to range constraints for general numeric queries in the interactive setting and make the following contributions:

- We provide a theoretical discussion on the desirable properties of range-adherent mechanisms and formalize a matrix representation of range-adherent mechanisms for general numeric queries.
- We propose two variants of a range-adherent linear programming mechanism using our discretized mechanism representation. By employing a user-independent criterion for optimal utility, we produce mechanisms that are useful to a broad range of users.
- We conduct experiments on our proposed mechanisms and a variety of existing mechanisms. We apply a number of measures of utility capturing the usability and information preservation of the mechanisms.
- We analyze the results of our experiments and discuss the implications of adherence to range constraints for the usability and information preservation of the mechanisms. We compare the performance of the mechanisms and demonstrate improvements in the utility of our linear programming mechanisms over the commonly used boundary-snapping method.

The remainder of the paper is structured as follows: In Section 2, we provide a review of the relevant literature on utility and adherence to constraints in differentially private mechanisms. In Section 3, we discuss the design of range-adherent mechanisms and formalize their representation. Within this formalized representation, we redefine the existing method of boundary-snapping and propose our linear programming mechanisms. Finally, in Section 4, we run experiments to compare a number of measures of utility over different range-adherent mechanisms as well as non-adherent variants.

2 Related Work

As the concept of differential privacy has been gaining momentum, the study of utility in differentially private mechanisms has become an increasingly important topic. Utility has often been considered in terms of distortion, relying on some measure of distance between true query responses and noisy query responses. Reduction of distortion has been a common goal in the design of mechanisms [6,26] and minimization of distortion has been used as a criterion for optimality [38].

It has also been proposed that rational users will make use of their prior knowledge and information about the randomization mechanism. They do this by applying Bayesian post-processing to noisy query responses in order to remap the noisy responses to values that minimize a loss function. For this class of users, a geometric mechanism [24] has been proposed which minimizes the expected loss for all users (i.e., users having any configuration of prior knowledge) when applied to counting queries. This result has been extended to risk-averse users (i.e., users who wish to minimize the worst-case for their expected loss in utility over all possible mechanism input values) [25].

Another study examining the same measure of expected loss shows that counting queries are essentially the only type of query for which such a universally optimal mechanism

can be designed [4]. Given this result, subsequent studies have focused on the design of non-universal optimality. A non-universally optimal mechanism has been proposed for restricted classes of Bayesian users [16] and a staircase mechanism [23] has been proposed to provide optimal utility for the class of risk-averse users who do not possess any prior knowledge about the query responses.

Due to the variety of utility measures available, a number of these measures have been studied within the context of axioms of utility [31]. A distinction is made between the utility of a mechanism in terms of usability (i.e., ease of use for specific tasks) and information preservation (i.e., how much of the original information is maintained in the noisy query responses). The results of the study show that distortion-based measures of utility do not satisfy all axioms of utility whereas measures of expected loss for Bayesian users do.

A commonly-used approach to improve utility in noisy query responses is to pose a large number of related queries in a batch (i.e., in the non-interactive setting) and to post-process the noisy responses in order optimize a utility goal subject to known relationships between the true query responses. In the context of the differentially private release of marginals (i.e., projections of a contingency table onto a subset of attributes), a transformation to the Fourier domain and the use of linear programming has been applied to ensure consistency between the marginals and generation of natural numbers as the noisy counts [2]. A subsequent study has shown that for values of ϵ needed to ensure a reasonable level of privacy, the noise induced by the aforementioned method leads to inadequate statistical inferences [17]. To improve upon this, post-processing has been proposed to remap a vector of noisy query responses to a new vector that satisfies known relationships between the elements of the vector while minimizing the L_2 distance between the vectors [27]. In [28], it is shown that more accurate responses can be obtained by instead minimizing the L_1 distance as this accounts for the distribution of the noise added to each element of the vector. Some studies in the non-interactive setting have also focused on the release of sanitized databases tailored to application-specific optimization problems. This has been studied in the context of objectives and constraints pertaining to optimal powerflow benchmarks [19] and mobility service scheduling [18].

While most optimization-based approaches in the non-interactive setting involve post-processing a set of noisy query responses, the matrix mechanism [29] handles optimization by remapping the queries themselves prior to posing them on the sensitive database. The underlying idea is to determine an alternate set of queries that offers more efficient access to the database (e.g., by eliminating redundancy) while retrieving sufficient information to answer the original set of queries. While this appears to be fundamentally different from other optimization approaches in the non-interactive setting, the authors show that [27] is in fact an instantiation of the matrix mechanism.

The approaches reviewed in the non-interactive setting share a common objective with our work in the use of optimization-based techniques for the maximization of utility in query responses subject to known constraints. However, they differ fundamentally from our work in their reliance on relationships between batched query responses. When moving to the interactive setting, such relationships are no longer present. Our work is therefore more closely related to interactive setting mechanisms that focus on improving utility through adherence to the range of the query being posed. A well-known instance of this is the truncated variant of the geometric mechanism [24, 25] which snaps out-of-bounds noisy responses of counting queries to the nearest valid value. This approach leads to inflated probability mass on the boundaries of the query range. An alternative, explicit fair, mechanism has been proposed to adhere to range constraints on counting queries while satisfying additional properties on the distribution of the probability mass [7]. These prop-

erties lead to better utility for query responses that are nearer to the center of the query range. Variants of the Laplace mechanism have also been studied in terms of truncation for adherence to the query range [32]. These variants include the snapping of out-of-bounds values in a manner analogous to the truncated geometric mechanism, and normalization of a truncated mechanism distribution. However, normalization alone on a truncated Laplace mechanism is not sufficient to preserve the differential privacy guarantee and adjustment of the calculation for the Laplace scaling parameter is also needed [9].

Other instances of truncated Laplace mechanisms have been proposed in the literature. However, their intent has not been adherence to a query range. In the context of approximate differential privacy [14], a truncated variant of the Laplace mechanism has been designed to maximize the probability mass decay rate of the distribution [21]. A truncated-uniform-Laplace mechanism has also been proposed for the purpose of uniformly most powerful tests on binomial data [1]. In both cases, truncation is based on mechanism parameters and is symmetric around the location parameter (i.e., the true query response). Thus, although these forms of truncation are related to utility, they do not enforce adherence to the query range.

3 Constraint-Aware Mechanisms

We open this section with a discussion on the desirable properties of range-adherent mechanisms. We then formalize a matrix representation of range-adherent mechanisms. We redefine boundary-snapping within this representation and study some of the properties it provides. Finally, we propose a linear programming approach for the creation of range-adherent mechanisms that optimize a user-independent criterion for utility.

3.1 Design Considerations

We propose that the design of mechanisms for the interactive setting should incorporate publicly known range constraints of queries by restricting the noisy output to fall within the valid range of the query. Our goal is to design more useful mechanisms in terms of downstream compatibility and utility. There are a number of ways to approach this task and the design decisions will impact the resultant utility of the mechanism. It is important to first consider the intended meaning of the term "utility". We therefore begin this section with a theoretical discussion on the design of range-adherent mechanisms. We then describe the impact that design decisions have on utility.

The "utility" of a mechanism may refer to either information preservation or usability [31]. The former refers to how much of the original information is carried in the noisy responses, whereas the later refers to how directly useful the noisy responses are to a particular user. The axiom of sufficiency [31] states that for any mechanism M_1 that can be composed with an arbitrary function to simulate another mechanism M_2, the information preservation of M_1 is greater than or equal to that of M_2. This is because any inferences that could be drawn from the behaviour of M_2 could equally be drawn from M_1 by simulating M_2 if desired. An important consequence of this axiom is that although post-processing of the noisy responses (without further access to the database) may improve usability, it cannot improve information preservation and may even be detrimental in that regard.

A commonly used method of adherence to range constraints is a step of post-processing to snap out-of-bounds query responses to the nearest valid value [25, 32]. We henceforth refer to this method as boundary-snapping. By the axiom of sufficiency, incorporation

of boundary-snapping into a mechanism produces a new mechanism with degraded information preservation. The non-constrained mechanism can simulate the one that uses boundary-snapping but the opposite is not true. Although adherence to range constraints is achieved, there is no other consideration made for how this benefits the utility of the mechanism.

An alternative method of post-processing which achieves adherence to range constraints is Bayesian post-processing [24]. This involves remapping noisy query responses to valid values in order to optimize a user-specific measure of utility. This form of post-processing relies on the specification of the user's prior knowledge of the sensitive data (represented as a probability distribution over the valid query responses) and the user's utility goal (represented as a loss function). Although this improves the usability of the responses, its reliance on specific user settings causes a degradation in information preservation for users operating under different settings. As such, this should be kept as a separate step which is either handled independently by the user or is provided as an optional operation on a per-user basis.

Although Bayesian post-processing applied as a separate step may be an appropriate solution in some situations, there are also scenarios where it cannot be applied. If the mechanism is used as a building block in a larger system that does not allow for the specification of additional user settings, post-processing cannot be properly applied. The use of default settings can be detrimental for the utility of many users and thus should not be used. In other cases, users may simply not yet be aware of the ideal objective function to apply for their data analysis and are thus unable to apply post-processing. Furthermore, only the remapping process is optimal with respect to the user's objective function; the underlying mechanism provides no such guarantee and may not be well-suited to a range-constrained setting. Ideally, a mechanism should be able to adhere to range constraints while optimizing a general goal for utility that is independent of user settings.

3.2 Formalization of Range-Adherent Mechanisms

To facilitate the incorporation of range-adherence into mechanism design, we next provide a formal representation of range-adherent mechanisms. To this end, we employ a matrix representation of mechanisms along with a set of requirements for range-adherence. We also consider the inclusion of optional requirements on the matrix which may be conducive to higher levels of utility.

A differentially private *mechanism* can be defined as a stochastic function that adds controlled noise to responses to queries posed on databases. That the noise must be controlled refers to the requirement for a privacy guarantee to be satisfied for all pairs of *adjacent* databases. For a set of valid databases $\mathbb{D}$, a pair of databases $D_1, D_2 \in \mathbb{D}$ are adjacent if they differ by a single record. More formally,

Definition 1. Let $\mathbb{R}$ be the set of real numbers, $f : \mathbb{D} \to \mathbb{R}$ be a query function, $K : \mathbb{R} \to \mathbb{R}$ be a randomization mechanism and ϵ be a *privacy parameter* selected by the data custodian. For K to satisfy the **differential privacy guarantee**, the following condition must hold for all pairs of adjacent databases D_1 and D_2:

$$\Pr\left(K\left(f\left(D_1\right)\right) = \mathrm{r}\right) \leq e^{\epsilon} \Pr\left(K\left(f\left(D_2\right)\right) = \mathrm{r}\right) \qquad \forall \mathrm{r} \in \mathbb{R}. \tag{1}$$

In practice, the range of a query may correspond to a particular subset of $\mathbb{R}$. We refer to such a correspondence as query range constraints. We wish to define requirements for the adherence of mechanisms to query range constraints. For this, we first consider the

treatment of the underlying *probability density function (PDF)* used by the mechanism to generate noise. We discretize the range of the PDF such that it can be mapped to a set of discretized response categories. Each response category represents a range of values such that reporting the category indicates a noisy query response that falls within the corresponding range. Let $P_{K(f(D))} : \mathbb{R} \to \mathbb{R}$ be the PDF used by the mechanism K for a query f posed on a database $D \in \mathbb{D}$ and let $\mathfrak{R} \in \mathbb{R}$ be the finite, continuous range that we wish to discretize. We define r_l and r_u to be the lower and upper bounds, respectively, of $\mathfrak{R}$.

Definition 2. Let R be a set of **discretized response categories** over a range $\mathfrak{R}$. Each $R_i \in R$ is a finite, continuous range, having lower and upper bounds ${r_i}^l$ and ${r_i}^u$ respectively such that $r_l \leq {r_i}^l \leq {r_i}^u \leq r_u$. The probability of a noisy response $R_i \in R$ is given by:

$$\Pr\left(K\left(f\left(D\right)\right) \in R_i\right) = \int_{{r_i}^l}^{{r_i}^u} P_{K(f(D))}(x)\, dx. \tag{2}$$

The discretization of a mechanism's PDF allows for the representation of the mechanism as a matrix having true query responses as columns, noisy query response categories as rows and conditional probabilities as the matrix entries (e.g., $M_{i,j}$ is the probability of reporting R_i given the true query response F_j). The matrix representation provides a convenient format for considering mechanism modifications for adherence to query range constraints.

Definition 3. Let $F \subset \mathbb{R}$ be the set of true responses for a query f and let R be the set of discretized noisy response categories. A **mechanism matrix** M is an $|R| \times |F|$ matrix such that its entries $M_{i,j}$ correspond to the conditional probability distribution, defined by a mechanism K as follows:

$$M_{i,j} = \Pr\left(R_i | F_j\right) = \Pr\left(K\left(F_j\right) \in R_i\right) \qquad i = 1, ..., |R|\,,\ j = 1, ..., |F|\,. \tag{3}$$

To avoid potential confusion, we emphasize the difference between the terms "mechanism matrix" and "matrix mechanism" [29]. The former refers to a matrix of mechanism probabilities as described in Definition 3 which we use throughout this paper. The later refers to a mechanism designed for the non-interactive setting which is largely unrelated to our work.

In order to provide a valid conditional probability distribution, a mechanism matrix must have a column for each possible true query response such that each of these columns constitutes a valid probability distribution. For a mechanism matrix to be considered a valid differentially private mechanism, its entries must additionally satisfy the differential privacy guarantee. To be considered range-adherent, we assert that the mechanism must use a discretization of noisy response categories that is finite and that covers all of the true query responses. We formalize these requirements for a mechanism matrix as follows:

- **R.1** The conditional probability distribution must be differentially private:

$$M_{i,j} \leq e^{\epsilon} M_{i,k} \quad \forall D_1, D_2 \in \mathbb{D} \text{ such that } f\left(D_1\right) = F_j,\ f\left(D_2\right) = F_k,\ |D_1 - D_2| = 1. \tag{4}$$

- **R.2** The noisy response categories must be disjoint and their union must be finite and must contain all of the true query responses:

$$R_i \bigcap R_j = \emptyset \qquad i = 1, ..., |R|\,,\ \ j = 1, ..., |R| \text{ such that } i \neq j, \tag{5}$$

$$r_l \neq -\infty, \quad r_u \neq \infty, \tag{6}$$

$$F_j \in \bigcup_{i=1}^{|R|} R_i \qquad j = 1, ..., |F|\,. \tag{7}$$

- **R.3** Each true query response must correspond to a column representing a valid probability distribution:

$$M_{i,j} \geq 0 \qquad i = 1, ..., |R|\,,\ \ j = 1, ..., |F|\,, \tag{8}$$

$$\sum_{i=1}^{|R|} M_{i,j} = 1 \qquad j = 1, ..., |F|\,. \tag{9}$$

We note that in some cases, one might additionally require that the union of all the noisy query responses covers the full span of the chosen range $\mathfrak{R}$. We omit this requirement for the sake of a more general representation as it may be desirable in some cases to allow for some portions of the range to never be reported. For example, with counting queries, it is reasonable to only report integer values as the noisy query responses.

The matrix representation also allows for the inspection of further mechanism properties which may help to improve utility in certain cases. We use the notation i_j to indicate the index of the noisy response category in which the true query response of index j falls (i.e., $F_j \in R_{i_j}$). We now consider properties that have been previously studied in the context of mechanisms designed for answering counting queries on constrained ranges [7]. We generalize these properties here in the context of mechanism matrices for general numeric queries:

- **Column Monotonicity -** For any fixed true response F_j, the probability of reporting a noisy response R_i given F_j monotonically decreases as the distance between between F_j and R_i increases:

$$M_{i,j} \leq M_{i+1,j} \qquad i = 1, ..., i_j - 1,\ \ j = 1, ..., |F|\,, \tag{10}$$

$$M_{i,j} \geq M_{i+1,j} \qquad i = i_j, ... |R| - 1,\ \ j = 1, ..., |F|\,. \tag{11}$$

- **Row Monotonicity -** For any fixed noisy response R_i, the probability of reporting R_i given a true query response F_j monotonically decreases as the distance between between R_i and F_j increases:

$$M_{i,j} \leq M_{i,j+1} \qquad i = 1, ..., |R|\,,\ \ j = 1, ..., |F| - 1 \text{ such that } F_{j+1} \leq r_i^u, \tag{12}$$

$$M_{i,j} \geq M_{i,j+1} \qquad i = 1, ..., |R|\,,\ \ j = 1, ..., |F| - 1 \text{ such that } F_j \geq r_i^l. \tag{13}$$

- **Symmetry -** The matrix has 2-fold rotational symmetry (equivalent to centrosymmetry in this context):

$$M_{i,j} = M_{|R|-i+1,|F|-j+1} \qquad i = 1, \dots |R|\,,\ \ j = 1, \dots |F|\,. \tag{14}$$

- **Fairness -** The probability of reporting the noisy response category containing the true query response is the same for all true query responses:

$$M_{i_j,j} = M_{i_{j'},j'} \qquad j = 1, \dots |F|\,,\ \ j' = 1, \dots |F|\,. \tag{15}$$

The use of a PDF that monotonically decreases as distance from the location parameter increases (referred to henceforth as a monotonic PDF) can be desirable from a data analysis perspective. This ensures that, as one would typically expect, the probability of noisy responses is inversely related to their distance to the true query response. The properties of column and row monotonicity guarantee that this concept holds in the context of a mechanism matrix. These properties are considered to help avoid certain pathological behaviours that may result from blindly following optimization of an objective function in the design of a mechanism [7]. For example, the geometric mechanism provides optimal utility for all Bayesian users [24]. Yet, its well-known truncated variant (which snaps out-of-bounds responses to the nearest valid value) does not maintain column monotonicity as it induces spikes in probability mass on the noisy response categories at the query range boundaries. This behaviour has been shown to lead to poor utility for cases where the true query responses are near the center of the range, particularly in the case of small databases [7].

The properties of fairness and symmetry enforce further regularity on the mechanism matrix to facilitate analysis. Continuing the example of the truncated geometric mechanism, the spike in probability mass on the boundary can lead to an over-representation of counts corresponding to the boundary response categories. The property of fairness helps to mitigate this by requiring that the probabilities of reporting the true query responses are all equal to each other. Similarly, the property of symmetry helps to mitigate bias towards certain noisy responses by requiring the mechanism matrix to be centrosymmetric. The example of the boundary-snapping Laplace mechanism in Table 1 demonstrates the spikes in probability mass that occur due to a lack of adherence to fairness and column monotonicity. The linear programming alternative shown in Table 2 adheres to row and column monotonicity, fairness and symmetry, thus avoiding such spikes in probability mass.

Under the assumption that the range of a mechanism spans infinitely, many mechanisms used in the interactive setting already satisfy these properties. Row and column monotonicity follow from the use of a monotonic PDF. Any data-independent mechanism will also satisfy the property of fairness since the true query response will not impact the probability of reporting the original value. These properties apply to well-known mechanisms such as the Laplace mechanism [12], the staircase mechanism [23] and the geometric mechanism [24]. However, the adaptation of a mechanism to a finite range necessitates modification to the discretized PDF in order to achieve requirement R.3. It is through such modification that the aforementioned properties may be lost.

3.3 Boundary-Snapping

Next, we redefine the concept of boundary-snapping in the context of a mechanism matrix. Using this representation, we prove certain properties that boundary-snapping mechanisms possess. The boundary-snapping method is commonly used when mechanism re-

sponses must conform to a known range in the interactive setting [7, 32]. This operation can be directly incorporated into the probability distribution of a mechanism with a finite range by reallocating all out-of-bounds probability mass to the nearest noisy response category. In the context of the discretization defined in Formula (2), this entails changing the lower bound of the first response category to negative infinity and the upper bound of the last response category to infinity.

Definition 4. For any mechanism K and discretized range R, the **boundary-snapping** mechanism matrix representation of K is an $|R| \times |F|$ matrix M with its entries defined as follows:

$$M_{i,j} = \begin{cases} \int_{-\infty}^{r_i^u} P_{K(F_j)}(x)\ dx & i = 1 \\ \int_{r_i^l}^{\infty} P_{K(F_j)}(x)\ dx & i = |R| \\ \int_{r_i^l}^{r_i^u} P_{K(F_j)}(x)\ dx & \text{otherwise} \end{cases} \qquad i = 1, ..., |R|\,,\ j = 1, ..., |F|\,. \tag{16}$$

Next, we derive several properties possessed by boundary-snapping mechanism matrices defined as in Formula 16.

Theorem 1. The boundary-snapping mechanism matrix of Definition 4 satisfies all three requirements of a valid differentially private, range-adherent mechanism when applied to any differentially private mechanism K for a discretized range R that covers a finite and continuous range $\mathfrak{R} \subset \mathbb{R}$ such that $F \subset \mathfrak{R}$.

Proof. This implementation of boundary-snapping is equivalent to the post-processing step of snapping the noisy responses to the nearest valid value. Since differential privacy is immune to privacy breaches induced by post-processing [15], requirement R.1 is satisfied, provided the original mechanism K is differentially private. Since R is a finite discretization of a range that covers all true query responses, R.2 is satisfied. Furthermore, as R is a discretization of a continuous range $\mathfrak{R}$, each matrix column j contains the full probability mass of a valid PDF $P_{K(F_j)}$ broken up over $|R|$ entries corresponding to integrals of $P_{K(F_j)}$. Since the probability density in a PDF can never drop below zero and the union of the integrals covers the full range $\mathbb{R}$ of $P_{K(F_j)}$, the column entries correspond to a valid probability distribution and satisfy requirement R.3. □

Theorem 2. Let K be a range-adherent mechanism produced through the application of boundary-snapping to a mechanism matrix for any symmetric and monotonic PDF. Under the assumptions that the discretized response categorizes are uniformly sized and the true query responses are uniformly spread across the range of the query, K preserves the properties of row monotonicity and symmetry.

Proof. For any discretization using uniformly sized response categories, each such category corresponds to a uniformly sized integral of the underlying PDF. If the PDF is monotonic, any sequence of such integrals to one side of the location parameter corresponds to a monotonic sequence of probability masses. Prior to the inflation of probability mass induced by boundary-snapping, any mechanism matrix produced under this configuration therefore provides row monotonicity. If the true query responses are uniformly spread across the range of the query, this initial matrix will also possess the property of symmetry. We must show that these properties are preserved after the inflation of probability mass in the boundary response categories.

Row Monotonicity: The new probability mass associated with any boundary response category is the sum of its original value and all probability mass beyond its outer boundary. Let x be the original value of a boundary response category and x' be the shifted probability mass that is added to it. Since the underlying PDF is monotonic, the value of x is determined by a monotonic function of the distance d between the boundary response category and the location parameter. Similarly, the newly added probability mass x' is determined by a monotonic function of the same distance d as this determines how much probability mass has been shifted. Since both values that contribute to the sum monotonically decrease as the distance from the location parameter increases, it follows that the new boundary response categories maintain the property of row monotonicity.

Symmetry: The dependence of the inflated probability mass on distance between the location parameter and the boundary also ensures that symmetry continues to hold. For an arbitrary response category $M_{1,j}$ along the lower boundary, the true query response is $j-1$ indices away from the lower boundary. We must show that $M_{|R|,|F|-j+1}$ is subject to the same increase in probability mass. This is a boundary response category on the upper boundary and the true query response is $j-1$ indices away from this boundary. Since the true query responses are uniformly spread across the range of the query, it follows by the definition of boundary-snapping that the increase in probability mass will be the same. The property of symmetry therefore holds. □

A consequence of applying boundary-snapping to a mechanism matrix is that column monotonicity may not be preserved. Boundary-snapping has the potential to increase the probability mass of a boundary response category to a value higher than that of its adjacent, non-boundary response category. This can occur when the true query response is near the boundary, causing a large amount of out-of-bounds probability mass to be shifted. If the true query response does not fall in the range covered by the boundary response category, the probability mass of the response categories no longer decreases monotonically as distance from the query response increases. This is in violation of column monotonicity.

Similarly, the property of fairness is not guaranteed to be preserved under the application of boundary-snapping. For every pair of true query responses $j, j' \in F$ such that $j \neq j'$, fairness requires that $M_{i_j,j} = M_{i_{j'},j'}$. However, if i_j is a boundary response category and $i_{j'}$ is not, the inflation of probability mass invalidates this condition. As such, fairness is not guaranteed to hold under the application of boundary-snapping.

3.4 Linear Programming

The adherence to range constraints achieved by boundary-snapping is an ad-hoc adaptation to mechanisms that were not designed for a range-constrained setting. As a consequence, it does not take into account the impact on the utility of the mechanism. Here, we wish to improve on this by proposing the use of linear programming to derive mechanisms that optimize a general objective function in order to provide a good level of utility for a broad range of users.

Outside of the context of counting queries, it is well-known that it is impossible to design mechanisms of universal optimality [4] (i.e., optimality for all users). We therefore cannot optimize utility for all Bayesian users. An alternative to optimality for Bayesian users is the notion of optimal risk-averse utility in which users wish to minimize the expected loss of the worst-case true query response [25,33]. However, the measure of expected loss is taken in terms of user-specific utility goals, making the design of universally optimal mechanisms difficult for this form of utility as well. Although the staircase mechanism [23] offers

optimal utility in general numeric queries for risk-averse users, it requires the assumption that users do not have any knowledge about range constraints of the query [22]. We note that there is an existing formulation of a linear program for risk-averse users posing counting queries [25]. However, this applies a user-specific objective function and is only used as an example rather than a proposed mechanism.

We therefore turn to an alternative criterion for optimality that abstracts from user-specific utility by considering the concentration of probability mass around the location parameter in the mechanism's PDF [38]. Intuitively, the concept is that for a given PDF, any rational user should prefer an alternative that can be derived from the given PDF by shifting probability mass closer to the location parameter as this would reduce the expected noise. To formalize this notion, an order was defined over PDFs based on the ability to obtain one PDF from another by shifting probability mass closer to the location parameter.

Definition 5. Let y_1 and y_2 be two PDFs having the same location parameter μ. The PDF y_1 is considered to be **smaller** than y_2, denoted as $y_1 \leq y_2$, if y_1 can be obtained from y_2 by shifting probability mass closer to μ.

Definition 6. For any class Y of PDFs, a PDF $y \in Y$ is considered to be **optimal** if it is minimal within Y.

Under the assumption of data-independence, an optimal, staircase-shaped, differentially private mechanism was derived for this criterion [38]. The assumption of data-independence was used to assert that the PDFs used for any true query response differ only in their location parameters. As a result, the PDFs used by the mechanism are all shaped the same, allowing for the derivation of an optimal mechanism to be based on the identification of a single optimal (and differentially private) PDF. In our setting, we cannot make the same simplification as each true query response sits at a different position relative to the boundaries of the valid range of noisy query responses. As a result, the mechanism must use a different PDF for each true query response. To reflect this, we propose an adapted optimality criterion.

Definition 7. For any range-restricted query f having a set F of true query responses, let P_1 and P_2 be finite, ordered sets of PDFs of size $|F|$ used by two different mechanisms that have been designed for f. The set P_1 is considered to be **smaller** than P_2, denoted as $P_1 \leq P_2$, if each element P_{1i} is smaller (by Definition 5) than its counterpart P_{2i} as shown in Formula 17.

$$P_1 \leq P_2 \quad \leftrightarrow \quad P_{1_i} \leq P_{2_i} \quad i = 1, \ldots |F| \,. \tag{17}$$

Without the property of data-independence, a mechanism can no longer be characterized by a single PDF. The ordering given by Definition 7 provides a natural extension to that of Definition 5 when dealing with mechanisms that are defined by a set of PDFs. Informally, a set of PDFs P_1 is preferable to another set P_2 if, for every true query response $F_i \in F$, the PDF P_{1_i} can be obtained from P_{2_i} by shifting probability mass closer to the true query response. As such, any rational user should prefer the use of P_1 over P_2 regardless of which PDF the true query response requires from the set. Based on this ordering, we define optimality in an analogous fashion to Definition 6.

Definition 8. For any class $\mathbb{P}$ of finite, ordered sets of PDFs, a set $P \in \mathbb{P}$ is considered to be **optimal** if it is minimal within $\mathbb{P}$.

An additional complication in the design of a data-dependent mechanism is the characterization of the form of the distributions that constitute an optimal set. This was handled in the data-independent setting by starting from a non-optimal PDF and shifting probability mass in a structured manner to produce the smallest possible PDF (under Definition 5) subject to the requirement for it to remain differentially private. This process resulted in the definition of a staircase-shaped PDF which gives rise to an optimal mechanism [38]. In our data-dependent setting, the optimization of a set of PDFs such that they remain differentially private with respect to each other presents a more challenging task. This is due to the fact that the PDFs cannot share a common form due to the query range constraint. To handle this, we leverage the discretized nature of the mechanism matrix representation to employ linear programming. By capturing optimality as a linear programming objective function, we need not explicitly define the shape of the PDFs that would give rise to an optimal discretized mechanism. We must therefore adapt our definition of optimality to a form that can be interpreted by a linear program. To make this change, we first show that the ordering over the elements of any class of finite, ordered sets of PDFs determined by the measure of expected loss using any monotonic loss function subsumes the ordering of Definition 7.

Theorem 3. Within a given class of finite, ordered sets of PDFs, the ordering of Definition 7 is subsumed by the ordering determined by the measure of expected loss over the sets of PDFs when using any non-decreasing loss function $L : \mathbb{R} \to \mathbb{R}$.

Proof. Let y_1, y_2 be two arbitrary PDFs from the same class Y. The inequality $y_1 \leq y_2$ implies that there are sets of ranges X and X' such that for each $x_i \in X$, some probability mass of y_2 can be shifted from this range to another range $x'_i \in X'$ that is nearer to the location parameter in order to produce y_1. For each such shifted range, the decrease in the distance to the location parameter combined with the non-decreasing loss function implies a decrease in the measure of expected loss. This leads to the following inequality:

$$\int_{x\in\mathbb{R}} L(x)y_1(dx) \leq \int_{x\in\mathbb{R}} L(x)y_2(dx). \tag{18}$$

By Definition 7, for any pair of finite, ordered sets of PDFs P_1 and P_2, such that $P_1 \leq P_2$, the above inequality must also hold for every pair of PDFs P_{1_i} and P_{2_i}. As a result, $P_1 \leq P_2$ implies an expected loss for P_1 that is less than or equal to that of P_2. It follows that the ordering determined by any non-decreasing loss function subsumes that of Definition 7. □

Corollary 1. For any class $\mathbb{P}$ of finite, ordered sets of PDFs, a set $P \in \mathbb{P}$ is **optimal** under Definition 8 if it minimizes expected loss for a non-decreasing loss function.

Proof. By Theorem 3, the ordering of Definition 7 is subsumed by the ordering determined by any non-decreasing loss function. As a result, any finite ordered set of PDFs that minimizes expected loss must also be minimal in the ordering of Definition 7, making it optimal under Definition 8. □

With this result, we are now ready to formulate the linear programming objective function. Given that any non-decreasing loss function can be applied to achieve the required ordering, we apply L_1 distance between the true query response and the center of the noisy response category. We accordingly set the objective function as the minimization of the expected L_1 distance. To ensure that the linear program produces a valid differentially private

mechanism, we apply requirements R.1 and R.3 of the matrix representation as constraints for the linear program. The inequalities of Formulae (4), (8) and (9) can be directly carried over without any changes in notation. The formulation of the linear program is as follows:

LP Mechanism Variant 1

$$\min \quad \sum_{i=1}^{|R|}\sum_{j=1}^{|F|} M_{i,j}\left|\frac{r_i^u + r_i^l}{2} - F_j\right|$$

$$\begin{aligned}
\textbf{s.t.} \quad & M_{i,j} \leq e^{\epsilon} M_{i,k} && \forall D_1, D_2 \in \mathbb{D} \text{ s.t. } f(D_1) = F_j,\, f(D_2) = F_k,\, |D_1 - D_2| = 1,\\
& M_{i,j} \geq 0 && i = 1, ..., |R|\,,\ j = 1, ..., |F|\,,\\
& \sum_{i=1}^{|R|} M_{i,j} = 1 && j = 1, ..., |F|\,.
\end{aligned}$$

We emphasize that this optimization-based approach differs fundamentally from those applied in the non-interactive setting. The linear program we formulate optimizes the probability distributions to be used by a differentially private mechanism. Contrary to this, optimization in the non-interactive setting typically involves post-processing applied to noisy responses which have already been generated by a randomization mechanism.

In some cases, users may wish to enforce additional properties in the mechanism in order to prevent certain pathological behaviours [7]. To address these concerns, we define an alternative linear program that is identical to the first but employs additional constraints to provide the properties of row and column monotonicity, symmetry and fairness. Once again, we can directly apply the inequalities and equations as specified in Formulae (10) - (15). To distinguish between these two linear program formulations, we refer to the formulation without the additional constraints as Variant 1 and the formulation with the additional constraints as Variant 2. The constraints used to extend the formulation are as follows:

LP Mechanism Variant 2

Extend Variant 1 with:

$$\begin{aligned}
M_{i,j} &\leq M_{i+1,j} && i = 1, ..., i_j - 1,\ j = 1, ..., |F|\,,\\
M_{i,j} &\geq M_{i+1,j} && i = i_j, ... |R| - 1,\ j = 1, ..., |F|\,,\\
M_{i,j} &\leq M_{i,j+1} && i = 1, ..., |R|\,,\ j = 1, ..., |F| - 1 \text{ s.t. } F_{j+1} \leq r_i^u,\\
M_{i,j} &\geq M_{i,j+1} && i = 1, ..., |R|\,,\ j = 1, ..., |F| - 1 \text{ s.t. } F_j \geq r_i^l,\\
M_{i,j} &= M_{|R|-i+1,|F|-j+1} && i = 1, ... |R|\,,\ j = 1, ... |F|\,,\\
M_{i_j,j} &= M_{i_{j'},j'} && j = 1, ... |F|\,,\ j' = 1, ... |F|\,.
\end{aligned}$$

We provide a small example to illustrate the resultant mechanism matrix for a counting query with a range of 0 - 5. For counting queries, it is natural to use the same set of integers from the possible true query responses as the set of noisy responses (i.e., $R = F$). The matrix for Variant 1 is shown in Table 3. Each column represents the probability distribution used by the mechanism given a particular true query response while each row corresponds to one of the potential noisy responses. A cell entry in row i, column j thus corresponds to the conditional probability $M_{i,j}$ of the mechanism. We note that Variant 1

has assigned a probability of zero to the noisy response categories on the boundaries of the valid range. Due to this, the mechanism will never report these responses, even in the event that the true query response falls in these boundary categories. This may seem at odds with the goal concentrating probability mass around true query responses. However, the optimization goal is to achieve the best concentration of probability mass when considering all probability distributions used by the mechanism. The mechanism has assumed this configuration in order to achieve a greater overall degree of expected utility across all true query responses. Furthermore, while the probabilities of $M_{0,0}$ and $M_{5,5}$ are 0 in this example, the probabilities of $M_{1,0}$ and $M_{4,5}$ are 0.771, meaning there remains a very high probability of reporting a noisy response that is close to a true query response in a boundary category.

	$f(D)=0$	$f(D)=1$	$f(D)=2$	$f(D)=3$	$f(D)=4$	$f(D)=5$
$K(f(D))=0$	0	0	0	0	0	0
$K(f(D))=1$	0.771	0.622	0.378	0.229	0.139	0.084
$K(f(D))=2$	0.090	0.149	0.245	0.149	0.091	0.055
$K(f(D))=3$	0.055	0.090	0.149	0.245	0.149	0.090
$K(f(D))=4$	0.084	0.139	0.229	0.378	0.622	0.771
$K(f(D))=5$	0	0	0	0	0	0

Table 3: Mechanism matrix for the linear programming Variant 1 solution of a counting query with a range of 0 - 5 and $\epsilon = 0.5$. Columns correspond to true query responses while rows correspond to noisy responses.

If properties such as 0-probability events are deemed undesirable, use of Variant 2 can provide a mechanism with a more constrained structure. This is shown for the same example in Table 4. Here, the additional constraints have forced the entries with a probability of zero in Variant 1 to now take on non-zero values. The additional properties of this variant may be desirable for some data analysis tasks but they come at the cost of less flexibility in the optimization of the objective function.

	$f(D)=0$	$f(D)=1$	$f(D)=2$	$f(D)=3$	$f(D)=4$	$f(D)=5$
$K(f(D))=0$	0.315	0.191	0.116	0.070	0.043	0.026
$K(f(D))=1$	0.315	0.315	0.191	0.116	0.070	0.043
$K(f(D))=2$	0.231	0.265	0.315	0.191	0.116	0.070
$K(f(D))=3$	0.070	0.116	0.191	0.315	0.265	0.231
$K(f(D))=4$	0.043	0.070	0.116	0.191	0.315	0.315
$K(f(D))=5$	0.026	0.043	0.070	0.116	0.191	0.315

Table 4: Mechanism matrix for the linear programming Variant 2 solution of a counting query with a range of 0 - 5 and $\epsilon = 0.5$. Columns correspond to true query responses while rows correspond to noisy responses.

4 Experimental Comparisons

In this section, we describe our experiments and discuss comparisons between our proposed mechanisms and our implementations of three other range-adherent mechanisms.

We first describe the different mechanisms and utility measures that we apply. This is followed by a detailed discussion on our experiments in which we give our analysis. We use both synthetic data as well as real data to gain insight into expected behaviour and practical performance.

4.1 Mechanisms

We compare our linear programming approach to boundary-snapping variants of two well-known mechanisms: the Laplace mechanism [12] (Mechanism 1) and the staircase mechanism [23] (Mechanism 2). Additionally, we include a range-adherent normalized Laplace mechanism [9] (Mechanism 3) in our comparisons. In the mechanism definitions, ΔF is used to denote the *query sensitivity*. This is defined as the maximum possible difference between the true query responses of any pair of adjacent databases.

Mechanism 1. Laplace [12] - For a database D, noise is drawn from a Laplace distribution using $f(D)$ as the location parameter and $\frac{\Delta F}{\epsilon}$ as the scaling parameter:

$$Lap\left(x|f(D), \frac{\Delta F}{\epsilon}\right) = \frac{\epsilon e^{-\frac{\epsilon|f(D)-x|}{\Delta F}}}{2\Delta F}. \tag{19}$$

Mechanism 2. Staircase [23] - For a database D, a mechanism drawing from a staircase-shaped PDF centered at $f(D)$ minimizes the expected distortion of the noisy query response:

$$Stair\left(x|f(D)\right) = \begin{cases} y & |f(D)-x| \in \left[0, \frac{\Delta F}{1+e^{\frac{\epsilon}{2}}}\right] \\ ye^{-\epsilon} & |f(D)-x| \in \left[\frac{\Delta F}{1+e^{\frac{\epsilon}{2}}}, \Delta F\right] \\ e^{-k\epsilon} Stair\left(x-k\Delta F\right) & |f(D)-x| \in [k\Delta F, (k+1)\Delta F] \text{ for } k \in \mathbb{N}, \end{cases} \tag{20}$$

$$y = \frac{1-e^{-\epsilon}}{2\Delta F\left(\frac{1}{1+e^{\frac{\epsilon}{2}}} + e^{-\epsilon}\left(1-\frac{1}{1+e^{\frac{\epsilon}{2}}}\right)\right)}. \tag{21}$$

The normalized Laplace mechanism (Mechanism 3) uses a PDF truncated to the valid range of the query. Due to the data-dependent operation of normalization, this mechanism requires a higher scaling parameter than the standard Laplace mechanism.

Mechanism 3. Normalized Laplace [9] - For a database D, noise is drawn from a Laplace distribution truncated to the range of $[r_l, r_u]$ and normalized within this range, using $f(D)$ as the location parameter and $\frac{2\Delta F}{\epsilon}$ as the scaling parameter:

$$NormLap\left(x|f(D), \frac{2\Delta F}{\epsilon}\right) = \begin{cases} \frac{2Lap\left(x|f(D), \frac{2\Delta F}{\epsilon}\right)}{2-e^{-\frac{\epsilon\left(f(D)-r_l\right)}{2\Delta F}} - e^{\frac{\epsilon(r_u-f(D))}{2\Delta F}}} & r_l \le x \le r_u \\ 0 & x < r_l, x > r_u. \end{cases} \tag{22}$$

4.2 Utility Measures

To compare the utility of the mechanisms, we employ measures of both usability and information preservation. Recall that usability indicates ease of use for specific tasks while

information preservation indicates how much of the original information is maintained in the noisy query responses. We therefore consider usability in terms of taking noisy query responses at face value (i.e., without applying any post-processing). See e.g., [23, 26] for instances of utility measured in this way. Specifically, we measure both the expected absolute error and the expected squared error of noisy query responses with respect to the true query response. These expected measures are calculated over all true query responses and over all possible noisy query responses for each true response. We assume a uniform distribution for the probabilities of the true responses. The probabilities of the noisy query responses are determined by the conditional probability of the mechanism given a particular true query response. The expected absolute error acts as a measure of the amount of distortion expected to be induced by the mechanism, giving preference to mechanisms that have low distortion in the expected case. The expected squared error assigns a penalty that grows rapidly as the amount of distortion increases. This gives preference to mechanisms that avoid allowing for high distortion in some cases in order to achieve low distortion in other cases. We have selected these measures as they provide a good representation of typical properties that are desirable in terms of the usability of a mechanism. The two measures are defined as follows:

$$Ex\_Err = \sum_{f \in F} \frac{\sum\limits_{r \in R} \Pr(r|f)\,|f - r|}{|F|}, \tag{23}$$

$$Ex\_Sqr\_Err = \sum_{f \in F} \frac{\sum\limits_{r \in R} \Pr(r|f)\,(f - r)^2}{|F|}. \tag{24}$$

We additionally measure utility for Bayesian users [24] having a uniform distribution over the true query responses as their prior knowledge. This models a common scenario in which the user initially has no additional information about distribution of the true query responses. Given a noisy response $r \in R$, a Bayesian user will remap this value to the response $f \in F$ that minimizes the expected loss according to their prior knowledge. The measure of expected loss under Bayesian post-processing is appropriate as a measure of information preservation [31] and is therefore useful as contrast against the measures of usability. For close correspondence between these measures, we employ instances of Bayesian post-processing for minimization of expected absolute error and expected squared error. We define these measures of loss as shown in Formulae (25) and (26).

$$Ex\_Baye\_Err = \sum_{r \in R} \Pr(r) \min_{f \in F} \sum_{f' \in F} \frac{\Pr(r|f')\,|f - f'|}{|F|}, \tag{25}$$

$$Ex\_Baye\_Sqr\_Err = \sum_{r \in R} \Pr(r) \min_{f \in F} \sum_{f' \in F} \frac{\Pr(r|f')\,(f - f')^2}{|F|}. \tag{26}$$

4.3 Experiments

To solve the linear programs needed for our mechanisms, we used WinGLPK [37], a Windows executable version of the GNU Linear Programming Kit [36]. In all experiments, we assume that the database sizes and attribute response categories are public knowledge. We compare our mechanisms against the alternatives using mean and max queries. We pose

the mean query on an integer-valued scalar attribute that can take on 5 possible values (0 through 4). We pose the max query on an integer-valued scalar attribute that can take on 10 possible values (0 through 9). Although we refer to the later as a max query for simplicity, it can equivalently be interpreted as a min or median query, which share the same query sensitivity and set of true query responses.

With respect to the discretization of the noisy query responses, it is necessary to specify a set of noisy response categories that covers the full valid range of the query (e.g., no gaps between noisy response categories) in order to obtain valid probability distributions from mechanisms that use continuous PDFs. When querying a mean value on a database D with an attribute taking on integer values from 0 to k, there are $k\,|D| + 1$ possible true query responses. Each of these responses are evenly spaced at a distance of $\frac{1}{k|D|}$ apart over the valid range of $[0, k]$. To handle the discretization, we set the size of each noisy response category to $\frac{1}{k|D|}$ and position them such that each noisy response category is centered on a true query response. When querying the max value on an attribute with integer values from 0 to k, there are $k + 1$ possible true query responses. Each of these responses are evenly spaced at a distance of 1 apart over the range $[0, k]$. In this case, we set the size of the response categories to 1 and again center them on the true query responses. For all distance measures, we consider the value of a noisy response category to be the midpoint of the span it covers.

In all experiments, we have measured expected absolute error and expected squared error both with and without Bayesian post-processing. Since the relationships between the plotted curves for expected squared error showed no significant differences from those of the expected absolute error, we only show the later for space considerations. In all experiments, lower values of loss indicate better utility.

4.3.1 Database Size

We begin our experimental analysis by plotting expected loss as a function of the database size for fixed values of the privacy parameter ϵ. For a max query, database size impacts neither the true query responses nor the query sensitivity. As a result, it has no influence over the measures of utility we employ. We therefore examine the impact of database size only for the mean query. Results are shown in Figure 1 using an ϵ value of 0.2 and in Figure 2 using an ϵ value of 0.5.

Through these experiments, we demonstrate that the most notable improvements in utility achieved by our mechanism occur at smaller database sizes. This is in line with the results on the explicit fair mechanism [7], which found that, in the context of counting queries, improvements in utility obtained through alternate methods of adherence to range constraints are most prominent for queries posed on small databases.

In many queries, such as the mean query studied here, the size of the database plays a major role in determining the amount of probability mass (for non-range-adherent mechanisms) that falls outside of the valid range of the query. For example, reduction of database size in a mean query increases the query sensitivity while leaving the valid range of the query unchanged. An increase in the ratio of query sensitivity to query range implies a greater amount of probability mass will fall outside the valid range in order to satisfy the differential privacy guarantee. Reduction of the privacy parameter ϵ has a similar effect. This can be seen in the more pronounced improvements in utility for linear programming in Figure 1 compared to Figure 2.

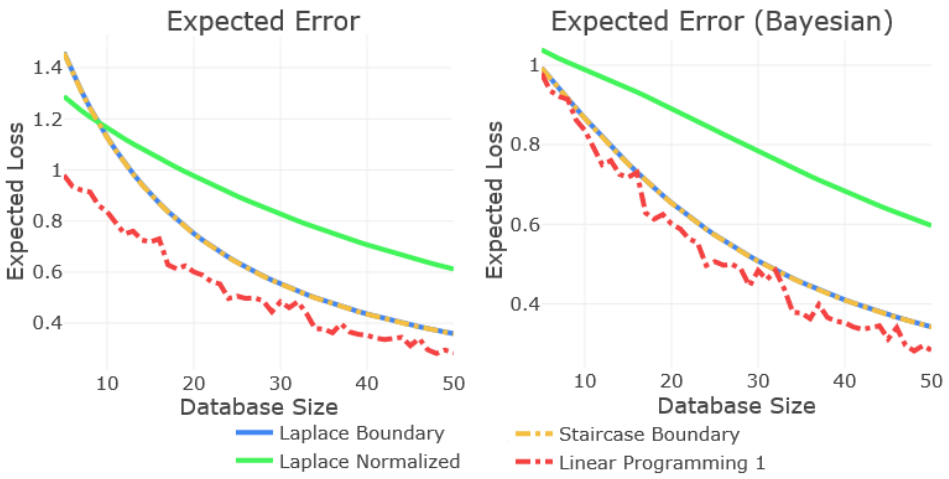

Figure 1: Expected loss as a function of the database size for a mean query with $\epsilon = 0.2$

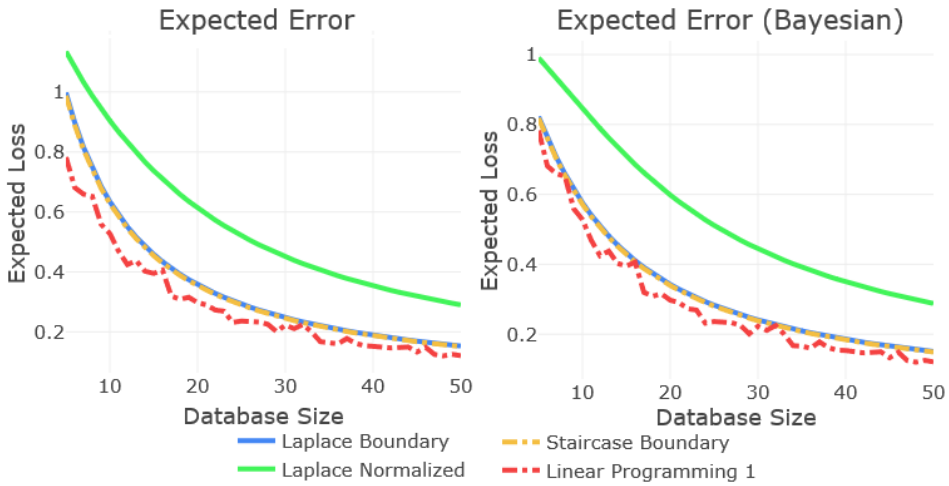

Figure 2: Expected loss as a function of the database size for a mean query with $\epsilon = 0.5$

Given these observations, we focus our other experiments on small databases (10 records) for the mean query in order to better observe the differences in the behaviours of the mechanisms. In practice, small datasets may arise in a number of instances. For example, studies on rare diseases, marginal groups of populations or small groups of participants can lead to small datasets. If studies involve expensive trials, cost may also be a limiting factor leading to small datasets. Privacy is particularly important when sample sizes are small. Yet, high ratios of query sensitivity to query range create a difficult scenario within which to achieve a good privacy/utility trade-off. This further highlights the importance of developing new mechanisms to improve utility in this setting.

4.3.2 Discretization

As we have chosen a specific method for determining the noisy response category size in the mechanism discretizations, we include experiments using different sizes for the noisy response categories to observe the effects that this may have on the utility of the mechanisms. Results are shown in Figure 3 for a mean query. Each plotted line shows a different size of discretized response categories. The numbers associated with each plot in the legends indicate a scaling factor for the noisy response category size. Here, lower numbers indicate finer granularity and higher numbers indicate coarser granularity. Surprisingly, the boundary-snapping Laplace mechanism achieves better levels of utility with a coarser granularity in high privacy settings. This is due to the fact that the high level of privacy leads to the probability mass being very spread out from the location parameter. Boundary-snapping then causes large amounts of out-of-bounds probability mass to pool at the boundaries of the valid range. The finer the granularity of the discretization is, the more concentrated the pooled probability mass becomes near the extremities of the valid range. This leads to greater expected distance between noisy query responses and true query responses. This suggests that boundary-snapping is a poor choice for queries posed on small databases under high privacy settings. To further illustrate this point, we include an additional mechanism that simply uses a uniform distribution over the noisy query responses in these experiments. In the high privacy setting, even this trivial mechanism performs better than the boundary-snapping variant.

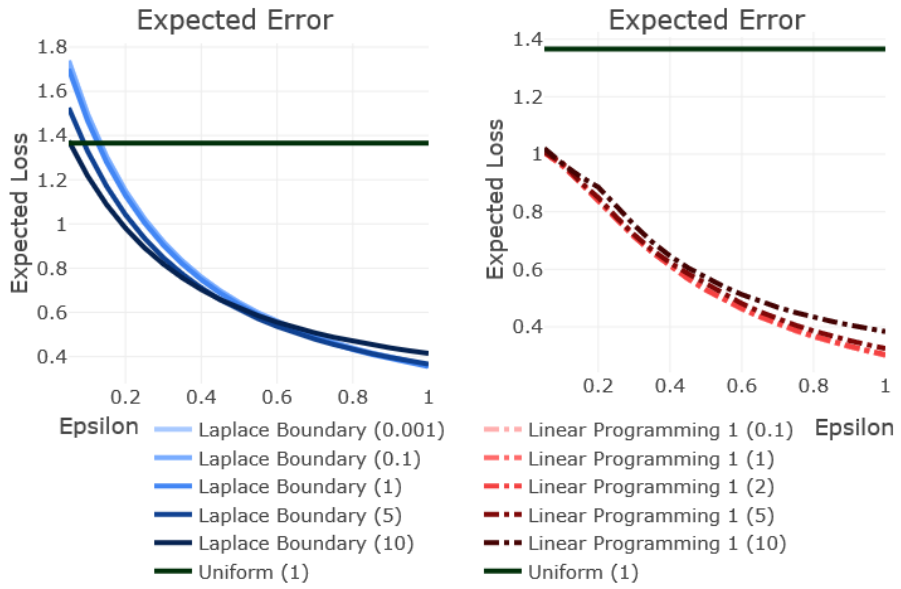

Figure 3: Expected loss as a function of ϵ for a mean query. Each plot uses noisy responses category sizes equal to the minimum distance between two different true responses scaled by the postfixed number in the legend (e.g., a postfix of 5 indicates categories 5 times larger).

For the discretization experiments on the linear programming mechanism, we observe the expected trend of coarser granularity leading to worse levels of utility. We note that for the three finest levels of granularity tested, the differences in the levels of utility are

almost negligible. Intuitively, the loss in query accuracy induced by the injected noise negates much of the benefit that could otherwise be achieved through the use of higher precision. This supports our decision to use noisy response categories having a size equal to the distance between true query responses in all other experiments. We omit graphs for the other measures as they display similar trends.

4.3.3 Expected Utility

With a fixed size for the database and the noisy query response categories, we are able to plot the levels of utility as a function of the privacy parameter ϵ to compare different types of mechanisms. This allows for a more comprehensive understanding of the behaviours of the mechanisms at varying levels of privacy. Our results are shown in Figure 4 for the mean query and in Figure 5 for the max query.

The linear programming Variant 1 consistently provides the best levels of utility, with the greatest improvements over other mechanisms appearing in the non-post-processing measures at high levels of privacy. For the linear programming Variant 2, although it does not reach the same levels of utility as Variant 1, it still outperforms the other mechanisms in the non-post-processing measures. In the measures of post-processing, Variant 2 performs on par with Variant 1 for the max query and on par with the boundary-snapping mechanisms for the mean query.

For min, max and median queries, the query sensitivity is equal the size of the range of true query responses. The high ratio of sensitivity to query range makes such queries ideal candidates for the improvements in utility attainable through the use of our linear programming mechanisms.

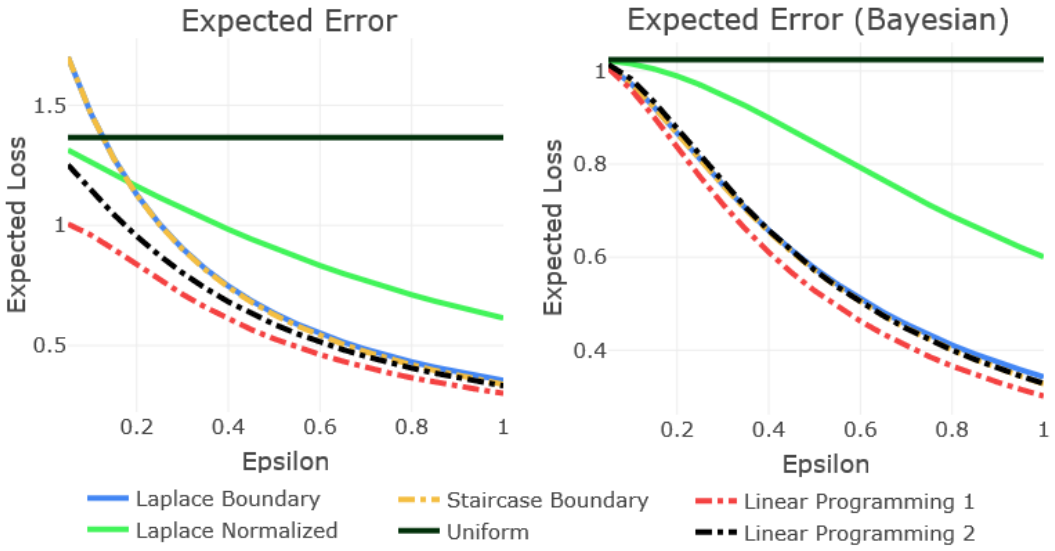

Figure 4: Expected loss as a function of ϵ for a mean query

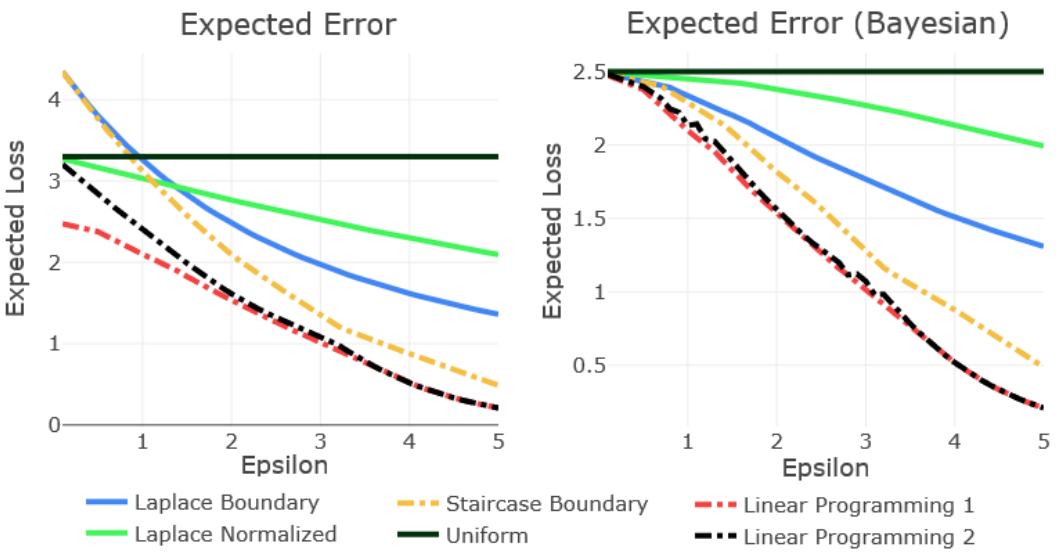

Figure 5: Expected loss as a function of ϵ for a max query

4.3.4 Extended Range

As we have hypothesized that the incorporation of range constraints into the design of a mechanism will have a positive impact on the level of utility, we provide a comparison between range-adherent mechanisms and non-adherent variants. Since any computer implementation of a mechanism must have a finite range, we apply a greatly widened range to model a lack of adherence to range constraints. To do so, we widen the range of noisy query responses to 300% of the constrained range, keeping this span centered on the valid range. Since probability mass is always focused near the true query response, any further widening of the range beyond 300% has little impact and does not provide additional insight. Results are shown in Figures 6 and 7. For visual clarity, we omit the boundary-snapping variant of the Laplace mechanism and Variant 2 of the linear programming mechanism from the graphs. The behaviour of the extended range versions of these omitted mechanisms matched those of the boundary-snapping staircase mechanism and the linear programming Variant 1, respectively.

The most notable difference in the range-extended variants is that the non-linear programming mechanisms show substantially worse levels of utility in the non-post-processing measures at high levels of privacy. In the measures of post-processing, the range-extended variants of the non-linear programming mechanisms perform on par with or marginally better than their range-adherent counterparts. Given that the application of boundary-snapping to a mechanism cannot improve its information preservation [31], it is not surprising to see a lack of improvement in the measures of post-processing. However, the large difference in the non-post-processing measures shows that adherence to range constraints clearly has a significant impact on utility.

The linear programming mechanisms show no differences in any measures of utility between their range-adherent and extended range variants. This is due to the fact that even when presented with response categories for an extended range, the optimization process will not assign probability mass to such responses.

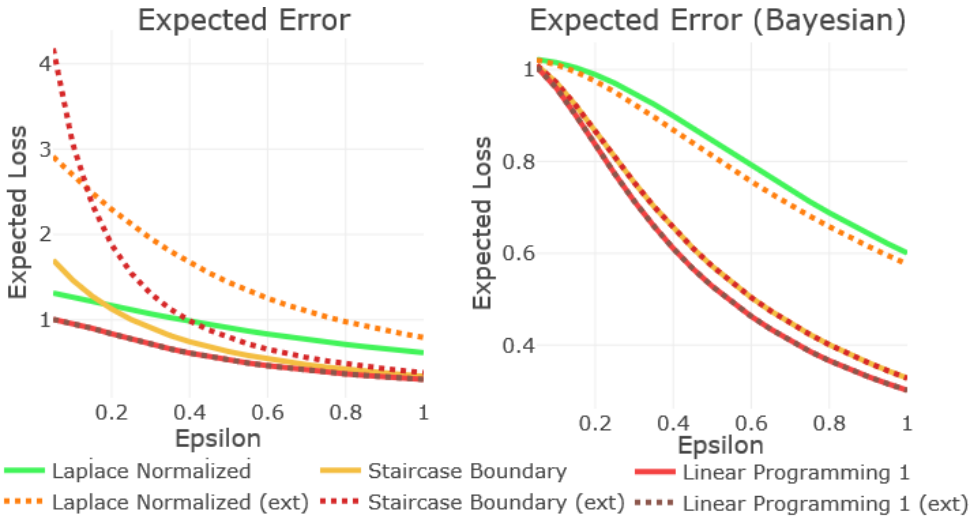

Figure 6: Expected loss as a function of ϵ for a mean query. Mechanism variants using an extended range are marked by the postfix *(ext)*.

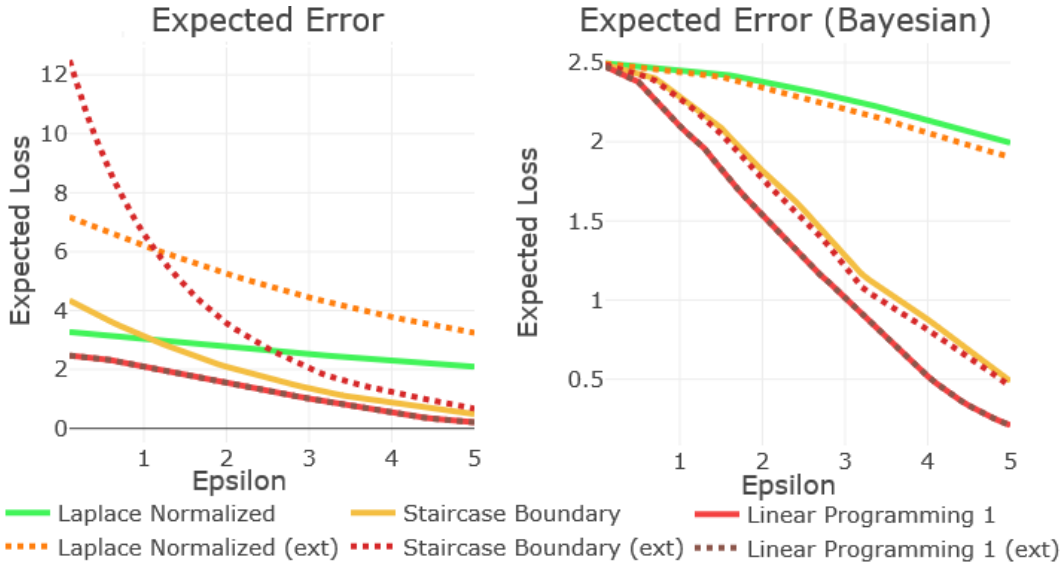

Figure 7: Expected loss as a function of ϵ for a max query. Mechanism variants using an extended range are marked by the postfix *(ext)*.

4.3.5 Real Data

Finally, to gain insight on the practical performance of the mechanisms, we have run experiments on real data. We have used the Statistics Canada 2011 National Household Survey public use microdata file [5] to query the family size of the survey respondents. The valid

range of responses are integers from 1 (no family) up to 7 (interpreted as 7 or more family members). The mean value over all groups for this query is 2.02. As required by Statistics Canada's data use regulations, we state that the results or views expressed here are not those of Statistics Canada. We have partitioned the records into groups of size 10, using only records from the province of Ontario. In total there are 341,253 such records. This experimental setup can be interpreted as a simulation of small-scale or finely-grained surveys conducted at the level of neighborhoods. Each group of records corresponds to a dataset collected by sampling a small number of households within a particular neighborhood.

To measure the utility of a query posed on a particular group, we employ each mechanism variant to generate a noisy response. For the measures of usability, the response is taken at face value, whereas for Bayesian users, the response is mapped to their best guess. In both cases, the resultant value is used to measure the loss (as absolute error and squared error) with respect to the true query response. Since the mechanisms are stochastic functions, this process is repeated 1000 times for each mechanism variant to obtain mean measures of utility. This is done over each of the groups and once again, mean values are computed across the groups.

We have run this experiment for both mean and max queries. Given the high sensitivity of the max query, we include the well-known concept of a smooth-sensitivity [35] mechanism for comparison in our experiments. Smooth sensitivity acts as an upper bound on local sensitivity (the maximum difference between the query response of a database and any of its neighbors). The main insight is that as a smooth function, smooth sensitivity is insensitive to the contents of the database, making it appropriate for use in the configuration of differentially private mechanisms. This allows for a reduction in the required sensitivity from the global level which is particularly high in queries such as max, min and median. To implement a smooth sensitivity mechanism for pure differential privacy (i.e., without any relaxations in the privacy guarantee), a Cauchy distribution can be used with a scaling parameter of $\frac{8S(D)}{\epsilon}$, where $S(D)$ is the smooth sensitivity of a database D. We then apply boundary-snapping to produce a range-adherent mechanism.

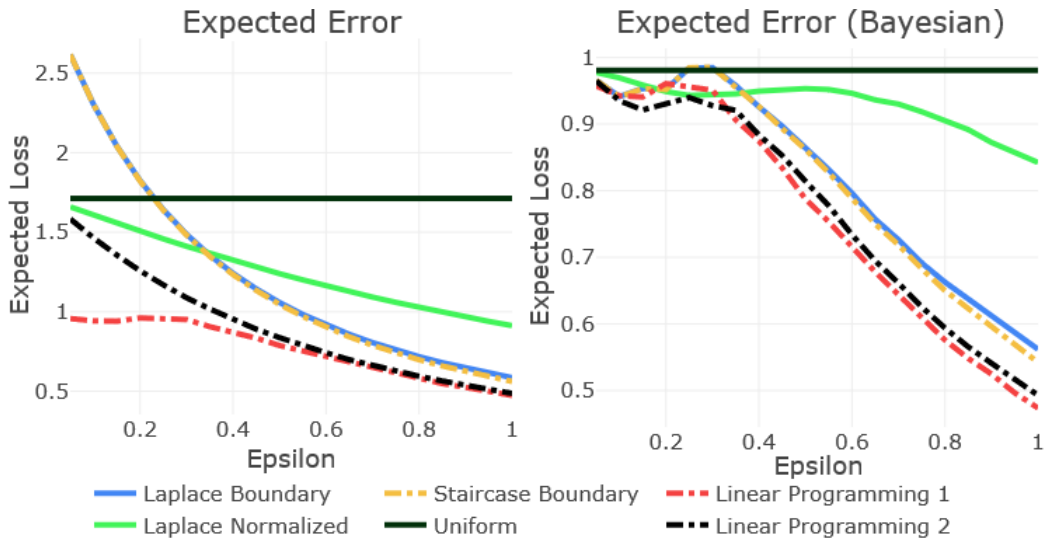

Figure 8: Expected loss as a function of ϵ for mean family size

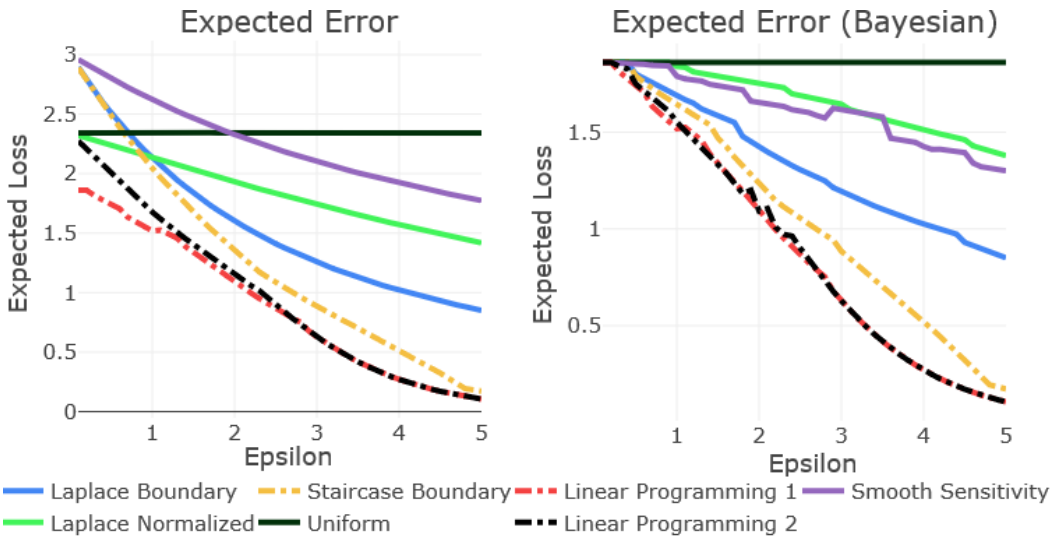

Figure 9: Expected loss as a function of ϵ for max family size

Note that, since we use real query responses from sets of records, the max query cannot be equivalently interpreted as a min or median query here as it could in the previous experiments. Results are shown in Figures 8 and 9.

The results for both queries mainly follow the same trends as in the experiments of expected utility on synthetic data. The linear programming Variant 1 performs best with Variant 2 next, most prominently at high levels of privacy. The only exception to this occurs with the measures of post-processing for the mean query where Variant 2 performs best at high levels of privacy. Since most of the data groups are expected to have similar true query responses, we hypothesize that the mechanism is simply able to perform well on the specific query responses that were commonly encountered despite having an overall lower level of expected utility than Variant 1. The mechanism using smooth sensitivity performs poorly compared to the other mechanisms. This is largely due to the use of a max query on small-sized databases which often results in relatively high local sensitivities. As a result, the mechanism cannot fully take advantage of the smooth sensitivity and suffers in utility due to the use of a less ideal underlying PDF compared to the other mechanisms. Analogous results can be expected for min queries.

4.4 Discussion

We conclude this section with a discussion on the practical implications of the results from our analysis. We assess the impact of adherence to range constraints on the utility of the mechanisms and discuss the performance of our linear programming mechanisms.

Our results indicate that adherence to query range constraints can indeed be beneficial for the utility of mechanisms, provided adherence is appropriately handled. We have observed that the improvements in utility are most pronounced for cases where the ratio of query sensitivity to query range is high. This often occurs for particular types of queries (e.g., min, max, median) or for general queries posed on small databases. For cases where queries are posed on small groups, high levels of privacy are arguably the most important setting as these individuals are likely to be more susceptible to unwanted inferences than those

who are hidden among larger group sizes. From the results of Section 4.3.4, we see that in this setting, adherence to range constraints greatly improves the measures of non-post-processing, indicating a significant improvement in usability. The correspondence in the measures of post-processing between the range-adherent mechanisms and the extended-range variants indicates that the mechanisms have not suffered in information preservation through the modifications required for adherence to range constraints. This suggests that the explicit adherence to range constraints offers a favourable improvement in utility with respect to a trade-off between usability and information preservation.

Overall, the linear programming Variant 1 performs the best both in terms of expected error and expected squared error. The mechanism achieved superior levels of utility in the non-post-processing measures and matched or slightly exceeded the levels of utility achieved by boundary-snapping variants in the measures of post-processing. While not matching the levels of utility achieved by Variant 1, the linear programming Variant 2 does match or exceed the levels of utility of all other mechanisms. These results are also visible in the experiments on real data with the exception some minor differences in the post-processing measures of the mean query at high levels of privacy. Given that we are interested both in usability as well as information preservation, we do not consider this discrepancy to outweigh the overall benefits of the linear programming Variant 1. For queries posed on small databases at mid-to-high levels of privacy, we recommend the linear programming Variant 1 for the best levels of utility. For users interested in mechanisms that offer the additional properties of row and column monotonicity, symmetry and fairness, the linear programming Variant 2 is a good alternative.

The number of variables in the linear programs is the product of the number of true query responses and the number of noisy response categories. Due to this, the computational expense of the linear program can grow rapidly as the size of the database increases, making it impractical to derive linear programming mechanisms for large cases. Despite this apparent shortcoming, we expect that this poses little practical significance. Our results suggest that for cases where the true query responses are impacted by database size, the utility levels of the different mechanisms begins to converge for large numbers of records. The major improvements in utility occurred either for small-sized databases or for queries invariant to database size. In such cases, the input sizes can be reasonably handled by a linear programming solver. We therefore recommend the use of the linear programming mechanisms for queries either unaffected by the size of the database or otherwise posed on small-sized databases. The exact sizes depend also on the chosen discretization and the computational power of the target machine. For other cases, we recommend the use of an alternative such as boundary-snapping applied to an appropriate mechanism. As an extension to our work, it may be of interest to study the applicability of approximation schemes to handle larger-sized databases.

Finally, we note that although we expressly chose an optimality criterion that abstracts from user-specific goals as the basis for the linear programming objective function, the selection of a specific loss function necessarily imposes some degree of user preference according to the selected measure of loss. The subsumed ordering of Definition 7 remains the same as long as a non-decreasing loss function is chosen. However, there are many pairs of finite, ordered sets of PDFs that are incomparable within this order but are comparable once a loss function is imposed. This implies that some user-specific preferences leak into the final objective function. As such, it may be the case that other loss functions or different objective function formulations may offer better results for general users.

5 Conclusions

Adherence to publicly known constraints on queries can be an effective method to improve utility in differentially private mechanisms. However, care must be taken to consider utility both in terms of information preservation as well as usability. In this work, we have studied the design of mechanisms that generate noisy query responses within the valid range of responses for the query. We have combined a formalized matrix representation of range-adherent mechanisms with a user-independent criterion for optimal utility in order to provide a basis for the derivation of a range-adherent linear programming mechanism. We have proposed two range-adherent linear programming variants. The first is subject only to the differential privacy requirement while the second is also subject to constraints providing row and column monotonicity, symmetry and fairness in the mechanism matrix. Through experimental comparisons, we have shown that the linear programming mechanisms are able to provide improvements in utility over boundary-snapping, a common choice when adherence to range constraints is required. We have observed that the most significant gains in utility occur for small-sized databases, which are more vulnerable to privacy breaches. We therefore recommend the use of boundary-snapping for cases of large input where the computational cost of a linear program becomes prohibitive. For smaller databases, particularly in mid-to-high levels of privacy, we recommend the use of the linear programming mechanisms for substantial improvements in utility.

Acknowledgments

We gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grants No. RGPIN-2020-06482, No. RGPIN-2016-06253 and No. CGSD2-503941-2017.

References

[1] J. Awan and A. Slavkovic. Differentially Private Uniformly Most Powerful Tests for Binomial Data. `https://arxiv.org/abs/1805.09236`, 2018.

[2] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release. In *Proceedings of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 273–282, 2007.

[3] A. Basu, T. Nakamura, S. Hidano, and S. Kiyomoto. k-Anonymity: Risks and the Reality. In *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 983–989, 2015.

[4] H. Brenner and K. Nissim. Impossibility of Differentially Private Universally Optimal Mechanisms. *SIAM Journal on Computing*, 43(5):1513–1540, 2014.

[5] Statistics Canada. Individuals File, 2011 National Household Survey (Public Use Microdata Files), 2014. `http://www5.statcan.gc.ca/olc-cel/olc.action?objId=99M0001X2011001&objType=46&lang=en&limit=0` Accessed: Mar, 2015.

[6] C.-L. Chen, R. Pal, and L. Golubchik. Oblivious Mechanisms in Differential Privacy: Experiments, Conjectures, and Open Questions. In *Proceedings of IEEE Security and Privacy Workshops*, pages 41–48, 2016.

[7] G. Cormode, T. Kulkarni, and D. Srivastava. Constrained Private Mechanisms for Count Data. In *Proceedings of the 34th IEEE International Conference on Data Engineering*, pages 845–856, 2018.

[8] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu. Empirical Privacy and Empirical Utility of Anonymized Data. In *Proceedings of the IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 77–82, 2013.

[9] W. Croft, J.-R. Sack, and W. Shi. Differential Privacy Via a Truncated and Normalized Laplace Mechanism. https://arxiv.org/abs/1911.00602, 2019.

[10] I. Dinur and K. Nissim. Revealing Information while Preserving Privacy. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, volume 22, pages 202–210, 2003.

[11] J. Domingo-Ferrer and V. Torra. Disclosure Control Methods and Information Loss for Microdata. *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 91–110, 2001.

[12] C. Dwork. Differential Privacy. In *Proceedings of 33rd International Colloquium on Automata, Languages, and Programming*, volume 4052, pages 1–12, 2006.

[13] C. Dwork. Differential Privacy: A Survey of Results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, volume 4978, pages 1–19, 2008.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, volume 3876, pages 265–284, 2006.

[15] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2014.

[16] E. ElSalamouny, K. Chatzikokolakis, and C. Palamidessi. Generalized Differential Privacy: Regions of Priors that Admit Robust Optimal Mechanisms. In *Proceedings of PrakashFest Conference*, volume 8464, pages 292–318, 2014.

[17] S. E. Fienberg, A. Rinaldo, and X. Yang. Differential Privacy and the Risk-Utility Tradeoff for Multi-Dimensional Contingency Tables. In *Proceedings of Privacy in Statistical Databases*, pages 187–199, 2010.

[18] F. Fioretto, C. Lee, and P. Van Hentenryck. Constrained-Based Differential Privacy for Mobility Services. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1405–1413, 2018.

[19] F. Fioretto and P. Van Hentenryck. Constrained-Based Differential Privacy: Releasing Optimal Power Flow Benchmarks Privately. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 215–231, 2018.

[20] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition Attacks and Auxiliary Information in Data Privacy. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 265–273, 2008.

[21] Q. Geng, W. Ding, R. Guo, and S. Kumar. Truncated Laplacian Mechanism for Approximate Differential Privacy. *CoRR*, 2018. http://arxiv.org/abs/1810.00877.

[22] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath. The Staircase Mechanism in Differential Privacy. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1176–1184, 2015.

[23] Q. Geng and P. Viswanath. The Optimal Mechanism in Differential Privacy. In *Proceedings of IEEE International Symposium on Information Theory*, pages 2371–2375, 2014.

[24] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally Utility-Maximizing Privacy Mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

[25] M. Gupte and M. Sundararajan. Universally Optimal Privacy Mechanisms for Minimax Agents. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 135–145, 2010.

[26] S. Haney, M. Graham, A. Machanavajjhala, M. Kutzbach, J.M. Abowd, and L. Vilhuber. Utility Cost of Formal Privacy for Releasing National Employer-Employee Statistics. In *Proceedings of*

ACM Special Interest Group on Management of Data, pages 1339–1354, 2017.

[27] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the Accuracy of Differentially Private Histograms Through Consistency. In *Proceedings of the Very Large Data Base Endowment*, pages 1021–1032, 2010.

[28] J. Lee, Y. Wang, and D. Kifer. Maximum Likelihood Postprocessing for Differential Privacy Under Consistency Constraints. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pages 635–644, 2015.

[29] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi. The Matrix Mechanism: Optimizing Linear Counting Queries Under Differential Privacy. *The VLDB Journal*, 24(6):757–781, 2015.

[30] T. Li and N. Li. On the Tradeoff Between Privacy and Utility in Data Publishing. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 517–526, 2009.

[31] B.-R. Lin and D. Kifer. Information Measures in Statistical Privacy and Data Processing Applications. *ACM Transactions on Knowledge Discovery from Data*, 9(4), 2015.

[32] F. Liu. Statistical Properties of Sanitized Results from Differentially Private Laplace Mechanism with Bounding Constraints. `https://arxiv.org/abs/1607.08554`, 2018.

[33] G. Loomes and R. Sugden. Regret Theory: An Alternative Theory of Rational Choice Under Uncertainty. *The Economic Journal*, 92(368):805–824, 1982.

[34] H. H. Nguyen, J. Kim, and Y. Kim. Differential Privacy in Practice. *Journal of Computing Science and Engineering*, 7:177–186, 2013.

[35] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth Sensitivity and Sampling in Private Data Analysis. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, pages 75–84, 2007.

[36] GNU Project. GNU Linear Programming Kit. Ver: 4.65 (2018-02-16), `https://www.gnu.org/software/glpk/`.

[37] H. Schuchardt. GLPK for Windows. Ver: 4.65 (2018-03-17), `http://winglpk.sourceforge.net/`.

[38] J. Soria-Comas and J. Domingo-Ferrer. Optimal Data-Independent Noise for Differential Privacy. *Information Sciences*, 250:200 – 214, 2013.