universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Maximizing Fisher Information in ϵ-Differential Privacy Mechanisms

verfasst von | submitted by

Matej Vedak

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von | Supervisor:	Assoz. Prof. Mag. Lukas Steinberger Bakk. BA PhD

Abstract

Local differential privacy (LDP) has emerged as a leading framework of personal information protection, even from untrusted data analysts. While many mechanisms satisfy ϵ-privacy constraints, their utility for statistical estimation varies significantly. To address this, we maximize Fisher information over the set of ϵ-LDP mechanisms, ensuring efficiency in subsequent statistical parameter estimates. Prior approaches relied on linear programming with exponential complexity $O(2^k)$ ($k = |\mathscr{X}|$, input alphabet cardinality), limiting practical applications to small input alphabets. We develop a projected gradient ascent algorithm with edge traversal (PGAET) that achieves polynomial complexity $O(k^9)$, enabling optimization in high-dimensional settings. We demonstrate its efficiency on discrete (Poisson) and continuous distributions (Cauchy) via discretization and approximation techniques. We extend previously known results to input alphabet sizes up to $k = 30$, a significant improvement over previously known results. Our work provides a method for researchers to investigate and construct privacy mechanisms with larger input alphabets that retain efficiency in statistical estimation.

Contents

1 Introduction

With the proliferation of computers, humans utilize and have access to an ever-growing amount of data. From data centres to mobile devices, the extent of data collection and utilization is ubiquitous. However, extensive data collection, while useful for companies and researchers, can be harmful to people whose data is being collected. Reasonably, concerns about privacy have become a growing topic in recent years.

The problem, though, is not so clear-cut. On one hand, researchers would like to have as much data as possible, even at the expense of user privacy. On the other hand, users want to preserve their privacy but still benefit from valuable tools such as recommendations and personalized search, which are built upon the analysis of often very private data. Is it possible to share private data in a way that is useful to researchers while preserving personal privacy?

An early attempt to address this balance was simple data anonymization. However, the dangers of relying on anonymization became apparent as attackers demonstrated its limitations. One of the most notable examples was the reconstruction attack on the 2010 U.S. Census data. Researchers showed that by combining aggregated census statistics with publicly available information, it was possible to reconstruct individual records with high accuracy, revealing sensitive demographic information [1]. This highlighted how anonymization alone could fail to protect privacy when auxiliary information was available.

Another famous case was the Netflix Prize dataset, where anonymized user data was released to facilitate research on recommendation systems. Despite the removal of direct identifiers, Narayanan and Shmatikov [2] demonstrated that individuals could be re-identified by correlating the dataset with publicly available IMDb reviews. This attack underscored the vulnerability of anonymization in the face of cross-referencing with external data.

The limitations of simple anonymization prompted the development of more sophisticated privacy-preserving techniques. Early methods in this field focused on techniques such as suppressing data from small groups (e.g., removing tables with fewer than a preset number of individuals) to reduce re-identification risk. Later, methods such as swapping were introduced, where entries most at risk of identification were swapped with others at random. More advanced methods include k-anonymity [3, 4] and l-diversity [5, 4]. Although these techniques offered incremental improvements, they lacked universality and were applicable only in limited situations. They failed to guarantee privacy in all situations and relied on specific sets of assumptions.

These challenges have pushed the evolution of privacy frameworks, culminating in modern approaches such as differential privacy (DP). Unlike earlier methods, differential privacy provides robust guarantees that hold even when attackers possess extensive auxiliary information. This robustness has enabled the widespread utility of differential privacy and its rapid adoption across diverse fields.

A central concern in many of these applications is the need to extract meaningful statistics from data while preserving privacy. At its core, statistics involves estimating parameters of a specified model to describe the underlying data-generating process. For example, researchers might aim to estimate the mean, variance, or other parameters of a distribution based on observed data. However, privatization mechanisms, particularly those adhering to differential privacy, inherently reduce the efficiency of parameter estimation — the more protected the data, the less informative it becomes.

This thesis focuses on a specific aspect of differential privacy: the privacy-utility tradeoff. While many mechanisms satisfy the criteria for differential privacy, their utility varies significantly. Here, we set out to find the mechanism that maximizes Fisher Information, a fundamental statistical concept that quantifies the amount of information a dataset provides about an unknown parameter.

1.1 Fisher Information

Fisher information is one of the most important concepts in statistics. It quantifies the amount of information an observable random variable X carries about an unknown parameter $\theta \in \mathbb{R}$ of a distribution that models X. More formally, it is defined as the variance of the score, which measures how sensitive the likelihood is to changes in θ.

Let X denote a random variable, and let $p_\theta(x)$ represent the probability density function (or probability mass function) of X, parametrized by θ. Suppose we have a sample of n independent and identically distributed (i.i.d.) observations, $X_1, X_2, ..., X_n$, drawn from this distribution. The *likelihood function*, which describes the joint probability of observing this sample, is given by:

$$\mathcal{L}(\theta; \underline{x}) = \prod_{i=1}^{n} p_\theta(x), \tag{1}$$

where $\underline{x} = (x_1, ..., x_n)$ is the vector of observed data. Since the likelihood function depends on θ, it provides a way to evaluate how plausible different parameter values are given the observed data.

The *log-likelihood function* is often more convenient to work with due to the product structure of the likelihood:

$$l(\theta; \underline{x}) = \log \mathcal{L}(\theta; \underline{x}) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{2}$$

The *score function* is defined as the gradient of the log-likelihood function with respect to the parameter θ:

$$s_\theta(\underline{x}) = \frac{\partial l(\theta; \underline{x})}{\partial \theta} = \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log p_\theta(x_i). \tag{3}$$

Intuitively, the score measures the sensitivity of the log-likelihood to changes in θ. It vanishes at critical points, such as maxima or minima, a key property exploited in maximum likelihood estimation (MLE).

Definition 1.1. The *Fisher information* [6] is defined as the second moment of the score under the probability distribution $p_\theta(x)$:

$$I_\theta^{(n)} = E_\theta[s_\theta^2]. \tag{4}$$

Substituting s_θ into the equation gives $I_\theta^{(n)} = E_\theta[\left(\sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log p_\theta(X_i)\right)^2]$. Using the fact that the variables are independent, the expected value of the cross terms vanishes during the square expansion, and we get

$$I_\theta^{(n)} = \sum_{i=1}^{n} E\left[\left(\frac{\partial}{\partial \theta} \log p_\theta(X_i)\right)^2\right] = n \cdot I_\theta^{(1)}.$$

This scaling reflects the additive nature of information from independent observations. Throughout this text, we write $I_\theta = I_\theta^{(1)}$.

Fisher information measures the curvature of the log-likelihood function with respect to θ: higher curvature implies more information and greater certainty about the parameter estimate, while lower curvature indicates greater uncertainty.

This property can be formalized as the *Cramér–Rao bound* [7], which establishes a fundamental limit on the variance of any unbiased estimator $\hat{\theta}$ of a parameter θ. It states that:

$$Var_\theta(\hat{\theta}) \geq \frac{1}{I(\theta)}, \tag{5}$$

where $I(\theta)$ is the Fisher information. This bound establishes the minimum possible variance of an unbiased estimator, which is inversely proportional to the Fisher information. Higher Fisher information indicates greater information about θ, lowering the variance of an efficient estimator, i.e., one that (nearly) attains the lower bound in 5. Additionally,

due to the fact that $I_\theta^{(n)} = nI_\theta^{(1)}$, increasing the sample size also lowers the variance of efficient estimators.

Example 1.2. Consider a $X \sim Bernoulli(\theta)$ random variable with probability mass function:

$$p_\theta(k) = \theta^k(1-\theta)^{1-k}, \ k \in \{0,1\} \tag{6}$$

where θ is the probability of success. The score function of a single observation is:

$$s(X;\theta) = \frac{X}{\theta} - \frac{1-X}{1-\theta}. \tag{7}$$

The Fisher information of a single observation is:

$$I(\theta) = E_\theta[s_\theta^2] = \frac{1}{\theta(1-\theta)}. \tag{8}$$

Due to the scaling property, the total Fisher information for a sample of size n is easily found to be $I_n(\theta) = n \cdot \frac{1}{\theta(1-\theta)}$. Furthermore, the Cramér–Rao bound lower bounds the variance of any unbiased estimator $\hat{\theta}$:

$$Var_\theta(\hat{\theta}) \geq \frac{\theta(1-\theta)}{n}. \tag{9}$$

Fisher information is minimized (and Cramér–Rao bound maximized) when $\theta = 0.5$, reflecting maximum uncertainty in the estimator $\hat{\theta}$ when both outcomes are equally likely. For extreme values of θ (close to 0 or 1), the Cramér–Rao bound rapidly reduces, reflecting better estimability of θ.

1.2 ϵ-Differential Privacy

ϵ-differential privacy (DP) has emerged as a leading framework for privacy protection. First introduced by Dwork et al. [8], it was initially tied to managing trade-offs between noise addition and dataset sensitivity. The framework was later formalized into its current form [9], revolutionising how privacy is approached in data analysis. Prior to DP, privacy frameworks often relied on specific assumptions about the attacker's capabilities and objectives. These frameworks were both challenging to model and prone to errors. In contrast, differential privacy requires no such assumptions. It ensures robust protection against any adversarial inference — whether the attacker aims to re-identify an individual, determine their presence in the dataset, or exploit prior knowledge about the data [10]. DP achieves this by ensuring that the inclusion or exclusion of any individual's data does not significantly alter the output of an analysis. This property guarantees that an adversary cannot reliably infer an individual's record in the database, even with access

to every other record. As a result, DP provides a strong, universal approach to privacy that transcends the limitations of earlier frameworks.

Differential privacy can be implemented in two primary contexts: global and local differential privacy.

Global differential privacy (GDP) applies in scenarios where a trusted data holder collects raw data and releases a sanitized (protected) version of the dataset to the public. For example, a statistical agency might release a noisy version of census data to protect individuals' privacy while enabling analysis. In this model, the data holder has access to the original raw data before applying the differential privacy mechanism.

In contrast, local differential privacy (LDP) ensures that data is sanitized at the data generator level—before it reaches the data holder. Each individual adds noise to their own data, ensuring that even an untrusted data holder cannot infer sensitive information. This model is particularly useful in scenarios where the data holder cannot be fully trusted, such as mobile app telemetry, browser usage analytics, or crowd-sourced surveys. For instance, a user's device might apply a randomized response mechanism to their input before sending it to a central server.

In this work, we focus on local differential privacy because it is robust in protecting individuals' privacy, even in environments with limited trust.

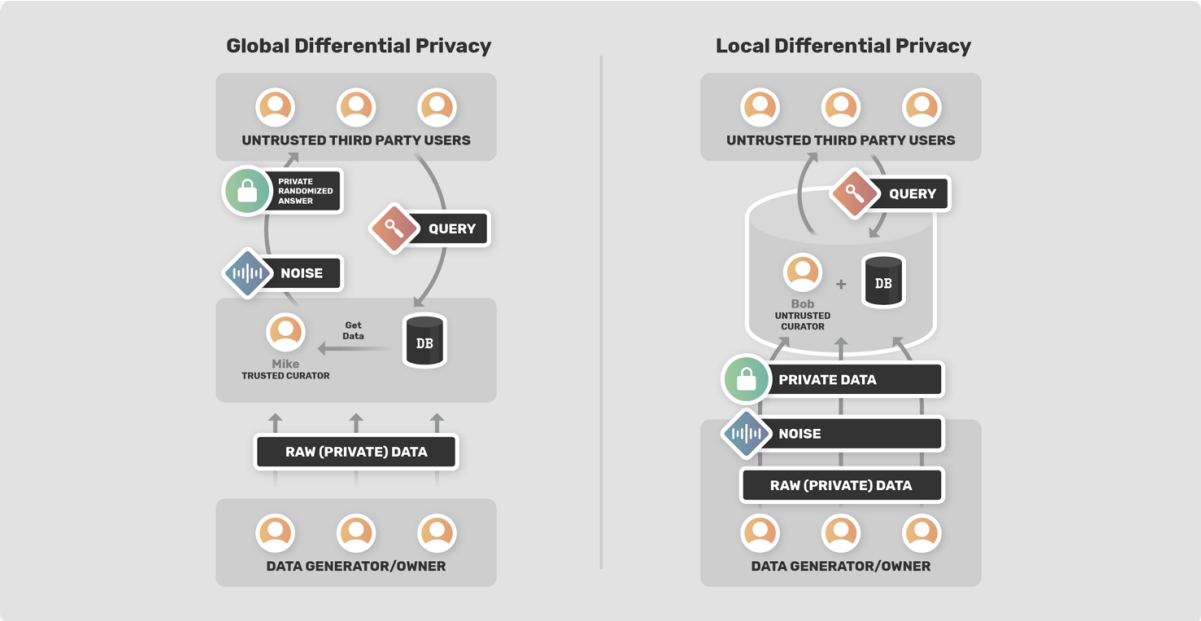

Figure 1: Illustration of the difference between local and global differential privacy. Taken from [11].

Formally, consider a setting where there are n data providers. Each owns some data X_i, defined on an input alphabet $\mathscr{X}$. Each X_i is independently sampled from some

distribution P_θ parametrized by θ. We can define a statistical privatization mechanism Q as a conditional distribution that maps $X_i \in \mathscr{X}$ stochastically to $Y_i \in \mathscr{Y}$, where $\mathscr{Y}$ is an output alphabet. It can be larger than $\mathscr{X}$. These Y_is are referred to as the protected (sanitized) views of X_is. The privatization mechanism Q takes the form of a column stochastic matrix that probabilistically maps input elements of $\mathscr{X}$ to the output elements $\mathscr{Y}$. Finally, a mechanism Q is *ϵ-locally differentially private* if

$$Q(y|x) \leq e^\epsilon \cdot Q(y|x'), \forall y \in \mathscr{Y}, x, x' \in \mathscr{X}. \tag{10}$$

where $Q(y|x) = \mathbb{P}(Y_i = y|X_i = x)$ is the privatization mechanism. ϵ is the privacy parameter - low values of ϵ correspond to high privacy regimes, while high values correspond to low privacy regimes (although it is worth noting that high and low privacy regimes depend a lot on the context and the particular application). In the extremes, the sanitized output is either independent of the private data ($\epsilon = 0$) or equal to the private data ($\epsilon = \infty$).

Example 1.3. To provide a concrete example, consider a privatization mechanism operating on an input alphabet of size 2. This mechanism is a 2×2 matrix that satisfies ϵ-privacy constraints and column-stochasticity. For instance:

$$Q = \frac{1}{1+e^\epsilon} \begin{pmatrix} e^\epsilon & 1 \\ 1 & e^\epsilon \end{pmatrix}. \tag{11}$$

This matrix, Q, is a Markov kernel that determines the conditional distribution $Z_i|X_i$. Here, Z_i represents the sanitized output, and X_i represents the protected/private input data.

If we label the input alphabet as $\{0, 1\}$, the columns of Q can be interpreted as the probabilities of mapping elements of the input alphabet to the corresponding elements of the output alphabet. For instance, if the private data is '0', then with probability $\frac{e^\epsilon}{1+e^\epsilon}$ the sanitized data will remain '0', and with probability $\frac{1}{1+e^\epsilon}$, the sanitized data will be flipped to '1'. Similarly, for the second element of the input alphabet ('1'), the mechanism outputs '1' with probability $\frac{e^\epsilon}{1+e^\epsilon}$ and flips it to '0' with probability $\frac{1}{1+e^\epsilon}$.

1.3 Our contribution

In the analysis of various statistical databases, researchers are interested in *statistics* of the data. Differential privacy applies randomness to the data, which can obfuscate the extracted statistics. However, given a ϵ-privacy requirement, there are many privacy mechanisms, some of which preserve statistical properties better than others.

Formally, we take a look at the set of all ϵ-differentially private mechanisms, $D_\epsilon = \{Q \in \mathbb{R}^{|\mathscr{Y}|\times|\mathscr{X}|} : \sup_{y\in\mathscr{Y},x,x'\in\mathscr{X}} \frac{Q(y|x)}{Q(y|x')} \leq e^\epsilon, \quad \sum_{x\in\mathscr{X}} Q(y|x) = 1, Q(y|x) \geq 0, \forall x \in \mathscr{X}, \forall y \in \mathscr{Y}\}$. Each mechanism must satisfy ϵ-differential privacy constraints and column stochasticity (columns sum up to 1 and all elements are non-negative). The private data is generated by a parametrized probability distribution p_θ, where θ is a single parameter. We assume that all clients use the same privatization mechanism and that each client's data is an i.i.d. sample from the parametrized distribution. Given all this information, we are interested in finding $Q \in D_\epsilon$ such that the Fisher Information of the sanitized data is maximized:

$$\sup_{Q\in D_\epsilon} I_\theta(Q) = \sup_{Q\in D_\epsilon} \sum_{y\in\mathscr{Y}} \frac{(\sum_{x\in\mathscr{X}} Q(y|x)\dot{p}_\theta(x))^2}{\sum_{x\in\mathscr{X}} Q(y|x)p_\theta(x)}. \tag{12}$$

We defer precise specification of terms in this equation to Section 3.3. Fisher information in this context is immensely important as it quantifies the information about the true parameter in the data generating statistical model. Differential privacy introduces a level of randomization to the original model, and we end up with another statistical model which produces the data we will get to see. This is the data we will use for estimation, and the estimation is the easiest if Fisher Information of the resulting data generating statistical model is large (c.f. Cramér–Rao bound, equation 5).

It is worth noting that this is an already solved problem. Kairouz, Oh and Viswanath [12] developed a linear programme that can solve this problem exactly. The issue with the linear programme is that it scales exponentially with the input alphabet cardinality $|\mathscr{X}|$, rendering it impractical for most applications. We provide a summary of the algorithm in section 2. We aim to develop a gradient ascent approach that scales better than the linear programme. To this end, we develop a projected gradient ascent algorithm, section 3. Finally, in section 4, we apply the projected gradient approach to the problems of finding efficient privacy mechanisms for the Cauchy and Poisson distributions. We apply the algorithm to previously unmanageable sizes of the input alphabet.

2 Linear Programme (extremal mechanisms)

This section is entirely based on [12].

The authors of the paper consider a generalized version of the maximization problem (12). Consider a client-server setting where each client with data X_i releases Y_i a sanitized version of the data, via a non-interactive ϵ-locally differentially private privatization mechanism Q. Assume all the clients use the same privatization mechanism Q, and each client's data is an i.i.d. sample from a distribution p_θ for some parameter θ. Given

the sanitized views $\{Y_i\}_{i=1}^n$, the data analyst would like to make inferences based on the induced marginal distribution

$$M_\theta(y) \equiv \sum_{x \in \mathscr{X}} Q(y|x) p_\theta(x), \tag{13}$$

for $y \in \mathscr{Y}$.

For an input alphabet $\mathscr{X}$ with $|\mathscr{X}| = k$, we represent the set of ϵ-locally differentially private mechanisms that lead to output alphabets $\mathscr{Y}$ with $|\mathscr{Y}| = l$ by

$$D_{\epsilon,l} = Q_{k\times l} \cap \{Q : \forall x, x' \in \mathscr{X}, y \subseteq \mathscr{Y}, \left| \ln \frac{Q(y|x)}{Q(y|x')} \right| \leq \epsilon\}, \tag{14}$$

where $Q_{k\times l}$ denotes the set of all $k \times l$ dimensional columns stochastic matrices. The set of all ϵ-locally differentially private mechanisms is given by

$$D_\epsilon = \cup_{l \in \mathbb{N}} D_{\epsilon,l}. \tag{15}$$

Given the entire set of ϵ-differential privacy mechanisms, a natural question to ask is how can we find the most useful privacy mechanism. Usefulness is formalized through the use of a utility function (which measures some important property of the data, for example, the Fisher Information), and the goal is to maximize said utility function. The utility function, denoted as $U : Q \to \mathbb{R}_+$, maps the privatization mechanism Q to non-negative real numbers. Additionally, U must be decomposable into a sum of *sublinear functions*.

Definition 2.1. A function $\mu(z) : \mathbb{R}^k \to \mathbb{R}$ is a sublinear function if

1. $\mu(\lambda z) = \lambda \mu(z)$ for all $\lambda \in \mathbb{R}_+$.
2. $\mu(z_1 + z_2) \leq \mu(z_1) + \mu(z_2)$ for all $z_1, z_2 \in \mathbb{R}^k$.

Let Q_y be the row of Q corresponding to $Q(y|\cdot)$ and μ be any sublinear function. Then, the maximization problem can be written as

$$\text{maximize}_Q \quad U(Q) = \sum_{y \in \mathscr{Y}} \mu(Q_y) \tag{16}$$

$$\text{subject to} \quad Q \in D_\epsilon. \tag{17}$$

16 is, in general, a complicated nonlinear program. We are maximizing a function in Q, where the dimension of Q might be unbounded. Luckily, it can be shown that one never needs an output alphabet larger than the input alphabet to achieve the maximum utility, vastly simplifying the optimization problem. Furthermore, it can be shown that the optimal solution must be at one of the vertices of the underlying feasible set of ϵ-differentially private matrices (this set forms an n-dimensional polyhedron).

Theorem 2.2. *For any sublinear function μ and any $\epsilon \geq 0$, there exists an optimal mechanism Q^* maximizing the utility in 16 over all ϵ-locally differentially private mechanisms, such that*

- *the output alphabet size is at most the input alphabet size, i.e. $|\mathscr{X}| \geq |\mathscr{Y}|$; and*
- *for all $y \in \mathscr{Y}$, and $x, x' \in \mathscr{X}$*

$$\left| \ln \frac{Q^*(y|x)}{Q^*(y|x')} \right| \in \{0, \epsilon\}. \tag{18}$$

This theorem establishes that there is an optimal mechanism with an extremal structure; the absolute value of the log-likelihood ratios can only take one of the two extremal values: 0 or e^ϵ. The authors refer to such mechanisms as staircase mechanisms and define the family of staircase mechanisms as

$$S_\epsilon \equiv \{Q | \text{satisfying } 18\}. \tag{19}$$

For a given input alphabet dimension k, we can construct the *Staircase Pattern Matrix*. It consists of all possible combinations of 1 and e^ϵ values. For example, when $k = 3$ there are $2^k = 8$ staircase patterns and the staircase pattern matrix is

$$S^{(3)} = \begin{pmatrix} 1 & 1 & 1 & 1 & e^\epsilon & e^\epsilon & e^\epsilon & e^\epsilon \\ 1 & 1 & e^\epsilon & e^\epsilon & 1 & 1 & e^\epsilon & e^\epsilon \\ 1 & e^\epsilon & 1 & e^\epsilon & 1 & e^\epsilon & 1 & e^\epsilon \end{pmatrix}$$

For any input dimension k, there will always be 2^k such patterns, and any row $Q^*(y|\cdot)$ of Q^*, the optimal privatization mechanism, is a scaled version of one of the columns of $S^{(k)}$. Using the pattern matrix, we can represent any staircase mechanism as

$$Q^T = S^{(k)}\Theta, \tag{20}$$

where $\Theta = \text{diag}(\theta)$ is a $2^k \times 2^k$ diagonal matrix and θ is a 2^k-dimensional vector representing the scaling of the columns of $S^{(k)}$.

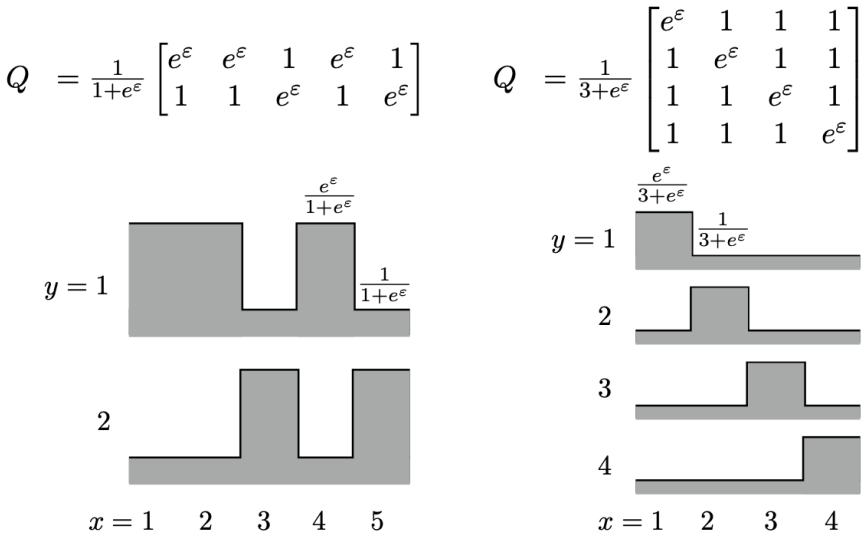

Figure 2: Examples of staircase mechanisms. On the left, the binary, on the right, the randomized response mechanism. Taken from [12] and slightly modified.

Theorem 2.3. *Given any sublinear function μ and $\epsilon > 0$, our optimization problem can be formulated as*

$$\max_{\theta \in \mathbb{R}^{2^k}} \quad \sum_{j=1}^{2^k} \mu(S_j^{(k)})\theta_j = \mu^T \theta \tag{21}$$

$$\text{subject to} \quad S^{(k)}\theta = \mathbb{1} \tag{22}$$

$$\theta \geq 0, \tag{23}$$

where $S_j^{(k)}$ refers to the jth column of the staircase matrix $S^{(k)}$. The solution of the linear programme is related to the optimal solution by $Q^T = S^{(k)}\Theta$.

Notice that in this formulation, both the number of variables θ and the number of constraints are dictated by the size of the staircase matrix $S^{(k)}$, which in turn grows exponentially with increasing k.

An important property of the objective function and differential privacy constraints is invariance under certain transformations, such as permutations and merging/splitting of outputs. This property is critical to understanding the flexibility of solutions generated by optimization algorithms (different algorithms might produce different solutions which are equivalent).

- Permutations: Rearranging the rows of a privatization mechanism Q results in a new mechanism Q' that is also ϵ-locally differentially private and achieves the same

utility. This is a result of the form of the utility under consideration (see Equation 16).

- Merging/splitting of rows: Consider two outputs y and y' that have the same pattern, $Q(y|\cdot) = CQ(y'|\cdot)$ for some positive constant C, and are a part of the output alphabet $\mathscr{Y}$. We can consider a new mechanism Q' obtained by merging the two rows corresponding to y and y' (reducing the size of the output alphabet by adding two rows of a privatization matrix, element-wise), name it y'' which is an element of a different output alphabet $\mathscr{Y}'$. It follows that Q' satisfies the differential privacy constraints and preserves the utility. Using the fact that $Q(y|\cdot) = CQ(y'|\cdot)$ we get

$$\mu(Q_y) + \mu(Q_{y'}) = \mu(CQ_{y'}) + \mu(Q_{y'}) = C\mu(Q_{y'}) + \mu(Q_{y'}) \tag{24}$$
$$= (1+C)\mu(Q_{y'}) = \mu((1+C)Q_{y'}) = \mu(Q'_{y''}), \tag{25}$$

  by the homogeneity property of μ ($\mu(\lambda z) = \lambda\mu(z)$), where the cardinality of the new output alphabet $\mathscr{Y}'$ is decreased by 1, $|\mathscr{Y}'| = |\mathscr{Y}| - 1$. It is important to mention that we can only merge rows if the resulting mechanism Q' remains column stochastic after the merge (i.e. the resulting elements cannot be greater than 1). The merging illustrates that multiple representations of equivalent mechanisms exist within the space of ϵ-differentially private mechanisms.

Naturally, we can define equivalence classes for mechanisms that are equivalent up to a permutation of rows and merging/splitting of rows with the same pattern:

$$[Q] = \{Q' \in D_\epsilon | \exists \text{ a sequence of permutations and merge/split of rows from } Q' \text{ to } Q\}.$$

Example 2.4. Consider an input alphabet of size three and a binary response privatization mechanism that achieves a certain value of the utility function:

$$Q^* = \frac{1}{1+e^\epsilon}\begin{pmatrix} 1 & 1 & e^\epsilon \\ e^\epsilon & e^\epsilon & 1 \end{pmatrix}.$$

However, we may split any row into two while maintaining the same ϵ-privacy structure to attain a privatization mechanism that achieves the same utility. For instance,

$$Q = \frac{1}{1+e^\epsilon}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2}e^\epsilon \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2}e^\epsilon \\ e^\epsilon & e^\epsilon & 1 \end{pmatrix}.$$

This invariance has significant implications for the projected gradient algorithm. The algorithm often identifies solutions that differ from those found by the linear programme but belong to the same equivalence class.

3 Projected Gradient Ascent

In this section, we present a modified version of gradient descent which aims to maximize Fisher information in ϵ-locally differentially private settings [12].

Before delving into the algorithm, we reiterate some useful properties of the optimization problem.

1. The output alphabet size is at most as big as the input alphabet size. This property allows us to limit the size of the privacy matrix. The claim is found in Theorem 21 and the proof in Section 7.1 of [12].

2. In the gradient ascent algorithm, we apriori restrict the search space by setting the output alphabet size equal to the input alphabet size. This includes all smaller output spaces as well, as reducing the alphabet size can be achieved by setting entire rows of the privatization matrix to 0.

3. The set of all ϵ-differentially private matrices with a fixed input alphabet size $|\mathscr{X}|$, $D_{\epsilon,|\mathscr{X}|}$, is a convex set. This claim can be found in Lemma 23, and the proof is in section 7.3 of [12].

4. Fisher information, as a function of the privatization matrix, Q, is of the form $I_\theta(Q) = \sum_{y\in\mathscr{Y}} \mu(Q_y)$ for a sublinear function μ as in Definition 2.1. Lemma 4.5 of [13] supports this claim.

5. The maximum value of the Fisher information is found at the vertices of the convex set of ϵ-differentially private matrices (Theorem 2.2, proof can be found in Section 7 of [12]).

3.1 Gradient Descent

Gradient descent is an algorithm for unconstrained optimization. The idea underlying the algorithm is fairly simple - utilize the information contained within the gradient of the function to move towards an optimum. It is a first-order local method, which means that it utilizes only the first derivative (gradient) of the function. The gradient is

combined with a step size (also called learning rate in the machine learning community) to iteratively move towards an optimum. A visual example is given in Figure 3.

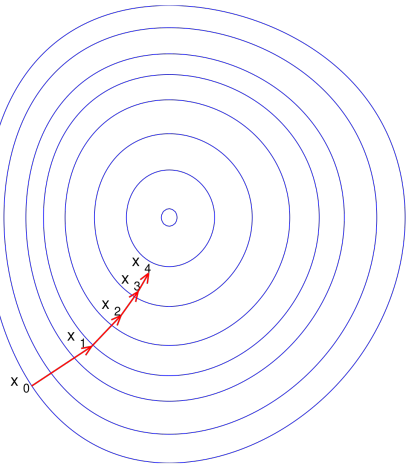

Figure 3: Illustration of gradient descent on a series of level sets. Taken from [14].

Specifically, look at a multivariable function $F(x)$ that is differentiable everywhere. In a neighbourhood of a point a, the function decreases the fastest if we move from a in the direction of the negative gradient of F at a, $-\nabla F(a)$. Combined with a step size λ_t, which may or may not be constant throughout, this procedure will yield a sequence of values:

$$x_{t+1} = x_t - \lambda_t \nabla F(x_t). \tag{26}$$

There is a variety of literature on properties of gradient descent (e.g. [15]). To ensure that gradient descent is an effective optimization method, it is crucial to establish conditions under which it converges to a minimum. One of the fundamental convergence results is as follows.

Theorem 3.1. *Consider the problem of minimizing a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, where f has a well-defined minimum, i.e. $\arg\min f \neq \emptyset$. The Gradient Descent algorithm with a constant stepsize defines a sequence $(x_t)_{t\in\mathbb{N}}$:*

$$x_{t+1} = x_t - \lambda \nabla f(x_t).$$

Assume that f is convex and L-smooth, for some $L > 0$. If the stepsize λ satisfies $0 < \lambda \leq \frac{1}{L}$, then for all $x^* \in \arg\min f$ and for all $t \in \mathbb{N}$

$$f(x_t) - \inf f \leq \frac{||x_0 - x^*||^2}{2\lambda t}.$$

Of course, there exist many extensions to the basic gradient descent algorithm. Generally, they concern step size adaptation [16], introduction of stochasticity [17], or preconditioning [18].

We utilize a slightly modified version of gradient descent. The problem we are interested in here is a maximization problem, so we are interested in gradient *ascent*. Alternatively, we could also minimize the negative of the objective function to arrive at the same result. The difference is irrelevant, and the same principles apply.

3.2 Feasible set visualization

As is often the case in mathematics, if possible, it is very instructive to visualize the problem you are working on in 2 or 3 dimensions. Here, we visualize the feasible set in 2-dimensions and try to exploit the properties of the space when dealing with higher-dimensional spaces.

Consider the case of a Bernoulli distribution. It is a simple distribution that describes the probability of success of some event. If X is a random variable with a Bernoulli distribution, then:

$$Pr(X = 1) = \theta = 1 - Pr(X = 0) = 1 - \theta. \tag{27}$$

Here, θ describes the probability of success. Naturally, $1 - \theta$ is the probability of failure. We can condense the probability mass function into

$$p_\theta(k) = \theta^k (1 - \theta)^{1-k} \tag{28}$$

where $k \in \{0, 1\}$ is the *support* of the distribution.

Given a privacy value ϵ and the parameter θ of the Bernoulli distribution, consider the problem of data privatization. Since the underlying data is discrete, the privatization mechanism takes on the form of a matrix. This matrix maps the input alphabet to an output alphabet (where the input alphabet, in this case, is $\{0, 1\}$). The output alphabet can be of any size. However, as outlined in Theorem 2.2, for functions we are interested in, it suffices to limit our search to output alphabets that are at most as large as the input alphabet. Therefore, the privatization matrix is a 2×2 matrix. The matrix needs to be a column stochastic matrix (that is, the columns sum up to 1 and are non-negative), and the ϵ-privacy criterion is enforced within each row.

Due to the column-stochasticity condition, this privatization matrix can be parametrized using two freely varying parameters:

$$Q = \begin{pmatrix} a & b \\ 1-a & 1-b \end{pmatrix} \quad 0 \leq a, b \leq 1. \tag{29}$$

In this Bernoulli setting, there are only two requirements for this matrix to be ϵ-differentially private:

1. $e^{-\epsilon}a \leq b \leq e^{\epsilon}a$,
2. $e^{-\epsilon}(1-a) \leq 1-b \leq e^{\epsilon}(1-a)$.

This set of inequalities can be solved analytically. The set takes on the form of a rhombus, with vertices $A = (0,0), B = (\frac{1-e^{\epsilon}}{e^{-\epsilon}-e^{\epsilon}}, \frac{e^{-\epsilon}-1}{e^{-\epsilon}-e^{\epsilon}}), C = (1,1), D = (\frac{1-e^{-\epsilon}}{e^{\epsilon}-e^{-\epsilon}}, \frac{e^{\epsilon}-1}{e^{\epsilon}-e^{-\epsilon}})$, see Figure 4.

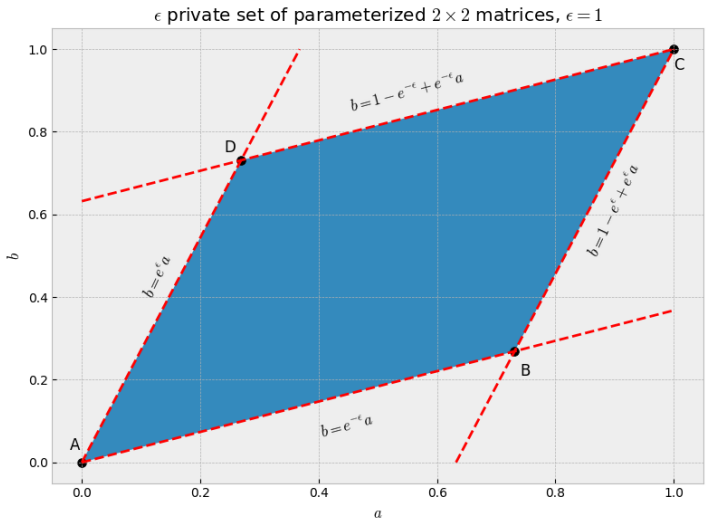

Figure 4: Visualization of a set of ϵ-differential private matrices for a simple Bernoulli experiment. In this simple example, the privatization mechanism can be written as $Q = \begin{pmatrix} a & b \\ 1-a & 1-b \end{pmatrix}$. Increasing the privacy parameter ϵ increases the size of this set, and in the limit of $\epsilon \to \infty$, the feasible set covers the entirety of available space, $[0,1] \times [0,1]$.

In higher dimensions, the feasible set of ϵ-differentially private matrices becomes significantly more complex, transitioning from a simple geometric shape like a rhombus in two dimensions to a high-dimensional polytope. While the feasible set for the Bernoulli case can be visualized as a rhombus in 2D, this simplicity does not carry over to higher-dimensional input alphabets. For larger input sizes ($|\mathscr{X}| > 2$), the privatization matrix Q must satisfy a set of linear constraints that extend the principles of column-stochasticity

and ϵ-differential privacy to multiple dimensions. Specifically, the constraints generalize as follows:

$$e^{-\epsilon}Q(y|x') \leq Q(y|x) \leq e^{\epsilon}Q(y|x'), \quad \forall x, x' \in \mathscr{X}, \forall y \in \mathscr{Y}.$$

Additionally, each column of Q must represent a valid probability distribution:

$$\sum_{x \in \mathscr{X}} Q(y|x) = 1, \quad Q(y|x) \geq 0, \quad \forall y \in \mathscr{Y}, \forall x \in \mathscr{X}.$$

The dimension of the feasible set is determined by the degrees of freedom of the matrix Q after accounting for column-stochasticity and differential privacy constraints. For a matrix with $|\mathscr{X}|$ rows and $|\mathscr{Y}|$ columns, there are $|\mathscr{X}| \cdot |\mathscr{Y}| - |\mathscr{Y}|$ degrees of freedom (subtracting the normalization constraints for each column). For every output element, there are $|\mathscr{X}| \cdot (|\mathscr{X}| - 1)$ ϵ-privacy constraints, for a total of $|\mathscr{Y}| \cdot |\mathscr{X}| \cdot (|\mathscr{X}| - 1)$ constraints. This set forms a convex polytope bounded by hyperplanes arising from the ϵ-privacy constraints. The number of constraints scales quadratically with input alphabet size $|\mathscr{X}|$ and linearly with output alphabet size $|\mathscr{Y}|$. Since the output alphabet size is at most the size of the input alphabet size, the resulting constraining set has $O(|\mathscr{X}|^3)$ constraints.

3.3 Projected Gradient Ascent

In this section, we develop the projected gradient approach, a well-known extension of the basic gradient descent algorithm with extensive literature on the subject. Depending on the context, it is also referred to as the *Gradient Projection Method* [19], *Proximal Gradient Method* [20], or *Projected Gradient Methods* [21].

This approach enjoys several desirable convergence properties, many of which have been well-established [15]. In particular, its fundamental convergence rate aligns with Theorem 3.1.

The objective function (Fisher Information of the sanitized model) depends on the privatization matrix Q and is given by

$$I_\theta(Q) = \sum_{y \in \mathscr{Y}} \frac{(\sum_{x \in \mathscr{X}} Q(y|x)\dot{p}_\theta(x))^2}{\sum_{x \in \mathscr{X}} Q(y|x)p_\theta(x)} \tag{30}$$

$$= \sum_{y \in \mathscr{Y}} \frac{(Q_y \cdot \dot{p}_\theta)^2}{Q_y \cdot p_\theta}, \tag{31}$$

where $Q_y = Q(y|\cdot)$ represents an entire row of the matrix Q and, with a slight abuse of notation, we write $p_\theta := (p_\theta(x))_{x\in\mathscr{X}} \in \mathbb{R}^{|\mathscr{X}|}$ for the probability vector of the prior distribution over the input alphabet $\mathscr{X}$ and $\dot{p}_\theta := (\frac{\partial}{\partial\theta}p_\theta(x))_{x\in\mathscr{X}} \in \mathbb{R}^{|\mathscr{X}|}$ for the vector of corresponding derivatives with respect to the parameter θ.

The gradient of the objective function, which will be used in the projected gradient updates, is given by:

$$\nabla I_\theta(Q)\big|_{(x,y)} = 2\dot{p}_\theta(x)\frac{Q_y \cdot \dot{p}_\theta}{Q_y \cdot p_\theta} - p_\theta(x)\frac{(Q_y \cdot \dot{p}_\theta)^2}{(Q_y \cdot p_\theta)^2}. \tag{32}$$

The objective function, the gradient, the set of feasible and optimal solutions are plotted in Figure 5. There are a few things we can immediately notice:

1. Vertices as Optimal Solutions: The optimal solutions are located at the vertices of the feasible solution set. In this simple example, they can be calculated analytically: $Q^* = \frac{1}{1+e^\epsilon}\begin{pmatrix} e^\epsilon & 1 \\ 1 & e^\epsilon \end{pmatrix}$, while the other solution is obtained by permuting the two rows. The optimal solutions are always located at the vertices of the feasible set, even in higher dimensions, as established in the previous section on "Extremal mechanisms".

2. Tradeoff Between Privacy and Information: The absolute highest values of the Fisher information (ignoring the feasible set constraints) are located at points $(1, 0)$ and $(0, 1)$. These points correspond to no privacy, as the data is either unchanged or the labels are switched. All finite values of ϵ lead to a reduction in Fisher information of the sanitized data, introducing a tradeoff between privacy and statistical efficiency.

We employ gradient ascent to iteratively maximize Fisher information while respecting differential privacy and stochasticity constraints. Since gradient ascent does not inherently respect differential privacy constraints, we enforce feasibility by projecting the iterates back onto the feasible set D_ϵ. This projection step is formulated as a quadratic optimization problem, ensuring that the updated matrix remains column-stochastic and satisfies the privacy constraints.

Specifically, given an intermediate iterate Q (that falls outside the feasible set), we solve:

$$P(Q) = \arg\min\{||Q - Z||_F : Z \in D_\epsilon\} \tag{33}$$

where $||\cdot||_F$ is the Frobenius norm, ensuring that $P(Q)$ is the closest feasible matrix to Q in terms of Euclidean distance.

This is a convex *quadratic programming (QP) problem*, with a quadratic objective and linear constraints. In practice, we solve it using *CVXPY* [22].

The computational complexity of the projection step depends on the choice of a solver. Interior-point methods, commonly used to solve quadratic programs, typically have a complexity of $O(n_v^3 n_c)$ (see [23] for extensive analysis), where n_v is the number of variables of the objective function and n_c is the number of constraints. The number of variables is dictated by the input alphabet dimension, given by $n_v = O(k^2)$, while the number of constraints scales as $n_c = O(k^3)$ leading to an overall complexity of $O(k^9)$ where k is the input alphabet size. Other solvers, such as active-set methods or first-order approaches, may have different complexities, but the key takeaway is that the problem remains polynomial-time solvable, scaling reasonably well with increasing matrix size.

The full projected gradient ascent update rule is then given by:

$$Q_{t+1} = P(Q_t + \lambda_t \nabla I_\theta(Q_t)). \tag{34}$$

The algorithm behaves like a standard gradient ascent until it reaches the boundary of the feasible set. Up to this point, Fisher information increases smoothly in each iteration. However, once the algorithm reaches the boundary, further improvements in Fisher information rely on iteratively stepping outside the feasible set and projecting back to a better feasible position. This process often resembles a zig-zagging motion along the constraint boundary. To facilitate this process, we map the boundary and periodically perform a large step along the boundary of the feasible set. The complete procedure is outlined in Algorithm 1, with a visualization provided in Figure 5, illustrating its behaviour on a simple Bernoulli problem.

Algorithm 1 Projected Gradient Ascent with Edge Traversal (PGAET)

Require: p_θ, $\dot{p_\theta}$, ϵ, max_iter, k-input alphabet size

Initialize Q_0 as a matrix of 1s, J_k, divided by the size of the input alphabet k, plus some random noise: $Q_0 = \frac{1}{k} J_k + N(\underline{0}, \underline{\sigma})$, where $\underline{\sigma}$ is a diagonal matrix with elements σ.
Project Q_0 onto the feasible set using ϵ-privacy and stochasticity constraints.
Set $Q_t \leftarrow Q_0$, $I_t \leftarrow \text{FisherInformation}(p_\theta, \dot{p_\theta}, Q_t)$.
Initialize $P_1 \leftarrow \text{None}, P_2 \leftarrow \text{None}$ (for boundary line search projections).
for $t = 1, \ldots, \text{max\_iter}$ **do**
    if $P_1 \neq \text{None}$ and $P_2 \neq \text{None}$ **then**
        Compute $\Delta \leftarrow P_2 - P_1$. ▷ Direction along the boundary.
        Perform boundary search: $Q_{t+1} \leftarrow \text{LineSearch}(Q_t, \Delta, \epsilon)$.
        ▷ Larger step along the boundary.
        Reset $P_1 \leftarrow \text{None}, P_2 \leftarrow \text{None}$.
    else
        Compute gradient $\nabla I_t \leftarrow \text{FisherGradient}(p_\theta, \dot{p_\theta}, Q_t)$.
        Normalize gradient: $\nabla I_t \leftarrow \nabla I_t / \max(1, \|\nabla I\|_{\text{Fro}}/C)$.
        Update $Q_{t+1} \leftarrow Q_t + \nabla I_t / \sqrt{t+1}$.
        if Q_{t+1} is not ϵ-private **then**
            Project Q_{t+1} onto the feasible set using ϵ-privacy and stochasticity constraints.
            if $P_1 \neq \text{None}$ **then**
                Set $P_2 \leftarrow Q_{t+1}$.
            else
                Set $P_1 \leftarrow Q_{t+1}$.
            end if
        end if
    end if
    Compute $I_{t+1} \leftarrow \text{FisherInformation}(p_\theta, \dot{p_\theta}, Q_{t+1})$.
    if $\text{Convergence}(Q_t, Q_{t+1}, I_t, I_{t+1}, \text{tol})$ **then**
        return Q_{t+1} ▷ Early exit for convergence.
    end if
    Update $Q_t \leftarrow Q_{t+1}, I_t \leftarrow I_{t+1}$.
end for
return Q_t.

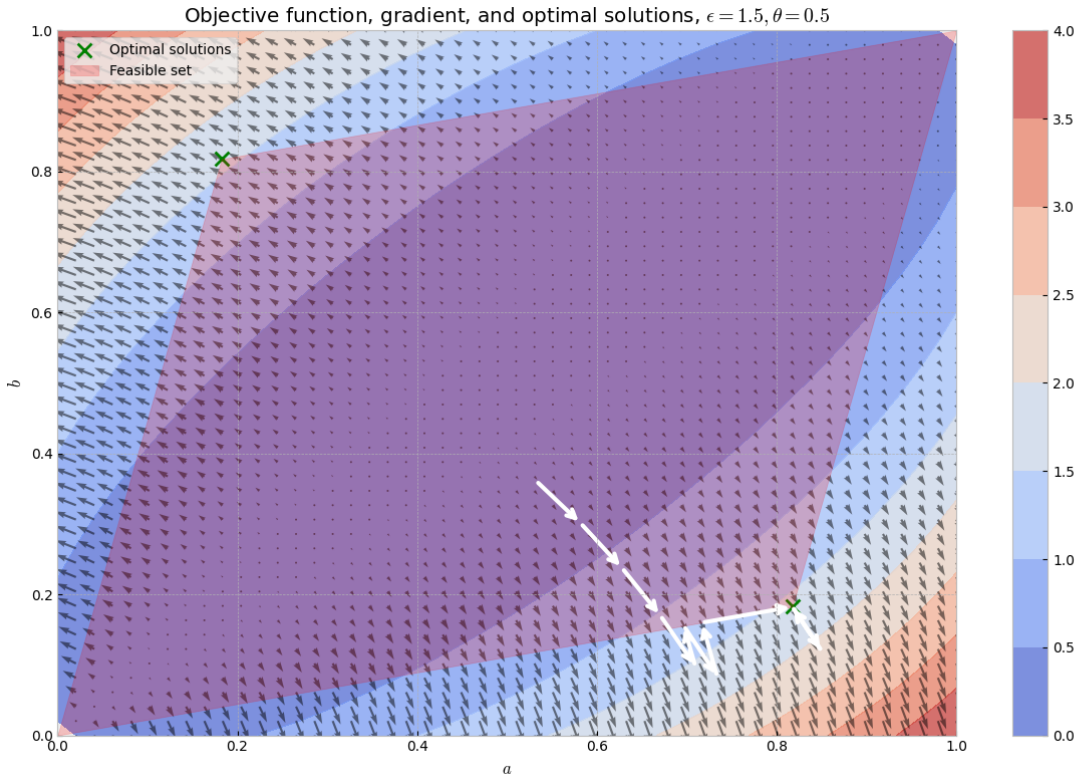

Figure 5: Fisher information (underlying colour-filled contours), the gradient of the Fisher information (black arrows), feasible solution set (pink rhombus), optimal solutions (green crosses), and a sample projected gradient ascent with edge traversal path (white arrows). Notice how successive projections push the solution towards the vertex of the feasible set. The edge traversal step greatly facilitates this process.

In practice, the algorithm may sometimes produce suboptimal solutions due to numerical precision errors and related computational limitations, or it may simply converge to a local maximum. To address this, we employ a multiple restart strategy, in which the algorithm is initialized from different starting points, and the best-performing solution is selected. We refer to this approach as *PGAETMultipleRestarts*.

3.4 Comparison and Performance

In this section, we compare the performance of the projected gradient ascent algorithm and the aforementioned linear programme (extremal mechanism). Since the linear programme is an exact algorithm, we use it as a baseline to compare with the gradient ascent results.

Throughout this section, it is assumed that the prior distribution is a binomial distribution:

$$p_\theta^{(m)}(x) = Pr(X = k) = \binom{m}{k} \theta^k (1-\theta)^{n-k} \tag{35}$$

where m denotes the number of independent Bernoulli trials, each with the same probability of success θ. Consequently, m dictates the dimension of the input alphabet in ϵ-differential privatization problems and can be read as the "dimensionality" of the privacy matrix.

We start with a simple showcase of sanitized Fisher information for various values of m in Figure 6. At all values of m, a significant drop in Fisher information can be observed compared to the non-sanitized setting. This is because we are dealing with a relatively high privacy regime where $\epsilon = 1.0$ (although high and low privacy are impossible to define and highly depend on a specific application).

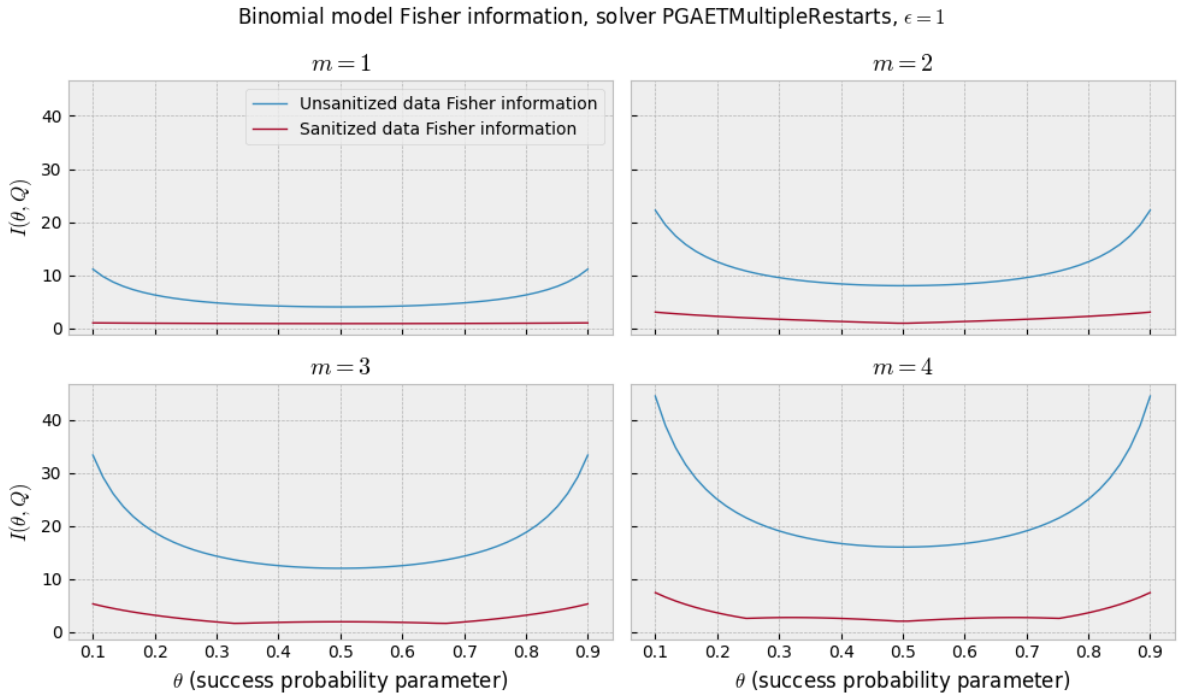

Figure 6: Unsanitized and sanitized Fisher information for values of $m \in \{1, 2, 3, 4\}$. Here, $\epsilon = 1$, corresponding to a relatively high privacy regime. Significant Fisher information loss can be seen in all plots.

In Figure 7, we depict PGAET's performance in a high privacy regime ($\epsilon = 1$) and in a more moderate setting ($\epsilon = 5$). In a high-privacy setting, PGAET continuously finds optimal solutions. In contrast, there is a consistent loss of accuracy in a lower-privacy setting, possibly due to rounding errors or imperfect convergence conditions.

The primary motivation for developing an alternative algorithm is the high time and space complexity of the exact linear programme. The extremal mechanism algorithm has an exponential memory complexity of $O(2^k)$, where k is the dimension of the input alphabet. This is due to the construction of the staircase matrix, which defines the number of constraints for the linear programme. Consequently, the time complexity is

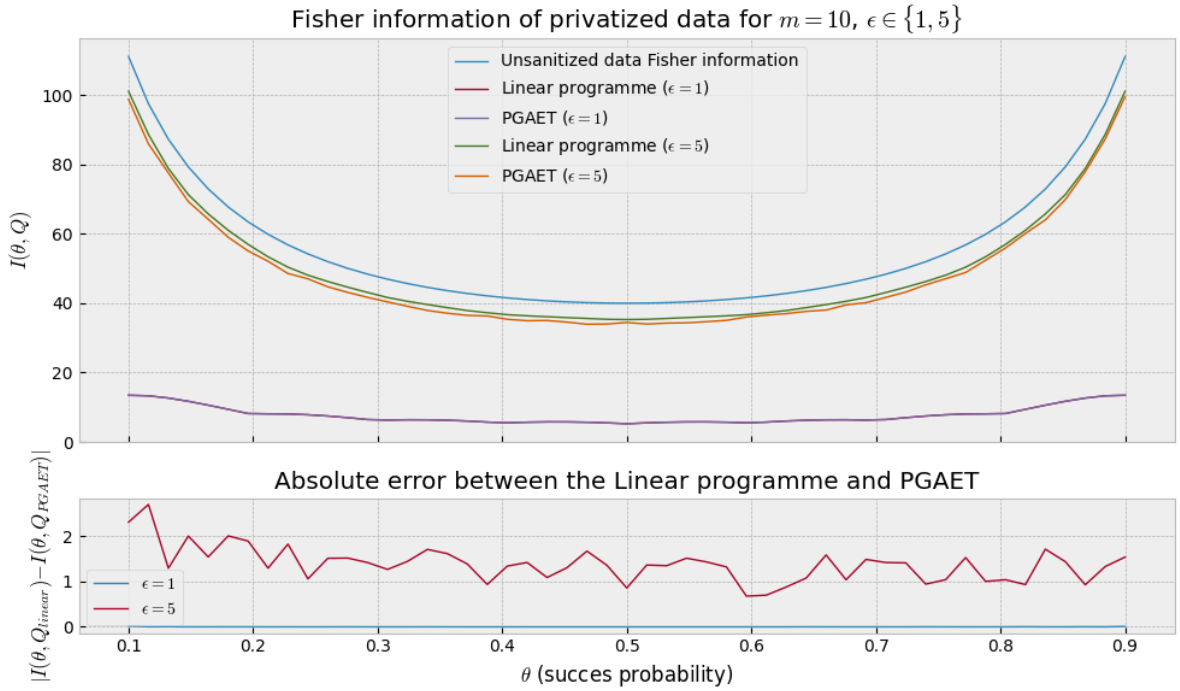

Figure 7: Comparison of the exact linear solver and our gradient ascent based algorithm on a binomial privatization problem with $n = 10, \epsilon \in \{1, 5\}$. The gradient ascent solver is restarted 10 times with different initial positions. In the lower privacy regime, the solver consistently fails to attain optimal solutions.

inherently constrained by the exponential number of constraints. Since all algorithms designed to solve linear programmes have at least linear complexity in the number of constraints, the overall time complexity is at least $O(2^k)$.

In contrast, the gradient ascent retains a polynomial complexity profile. The most computationally intensive step is the projection subroutine, which scales as $O(k^9)$ in the input alphabet dimension. This polynomial complexity is highly valuable, allowing us to run the algorithm efficiently even for relatively large values of k, whereas the linear programme becomes infeasible beyond $k = 15$ on a standard retail computer. In Figure 8, we compare the runtime of both algorithms, demonstrating better scalability of the gradient ascent approach.

4 Applications

In previous sections, we explored the algorithm's performance on a discrete, binomial probability distribution. While certainly useful for discrete probability distributions, applying the algorithm to continuous probability distributions via discretization is also

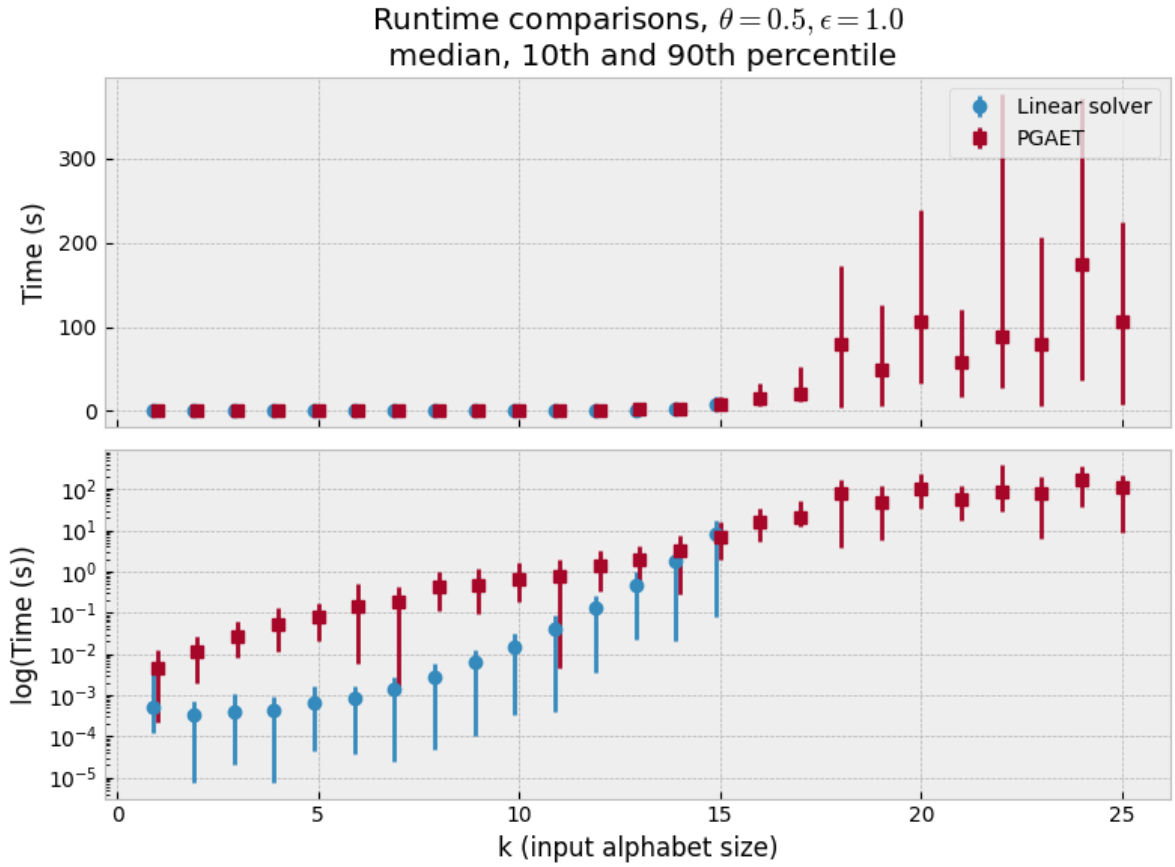

Figure 8: Runtime comparison of the exact linear programme (red) and the gradient ascent approach (blue). Points are median runtimes over 10 experiments, while lower and upper error bars represent the 10th and 90th percentiles, respectively. The linear programme exhibits exponential time complexity, while PGAET retains the polynomial profile.

possible. Steinberger [13] gives an example for the Gaussian location model (chapter 5.2). Here, we apply the algorithm to the Cauchy distribution.

Consider a random variable $X \sim Cauchy(\theta)$. The probability density function is given by

$$p_\theta(x) = \frac{1}{\pi \left[1 + (x - \theta)^2\right]}. \tag{36}$$

The cumulative distribution function is

$$F_\theta(x) = \frac{1}{\pi} \arctan\left(x - \theta\right) + \frac{1}{2} = F_0(x - \theta) \tag{37}$$

and the quantile function, which is the inverse of the cumulative distribution function, is

$$F_\theta^{-1}(\alpha) = \theta + \tan \pi \left(\alpha - \frac{1}{2}\right) = \theta + F_0^{-1}(\alpha). \tag{38}$$

The Fisher information of the location parameter θ can be expressed as $I_\theta = \frac{1}{2}$.

Finding an optimal privatization mechanism for a continuous probability distribution is practically an impossible task. It would be a Markov kernel $Q : \mathcal{G} \times X \to [0, 1]$ acting on a the probability measure P_θ that is generating the private data from a measurable space $(X, \mathcal{F})$ and transforming it to a probability measure QP_θ for the sanitized data on $(Z, \mathcal{G})$, via

$$[QP](dz) = \int_X Q(dz|x)P(dx). \tag{39}$$

The feasible region for the optimization problem would thus be the set $\mathscr{M}_\epsilon(X \to Z)$ of all Markov kernels from (X, F) to (Z, G)) that also satisfy the differential privacy constraint

$$Q(A|x) \leq e^\epsilon \cdot Q(A|x'), \quad \forall A \in \mathcal{G}, x, x' \in X.$$

This feasible set is an infinite-dimensional space, and it is currently not clear how to run an optimization algorithm on it.

To sidestep the problem of infinite dimensionality, we can discretize the probability distribution and apply the aforementioned algorithms. Steinberger [13] (Lemma 5.2) introduces a consistent quantizer for a probability distribution. For clarity, here we write a simplified version.

Definition 4.1. Let $k \in \mathbb{N}$ be an integer denoting the desired number of bins. Define $B_j(\theta_0) = (F^{-1}_{\theta_0}(\frac{j-1}{k}), F^{-1}_{\theta_0}(j/k)]$ and $r_{\theta,\theta_0}(j) = P_\theta(B_j(\theta_0)) = \int_{B_j(\theta_0)} p_\theta(x)dx$, $j = 1, ..., k$. Finally, $T_{k,\theta_0}(x) = \sum_{j=1}^k j\mathbb{1}_{B_j(\theta_0)}(x)$ is a "labeller function" - it assigns a numerical label depending on which bin x falls in to. As $k \to \infty$ this discrete version converges towards the continuous probability distribution.

Now the objective function in the discretized model $(r_{\theta,\theta_0})_{\theta\in\mathbb{R}}$ reduces to

$$I_\theta(QT_{k,\theta_0}) = \sum_{i=1}^k \frac{(\sum_{j=1}^k Q_{ij}\dot{r}_{\theta,\theta_0}(j))^2}{\sum_{j=1}^k Q_{ij} r_{\theta,\theta_0}(j)}, \tag{40}$$

where $\dot{r}_{\theta,\theta_0}(j) = \int_{B_j(\theta_0)} \frac{\partial}{\partial\theta} p_\theta(x)dx = p_\theta \circ F^{-1}_{\theta_0}(\frac{j-1}{k}) - p_\theta \circ F^{-1}_{\theta_0}(\frac{j}{k})$ is the derivative with respect to θ. Thus, if we set $\theta_0 = \theta$, we get $r_{\theta,\theta}(j) = \frac{1}{k}$, $\dot{r}_{\theta,\theta}(j) = p_0 \circ F^{-1}_0(\frac{j-1}{k}) - p_0 \circ F^{-1}_0(\frac{j}{k})$ and

$$I_\theta(QT_{k,\theta}) = \sum_{i=1}^k \frac{(\sum_{j=1}^k Q_{ij}\dot{r}_{\theta,\theta}(j))^2}{\frac{1}{k}\sum_{j=1}^k Q_{ij}}. \tag{41}$$

In Figure 9, we depict Fisher information of the sanitized quantized Cauchy distribution, where the number of bins is the x-axis. We plot a high privacy regime ($\epsilon = 1$) in which the maximum Fisher information is already obtained at two bins. As the number of bins

increases, the Fisher information slowly converges towards this optimum. In this high-privacy regime, both linear solver and PGAET successfully find optimal solutions at all numbers of bins.

On the other hand, the situation in the lower privacy regime ($\epsilon = 5$) is vastly different. Optimal Fisher information is not immediately obtained, and we can observe slower increasing convergence toward the maximum value. PGAET successfully finds optimal solutions in low numbers of bins but quickly diverges from optimal solutions produced by the linear programme as the number of bins increases.

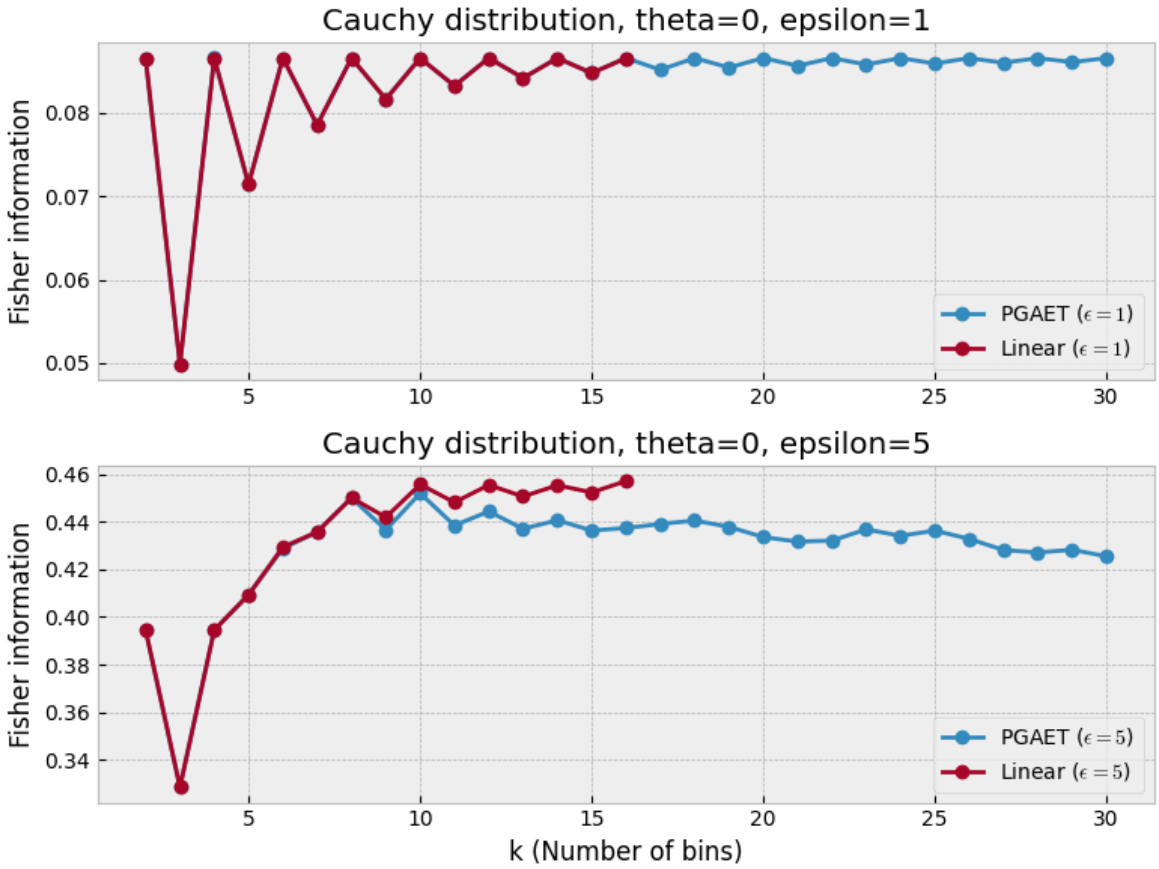

Figure 9: Optimal sanitized Fisher information of the discretized Cauchy distribution. The high privacy ($\epsilon = 1$) case is depicted in the top figure, and the lower privacy ($\epsilon = 5$) regime is in the bottom figure. The optimum value of the Fisher information is immediately obtained in the high privacy regime, while the low privacy regime exhibits slower gradual convergence. PGAET successfully finds optimal values in the high privacy regime while it remains suboptimal in the lower one.

Similarly, we can apply PGAET to the problem of finding efficient privacy mechanisms for sensitive data that come from a Poisson distribution, even though the input alphabet $\mathscr{X}$ is of infinite dimension in that case. The Poisson distribution is given by $p_\lambda(x) = \frac{e^{-\lambda}\lambda^k}{k!}$, while the binomial is $p_\theta^{(m)}(x) = \binom{m}{x}\theta^x(1-\theta)^{1-x}$. For m sufficiently large and θ sufficiently small, we can approximate the binomial distribution with Poisson $p_\theta^{(m)}(x) \approx p_{\lambda=m\theta}(x)$.

Of course, this relationship goes both ways and we can approximate the Poisson with the binomial distribution

$$p_\lambda(x) \approx p^{(m)}_{\theta=\frac{\lambda}{m}}(x).$$

The Fisher information of the Poisson distribution is $I_P(\lambda) = \frac{1}{\lambda}$, while the Fisher information of $Binom(m, \frac{\lambda}{m})$ with respect to λ is

$$I_B(\lambda) = \frac{1}{\lambda} + \frac{1}{m} + \frac{\lambda}{m(m-\lambda)}.$$

In the limit $m \to \infty$ we recover the Fisher information of the Poisson distribution, $\lim_{m\to\infty} I_B(\lambda) = I_P(\lambda)$.

To approximately compute an efficient privacy mechanism for the Poisson problem, we can solve the approximating binomial problem. In Figure 10, we depict the optimal Fisher information of the approximated Poisson distribution ($\lambda = 1$), with the number of trials in the approximating Binomial distribution composing the x-axis. We plot two privacy regimes, a high privacy regime ($\epsilon = 0.5$) and a low privacy regime ($\epsilon = 10$). Both regimes exhibit similar convergence properties. In this problem, PGAET successfully attains optimal solutions in both privacy regimes. The Fisher information of the approximating binomial problem converges from above to its limiting value as the number of trials m increases, in stark contrast to the behaviour of the Cauchy distribution. Additionally, we plot the unsanitized Poisson Fisher information, which is an upper bound to the limiting value for all privatization mechanisms with finite ϵ. While the limiting value of the Fisher information of the approximating binomial problem always remains below that of the unsanitized Poisson distribution, the gap narrows as ϵ increases.

A key limitation of the algorithm developed in this thesis is its reliance on knowing the underlying parameter θ of a parametrized distribution P_θ. In practice, this is highly unrealistic — statistical inference is about estimating θ, not assuming it is given. If θ were already known, the data itself would be of little interest.

Steinberger [13] introduces a procedure that solves this problem. The procedure first estimates θ in a ϵ-differentially private way with an estimator $\hat{\theta}_{n_1}$ based on a subsample $Z_1, ..., Z_{n_1}$ generated by some preliminary privacy mechanism Q_0. The remaining individuals (the second part of the sample) then use $\hat{Q}_{\hat{\theta}_{n_1}}$ to generate $Z_{n_1+1}, ..., Z_n$ in a more efficient ϵ-private way.

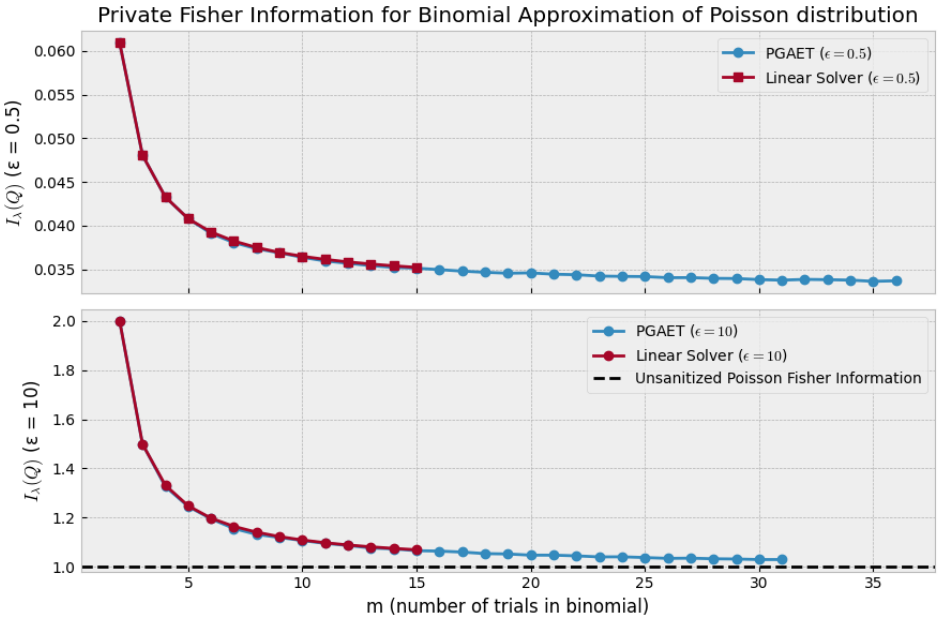

Figure 10: Optimal sanitized Fisher information of Poisson distribution ($\lambda = 1$) where the Poisson distribution is approximated with the Binomial: $p_\lambda(x) \approx p^{(m)}_{\theta=\frac{\lambda}{m}}(x)$. The high privacy regime ($\epsilon = 0.5$) is depicted in the top figure, and the low privacy regime ($\epsilon = 10$) in the bottom figure. Both regimes exhibit similar convergence behaviour. PGAET attains optimal solutions in both privacy regimes.

5 Conclusion

In this thesis, we explore an aspect of ϵ-differentially privacy. Namely, given an ϵ-privacy requirement, we aim to find the privacy mechanism that maximizes Fisher information of the sanitized data, ensuring efficiency in any subsequent statistical estimation procedures:

$$\sup_{Q \in D_\epsilon} I_\theta(Q) = \sup_{Q \in D_\epsilon} \sum_{y \in \mathscr{Y}} \frac{(Q_y \cdot \dot{p_\theta})^2}{Q_y \cdot p_\theta},$$

where D_ϵ is the set of all ϵ-differentially private mechanisms, Q is a privatization mechanism, p_θ is the probability vector of the prior private distribution over the input alphabet $\mathscr{X}$, $\dot{p_\theta}$ vector of derivatives with respect to the parameter θ, and $I_\theta(Q)$ is the Fisher information of the resulting sanitized model produced by the mechanism Q.

Previous work was constrained to low dimensions of the input alphabet, limited by the exponential complexity $O(2^k)$ in the input alphabet dimension of the linear programme, $k = |\mathscr{X}|$. The main limitation is the step of explicit construction of the staircase matrix $S^{(k)}$, which holds all possible patterns the optimal matrix is constructed from. The staircase matrix grows exponentially in size and dictates how many variables and constraints the linear programme uses.

We take a different approach by applying projected gradient ascent with edge traversal (PGAET). Utilizing gradient ascent, we sidestep the exponential complexity as gradient based methods utilize only locally available information of the optimization problem. To enforce ϵ-privacy and stochasticity constraints and remain within the feasible set, we project infeasible intermediate solutions:

$$Q_{t+1} = P_{D_\epsilon}(Q_t + \lambda_t \nabla I_\theta(Q_t)).$$

Additionally, in the case of multiple successive projections, we utilize the previous projections to construct a larger feasible step along the boundary of the feasible set (we call this edge traversal), speeding up the overall process. This approach achieves significantly improved polynomial complexity of $O(k^9)$, making it more scalable than the exponential-time linear programme. The gradient ascent approach successfully finds optimal Fisher information mechanisms in high-privacy regimes while falling slightly suboptimal in low-privacy regimes.

We apply PGAET to the problems of finding efficient differentially private mechanisms for the Poisson and Cauchy distributions, extending previously known findings to higher-dimensional alphabet cardinalities up to $k = 30$.

To run the algorithms, we discretize the Cauchy distribution and apply the optimization algorithms to the approximating discretized distribution. In high privacy regimes, PGAET finds optimal solutions and extends the convergence properties to bigger input alphabet sizes. In low privacy regimes, PGAET falls short of optimality.

Even though the Poisson distribution is discrete, its support is infinite in size, requiring modification before currently known algorithms can be applied. We use the binomial approximation of the Poisson distribution to reduce the input alphabet to a finite size. Surprisingly, PGAET finds optimal solutions in all privacy regimes.

Future research could focus on improving optimization techniques for low-privacy settings and on establishing theoretical guarantees of the optimality of PGAET. One promising direction is the removal of hard feasible set boundaries and replacement with soft barrier functions. These barriers need to be chosen carefully to retain useful properties of the underlying optimization problem, such as convexity. Furthermore, a variety of other promising optimization algorithms may be utilized, such as interior point methods.

To conclude, this work presents a scalable and computationally efficient alternative to extremal mechanisms (linear programme) for optimizing Fisher information in differential privacy settings. Using projected gradient ascent, we apply an algorithm that remains feasible to run in high-dimensional spaces where previous approaches fail. Although

challenges remain in low-privacy regimes, our approach opens avenues for further research and application of privatization mechanisms with larger input alphabets than previously possible.

A Source Code

All supplementary material and complete code can be found at https://github.com/MatejVe/locally_efficient_differential_privacy.

References

[1] John M. Abowd, Matthew J. Schneider, and Lars Vilhuber. The U.S. Census Bureau Adopts Differential Privacy. *Proceedings of the National Academy of Sciences*, 116(20):9794–9803, 2018.

[2] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 111–125. IEEE, 2008.

[3] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[4] Mariana Cunha, Ricardo Mendes, and João P. Vilela. A survey of privacy-preserving mechanisms for heterogeneous data types. *Computer Science Review*, 41:100403, 2021.

[5] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3–es, March 2007.

[6] Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer, New York, 2 edition, 1998.

[7] Harald Cramér. *Mathematical Methods of Statistics*. Princeton Mathematical Series. Princeton University Press, Princeton, 1946.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[9] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[10] Damien Desfontaines. Why differential privacy is awesome. https://desfontain.es/blog/differential-privacy-awesomeness.html, 07 2018. [Online; accessed 31-December-2024].

[11] OpenMined. Local vs global differential privacy. https://blog.openmined.org/basics-local-differential-privacy-vs-global-differential-privacy/, 2021. [Online; Accessed 31-December-2024].

[12] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy, 2015.

[13] Lukas Steinberger. Efficiency in local differential privacy, 2024.

[14] Wikipedia contributors. Gradient descent — Wikipedia, the free encyclopedia, 2024. [Online; accessed 31-December-2024].

[15] Guillaume Garrigos and Robert M. Gower. Handbook of convergence theorems for (stochastic) gradient methods, 2024.

[16] Daniela di Serafino, Valeria Ruggiero, Gerardo Toraldo, and Luca Zanni. On the steplength selection in gradient methods for unconstrained optimization. *Applied Mathematics and Computation*, 318:176–195, 2018. Recent Trends in Numerical Computations: Theory and Algorithms.

[17] Shun ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4):185–196, 1993.

[18] Xi-Lin Li. Preconditioned stochastic gradient descent. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1454–1466, 2018.

[19] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[20] Shiqian Ma. Alternating proximal gradient method for convex minimization. *Journal of Scientific Computing*, 68(2):546–572, 2016.

[21] Paul H. Calamai and Jorge J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987.

[22] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016. To appear.

[23] Florian A. Potra and Stephen J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1):281–302, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

B German Abstract

Lokale differentielle Privatsphäre (LDP) hat sich als führendes Framework zum Schutz persönlicher Informationen etabliert, selbst gegenüber nicht vertrauenswürdigen Datenanalysten. Obwohl viele Mechanismen die ϵ-LDP-Bedingungen erfüllen, variiert ihr Nutzen für die statistische Schätzung erheblich. Um dies zu adressieren, maximieren wir die Fisher-Information innerhalb der Menge der ϵ-LDP-Mechanismen, wodurch die Effizienz nachfolgender statistischer Parameterschätzungen sichergestellt wird. Bisherige Ansätze basierten auf linearer Programmierung mit exponentieller Komplexität $O(2^k)$ (wobei $k = |\mathscr{X}|$ die Kardinalität des Eingabealphabets ist), was die praktische Anwendung auf kleine Eingabealphabete beschränkte. Wir entwickeln einen projizierten Gradientenanstiegsalgorithmus mit Kantendurchlauf (PGAET), der eine polynomiale Komplexität von $O(k^9)$ aufweist und somit Optimierungen in hochdimensionalen Kontexten ermöglicht. Wir demonstrieren seine Effizienz anhand diskreter (Poisson) und kontinuierlicher Verteilungen (Cauchy) durch Diskretisierungs- und Approximationsverfahren. Unsere Ergebnisse erweitern die bisher bekannten Resultate auf Eingabealphabetgrößen bis $k = 30$, was eine deutliche Verbesserung gegenüber früheren Arbeiten darstellt. Diese Arbeit bietet Forschenden eine Methode, um Datenschutzmechanismen mit größeren Eingabealphabeten zu entwickeln, die gleichzeitig effiziente statistische Schätzungen ermöglichen.