Bachelor Thesis BSc^2 Econometrics/Economics

Differentially private convex optimization with piecewise affine objectives: some subgradient approaches

Rijk van Oostenbrugge

Student ID number: 432778

Erasmus School of Economics, Erasmus University Rotterdam

July 7, 2019

Supervisor: prof. dr. S.I. Birbil

Second assessor: drs. U.C. Cakmak

Abstract

In this paper we propose two new private subgradient methods: the bootstrapped private subgradient method and the private weighted average subgradient method. These are compared via a simulation study with the Laplace mechanism, the exponential mechanism, the private subgradient method and random uniform variables. We find that the exponential mechanism and the private weighted average subgradient method perform best depending on the situation, as instances can be generated for both mechanisms in which one performs better than the other and vice versa. However, the exponential mechanism seems to outperform the private weighted subgradient method in most cases treated in this paper.

The views stated in this thesis are those of the author and not necessarily those of Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction

Data has become much more prevalent since the rise of Web 2.0, as the use of social media, online shopping, streaming services, et cetera have increased (Acquisti et al., 2016). The use of these services all generate data, which may be very personal. This added dimension of data is two-edged, as publication of private data is undesirable from an ethical standpoint. However, for example in the context of systems operation, the availability of user data may increase the optimality of the system performance (Han et al., 2014), which leads to a tradeoff between privacy and optimality. Therefore, differential privacy may lead to a more optimal solution in this tradeoff, as data can not be traced back to a single person if differential privacy is ensured.

Statistical disclosure control, being able to disclose statistics of a population, while ensuring the privacy of the individuals within the population, has received interest from the cryptography and database communities (Dwork et al., 2006). Within statistical disclosure there exist two main types of computational models, non-interactive models in which the user can not receive any more information than is given and which allows the owner of the database to delete the data, as no other operations will be performed on the dataset, and interactive models in which the user may query the database, usually via a privacy mechanism.

Han et al. (2014) have applied differential privacy mechanisms in a relatively simple setting i.e. the affine case within a hypercube. In this setting the authors found that the private subgradient mechanism is the least suboptimal mechanism when comparing that mechanism to the exponential and Laplace mechanisms. In this paper, we will replicate Han et al. (2014) and propose some extensions. Firstly, Section 2 will introduce the treated problem formally, which will be followed by Section 3 which will treat the theoretical background of this problem. Next, we discuss the methodology in Section 4 and the results in Section 5. Finally, we conclude the paper in Section 6, which discusses the obtained results.

2 Problem Statement

Differential privacy in statistical databases ensures that adding or removing an item in the database will not affect the outcome of the analysis (Dwork, 2011). Private user data is contained by a set D, which is called a database and $\mathcal{D}$ contains all databases of interest. Suppose we would like to obtain information from $D \in \mathcal{D}$, that information is mapped by a query $q : \mathcal{D} \rightarrow \mathcal{Q}$ for some target domain $\mathcal{Q}$. Making $q(D)$ public may be undesirable, as database D contains private information. Therefore, to preserve privacy we need a mechanism $M : \mathcal{D} \rightarrow \mathcal{Q}$ that approximates q for any given query q. At its core, differential privacy translates the privacy of an individual user into the difference that this user makes in the database. These changes are defined by an adjacency relation, denoted by $Adj(\cdot, \cdot)$, and any two databases that satisfy the adjacency relation are called adjacent databases. The level of differential privacy that a mechanism delivers is quantified with ϵ-differential privacy.

Definition 1. ϵ-differential privacy (Han et al., 2014). A mechanism $M : \mathcal{D} \rightarrow \mathcal{Q}$ preserves ϵ-differential privacy if

$$\mathbb{P}(M(D) \in \mathcal{R}) \leq e^{\epsilon}\mathbb{P}(M(D') \in \mathcal{R}) \tag{1}$$

holds for all $\mathcal{R} \subseteq \mathcal{Q}$ and all pairs of adjacent databases D and D'.

In this context, $\epsilon > 0$ indicates the level of privacy, where a smaller ϵ means a higher level of privacy. Furthermore, we see from this formula that it is necessary that the output of a differentially private mechanism of two adjacent databases is not too different.

The first goal of the paper is to replicate the paper written by Han et al. (2014) after which we try to extend on the paper. In Han et al. (2014), the authors study differentially private convex optimization with piecewise affine objectives, with the goal to minimize $f : \mathbb{R}^d \rightarrow \mathbb{R}$, where f is convex and piecewise affine i.e.

$$f(x) = \max_{i=1,2,...,m} a_i^T x + b_i, \tag{2}$$

for some $\{a_i \in \mathbb{R}^d, b_i \in \mathbb{R}\}_{i=1}^m$, within convex polytope $\mathcal{P}$ thus leading to the optimization problem:

$$\min_x \quad f(x) \qquad s.t. \quad x \in \mathcal{P}. \tag{3}$$

We consider the case in which $\{b_i\}_{i=1}^m$ contains user information, therefore $D = \{b_i\}_{i=1}^m$. The other data, $\{a_i\}_{i=1}^m$ and polytope $\mathcal{P}$ are public and fixed. Two databases $D = \{b_i\}_{i=1}^m$ and $D' = \{b_i'\}_{i=1}^m$ are adjacent when Equation 4 holds.

$$Adj(D, D') \quad \text{if and only if} \quad \max_{i \in \{1,2,...,m\}} \mid b_i - b_i' \mid \leq b_{max} \tag{4}$$

The arising problem is to find a mechanism M which outputs a near optimal solution, while preserving ϵ-differential privacy for adjacent databases D and D' i.e. for mechanism M, Equation 1 must hold for all $\mathcal{R} \in \mathcal{P}$ and all adjacent databases D and D'.

3 Theoretical Background

3.1 Economic relevance

An economic interpretation can be found in the concept of minimax regret in market entry. Assume a company wants to enter a market that contains some competitors. Then the company may apply minimax regret to determine its entry and the level of attributes of the product (Lin and Ng, 2011). Assume a set of attributes X_1,X_2,...,X_n with finite domains, defining a set of outcomes $\mathbf{X} = X_1 \times X_2 \times ... \times X_n$. The utility of a consumer is given by a function $u : \mathbf{X} \rightarrow \mathbb{R}$. In this case we want to minimize the maximum regret. The regret function, the maximum regret function and the minimax regret functions are defined in equations 5, 6 and 7 respectively (Boutilier et al., 2003).

$$R(x, x', \mathcal{U}) = \max_{u \in \mathcal{U}} u(x') - u(x) \tag{5}$$

$$MR(x, \mathcal{U}) = \max_{x'} R(x, x', \mathcal{U}) \tag{6}$$

$$MMR(\mathcal{U}) = \min_{x} MR(x, \mathcal{U}) = \min_{x} \max_{x'} \max_{u \in \mathcal{U}} u(x') - u(x) \tag{7}$$

Now suppose a company wants to compete against a certain product with fixed characteristics x' and that the utility functions of its possible consumers are known. Furthermore, suppose the utility that consumers receive from x' is unknown due to hidden characteristics e.g. marketing and branding. Then in the case that $-u(x)$ is linear and x' is fixed but unknown, we can translate this problem to Equation 2 where $u(x')$ is equivalent to b_i and $-u(x)$ is equivalent to $a_i^T x$. However this can be more generalized as long as $-u(x)$ is convex. This could for example be a square root type function, as the negative square root function is convex and diminishing marginal returns would hold.

3.2 Laplace mechanism

The Laplace mechanism is convenient when the range of query $\mathcal{Q}$ is $\mathbb{R}$. This can be generalized to $\mathbb{R}^d$ (Han et al., 2014): If the sensitivity of query q, as defined in Equation 8, is bounded, then ϵ-differential privacy can be achieved by adding i.i.d. Laplace noise $Lap(d\Delta/\epsilon)$ to each element of q. Where Δ is defined as in Equation 8.

$$\Delta := \max_{D, D'} \left\| q(D) - q(D') \right\|_{\infty} . \tag{8}$$

The Laplace mechanism is very useful due to its ability to deliver all users as much utility as they would have when a mechanism would be tailored for each user individually (Ghosh et al., 2012). Furthermore, Koufogiannis et al. (2015) have studied differential privacy in metric spaces. The authors have proven that the Laplace mechanism returns the lowest mean squared error when compared to other single-dimensional mechanisms that add noise independent of the input.

3.3 Exponential mechanism

The exponential mechanism uses a scoring function $u : \mathcal{Q} \times \mathcal{D} \to \mathbb{R}$. In the case of a minimization problem, the negative objective function can be a candidate for the scoring function. McSherry and Talwar (2007) have shown that the exponential mechanism $M_E(D; u)$ is ϵ-differential private if q is randomly reported following the probability density function in Equation 9

$$\frac{exp(\epsilon u(q, D)/2\Delta_u)}{\int_{q' \in \mathcal{Q}} exp(\epsilon u(q', D)/2\Delta_u) dq'}, \tag{9}$$

where

$$\Delta_u := \max_{x} \max_{D,D':Adj(D,D')} \mid u(x,D) - u(x,D') \mid . \tag{10}$$

Huang and Kannan (2012) have researched the application of the exponential mechanism in mechanism design problems, which aim to maximize social welfare. The authors show that using the exponential mechanism will return a truthful, differentially private solution, which they confirm by showing that the mechanism can be applied to three problems: the combinatorial public project problem, the multi-item auction and the procurement auction for a spanning tree.

3.4 Private subgradient method

Hsu et al. (2014b) have proposed a subgradient method in a differentially private context. A subgradient taken at x_0 in Equation 11 can be a_k. However, to calculate the subgradient access to the private data $\{b_i\}_{i=1}^{m}$ is needed. Therefore, it is required to privatize the computation at each iteration and to limit the number of iterations (Hsu et al., 2014b). The exponential mechanism is able to privatize the calculation of subgradients using the objective function for u, as described in Section 3.3, which is referred to as u_{sub} in the private subgradient method for clarity.

$$a_k^T x_0 + b_k = \max_{i=1,2,\ldots,m} \{a_i^T x_0 + b_i\} \tag{11}$$

We denote the sensitivity of u_{sub} at x_0 as $\Delta_{u_{sub}}(x_0)$, which is defined in Equation 12.

$$\Delta_{u_{sub}}(x_0) := \max_{i \in \{1,2,\ldots,m\}} \max_{D,D'} \mid u_{sub}(i;x_0,D) - u_{sub}(i;x_0,D') \mid = b_{max}. \tag{12}$$

Two algorithms are needed for the calculation of the private subgradient solution, Algorithm 1 calculates the ϵ-differential private subgradient. Algorithm 2 shows that that the mechanism is ϵ-differential private. (ϵ/k)-private subgradients are calculated for k iterations due to sequential composition, in this case that implies that when k ϵ-private queries are performed, the whole mechanism will be $k\epsilon$-private (McSherry and Talwar, 2007).

Algorithm 1 ϵ-differentially private subgradient (Han et al., 2014).

```
Choose scoring function $u : \{1,2,...,m\} \to \mathbb{R}$ as $u_{sub}(i;x_0,D) = a_i^T x_0 + b_i$.
Select the index $i^*$ using the exponential mechanism: $\mathbb{P}(i^* = i) \propto \exp(\epsilon u_{sub}(i;x_0)/2b_{max}$
Output $a_{i^*}$
```

3.5 Bootstrapped private subgradient method

We propose a method in which we adapt Algorithm 1 to return a bootstrapped private subgradient, which is shown in Algorithm 3. This algorithm returns an $l\epsilon$-differentially private subgradient due to sequential

Algorithm 2 ϵ-differentially private subgradient method (Han et al., 2014).

```
Set the number of iterations $k$, step sizes $\{\alpha_i\}_{i=1}^{k}$ and $x^{(1)} \in \mathcal{P}$.
for  $i = 1, 2, ..., k$ do
    Obtain an $(\epsilon/k)$-private subgradient $g^{(i)}$ using Algorithm 1
    Update $x^{(i+1)} := x^{(i)} - \alpha_i g^{(i)}$.
Output $x^{(k+1)}$ as the $\epsilon$-differentially private solution.
```

composition, as the average of l subgradients is taken. Therefore, a slight adjustment in Algorithm 2 is needed, as an $\epsilon/(lk)$-private subgradient will be required to return an ϵ-differentially private solution. Moreover, the bootstrapped private subgradient method is equivalent to the private subgradient method when $l = 1$.

Algorithm 3 Bootstrapped $l\epsilon$-differentially private subgradient (Han et al., 2014).

```
Choose scoring function $u : \{1, 2, ..., m\} \to \mathbb{R}$ as $u_{sub}(i; x_0, D) = a_i^T x_0 + b_i$.
for  $i = 1, 2, ..., l$ do
    Select the index $i^*$ using the exponential mechanism: $\mathbb{P}(i^* = i) \propto \exp(\epsilon u_{sub}(i; x_0)/2b_{max}$
    Save $a_{i^*}$ in a list containing all selected values of $a_{i^*}$
Set $a_{bootstrap}$ as the average vector from the list containing all $a_{i^*}$
Output $a_{bootstrap}$
```

3.6 Private weighted average subgradient method

We propose a method in which we adapt Algorithm 1 to use more information for Algorithm 2 by returning the weighted average of the subgradients at a point x. In this case, the weights are determined by the exponential mechanism. This will lead to a less private subgradient algorithm, as information for every b_i is released. This implies a factor m which needs to be accounted for in Algorithm 2. Therefore, $\epsilon/(km)$-differentially private weighted average subgradients are required to be computed by Algorithm 4 to return an ϵ-differentially private solution to the problem.

Algorithm 4 $m\epsilon$-differentially private weighted average subgradient.

```
Choose scoring function $u : \{1, 2, ..., m\} \to \mathbb{R}$ as $u_{sub}(i; x_0, D) = a_i^T x_0 + b_i$.
Give each index $j$ a weight using the exponential mechanism: $\mathbb{P}(j = i) \propto \exp(\epsilon u_{sub}(i; x_0)/2b_{max}$
Define $\overline{a} := \sum_{j=1}^{m} \mathbb{P}(j = i) a_j$
Output $\overline{a}$
```

3.7 Selection of ϵ

Hsu et al. (2014a) look at the setting of ϵ from an economic perspective and model the conflicting objectives of the data analyst, who wants the most accurate data for better inferences, versus the participant, who may not necessarily want all their data included into the analysis. Furthermore, the ranges under which ϵ is tested in algorithms seem to be rather wide in the literature, with values of ϵ ranging from 0.01 to 7 (Hsu et al., 2014a).

In practice these values seem to vary more for example, Apple seems to apply higher levels of ϵ, with ϵ ranging from 2 to 8, however as some data may be retrieved more often this will lead to higher levels of ϵ due to sequential composition, as explained in Section 3.4, whereas Google applies an ϵ of 2 for uploaded data with a maximum privacy budget of 8-9 over the user's lifetime (Erlingsson et al., 2014). This is closer to the aforementioned range under which ϵ is studied, however the privacy levels still differ relatively much due to the exponential nature of ϵ.

4 Methodology

The paper will follow the methodology of the simulation study of Han et al. (2014), in which the authors simulate the data and apply privacy mechanisms with $\epsilon = 0.1$, after which they compare the levels of suboptimality of these privacy mechanisms under varying levels of c and m. The privacy mechanisms which are compared are the Laplace mechanism, the exponential mechanism and the private subgradient method. We deviate from Han et al. (2014) by varying the levels of ϵ and adding the bootstrapped subgradient method, the private weighted average subgradient method and uniform random drawn variables to the comparison. Moreover, we will look at some special cases to gain more insight in the compared mechanisms.

4.1 Data generation

The data $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ are generated from i.i.d. Gaussian distributions with mean 0 and standard deviation of 1, which are constrained within a d-dimensional hypercube, centered at the origin $\mathcal{P} = \{x : -c \preccurlyeq x \preccurlyeq c\}$. In our base case the values for c, m, ϵ, d are respectively 2, 10, 0.1, and 2, the effect of all variables except for d will be looked at in Section 5. For these cases we run the mechanisms 1000 times each for 100 sets of $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$. For the special cases, the data $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ are manipulated to gain some insights in a somewhat simpler setting. In the parallel case $\{a_i\}_{i=1}^m$ is a vector of ones, while $\{b_i\}_{i=1}^m$ are generated as in the regular case. However the mechanisms will be run for 100 times each for 50 sets of$\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$. In the cases with one vector being perpendicular to the other parallel vectors and with half the vectors being perpendicular we will follow the methodology of the parallel case, where first the parallel vectors are generated and then appended with a vector of negative ones.

4.2 Solving the problem without privacy constraints

The problem without the privacy constraints is solved using the SLSQP method in the minimize function of SciPy.optimize (Jones et al., 2001) in Python, due to its ability to minimize functions with constraints.

4.3 Laplace mechanism

Laplace noise is implemented by generating $\{w_i\}_{i=1}^{d}$ from the exponential distribution, after which $\overline{w} = \sum_{i=1}^{m} w_i$, followed by generating the direction $\hat{e}$ from the normalization of a d-dimensional Gaussian drawing with mean 0 and standard deviation of 1. The noise will be added to the solution and the data set respectively.

4.4 Exponential mechanism

The exponential mechanism is implemented by approximating the integral in Equation 9 using the Metropolis algorithm with a multivariate Gaussian proposal distribution with mean 0 and covariance matrix $\Sigma = \eta c I_{d\times d}$, where $\eta = 0.1$ and $I_{d\times d}$ is a $d \times d$ identity matrix. In the Metropolis algorithm we assume that the Monte Carlo Markov Chain will be stationary after 5000 iterations, after which the following values will be returned, as this value should be distributed according to the distribution. The negative objective function will be used as the scoring function u.

4.5 Private subgradient method

The private subgradient method will run for 100 iterations, following the methodology of Han et al. (2014). α_i is set at $\frac{1}{i^{1.25}}$, which satisfies the step size rules (Boyd, 2003). The level of α_i has been determined by grid search, after selecting a few values and selecting the value which returned the lowest average objective value, as can be seen in Appendix A from Table 1.

4.6 Bootstrapped private subgradient method

We will run the bootstrapped private subgradient for 100 iterations and set α_i at $\frac{1}{i^{1.25}}$, as the method is similar to the private subgradient method. Furthermore, we will run the method for l set at 2, 4 and 10, however only results for $l = 10$ will be shown in Section 5, as that number of bootstrapped values generally outperforms the other two as can be seen from Appendix B.

4.7 Private weighted average subgradient method

The private weighted average subgradient method will run for 100 iterations, similar to the private subgradient method. α_i is set at $\frac{1}{i^{1.4}}$, which satisfies the step size rules (Boyd, 2003). The level of α_i has been determined by grid search, after selecting a few values and selecting the value which returned the lowest average objective value, as can be seen in Appendix A from Table 2.

4.8 Uniform distributed random variable

To compare the performance of the privacy mechanisms we add the uniform distributed random variable to the comparison, as these variables do not use any information from the database except for $\mathcal{P}$. For each dimension a random uniform number is drawn from $[-c, c)$. The half-open interval is caused by the properties of generating random uniform numbers with NumPy (Jones et al., 2001), however this should not have too much of an effect on the results, as the probability of a random variable at a specific value is equal to 0 in the case of continuous distributions.

5 Results

5.1 Varying levels of c

When c increases with fixed $\epsilon = 0.1$, $m = 10$ and $d = 2$, we observe from Figure 1 that the optimal value decreases, which makes sense, as the optimal value is more likely to be within the feasible region. Furthermore, it can be noticed that the Laplace mechanism on both the solution and on the data increases rather sharply. Moreover, the other methods increase in c at a somewhat similar rate, slower than both applications of the Laplace mechanism. However, the private weighted average subgradient method seems to perform best in this setting, as it is the least suboptimal method over the tested values of c, however it is very closely followed by the bootstrapped private subgradient method.

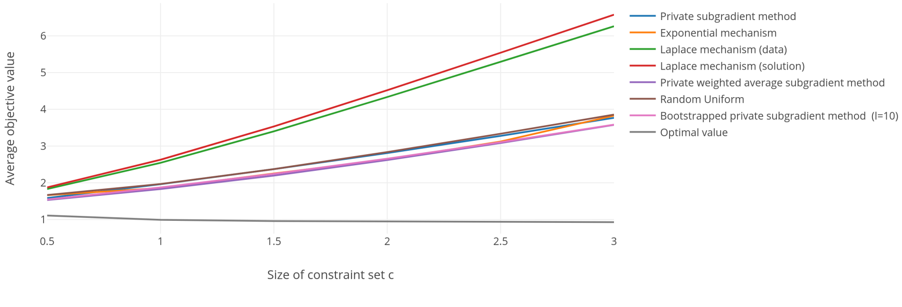

Figure 1: The average objective value for different values of c

5.2 Varying levels of m

We observe in Figure 2 that the average objective values increase for all mechanisms as m increases. This result makes intuitive sense, as more affine functions are being added to the problem, which could increase the objective values. We see that all mechanisms show similar increases, where the exponential mechanism outperforms all other mechanisms once m exceeds the base case of $m = 10$. Furthermore, it can be seen that the private weighted average subgradient method outperforms the other subgradient methods and that the Laplace mechanisms perform worst. Moreover, it should be noticed that the private weighted average subgradient method shows the best performance of the subgradient methods.

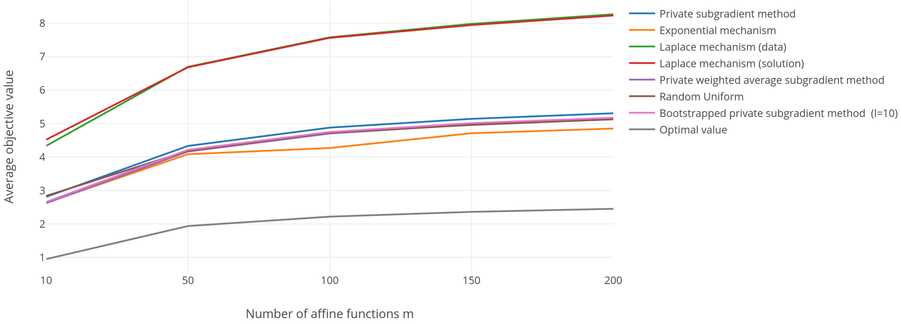

Figure 2: The average objective value for different values of m

5.3 Varying levels of ϵ

When a higher ϵ is set, then the average objective values of most mechanisms monotonically decrease in ϵ, as can be seen from Figure 3, except for the exponential mechanism, which shows irregularities between values of 0.001 and 0.1. The private subgradient method shows a slight decrease, while the Laplace mechanisms seem to decrease fastest once epsilon exceeds values of 0.1. Furthermore, we see that the private weighted average subgradient method and the bootstrapped private subgradient method are relatively invariant to the change of epsilon in this range of values. Lastly, we see that the exponential mechanism is the least suboptimal mechanism in most cases, except for the cases where $\epsilon = 0.001$ and 0.1 in which the private weighted average subgradient method outperforms the exponential mechanism.

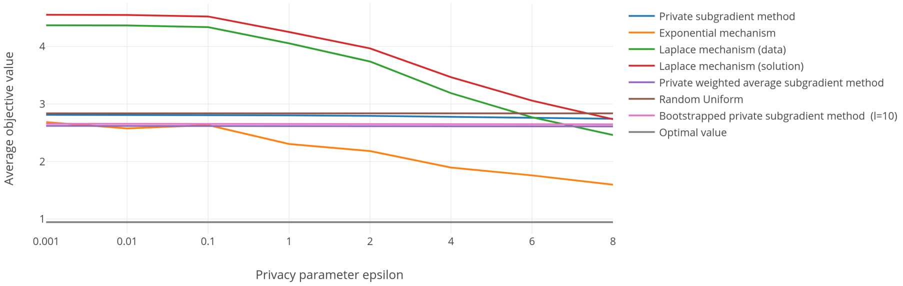

Figure 3: The average objective value for different values of ϵ

5.4 Parallel slope case

In the parallel case, we observe from Figures 4, 5 and 6 that the mechanisms perform in roughly the same patterns. The uniform case generally performs worst, while the Laplace mechanism on the data performs best. Please note that the lines for the Laplace mechanism coincide with the lines for the optimal value, as noise is added to the intercepts, after which that problem will be optimized, leading to the optimal solution, which is the corner solution due to all functions being parallel. Furthermore, the lines for the bootstrapped private subgradient method and the private subgradient method coincide, due to these mechanisms having the same step sizes and these algorithms picking the same subgradient, as the subgradients are all equal.

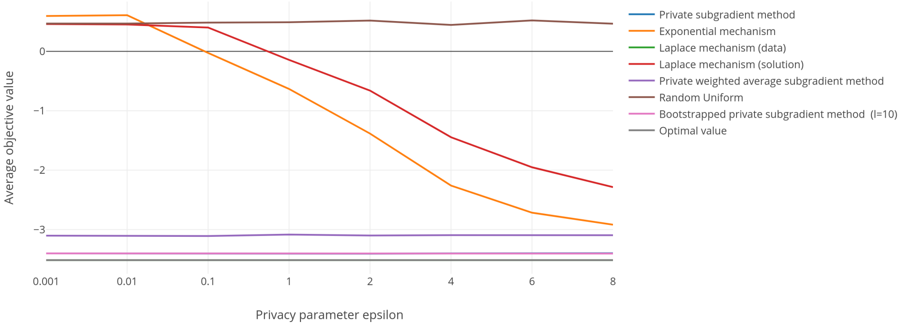

Figure 4: The average objective value for different values of ϵ with m=2

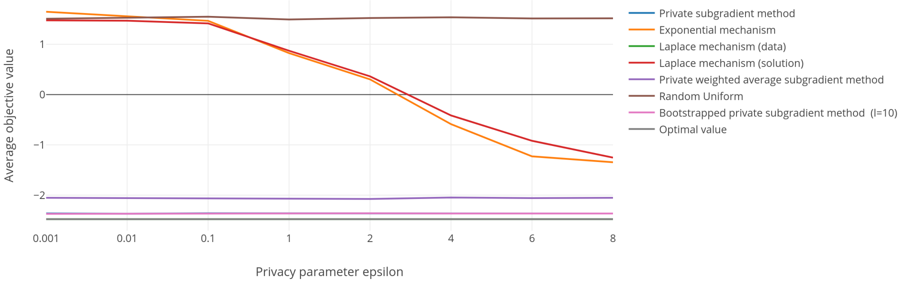

Figure 5: The average objective value for different values of ϵ with m=10

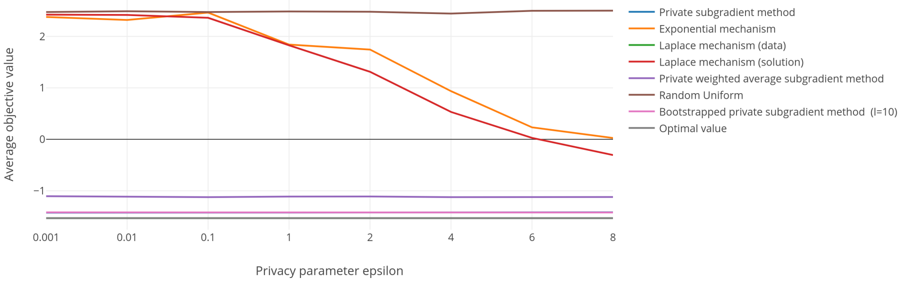

Figure 6: The average objective value for different values of ϵ with m=100

5.5 Parallel slopes with 1 perpendicular slope case

In this case all a_i except for one are set at 1, while the leftover a_i is set at -1. We observe in this case from Figures 7, 8 and 9 that the exponential mechanism is generally the best performing mechanism, while the random uniform variable seems to perform rather well, as it is not consistently outperformed by other mechanisms other than the exponential mechanism when m is not equal to 2. The subgradient methods are performing relatively bad in this setting. This could be attributed to the amount of b_i, which could throw off the probabilities in the algorithms, possibly leading to inaccurate choices of subgradients. Also, the subgradient mechanisms are relatively invariant to the change in ϵ, similar to the previous cases. Furthermore, we observe again that the Laplace mechanisms show relatively much improvement as ϵ increases. Lastly, we see that the exponential mechanism shows an irregularity between values of ϵ of 0.001 and 0.1.

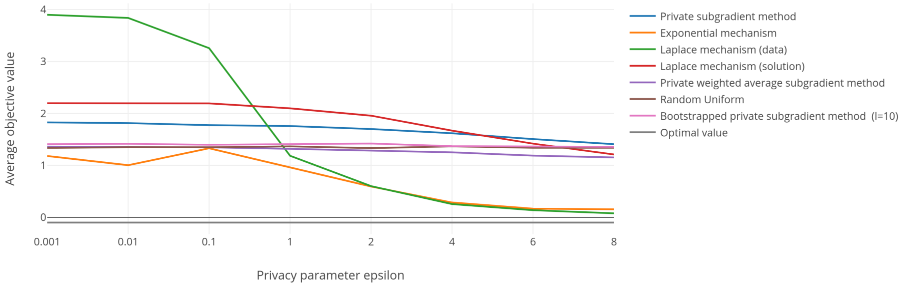

Figure 7: The average objective value for different values of ϵ with m=2

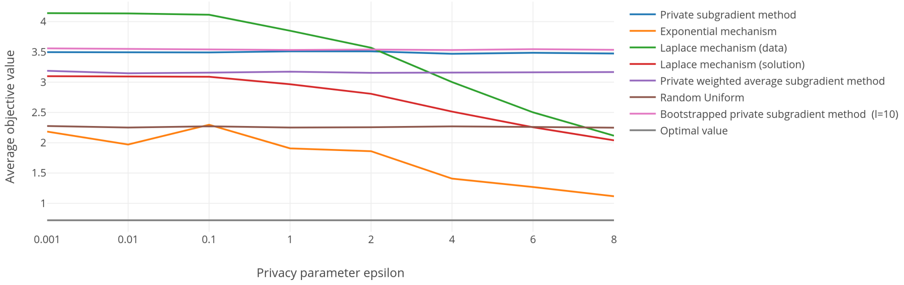

Figure 8: The average objective value for different values of ϵ with m=10

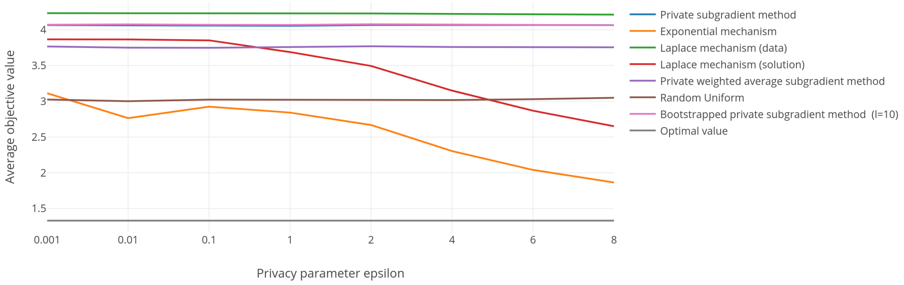

Figure 9: The average objective value for different values of ϵ with m=100

5.6 Parallel slopes mixed with perpendicular slopes case

In this case half of the a_i are set to 1 with the other half of the a_i are set at -1. We observe from Figure 10, 11 and 12 that the exponential mechanism is the least suboptimal mechanism. The exceptions are when ϵ is larger than 2 and $m = 2$ in which the Laplace mechanism on the data is less suboptimal and in the case where $m = 100$ and $\epsilon = 0.001$ in which both the uniform random variables and the private subgradient weighted average method slightly outperform the exponential mechanism. Furthermore, the private subgradient method performs worst of all subgradient methods, while the private weighted average subgradient method performs best of these subgradient methods.

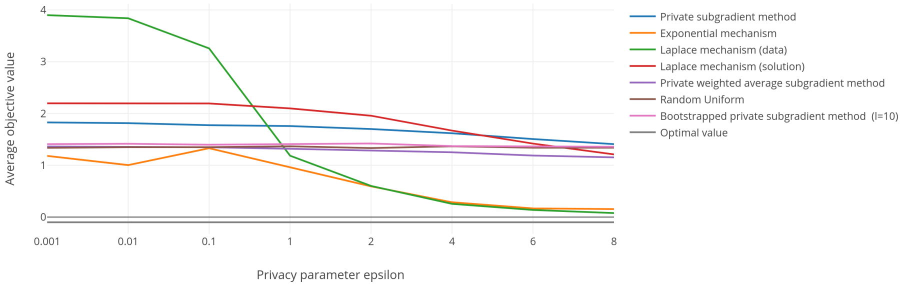

Figure 10: The average objective value for different values of ϵ with m=2

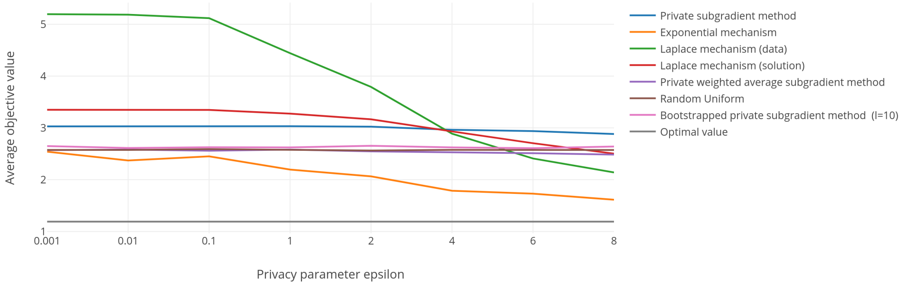

Figure 11: The average objective value for different values of ϵ with m=10

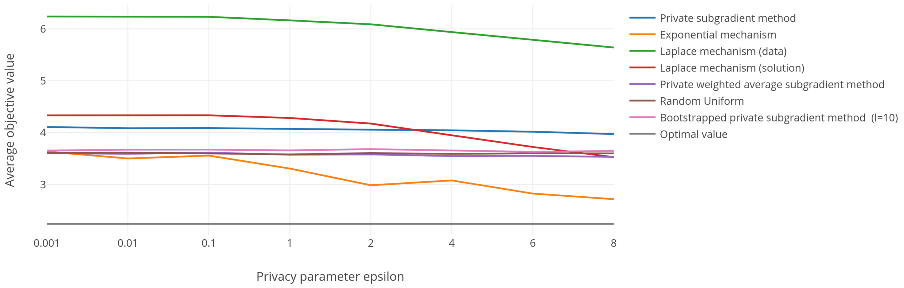

Figure 12: The average objective value for different values of ϵ with m=100

6 Discussion

In this paper we propose two new subgradient methods and compare these with the methods in Han et al. (2014) and uniform random variables by simulating data. We observe that the exponential mechanism works least suboptimal in most cases. However, cases can be devised in which this mechanism performs worse compared to the other mechanisms, for example in the case with parallel a_i. Additionally, we see that the private weighted average subgradient method and the Laplace mechanism on the data perform best in some cases, however from the cases in this paper, we see that this depends on the setting.

Future research may look into generalizing the problem statement, for example by generalizing $f(x)$ to other convex functions or by changing the shape, size and position of $\mathcal{P}$. Furthermore, other mechanisms could be taken into account of this comparison or the mechanisms treated in this could be refined, i.e. an optimization of the parameter l in the bootstrapped private subgradient method. Likewise, these mechanisms could be tested in different instances of $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ to gain more insight in the performance of these mechanisms. Moreover, the hyperparameters α_i and k could possibly be more optimized, as we may have reached a local minimum in our grid searches. Also, some type of cross validation could be applied to ensure the generalizability of the methods. For example in the parallel case, the subgradient methods should be able to reach the optimal solution, as there is effectively one subgradient, which leads to the optimal solution. In addition, there could be more research on the behavior of the exponential mechanism with respect to ϵ, as the irregularity seems to be caused by ϵ being on both sides of the division in Equation 9, however this might require some formal analysis to see if this could be manipulated in a favourable manner.

7 Acknowledgements

I would like to thank prof. dr. S.I. Birbil for introducing me to the field of differential privacy, guiding me throughout the writing of this thesis and for the helpful discussions we had with regards to this topic.

References

Acquisti, A., Taylor, C., and Wagman, L. (2016). The economics of privacy. *Journal of Economic Literature*, 54(2):442–92.

Apple. Differential privacy overview. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf. Accessed: 2019-06-17.

Boutilier, C., Patrascu, R., Poupart, P., and Schuurmans, D. (2003). Constraint-based optimization with the minimax decision criterion. In *International Conference on Principles and Practice of Constraint Programming*, pages 168–182. Springer.

Boyd, S. (2003). Subgradient methods.

Dwork, C. (2011). Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM.

Ghosh, A., Roughgarden, T., and Sundararajan, M. (2012). Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693.

Han, S., Topcu, U., and Pappas, G. J. (2014). Differentially private convex optimization with piecewise affine objectives. In *53rd IEEE Conference on Decision and Control*, pages 2160–2166. IEEE.

Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., and Roth, A. (2014a). Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410. IEEE.

Hsu, J., Roth, A., Roughgarden, T., and Ullman, J. (2014b). Privately solving linear programs. In *International Colloquium on Automata, Languages, and Programming*, pages 612–624. Springer.

Huang, Z. and Kannan, S. (2012). The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 140–149. IEEE.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.

Koufogiannis, F., Han, S., and Pappas, G. J. (2015). Optimality of the laplace mechanism in differential privacy. *arXiv preprint arXiv:1504.00065*.

Lin, J. and Ng, T. S. (2011). Robust multi-market newsvendor models with interval demand data. *European Journal of Operational Research*, 212(2):361–373.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy.

A Step size α_i grid search

Table 1: Average objective values in a grid search over α_i in the private subgradient method

α_i	Average objective value
0.51	3.216862
0.55	3.193634
1	2.843963
1.1	2.819991
1.15	2.814023
1.2	2.810908
1.225	2.810166
1.25	2.809884
1.275	2.809988
1.3	2.810431

Table 2: Average objective values in a grid search over α_i in the private weighted average subgradient method

α_i	Average objective value
0.51	3.143295
1.2	2.632173
1.25	2.624945
1.3	2.620758
1.35	2.618912
1.4	2.61876
1.45	2.619904
1.5	2.622033

B Selection of l in the bootstrapped private subgradient method

Table 3: Average objective values of the bootstrapped private subgradient method for different values of c

	Average objective value		
c	$l = 2$	$l = 4$	$l = 10$
0.5	1.548087	1.549246313	1.546318958
1	1.90986	1.896081277	1.873503252
1.5	2.314152	2.279665948	2.244296388
2	2.739694	2.69421348	2.656434942
2.5	3.189795	3.141127662	3.104874732
3	3.663412	3.615561916	3.581757256

Table 4: Average objective values of the bootstrapped private subgradient method for different values of m

	Average objective value		
m	$l = 2$	$l = 4$	$l = 10$
10	2.739694	2.694213	2.656435
50	4.329233	4.261848	4.216263
100	4.886827	4.799117	4.748728
150	5.161039	5.080252	5.01459
200	5.315335	5.234949	5.175692

Table 5: Average objective values of the bootstrapped private subgradient method for different values of ϵ

	Average objective value		
ϵ	$l = 2$	$l = 4$	$l = 10$
0.001	2.740295	2.69451	2.656573
0.01	2.740199	2.694461	2.65656
0.1	2.739694	2.694213	2.656435
1	2.734714	2.691475	2.655392
2	2.728875	2.688136	2.654064
4	2.716863	2.682028	2.651582
6	2.705177	2.675904	2.648963
8	2.693812	2.669973	2.646622

C Explanation of programming code

In these programs we simulate the problems that are discussed in the paper. In Complete Thesis code.ipynb we run the 'regular' case in which a_i and b_i are generated via standard Gaussian distributions. In Parallel Complete Thesis code.ipynb we run the case where all a_i carry the value 1. In 1 perpendicular Complete Thesis code.ipynb we run the case where all a_i except for one carry the value 1, while the last a_i carries the value -1. In Half perpendicular Complete Thesis code.ipynb we run the case where half of the a_i carry the value 1 while the other half carry the value -1. These programs have the same structure and therefore are explained all at once. All parameters can be changed where they are defined. Changing the variable dimensions takes somewhat more effort, as the amount of bound vectors in bnds needs to be equal to the amount of dimensions. Furthermore, the variable alpha needs to be changed inside the definition of the method. The average objective values that will be returned by the program are in the following order: Private subgradient method, Exponential mechanism, Laplace mechanism (data), Laplace mechanism (solution), Private weighted average subgradient method, Bootstrapped private subgradient method (l=2), Random Uniform, Bootstrapped private subgradient method (l=4), Bootstrapped private subgradient method (l=10). This is followed by the average optimal value. Furthermore, the average standard deviation will be returned as well in the same order as the average objective values.