King Abdullah University
of Science and Technology

Homework 1

Konstantin Burlachenko

Assignment in Contemporary Topics in Computer Security.
Last updated on September 25, 2021

Contents

1 Task 1

If $Z \sim \text{Lap}(\lambda) \implies Z \sim f(z) = \frac{1}{2\lambda}\exp\left(-\frac{|z|}{\lambda}\right)$, where $f(z)$ is a probability distribution function (pdf) of random variable (r.v.) Z.

$$\mathbf{E}(Z^2) = \int_{-\infty}^{+\infty} z^2 \cdot f(z)dz = \int_{-\infty}^{+\infty} z^2 \cdot \frac{1}{2\lambda}\exp\left(-\frac{|z|}{\lambda}\right)dz = |\text{because function is even}| =$$
$$2\int_{0}^{+\infty} z^2 \cdot \frac{1}{2\lambda}\exp\left(-\frac{|z|}{\lambda}\right)dz = \int_{0}^{+\infty} z^2 \cdot \frac{1}{\lambda}\exp\left(-\frac{z}{\lambda}\right)dz =$$
$$\int_{0}^{+\infty} \left(\frac{z}{\lambda}\right)^2 \cdot \lambda\exp\left(-\frac{z}{\lambda}\right)dz = |\text{replace } t = z/\lambda, \lambda > 0| =$$
$$\int_{t=0/\lambda}^{t=+\infty/\lambda} t^2 \cdot \lambda\exp(-t)d(t\lambda) = \int_{t=0/\lambda}^{t=+\infty/\lambda} t^2 \cdot \lambda^2\exp(-t)dt =$$
$$-\int_{t=0}^{t=+\infty} t^2 \cdot \lambda^2 d(\exp(-t)) = \int_{+\infty}^{0} t^2 \cdot \lambda^2 d(\exp(-t)) =$$
$$\lambda^2\exp(-t)t^2|_{t=+\infty}^{t=0} - \lambda^2\int_{t=+\infty}^{t=0}\exp(-t)d(t^2) = 2\lambda^2\int_{0}^{+\infty}\exp(-t)dt = -2\lambda^2\exp(-t)|_{t=0}^{+\infty} = 2\lambda^2$$

And finally take square root we can obtain: $\sqrt{E(Z^2)} = \sqrt{2}\lambda$

Comment: In the middle of derivation we have used implicitly the fact that: $\lim\limits_{t\to\infty} exp(-t)\cdot t^2 = \lim\limits_{t\to\infty}\frac{t^2}{exp(t)} = \lim\limits_{t\to\infty}\frac{2}{exp(t)} = 0$. Here in the prove we applying L'Hopialle rule for that limit, and take derivative twice of numerator and denominator, and obtaining value for limit equal to 0.

Now let's prove that $z \sim \text{Lap}(\lambda) \implies \mathbf{P}(z > \lambda t) \leq \exp(-t)$.

By definition $\lambda > 0$. First of all, if $t \leq 0 \implies 1 \leq \exp(-t)$, but by definition of probability measure $\mathbf{P}(z > \lambda t) \leq 1$. But $\mathbf{P}(z > \lambda t) \leq 1 \leq \exp(-t) \implies \mathbf{P}(z > \lambda t) \leq \exp(-t), \forall t \leq 0$.

Now let's consider the case $t > 0$ and use p.d.f. of Z to measure probability of the event $\{z > \lambda t\}$:

$$\mathbf{P}(z > \lambda t) = \int_{\lambda t}^{+\infty} f(z)dz = \frac{1}{2\lambda}\int_{\lambda t}^{\infty}\exp\left(-\frac{|z|}{\lambda}\right)dz = |\text{beause } \forall z \in (\lambda t, +\infty], z > 0| =$$
$$\frac{1}{2\lambda}\int_{\lambda t}^{\infty}\exp\left(-\frac{z}{\lambda}\right)dz = -\frac{1}{2}\exp(-z/\lambda)|_{z=\lambda t}^{+\infty} = 0 - (-1/2\exp(-t)) = \frac{1}{2}\exp(-t)$$

So we have: $\mathbf{P}(z > \lambda t) = \frac{1}{2}\exp(-t) \implies \mathbf{P}(z > \lambda t) \leq \exp(-t)$

Comment The proved statement $\mathbf{P}(z > \lambda t) = \frac{1}{2}\exp(-t)$ can be used to prove slightly another bound. Because f is even (i.e. $f(z) = f(-z), \forall z \in \mathbb{R}$) we have: $\mathbf{P}(|z| > \lambda t) = 2 \cdot \int_{\lambda t}^{+\infty} f(z)dz = 2 \cdot \frac{1}{2}\exp(-t) \implies \mathbf{P}(|z| > \lambda t) = \exp(-t)$.

2 Task 2

The important concept of the L_1 *global sensitivity* of a function $f : \mathcal{X}^n \to \mathbb{R}^k$ is defined in Lecture 3 as:

$$GS_f \coloneqq \max_{D \sim D'} \|f(D) - f(D')\|_1$$

Where $D \sim D'$ means that datasets are different in only one data point. Such D, D' are called also as *neighbouring datasets*. This definition can be motivated by upper bounding how the function can change in the worst case by modifying a single data element.

(a) The high dimensional mean: $f(D) = \frac{1}{n}\sum_{i=1}^{n} x_i, \mathcal{X} = \{v \in \mathbb{R}^d : \|v\|_1 \leq 1\}$. The maximum possible change in one datapoint i can be bound as $\|x_i - x_i'\|_1 = \|(x_i - 0) + (0 - x_i')\|_1 \leq \|x_i\|_1 + \|x_j\|_1 = 2$, this expression does not depend on i.

$$GS_f \coloneqq \max_{D \sim D'} \|f(D) - f(D')\|_1 = \frac{1}{n} \max_{D \sim D'} \| \sum_{j=1}^{n} (x_j - x_j')\|_1 = \frac{1}{n} \max_{D \sim D'} \|(x_i - x_i') + 0\|_1 = \frac{2}{n}$$

In the middle of derivation we used that for $j \neq i$ we have $x_j = x_j'$. The bound is attained when $x_i = [1, 0, \ldots, 0]$ to $x_i' = [-1, 0, \ldots, 0]$ and $x_{j,j\neq i} = x_{j,j\neq i}$ are arbitrarily. Without any restriction to D, D' this bound $\frac{2}{n}$ is not improvable for single data point and in that sense the bound is tight.

(b) The unnormalized covariance matrix:
$f(D) = \sum_{i=1}^{n} x_i x_i^\top, \mathcal{X} = \{v \in \mathbb{R}^d : \|v\|_1 \leq 1\}$,

$$GS_f \coloneqq \max_{D \sim D'} \|f(D) - f(D')\|_1 \coloneqq \max_{D \sim D'} \|vec(\sum_{j=1}^{n} x_j x_j^\top - \sum_{j=1}^{n} x_j' x_j'^\top)\|_1 = \max_{D \sim D'} \sum_{r=1,c=1}^{r=d,c=d} \left| \left[\sum_{j=1}^{n} \left(x_j x_j^\top - x_j' x_j'^\top \right) \right]_{rc} \right|$$

Where in last equation:

1. Operator *vec* unroll matrix into column vector be definition from the task.
2. The L_1 norm of a matrix in vectorized form can be considered as sum of absolute values of the matrix, and becuase summation is commutative and associative we can perform summation in any order

During considering datasets D, D' we know that they are different in exactly one datapoint, let assume index of that point $i \in \{1, \ldots, n\}$. Because (1) for $j \neq i$ we know that $x_j = x_j'$; (2) for outer product $(v \cdot u)_{i,j} = v_i \cdot u_j, \forall v, d \in \mathbb{R}$; (3) Also we have constraint

$\|x_i\|_1 \leq 1$, and so we can look into GS_f as:

$$GS_f = \max_{D\sim D'} \sum_{r,c} |(x_i)_r(x_i)_c + (-(x'_i)_r)(x'_i)_c|] \leq \max_{D\sim D'} \sum_{r,c} |(x_i)_r||(x_i)_c| + |(x'_i)_r||(x'_i)_c| =$$

$$\max_{D\sim D'} \sum_r \sum_c |(x_i)_r||(x_i)_c| + \sum_r \sum_c |(x'_i)_r||(x'_i)_c| =$$

$$\max_{D\sim D'} \sum_r \left(|(x_i)_r| \left(\sum_c |(x_i)_c| \right) \right) + \sum_r \left(|(x'_i)_r| \left(\sum_c |(x'_i)_c| \right) \right) \leq$$

$$\max_{D\sim D'} \sum_r \left(|(x_i)_r| \cdot 1 \right) + \sum_r \left(|(x'_i)_r| \cdot 1 \right) = 2$$

For example for vector $x_i = [1, 0, \ldots, 0]$ the result outer product $x_i x_i^\top$ is matrix filled with zeros for all elements, except element $[x_i x_i^\top]_{11} = 1$. For vector $x'_i = [0, 1, 0, \ldots, 0]$ the result outer product $x'_i x'^\top_i$ is zero matrix, except element $[x'_i x'^\top_i]_{22} = 1$. Result of vectorization will be vector $[1, -1, 0, \ldots, 0]^T$ with L_1 norm of that vector equal 2. This examples demonstrates that derived bound is tight and not improvable.

(c) $f(D) = median(x_1, \ldots, x_n), \mathcal{X} = [0, 1]$

$$GS_f \coloneqq \max_{D\sim D'} \|f(D) - f(D')\|_1 = \max_{D\sim D'} |median(D) - median(D')|$$

Because $D \in [0, 1]^n, D' \in [0, 1]^n$ the maximum possible attained $median(D)$ is 1, and minimum possible attained median $median(D')$ is 0. This is because $x_i \in [0, 1]$ and median have to be inside $[\inf x_i, \sup x_i] \subset [0, 1]$. Without constraint $D \sim D'$ the maximum $|f(D) - f(D')| = 1$. And this is attained for $D = [1, 1, \ldots, 1]$, and $D' = [0, 0, \ldots, 0]$.

Extra constraint $D \sim D'$ can only decrease value $\max |f(D) - f(D')|$, because addition constraints can only decrease the feasible set of D, D' for that optimization problem.

But we can find D, D' such that $D \sim D'$ and $median(D) = 1, median(D') = 0$. For example for $n = 5$ we have $median([0, 0, 0, 1, 1]] = 0$ and $median([0, 0, 1, 1, 1]] = 1$. This two datasets are different only in position 3, and by definition are neighborhood datesets. So finally for the *median* function $GS_f = 1$.

(d) Dataset is a list of edges in the graph with a fixed number of vertices V. A *connected component* of an undirected graph is a subgraph in which each pair of nodes is connected with each other via a path. Function $f(D)$ provide number of components.

If a number of vertices V is zero, then a number of connected components are zero. Except this degeneranive case $f(D) \geq 1$. The function $f(D) = 1$ is achieved when the graph is connected. Next, it's possible to have $f(D) = V$ when there are no edges in the graph. It's impossible to have $f(D) > V$ because each connected component should be at least one vertex.

Suppose D is a dataset of edges, and D'' is a neighborhood dataset that differs only in one edge number i.

First of all, let's consider dataset D' as a set of edges without edge i. It can be several scenarios for value of function $f(D')$ in that case:

1. Edge i connects vertices $v_k, v_z \in V$ in single connected component C in dataset D. But before deletion this edge (v_k, v_z) another path that connects $v_k, v_z \in V$ was exist. So deletion does not effect value of function f and $f(D') = f(D)$.
2. Edge i connects vertices $v_k, v_z \in V$ in single connected component C in dataset D. And before deletion this edge was the only way go from v_k to v_z. In that case deletion of edge i breaks connected component C into 2 connected components. All vertices reachable from v_k will be in one connected component, and vertices reachable from v_z will be another component. This affects value of function $f(D')$ in that way: $f(D') = f(D) + 1$.
3. Edge i existed, but vertices $v_k, v_z \in V$ are not in single connected component. It's an impossible case. If edge (v_k, v_z) exist that it already means that v_k and v_z are in the same connected component because they are connected with path of length 1 presented by edge i.

As a second step, we add another edge, not the same as edge i, to the set of edges D' and obtaining the final set D''. The set D'' by our definition of neighboring of a set D, because it differs in exactly one edge from D. If we add a new edge to previously constructed D', then there are two scenarios:

a The new edge (v_t, v_y) connects previously connected vertices which already contains a path in D'. In that case $f(D'') = f(D')$.

b The new edge (v_t, v_y) connects previously unconnected vertices v_t, v_y. In that case all reachable vertices from v_y and all reachable vertices in v_t will be grouped into single connected component in D''. In that case. $f(D'') = f(D') - 1$.

By our construction $D'' \sim D$. Because the number of possible actions is finite to construct D'' we can enumerate all possible scenarios of constructing $D \to D' \to D''$:

- $(a) - (1) \implies f(D'') - f(D) = 0$
- $(b) - (1) \implies f(D'') - f(D) = f(D') - 1 - f(D) = -1$
- $(a) - (2) \implies f(D'') - f(D) = f(D') - f(D) = 1$
- $(b) - (2) \implies f(D'') - f(D) = (f(D') - 1) - f(D) = (f(D) + 1 - 1) - f(D) = 0$
- $(3) - *$ is impossible case.

This analysis demonstrates that if actions $(b) - (1)$ or $(a) - (2)$ can be applied for the graph then:

$$GS_f \coloneqq \max_{D \sim D''} \|f(D) - f(D'')\|_1 = \max_{D \sim D''} |f(D) - f(D'')| = 1$$

We can not derive anything extra from the task description about the graph structure, so we can upper bound $GS_f = 1$. The graph for which $GS_f = 1$ is exist, example is in Figure 1. That demonstrates that this bound $GS_f = 1$ is tight.

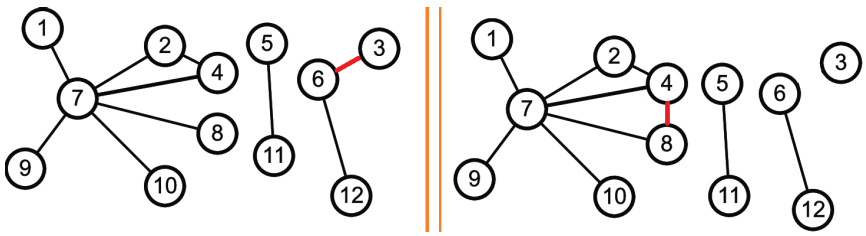

Figure 1: Example of two neighboorhood graphs. Graph (b) is transformation of graphs (a) achieved by changing single red edge. Number of connected components in (a) is 3, and number of connected components in (b) is 4. Global sensitivity of such change is 1.

3 Task 3

In the reconstruction attacks that we have considered in the lecture, we consider the case when the attacker asks k queries, and he/she obtains k responses for them, which are linear functions for that queries, but with extra noise. The noise is bounded in the following way: $(\langle f_i, s\rangle - a_i) \leq \alpha n$.

Where: $f_i \in \{0,1\}^n$ is a query vector, $s \in \{0,1\}^n$ is a secret vector, $a_i \in \{0,1,\ldots,n\}$ is obtained (noisy) answer, and αn is a bound which we use for a purpose try to hide $\langle f_i, s\rangle$ from analytic.

3.1 Theorem derivation

Now we consider the following version of **Lemma 2.4**:
If $t \in \{-1,0,1\}^n$ be a vector with at least m non-zero entries and $u \in \{0,1\}^n \sim U^n$, and $2 \leq w \ll 2^m$ we have:

$$\mathbf{P}\left(|u \cdot t| \geq \frac{\sqrt{m\log(w)}}{10}\right) \geq \frac{1}{w}$$

Or equivalently:

$$\mathbf{P}\left(|u \cdot t| < \frac{\sqrt{m\log(w)}}{10}\right) \leq 1 - \frac{1}{w}$$

Theorem to prove: If we ask $n^2 < k \ll 2^n$ queries, and all queries are bounded by αn, then with extremely high probability the reconstruction error is $\mathbf{O}\left(\frac{\alpha^2 n^2}{\log(k/n)}\right)$

Let's suppose that reconstructed answer $\widetilde{s}$ differs from the true answer s in at least $m = T^2\frac{\alpha^2 n^2}{\log(k/n)}$ positions. Let's define set $\mathbf{B}$ in a similar way to *Lecture notes 2* as:

$$\mathbf{B} = \{\widetilde{s} : |\widetilde{s} - s|_1 \geq m\}$$

We will show that with high probability: $\exists i \in [k] : |\langle f_i, s - \widetilde{s}\rangle| \geq 4\alpha n$.

Because s and $\widetilde{s}$ are differ in at least m positions it implies that vector $s - \widetilde{s}$ has at least m not zero elements equal to -1 or $+1$. Because we construct queries as as uniform vectors from $\{0,1\}^n$ we can apply Lemma 2.4 for which we specify $u = f_i, t = s - \widetilde{s}$:

$$\mathbf{P}(\forall i \in [k] : |\langle f_i, s - \widetilde{s}\rangle| < 4\alpha n) \leq \mathbf{P}(\forall i \in [k] : |\langle f_i, s - \widetilde{s}\rangle| < \frac{\sqrt{m\log(w)}}{10}) \leq (1 - 1/w)^k \leq 2^{-2n}$$

To state this inequality we require that $f_i, f_j, i \neq j$ and two extra constraints hold:

$$(1 - 1/w)^k \leq 2^{-2n} \tag{1}$$

$$4\alpha n \leq \frac{\sqrt{m\log(w)}}{10} \tag{2}$$

Now let's analyze inequality 1:

$$(1-1/w)^k \le 2^{-2n} \iff k\log_2(1-1/w) \le -2n \iff \ln(1-1/w)/\ln(2) \le -2n/k \iff \ln(1+(-1/w)) \le -2\ln(2)n/k$$

It can be shown from convexity of $\exp(x)$ at $x_0 = 0$ the following inequality holds:

$$ln(1+x) < x, \forall x > -1.$$

So if $w > 1, (-1/w) \le -2\ln(2)n/k \implies \ln(1+(-1/w)) \le (-1/w) \le -2\ln(2)n/k$.

So we can use that condition and obtains:
$1/w \ge 2\ln(2)n/k \iff \ln(4)w \le k/(n) \iff \log\ln(4) + \log(w) \le \log(k/n)$.

Let's assume $v \cdot \log(w) = \log(k/n)$, and if v sufficiently large and all logarithms are non-negative it's possible to satisfy this inequality.

Now let's analyze Inequality 2:

$$4\alpha n \le \frac{\sqrt{m\log(w)}}{10} = T\alpha n\sqrt{\log(w)/\log(k/n)}/10 \iff$$
$$40 \le T\sqrt{\log(w)/\log(k/n)} \iff 1600\log(k/n)/\log(w) \le T^2$$

And for $1600\log(k/n)/\log(w) \le T^2 \iff 1600 \cdot v \le T^2$

So if we have control over T and v we can make Equations 1, 2 satisfiable. For example, we can take $T^2 = 1600 \cdot v$. The residual of the prove repeats Lecture Notes 2:

$$\mathbf{P}(\exists \widetilde{s} \in \mathbf{B}, \forall i \in [k] : |\langle f_i, s-\widetilde{s}\rangle| < 4\alpha n) \le \sum_{\widetilde{s}\in\mathbf{B}} \mathbf{P}(\forall i \in [k] : |\langle f_i, s-\widetilde{s}\rangle| < 4\alpha n) \le 2^n \cdot 2^{-2n} = 2^{-n}$$

So in probabilistic sense with extremely high probability for $\widetilde{s} \in \mathbf{B}$ we have $\exists i \in [k] : |\langle f_i, s-\widetilde{s}\rangle| \ge 4\alpha n$.

But if solution has that property $|\langle f_i, s-\widetilde{s}\rangle| \ge 4\alpha n$, it implies:
$|\langle f_i, \widetilde{s}-a_i\rangle| \ge |\langle f_i, s-\widetilde{s}\rangle| - |\langle f_i, a_i - s\rangle|$ (from triangle inequality)

But because: $|\langle f_i, a_i - s\rangle| \ge -\alpha n$ we can bound $|\langle f_i, \widetilde{s}-a_i\rangle| > 3\alpha n$.

And this means that $\widetilde{s}$ can not be the result of the reconstruction attack because $\widetilde{s}$ is not even feasible. This steps repeat considerations (2.11,2.12,2.13) from lecture notes 2.

3.2 Theorem comparisons

Now we have the three theorems (the first two from Lecture Notes 2 and the last one that we have shown):

1. The Theorem 2.2 provide guarantee that Algorithm 1 will give us $\widetilde{s}$, for which reconstruction error $|s-\widetilde{s}|_1 \le 4\alpha n$, given that we can ask 2^n queries with bounded error: $(\langle f_i, s\rangle - a_i) \le \alpha n$.

2. The Theorem 2.3 provide guarantee that Algorithm 2 will give us $\widetilde{s}$, for which reconstruction error $|s - \widetilde{s}|_1 \leq 256\alpha^2 n^2$ **with high probability**, given that we can ask $k = 20n$ queries with bounded error: $(\langle f_i, s\rangle - a_i) \leq \alpha n$.
3. Variation of Theorem 2.3 provide guarantee that Algorithm 2 will give us $\widetilde{s}$, for which reconstruction error $|s - \widetilde{s}|_1 \leq \dfrac{(1600v)\alpha^2 n^2}{\log(k/n)}$ **with high probability**, given that we can ask $k \in [n^2, 2^n]$ queries with bounded error: $(\langle f_i, s\rangle - a_i) \leq \alpha n$.

Case for $k \approx n^2$ Theorem 2.2 (1) - is not applicable, Theorem 2.3 (2) has requirement for $k \approx n$ and extra queries are not part of analysis for Theorem 2.3 (2). It provides bound on reconstruction error as $256\alpha^2 n^2$. The variation of Theorem 2.3 (3) give us reconstruction error $\mathbb{O}(\alpha^2 n^2 / \log(n))$. So Theorem 2.3 (3) provides formally better bound if $\log(n)$ sufficiently big. But if $n = 2^{32} \approx 4\,billions$ then $\ln(2^{32}) \approx 22$ and it's worthwhile to try both algorithms (2), (3) because some improvement may be hidden in assymptotic constants. For practical purpose there is no big difference in algorithm in asymptotic rates, because $\log(n)$ is pretty low growing function, even for big n.

Case for $k \approx 2^{\sqrt{n}}$.

Theorem 2.3 (2) provides bound on reconstruction error as $256\alpha^2 n^2$, and Theorem 2.3 (3) provides bound on reconstruction error as

$$\mathbb{O}\left(\frac{\alpha^2 n^2}{\log(k/n)}\right) = \mathbb{O}\left(\frac{\alpha^2 n^2}{\log(2^{\sqrt{n}}) - \log(n)}\right) = \mathbb{O}\left(\frac{\alpha^2 n^2}{\sqrt{n}\log(2) - \log(n)}\right) = \mathbb{O}\left(\alpha^2 n^{1.5}\right)$$

Here we observed $log(n)$ term into $\mathbb{O}(\sqrt{n})$, because $\sqrt{(}n)$ increasing faster then $\log(n)$ due to L'Hopialle rule.

The Theorem 2.3 (3) in that case provide majority improvement in the decreasing asymptotic rate of reconstruction error $\mathbb{O}\left(\alpha^2 n^{1.5}\right)$, that is better then for Theorem 2.3 (2) $\mathbb{O}\left(\alpha^2 n^2\right)$.

Case for $k \approx 2^n$ The Theorem 2.3 (2) provide guarantee that Algorithm 2 will give us $\widetilde{s}$, for which reconstruction error $\mathbb{O}\left(\alpha^2 n^2\right)$, but Theorem 2.3 (3) provide us reconstruction error: $\mathbb{O}\left(\dfrac{\alpha^2 n^2}{\log(2^n) - \log(n)}\right) = \mathbb{O}\left(\alpha^2 n\right)$

This is far better bound than Theorem 2.3 (2) can give us.

Interestingly, in the regime $k = 2^n$ Theorem 2.2. can also be applied if make $k = 2^n$ and the Theorem 2.2. provide us reconstruction error as $\mathbb{O}(\alpha n)$. In case if $\alpha \in (0, 1]$ the Theorem 2.3 (2) may give us better results asymptotically, but it's important to distinguish that specifically in that mode Theorem 2.2. is deterministic, and Theorem 2.3 (2) has a probabilistic nature in its formulation.

Algorithm	Number of experiments	Error estimation
Random response	1000	-0.01079273
Random response	100000	0.00209399
Random response	500000	-0.00209399
Laplacian mechanism	1000	-0.0016794
Laplacian mechanism	100000	-0.00027954
Laplacian mechanism	500000	-0.00018496

Table 1: Verification of unbiasedness for DP mechanisms

4 Task 4

In that task we implement two mechanisms (Generalized) Random Response and Laplacian Mechanism for $f(x) = \frac{1}{n}x_i, x_i \in \{0, 1\}$. The $GS_f = 1/n$. All experimental code is locating in *experiment.py* and presents experiments. To launch it, it is enough to have a Python interpreter and install for it two extra libraries: NumPy and matplotlib.

4.1 Test unbiasedness of random estimators

The first test is sanity check that (Generalized) Random Response and Laplacian Mechanism are unbiased estimators of $f(x)$. For that purpose we fixed $\varepsilon = 0.01$ and vector $x \in \mathbb{R}^{\kappa}$ equal to elementwise to 1, i.e. $x_i = 1$. After that we carry 1000, 100000, 500000 experiments and estimate expectation of error of this mechanism, i.e. $f(x) - \phi(x)/n$ over specific number of experiments. Results are presented in 1. The variance of expectation estimator based on series of experiments decreases the variance of random variables by factor $\frac{1}{\#experiments}$ from the Central Limit Theorem in Chebychev form (or Weak Law of Large Numbers). This test convinces that extra scaling for the Random Response mechanism presented in "Theorem 3.2, Lecture 3: Definition of Differential Privacy" is necessary.

4.2 Test dependency on ε

In that experiment we vary $\varepsilon \in [0.01, 2.0]$, smaller ε corresponds to stronger privacy. We fixed input size $n = 1000$. For each ε we sample $x \sim U(0, 1^n)$ and apply DP mechanism on top of it. After that we compute error from the true value as $error(x) = f(x) - DP_{mechanism}(x; \varepsilon)$. But because Mechanism is randomized the error is random variables. We estimate it's mean and variance across 900 repetition of experiments. Results are presented in Figure 2. The error bars are of height of two estimated standard deviations. Theory predict us that error in function value in absolute value for Random Response is bounded by $\mathbb{O}\left(\frac{1}{\sqrt{n}\varepsilon}\right)$, and error for Laplacian Mechanism for that function is bounded by $\mathbb{O}\left(\frac{1}{n\varepsilon}\right)$. Note that as ε gets closer to 0 (corresponding to stronger privacy), the error increases if the sample complexity is fixed.

To observe asymptotic behavior in practice it's a possibly to apply *log* and observe $\log \mathbb{O}(1/\varepsilon^{-1}) = const - 1 \cdot \log(\varepsilon)$. As we see for two algorithms the slope in the log-log

plot, 2 (the bottom graphic) is the same, and this exactly what theory say for us.

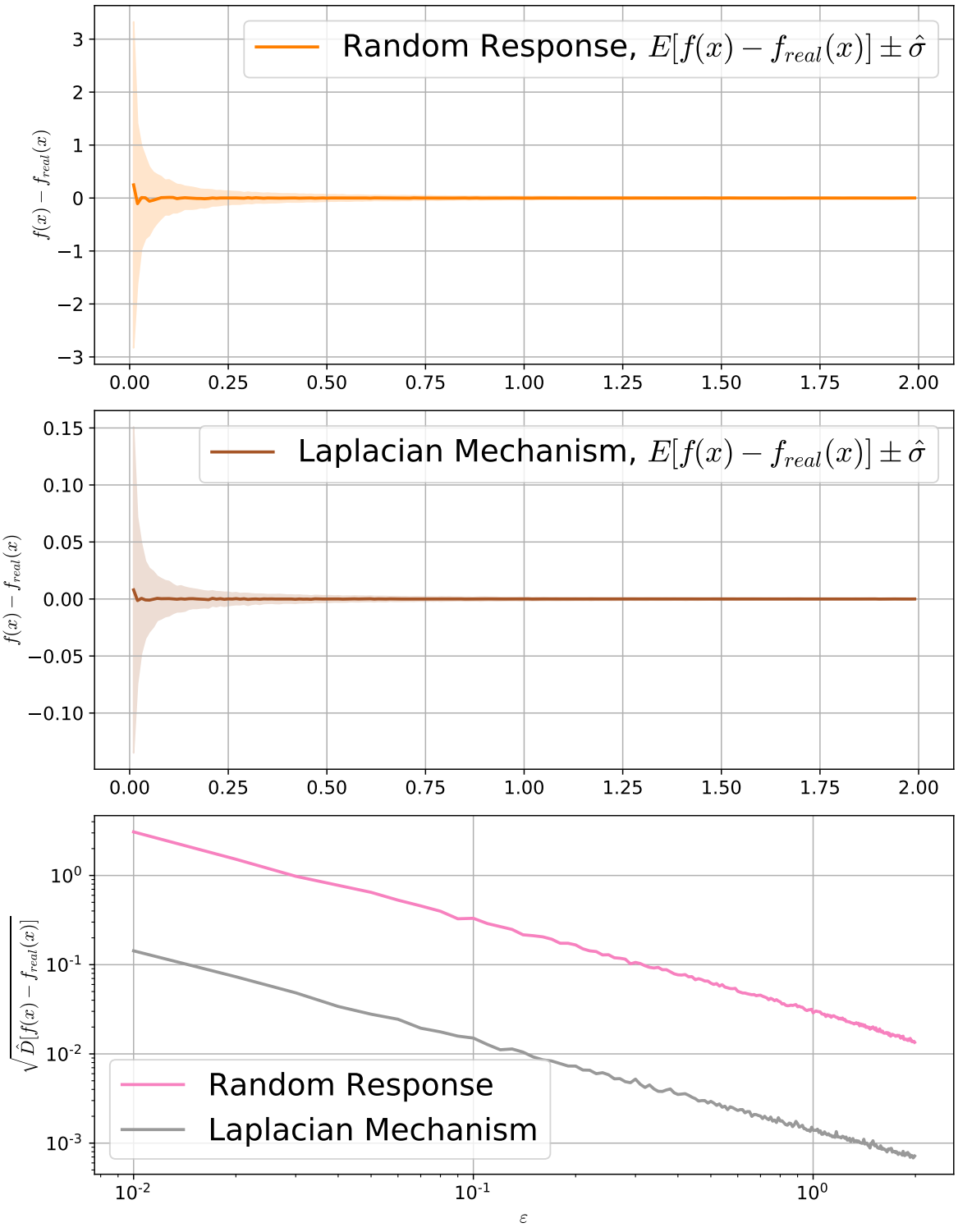

Figure 2: Dependency of errors in DP mechanism as a function of ε. First two plot contains error of for computing $f(x)$ with error bars. The last plot contains estimated standard deviation of error during computing $f(x)$ in logarithmic scale in OX, OY axis.

4.3 Test dependency on n

In that experiment we fix $\varepsilon = 0.001$ and we vary input size $n \in [100, 10\,000]$. For each n we sample $x \sim U(0, 1^n)$ and apply DP mechanism on top of it. After that we compute error from the true value as $error(x) = f(x) - DP_{mechanism}(x; \varepsilon)$. Similar to the previous section description, because Mechanism is randomized the error is random variables. We estimate it's mean and variance across 900 repetition of experiments. Results are presented in Figure 3. The error bars are of height of two estimated standard deviations.

As we see the asymptotic behavior for two algorithms is completely different. The theory predict us that error in estimation value of function f, for Random response is $\mathbb{O}\left(\frac{1}{\sqrt{n}\varepsilon}\right)$, and error for Laplacian Mechanism for that function is $\mathbb{O}\left(\frac{1}{\sqrt{n}\sqrt{n}\varepsilon}\right)$. Theoretical justification is in Theorem 3.2, 3.5 in Lecture Note 3 for the course.

From figure 3 we can observe the following:

1. Laplacian mechanism for all size of tested inputs n, and for $x \sim \{0, 1\}^n$ provides fat less variance, for the same level ε.
2. The rate with the Laplacian mechanism is decaying is far more rapid, as we can in the log-log plot at Figure 3. For example, for $n = 10\,000$, the error estimated with standard deviation is near 100 smaller than for Random Response. This is a factor that theory predicts us as $\mathbb{O}\left(\frac{1}{\sqrt{n}}\right)$.
3. If we fix the privacy guarantee we would like to have, we need more data to decrease the error.
4. In the Figure 3 the OX, OY axis has log-log plot. So the value which we observe in OX,OY axis are power of 10. Next because $\log(\mathbb{O}(n^{-0.5})) = const_a - 0.5 \cdot \log(n)$, and $\log(\mathbb{O}(n^{-1})) = const_b - 1 \cdot \log(n)$, if the theory is correct we should observe the *twice faster* decreasing (slope in absolute sense is twice bigger) in log-log plot fo Laplacian mechanism in comparison to Random Response. And this exactly what we observe from log-log plot at the Figure 3, bottom graphic.

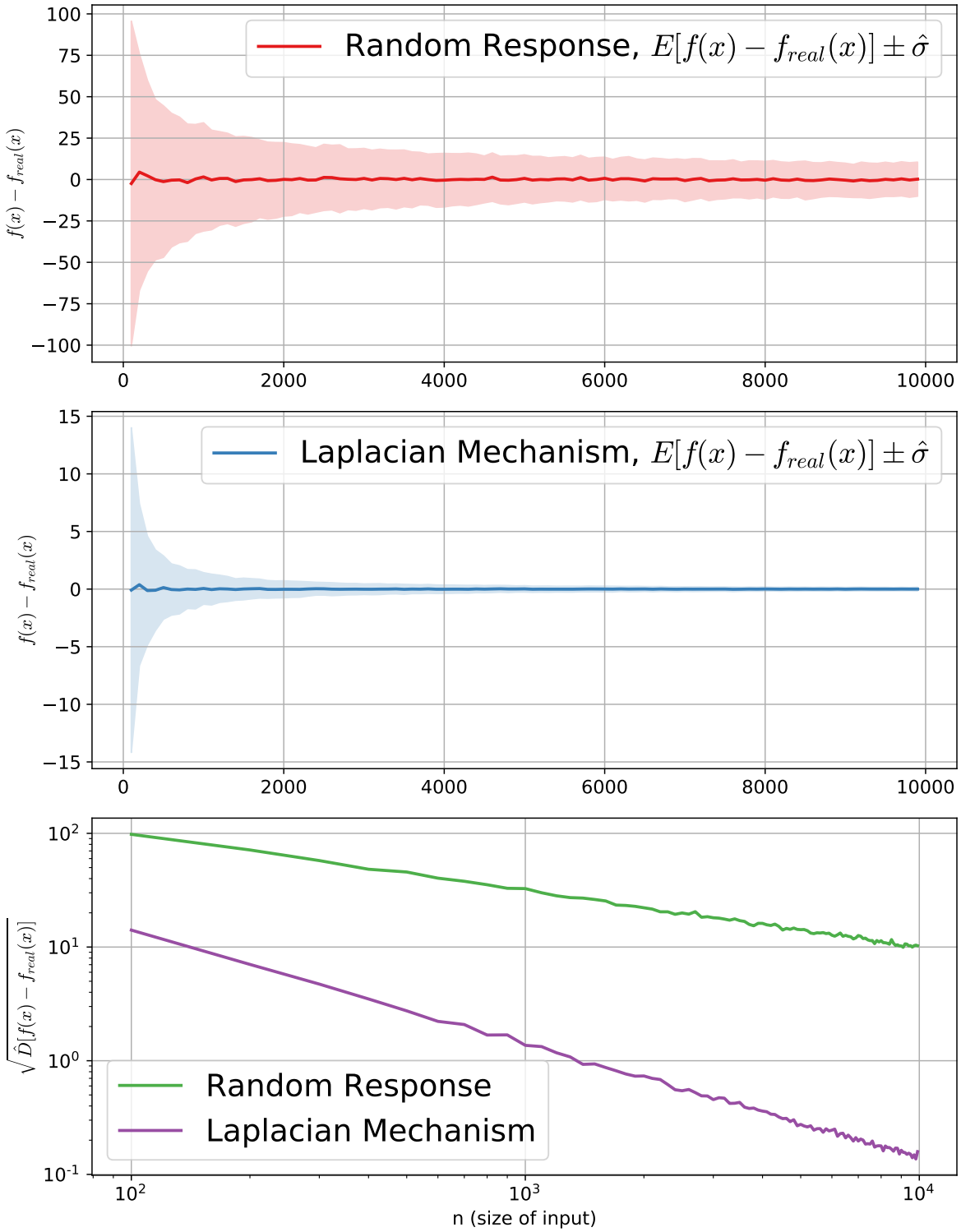

Figure 3: Dependency of errors in DP mechanism as a function of n. First two plot contains error of for computing $f(x)$ with error bars. The last plot contains estimated standard deviation of error during computing $f(x)$ in logarithmic scale in OX, OY axis.