CS394S Assignment 1

Siyuan Chen 174817

1. Properties of Laplace Distributions

1. Properties of Laplace Distributions

- Prove $\sqrt{E(z^2)} = \sqrt{2}\lambda$

$$E(z^2) = \int_{-\infty}^{+\infty} \frac{y^2}{2\lambda} \exp(-\frac{|y|}{\lambda})\, dy$$

$$= \int_0^{\infty} \frac{y^2}{\lambda} \exp(-\frac{y}{\lambda})\, dy$$

$$= \int_0^{\infty} y^2\, d(-\exp(-\frac{y}{\lambda}))\, dy$$

$$= \underbrace{-y^2 \cdot \exp(-\frac{y}{\lambda}) \Big|_0^{\infty}}_{=0} + \int_0^{\infty} \exp(-\frac{y}{\lambda})\, dy^2$$

let $u = \frac{y}{\lambda}$, $y = u\lambda$

$$\int_0^{\infty} \exp(-\frac{y}{\lambda})\, dy^2 = \int_0^{\infty} 2y \cdot \exp(-\frac{y}{\lambda})\, dy$$

$$= \int_0^{\infty} 2u\lambda \exp(-u)\, d(u\lambda)$$

$$= \lambda^2 \int_0^{\infty} u^2 \exp(-u)\, du$$

$$= \lambda^2 \cdot \Big[\underbrace{-u^2\exp(-u)}_{0} \underbrace{- 2u \cdot \exp(-u)}_{0} \underbrace{- 2e^{-u}}_{2}\Big]\Big|_0^{\infty}$$

$$= 2\lambda^2$$

Therefore. $\sqrt{E(z^2)} = \sqrt{2}\lambda$.

- Prove For every $t>0$: $P(Z>\lambda t) \leq \exp(-t)$

by integrating the PDF of Laplace distribution:

$$h_\lambda(y) = \frac{1}{2\lambda}\exp\left(-\frac{|y|}{\lambda}\right).$$

$$P(Z>\lambda t) = \int_{\lambda t}^{\infty} \frac{1}{2\lambda}\exp\left(-\frac{|y|}{\lambda}\right) dy. \quad (\text{Because } t>0)$$

$$= \int_{\lambda t}^{\infty} \frac{1}{2\lambda}\exp\left(-\frac{y}{\lambda}\right) dy$$

let $u = \frac{y}{\lambda}$, $y = u\lambda$

$$= \int_{t}^{\infty} \frac{1}{2\lambda}\exp(-u) \cdot du\lambda$$

$$= \int_{t}^{\infty} \frac{1}{2}\exp(-u)\, du$$

$$= \frac{1}{2} -\exp(-u) \Big|_{t}^{\infty}$$

$$= \frac{1}{2}\left[0 + \exp(-t)\right] = \frac{1}{2}\exp(-t)$$

$$\leq \exp(-t).$$

2. Global Sensitivity

2. Global Sensitivity.

(a). Let $u = f(D)$ $u' = f(D')$ where D' is the neighbour dataset of D.

Global sensitivity of $f = \frac{1}{n}\sum_{i=1}^{n} x_i$

$$\Delta f_{GS} = \max_{D,D'} \| f(D) - f(D') \|_1$$

$$= \frac{1}{n} \max_{D,D'} \| x_j - x_j' \|$$ ← x_j and x_j' is the exact different element.

Since $D \in \mathcal{X}^n = \{ v \in \mathbb{R}^d : \|v\|_1 \leq 1 \}$

Therefore $\max_{D,D'} \| x_j - x_j' \| \leq 2$

$$\Delta f_{GS} = \frac{1}{n} \max_{D,D'} \| x_j - x_j' \| \leq \frac{2}{n}$$

(b) $\Delta f_{GS} = \max_{D,D'} \| f(D) - f(D') \|_1$

$$= \max_{D,D'} \| \sum_{i=1}^{n} x_i x_i^T - \sum_{i=1}^{n} x_j x_j^T \|$$

For instance, Assume that x_1 is the exact different element.

$$\Delta f_{GS} = \max_{D,D'} \| (x_1 - x_1') \sum_{i=1}^{n} x_i \|$$

Because $\mathcal{X} = \{v \in \mathbb{R}^d, \|v\|_1 \le 1\}$

Therefore $\|X\|_1^n = |x_1| + |x_2| + |x_3| \dots + |x_n| \le 1$

$$\Delta f_{GS} = \max_{D, D'} \|(x_1 - x_1)' \sum_{i=1}^{n} x_i\| \le \max_{D, D'} |x_1 - x_1'| \le 2$$

(C). $\Delta f_{GS} = \max_{D, D'} \|f(D) - f(D')\|$

$f(D) = \text{median}(x_1 \dots x_n)$ and $\mathcal{X} = [0, 1]$.

Therefore: $f(D) = [0, 1]$, $f(D') = [0, 1]$

$$\Delta f_{GS} = \max \|f(D) - f(D')\| \le 1$$

(d) $\Delta f_{GS} = \max_{D, D'} \|f(D) - f(D')\|$

Assume that the original graph D is a (V, E) graph, and the

G_D is the resulting Graph $(V, \hat{E})$

The number of subgraph of D is maximum:

$f(D) \le C_E^1 \cdot 2^{n-2} + C_E^2 \cdot 2^{n-3} + \dots + C_E^{E-1} \cdot 2^1 + 1 \cdot 2^0$

E is maximum: $C_n^2 = \frac{n \times (n-1)}{2} = \frac{n^2 - n}{2}$

when remove or add one edge $\tilde{e}$ from the original graph E.

$$\Delta f_{GS} = \max_{D, D'} \| f(D) - f(D') \| = O\left(\frac{n^2 - n}{2} \cdot 2^{n-2} \right) = O(n^2 \cdot 2^n)$$

which is unbounded.

$\Delta f_{GS} \sim \infty$

3. Reconstruction Attacks

Our goal is to show any vector $\tilde{s} \in \{0,1\}^n$ that disagree with s on more than $\frac{\alpha^2 n^2}{\log(k/n)}$ entries cannot satisfy : $|F_i\tilde{s} - a_i| \leq \alpha n$

And cannot be the output of reconstruction attack.

we now fix the true secret vector $s \in \{0,1\}^n$, let

$$B = \{\tilde{s} : \tilde{s} \text{ and } s \text{ disagree on at least } \frac{\alpha^2 n^2}{\log(k/n)} \text{ entries}\}$$

Our goal is to show that the reconstruction attack does not output any vector in B.

We fix some $\tilde{s} \in B$, and show that it is eliminated with extreme high probability. Suppose $\tilde{s} \in (0,1)^n$ differs from s on at least $m = \frac{\alpha^2 n^2}{\log(k/n)}$

Use lemma 1: let $t \in \{-1,0,1\}^n$, with at least m, nonzero entries and $u \in \{0,1\}^n$ be a uniformly random vector.

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

$$\Rightarrow P\left(|u \cdot t| \leq \frac{\sqrt{m \log w}}{10}\right) \leq 1 - \frac{1}{w}. \qquad \text{lemma (2)}$$

Because $\frac{\sqrt{m \log w}}{10} \leq 4\alpha n$ according to Lecture 2 notes,

$$\log w \leq \frac{1}{m} 1600 \alpha^2 n^2$$

$$w \leq \exp\left(\frac{1}{m} 1600 \alpha^2 n^2\right) \qquad m = \frac{\alpha^2 n^2}{\log(k/n)}$$

$$\frac{1}{w} \geq \exp\left(-\frac{1}{m} 1600 \alpha^2 n^2\right)$$

$$1 - \frac{1}{w} \leq 1 - \exp\left(-\frac{1}{m} 1600 \alpha^2 n^2\right) = 1 - \exp\left(-\frac{\log(k/n)}{\cancel{\alpha^2 n^2}} \cdot 1600 \cancel{\alpha^2 n^2}\right)$$

$$= 1 - \exp\left[-1600 \log\left(\frac{k}{n}\right)\right]$$

Therefore:

$$P\left(\forall i \in [k], \left| F_i \cdot (s - \tilde{s}) \right| \leq \frac{\sqrt{m \log w}}{10}\right) \qquad m = \frac{\alpha^2 n^2}{\log(k/n)}$$

$$P\left(\forall i \in [k] \left| F_i (s - \tilde{s}) \right) \leq \frac{\alpha n}{10} \cdot \sqrt{\log\left(w - \frac{k}{n}\right)}\right) \leq \left(1 - \frac{1}{w}\right)^k$$

$$= \left[1 - \exp\left(-1600 \log \frac{k}{n}\right)\right]^k$$

since $n^2 << k << 2^n$.

$$\left[1 - \exp(-1600 \log n)\right]^n \leq \left[1 - \exp\left(-1600 \log \frac{k}{n}\right)\right]^k \leq \left[1 - \exp\left(-1600 \log \frac{2^n}{n}\right)\right]^{2^n}$$

Therefore: with $n^2 << k << 2^n$, the probability that reconstruction error is at most $O\left(\frac{\alpha^2 n^2}{\log(k/n)}\right)$ is very high

4. Random Response and Laplacian Mechanism

We need to first generate the data:

```
import numpy as np
import random
def generate_data(n):
    out = []
    for i in range(n):
        out.append(random.randint(0,1))
    return out
```

The definition of the query is

$$f(D) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

```
def f(D):
    return np.mean(D)
```

Therefore, the definition of Random Response is for each individual, to roll a dice, if result is 1, 2, 3 or 4: report true value. If 5 or 6 report opposite value, the code implementation is (This function returns the response to a query):

```
def RandomResponse(D):
    responses = []
    # Roll a dice
    for i in range(len(D)):
        dice = random.randint(1,6)
        if dice in [1,2,3,4]:
            responses.append(D[i])
        else:
            responses.append(0 if D[i]==1 else 1)
    return f(responses)
```

The definition of Laplacian mechanism is to add a Laplace noise on the true result. The independent Laplace(Δ/ε) random variables is (1/nε) when Δ = 1/n, the code implementation of the Laplacian mechanism is (This function returns the response to a query):

```
def Laplacian(D,e,n):
    true_result = f(D)
    laplac_noise = np.random.laplace(0, (1/(e*n)))
    out = true_result + laplac_noise
    return out
```

We first generate the n from [10,50,100,500,1000,2000,3000,5000,10000] and e from [0.1,0.2,0.3,0.5,1,2,3,5,10]

```
n_list = [10,50,100,500,1000,2000,3000,5000,10000]
e_list = [0.1,0.2,0.3,0.5,1,2,3,5,10]

error_dic = {}
error_dic["Random Response"] = []
for e in e_list:
    error_dic["Laplacian with e = {}".format(e)] = []
```

Then for all the n and e, we calculate the relative error between the actual response and the noised response:

```
for n in n_list:
    print("=> n: {}".format(n))
    true_dataset = generate_data(n)
    true_f =  f(true_dataset)
    random_response_f = RandomResponse(true_dataset)
    error_dic["Random Response"].append(abs(true_f - random_response_f))

    for e in e_list:
        laplace_response_f = Laplacian(true_dataset,e,n)
        error_dic["Laplacian with e = {}".format(e)].append(abs(true_f -
laplace_response_f))
```

The error list is plotted as follows, as can be seen:

- The larger the number of sample size, the less the error would be, therefore the higher the utility
- The error would be less once the Laplacian e is set to be very high
- The Random Response method is less accurate compared to Laplacian noise whith e >0.5