Data Privacy and Security: Homework #3

Due on December 29, 2020 at 11:59pm

Professor: Mehmet Emre Gürsoy
TA: Çağhan Köksal

Muhammet Soytürk

Problem 1

(a) The sensitivity of the given query $S(q)$, where q = number of location readings exist in Koc University campus, would be: $S(q) = T^{MAX}$. Since T^{MAX} could potentially go to infinity, $S(q) = \infty$

Individual	Trajectory
T1	[(-1.618,41.141), (-1.618,41.142), (-1.618,41.143), ...(∞,∞)]
T2	[(-2.618,41.141), (-2.618,41.142), (-2.618,41.143)]
T3	[(-3.618,41.141), (-3.618,41.142), (-3.618,41.143)]
T4	[(-4.618,41.141), (-4.618,41.142), (-4.618,41.143)]
T5	[(-5.618,41.141), (-5.618,41.142), (-5.618,41.143)]
...	...
TD	[(-D.618,41.141), (-D.618,41.142), (-D.618,41.143)]

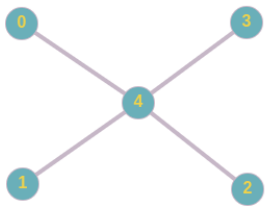
If we remove T1 from the dataset, then $S(q) = \infty$.

(b) Since we remove or add an individual to get the neighboring dataset, S(q) = 1. Consider the following dataset:

Owner	Id	Contents
x1	t1	{a}
x2	t2	{b}
x3	t3	{c}
...	...	
xD	TD	{d}

If we remove t1 from the dataset, then S(q) = 1

(c) S(q) = 2 * Number of vertices. For example if we have a graph like the following:



We have 1 + 1 + 1 + 1 + 4 = 8 total edges in the graph. If we remove vertex 4 from the graph, then we would have 0 total edges.

(d) Since we remove or add an individual to get the neighboring dataset, S(q) = 1. Consider the following dataset:

Id	Searches
1	{fenerbahçe}
2	{fenerbahçe}
3	{fenerbahçe}
...	...
D	{fenerbahçe}

If we add or remove an individual, number of users whose search history includes fenerbahçe would increase or decrease by one at most.
(e) S(q) = 1. Let's consider the following example:

Job	Nationality	Gender	Disease
Accountant	American	*	Flu
Salesperson	Western Europe	Male	HIV

If we use variational distance, dist(Q, Pq*) = 1. If we remove one of the records, new variational distance is 0. Hence, S(q) = 1.

Problem 2

Problem 3

(a) It is not ϵ-DP. Counter example:

Record
$e^\epsilon - 1$
$e^\epsilon + 1$

$\forall$ D,D', $\forall$O, it should satisfy the following:

$$\frac{Pr[A(D) = O]}{Pr[A(D') = O]} \leq e^\epsilon$$

If we remove the second record to get the neighboring dataset and check for output "large":

$$\frac{1/2}{0^+} \leq e^\epsilon$$

Since the left side of the equation goes to infinity, the equation does not hold. Hence, the algorithm is not ϵ-DP.

(b) It is not ϵ-DP. Counter example: k=1 and D is:

Job	Nationality	Gender	Disease
Salesperson	Western Europe	Male	HIV

If we get the neighboring dataset by removing the only record, the equation does not hold. Hence, the algorithm is not ϵ-DP.

$$\frac{1}{0^+} \leq e^\epsilon$$

(c)

Problem 4

(a)

1. Compute the real answer for query: q(D). This step is done in the query_all function of LaplaceMechanism class in q4.ipynb. It basically returns the number of people in each education level.

2. Find sensitivity of q: S(q). Since we can only add or remove one individual, it's 1. self.sensitivity represents this value in LaplaceMechanism class.

3. Draw a random sample from Lap(0, S(q)/€). I have used np.random.laplace function to achieve this. Check draw_samples function in LaplaceMechanism. It returns a sample for each education level

4. Add the random sample r to real answer. This step is done in get_noisy_data function of LaplaceMechanism.

5. Return the noisy data. This is also done in get_noisy_data function.

Scale value is represented as an attribute of LaplaceMechanism class and equals to sensitivity / epsilon.

(b) For implementation check q4.ipynb.

(c) Since epsilon is small, the query result I receive is a bit different from the original results. While the original results are: [3178, 2416, 1990, 1393, 592, 504, 399, 398, 89, 82, 55, 43, 38, 22, 8, 1], what I receive after applying laplace mechanism is: [3140.29338679 2316.10846544 1894.88086299 1236.78752685 891.79108527 537.19163221 450.22931349 648.19074704 106.30116098 152.87497005 -36.28289116 87.09409537 -49.37620888 -159.58829914 257.30234815 -11.22286503]. It might not be a big problem for big values such as 3178 but for other values I sometimes receive negative values. I suppose we would convert negative values to 0 or not answer at all.

(d) As we increase the epsilon, we get lower error. This is expected because when we increase the epsilon, we decrease privacy which means we add less noise to the result which decreases error. Smaller epsilon values are preferred for privacy.

Epsilon	Err
0.01	110.1746
0.05	26.0734
0.1	8.9587
0.5	1.8041
1	1.2759

Problem 5

(a) Check q5.ipynb.

(b)

1. Determine the appropriate score function q_f. This step is done in ExponentialMechanism's query function. I choose number of people in an education class / 10 as the score function for programatical reason.

At the beginning I chose number of people as the score function but it was causing overflows. That's why I scaled it down by 10. I also tried to divide it by mean or median but it was not performing well in terms of accuracy. The sensitivity of the function is 1 since we can only add or remove one individual.

2. Find sensitivity of q: S(q). Since we can only add or remove one individual, it's 1. self.sensitivity represents this value in LaplaceMechanism class.

3. Draw a random sample from Lap(0, S(q)/€). I have used np.random.laplace function to achieve this. Check draw_samples function in LaplaceMechanism. It returns a sample for each education level

4. Add the random sample r to real answer. This step is done in get_noisy_data function of LaplaceMechanism.

5. Return the noisy data. This is also done in get_noisy_data function.

(c) Check q5.ipynb

(d) For ExponentialMechanism, the following are the results:

Epsilon	Accuracy
0.0001	0.02
0.001	0.07
0.01	0.15
1	0.84

Since LaplaceMechanism does not output the real result (it adds noise to the result), accuracy value cannot be reported. Laplace based algorithm is more preferrable because we are returning numerical values.

Problem 6

(a) Check User class in q6.ipynb

(b) Check GRR class q6.ipynb

(c) Amount error decreases as we increase the epsilon value. This is expected because as we increase epsilon we relax the privacy. Hence, we get results which are more similar to the real dataset.

Epsilon	Error
0.5	3471.3736
1.0	1316.4754
2.0	473.8173
4.0	74.0072

(d) Check User Class in q6.ipynb

(e) Check SimpleRAPPOR class in q6.ipynb

(f) Amount error decreases as we increase the epsilon value. This is expected because as we increase epsilon we relax the privacy. Hence, we get results which are more similar to the real dataset.

Epsilon	Error
0.5	1050.3074
1.0	552.8131
2.0	235.6007
4.0	102.5344

(g) Usually Simple RAPPOR is preferable since it does not depend on value 'd' but for this dataset d is not too big and it's computationally more efficient.

(h) For both the user side steps would be identical because we don't have control over the user's information but on the server side of things, instead of return all age groups' recovered data, I would only return the mean of those age groups.