Homework 3
COMP530 Fall 2020 - Data Privacy and Security

Erhan Tezcan 0070881

22.12.2020

Answers

Answer 1:

(a) Imagine D has a record with 1 location reading ($|D| = 1$), and a neighboring data set D' has that same record and a record with T^{max} location readings. As a result, querying the number of locations readings has sensitivity $T^{max} - 1$.

(b) Support of an itemset X in D is:

$$supp(X) = \frac{|\{t \in D : X \subseteq t\}|}{|D|}$$

Depending on the definition of a neighboring dataset, the answer may change, so we will look at all of them.

- $D' = D \cup \{\text{customer}\}$ where at most N transactions with various items can be added. Fix an itemset X with support $supp(X) = s/|D|$. We can add N transactions that for every newly added transaction t^*, $X \subseteq t^*$. Now our neighboring dataset D' has $|D'| = |D| + N$ and $supp(X) = (s + N)/(|D| + N)$. The sensitivity of this query is thus:

$$\left| \frac{s+N}{|D|+N} - \frac{s}{|D|} \right|$$

- $D' = D \cup \{\text{transaction}\}$ where at most m distinct items can be added in a new transaction of a customer. Fix an itemset X where $i \in X$ and let $supp(X) = s/|D|$. We can have a new transaction t^* where $X \subseteq t^*$ and the support becomes $(s + 1)/(|D| + 1)$. The sensitivity of this query is thus:

$$\left| \frac{s+1}{|D|+1} - \frac{s}{|D|} \right|$$

- $D' = D \cup \{\text{item}\}$ where only an item is added to a certain transaction. Call this new item $i \in I$ and fix an itemset X where $i \in X$ and let $supp(X) = s/|D|$. We can increment s by adding item i to some transaction t where $X \not\subseteq t$ but $X \subseteq t \cup \{i\}$, and creating t'. Consequently $X \subseteq t'$ and

thus the support has increased by $1/|D|$. The sensitivity of this query is thus $1/|D|$.

(c) If we define neighboring graph to have a new edge, then sensitivity is 2. This is because our added edge will increase the degree of two vertices (not considering the loop, however that wouldn't matter) and thus the sum increases by 2.

If we define neighboring graph to have a new vertex, we can add one such that it is connected to every other vertex, thus it has degree V (V is the original vertex count). This increases the sum by $2V$ where V comes from the degree of new vertex and the other V comes from the increase in the degree of every other vertex by 1.

(d) A newly added individual either has the keyword "fenerbahce" in it's search history or it does not ($p \wedge p' \implies true$ from a logical perspective). If it does, the query would consider that, so the sensitivity is 1.

(e) Our newly added record would belong to some certain EC in the dataset, never to 2 different EC's at once. A newly added record in an EC can either break $t-closeness$ (w.r.t what value it had before) or not break it. If it does not break it, the query returns whatever it returned before. But if it breaks, now it returns 1 less than what it returned before, and as a result the sensitivity of this query is 1.

Answer 2:

(a) *Proof.* The identity function fits this definition. Let $\mathcal{A}(D) = D$ and therefore for all possible datasets the output is the dataset itself, therefore different than any other output. This definition alone shows that such an algorithm has no privacy to speak of, however to give a more formal proof we can say that

$$\frac{\Pr[\mathcal{A}(D) = O]}{\Pr[\mathcal{A}(D') = O]}$$

is either 0 $(0/1)$ or infinite $(1/0)$, which in the latter case breaks ϵ-DP. □

(b) *Proof.* If $\mathcal{A} : D \to \mathbb{R}$ is ϵ-DP, then

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

Since $f : \mathbb{R} \to \mathbb{R}$ defines a one-to-one mapping, for every such D, O pair given above there exists one and only one $D, f(O)$ pair, where f maps O to $f(O)$ and probabilities are still the same. As a result, $f(\mathcal{A}(D))$ is ϵ-DP. □

(c) *Proof.* Here we are told that:

$$\frac{\Pr[\mathcal{A}_1(D) = O]}{\Pr[\mathcal{A}_1(D') = O]} \leq e^{\epsilon_1}$$

$$\frac{\Pr[\mathcal{A}_2(D) = O]}{\Pr[\mathcal{A}_2(D') = O]} \leq e^{\epsilon_2}$$

First let us define what it means to have $\mathcal{A}_{1,2}(D) = O$. With this, we mean that $\mathcal{A}_{1,2}(D) = (\mathcal{A}_1(D), \mathcal{A}_2(D)) = (O_1, O_2)$ and $O_1, O_2 \in \mathbb{R}^n$. We want to show that

$$\frac{\Pr[\mathcal{A}_{1,2}(D) = O]}{\Pr[\mathcal{A}_{1,2}(D') = O]} \leq e^{\epsilon_1 + \epsilon_2}$$

Since the algorithms are independent, we can treat this as:

$$\frac{\Pr[\mathcal{A}_1(D) = O_1 \wedge \mathcal{A}_2(D) = O_2]}{\Pr[\mathcal{A}_1(D') = O_1 \wedge \mathcal{A}_2(D') = O_2]} \leq e^{\epsilon_1 + \epsilon_2}$$

where $O = (O_1, O_2)$. This further gives:

$$\frac{\Pr[\mathcal{A}_1(D) = O_1]}{\Pr[\mathcal{A}_1(D') = O_1]} \frac{\Pr[\mathcal{A}_2(D) = O_2]}{\Pr[\mathcal{A}_2(D') = O_2]} \leq e^{\epsilon_1 + \epsilon_2}$$

From the first two inequalities we were provided, we can say that:

$$\frac{\Pr[\mathcal{A}_1(D) = O_1]}{\Pr[\mathcal{A}_1(D') = O_1]} \frac{\Pr[\mathcal{A}_2(D) = O_2]}{\Pr[\mathcal{A}_2(D') = O_2]} \leq e^{\epsilon_1} \times e^{\epsilon_2} \leq e^{\epsilon_1 + \epsilon_2}$$

thus proving that $\mathcal{A}_{1,2}$ is $(\epsilon_1 + \epsilon_2)$-DP. □

(d) *Proof.* Let D be a dataset and D_i denote a dataset that differs by i individuals. We are told that an algorithm $\mathcal{A}$ satisfies ϵ-DP, so:

$$\frac{\Pr[\mathcal{A}(D) = O]}{\Pr[\mathcal{A}(D_1) = O]} \leq e^{\epsilon}$$

In fact, notice that $D{=}D_0$. Now, reformulate this expression as:

$$\frac{\Pr[\mathcal{A}(D_{i-1}) = O]}{\Pr[\mathcal{A}(D_i) = O]} \leq e^{\epsilon}$$

without loss of generality (because D_i could be in numerator and D_{i+1} could be in the denominator). Looking at the expression:

$$\prod_{i=1}^{k} \frac{\Pr[\mathcal{A}(D_{i-1}) = O]}{\Pr[\mathcal{A}(D_i) = O]} \leq \prod_{i=1}^{k} e^{\epsilon}$$

It is easy to see that $\prod_{i=1}^{k} e^{\epsilon} = e^{k\epsilon}$. For the left side, notice that every term's denominator gets cancelled out by the next term's numerator, and we are left with

$$\frac{\Pr[\mathcal{A}(D_0) = O]}{\Pr[\mathcal{A}(D_k) = O]} \leq e^{k\epsilon}$$

Well, this is the exact definition of a $(k\epsilon)$-DP private algorithm, thus it has been shown. □

Answer 3:

(a) $\mathcal{A}$ does not satisfy differential privacy.

Proof. Let D be a dataset where $|D| = e^\epsilon$. Let D' be a neighboring dataset such that $|D'| = |D| + 1$ and thus $|D'| > e^\epsilon$. Looking at the expression:

$$\frac{\Pr[\mathcal{A}(D') = \text{“large”}]}{\Pr[\mathcal{A}(D) = \text{“large”}]} \leq e^\epsilon$$

this expression wrong, because the left side actually equals to $1/0$ and that can be treated as ∞, which is greater than any e^ϵ. □

(b) $\mathcal{A}$ does not satisfy differential privacy.

Proof. Let D be a k-anonymous dataset. This means that $\forall r \in D$, there exists at least $k-1$ records with the same QI values. Without loss of generality, fix an $r \in D$, and let K denote the set of all records that have same QI values with r, r included. By definition of k-anonymity, we know that there can be a K such that $|K| = k$. Note that $K \subseteq D$. Again without loss of generality, fix an $r' \in K$ where $r' \neq r$. Let $D' = D \backslash \{r'\}$. Notice that now in D' there exists $k-2$ other records identical to r with respect to it's QIs, and therefore D' is $(k-1)$-anonymous. With this in mind, we look at the expression below:

$$\frac{\Pr[\mathcal{A}(D, k) = \text{TRUE}]}{\Pr[\mathcal{A}(D', k) = \text{TRUE}]} \leq e^\epsilon$$

this expression wrong, because the left side actually equals to $1/0$ and that can be treated as ∞, which is greater than any e^ϵ. □

(c) $\mathcal{A}$ satisfies differential privacy.

Proof. Suppose D has no records named John (without loss of generality) and aged above 40. Let D' have all the records in D and an extra record of a person named John with age above 40. In this case, the case where it may break ϵ-DP is when:

$$\frac{\Pr[\mathcal{A}(D, \text{John}) = \text{FALSE}]}{\Pr[\mathcal{A}(D', \text{John}) = \text{TRUE}]} \leq e^\epsilon$$

$\mathcal{A}(D, \text{John})$ returns $0 + r$ where $r = Lap(0, \epsilon)$ and $\mathcal{A}(D, \text{John})$ returns $1 + r'$ where $r' = Lap(0, \epsilon)$. For this to happen, $r < 1$ and $r' > 0$. So in other words:

$$\frac{\Pr[r < 1]}{\Pr[r' > 0]} \leq e^\epsilon$$

Let C denote the CDF, then $\Pr[r < 1] = C(1)$ and $\Pr[r' > 0] = 1 - C(0)$. The CDF of laplace distribution with 0 mean, ϵ scale and $x \geq$ mean:

$$C(x) = \frac{1}{2} e^{-\frac{x}{\epsilon}}$$

As a result, $C(0) = 1$ and $C(1) = \frac{1}{2} e^{-1/\epsilon}$. Putting these back in our inequality:

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

$$e^{\frac{-1}{\epsilon}} \leq e^{\epsilon}$$

$$\frac{-1}{\epsilon} \leq \epsilon$$

is true. Thus, $\mathcal{A}$ is ϵ-DP. $\square$

Answer 4:

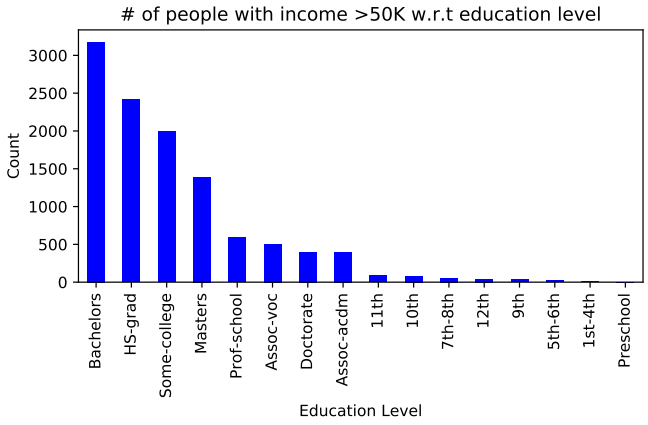
(a) The steps of this algorithm is described as:
   (a) Compute the histogram
   (b) Add noise from $Lap(0, 1/\epsilon)$ to each bin in the histogram. Here, Lap is laplace distribution, 0 is the mean and $1/\epsilon$ is the scale. The numerator 1 comes from the sensitivity of this histogram query.
   (c) Return noisy histogram.

   This works because of the parallel composition property of DP. Each bin is actually a subset of the dataset, and these bin's are all disjoint.

(b) Source code attached.

(c) The normal histogram and noisy histogram are given in figures 1 and 2 respectively. We see that they are actually not so far

FIGURE 1. Number of people with income > 50K with respect to their education level.



apart, we can still "have an idea" about the overall distribution of education and income, however, in a finer granularity we can see errors, for example in the noisy histogram there is a negative value! That would not happen in a normal histogram, which counts number of occurences, and occurence cant be negative. However like I stated before, this does not prevent us to have a general idea regarding the histogram.

(d) Error and epsilon plot is given in figure 3.

We can see that error rises exponentially when we decrease epsilon. This is expected, smaller epsilon means more noise, and more noise means more error. The scale parameter of laplace distribution increases

FIGURE 2. Number of people with income > 50K with respect to their education level with added Laplace noise.

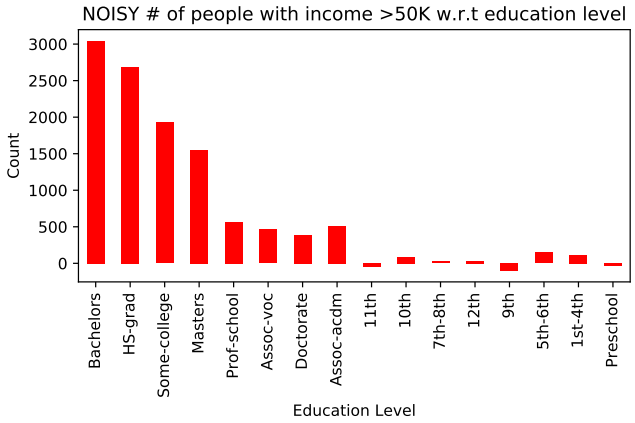
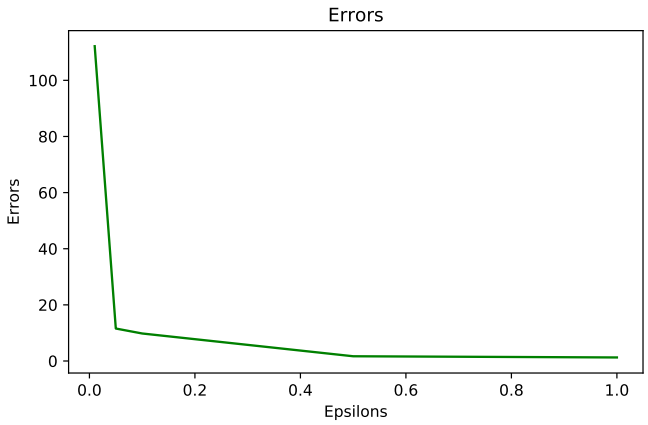


FIGURE 3. Error and Epsilon.



when epsilon decreases, and as a result higher noises start to have higher probability. This reflects to our results as high error.

Answer 5:

(a) Source code attached.

(b) First we define a score function $q : D \times R \to \mathbb{R}$ where D is the dataset, R is the domain of discrete outputs (i.e. possible values of "education" QI) and $\mathbb{R}$ is as always the set of real numbers. In this homework, I've chosen a score function $q(D, r)$ that returns the number of records in D that have some value $r \in R$. Notice that this is exactly a histogram, when you calculate it for all $r \in R$. Likewise, the sensitivity of this score function is 1.

Then, we choose a random $r^* \in R$ with probability:

$$\Pr[r^* \text{is returned}] = \frac{e^{\frac{\epsilon q(D,r^*)}{2}}}{\sum_{t \in R} e^{\frac{\epsilon q(D,t)}{2}}}$$

(c) By using the post-process property of DP, we can use the noisy histogram obtained in the previous question and run this query on it, by selecting the bin with the highest number. This will still satisfy ϵ-DP of what the noisy histogram satisfied.

(d) Results are given in table 1. With smaller epsilon, the accuracy drops significantly. By looking at the table, we can see that Laplace-based method performs better, on almost all epsilon values. Only at $\epsilon = 0.0001$ it falls a bit short, but at such low accuracy that would not matter much anyway.

	$\epsilon = 0.0001$	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 0.1$
Exponential Mechanism	0.06	0.12	0.96	1.0
Laplace-Based	0.05	0.47	1.0	1.0

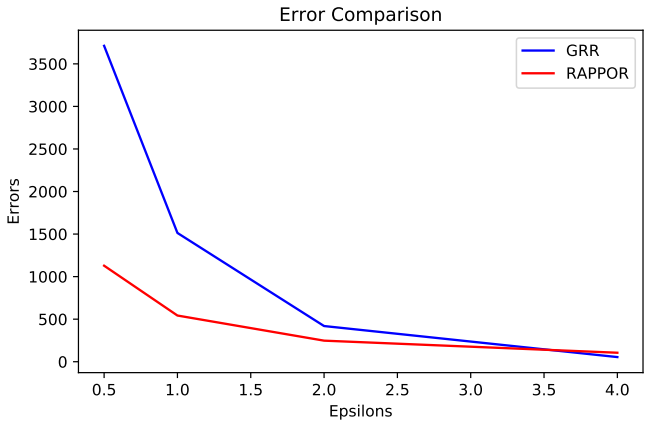
TABLE 1. Table of accuracies for each DP algorithm, averaged over 100 runs for each epsilon.

Answer 6:

	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 4.0$
GRR	3712.095	1513.115	419.282	54.586
Simple RAPPOR	1128.314	542.781	247.537	105.565

TABLE 2. Table of errors for GRR and SimpleRAPPOR.

FIGURE 4. Error and Epsilon for GRR and SimpleRAPPOR.



(a) Source code attached.
(b) Source code attached.
(c) Errors for GRR are given in table 2. We see that as epsilon gets larger, error rate drops signficiantly. In fact, we can see that this inverse correlation is exponential. This is due to the fact that the epsilon is used as an exponent in p, which is the probability of giving honest answers. As epsilon gets larger, the probability of telling the truth rises exponentially.
(d) Source code attached.
(e) Source code attached.[1]
(f) Errors for SimpleRAPPOR are given in table 2. Similar to GRR, we observe that as epsilon gets larger, error rate drops signficiantly, in a similar inversely correlated way. Here too, as epsilon gets larger, the probability of telling the truth rises exponentially.

[1]Note that the program may take some time to run, however be assured that it will not be stuck whatsoever.

(g) SimpleRAPPOR seems to perform better, with significantly lower error compared to GRR when epsilon is small, also shown in figure 4. In fact, for $\epsilon = 0.5$ SimpleRAPPOR is more than 3 times better! For these reasons, I would prefer SimpleRAPPOR. Note that SimpleRAPPOR can be extended to actually include arbitrary values, rather than our fixed domain of $\{1, 2, \ldots, d\}$, so that is yet another reason to chose RAPPOR.

(h) With the noisy histogram at our hands that show the smartphone users per age, we can make a heuristic. In the most naive approach, we would assume the whole population is usinga smartphone and reports back their age, and in that case we can just take the average of all values in our histogram. Since the histogram is ϵ-LDP, due to post-processing property of LDP this average result also does not violate privacy. This would apply to both GRR and SimpleRAPPOR.