university of groningen

Data Privacy - Assignment 1

Kyriakos Antoniou (S5715881), Sumanth Seeram(S5652588), Narjes Sharafi (S5697832)

August 29, 2025

---

1 Q1. k-anonymity. The organization decides to make this data public after some anonymization, namely suppression and generalization. The original data is the following obtained from the survey is provided in Table 1.

Age	Zip Code	Profession	Alignment
25	56601	Engineer	PartyX
34	56602	Doctor	PartyY
66	56601	Teacher	PartyX
24	56604	Engineer	PartyZ
47	56603	Doctor	PartyZ
51	56602	Student	PartyY
22	56603	Student	PartyX
28	56601	Doctor	PartyY
31	56603	Engineer	PartyZ
34	56602	Student	PartyX

Table 1: Political tendencies (related to question 1)

1.1 Identify quasi-identifiers (equivalence classes) and explain how you obtained it.

Quasi-identifiers are attributes that, when combined, can link an anonymized dataset to other datasets. They are pieces of information that are not unique identifiers by themselves, but when combined with other quasi-identifiers they end up as unique identifiers.

Like for example, first name and last name can be combined to identify an individual in a better way, if we add DOB to it, it can help us uniquely identify the individual.

As a result, the individuals can be identified with their quasi-identifiers. In table 1, ”Age”, ”Zip Code”, and ”Profession” are the quasi-identifiers as they need to be combined in order to act as unique identifiers. The records in the table vary in these attributes and can help us uniquely identify the individuals. Equivalence classes are groups of records with indistinguishable attributes. In this table, before any anonymization operation, each record forms one distinct equivalence class.

The attribute ”Alignment” is the sensitive attribute that is released publicly and this is the research topic.

1.2 By employing suppression on any combination of columns/rows, make the released table 2-anonymous (keep the data in the table as much as possible).

In table 2 the result of applying suppression to the quasi-identifiers is presented. Suppression involves hiding individual attributes to prevent individual identification. In this table, for just two records with the same ”Age” attribute and ”Zip Code”, only the ”Profession” is suppressed, while for other records, the ”Age” attribute is hidden. The ”Zip Code” attribute is suppressed for the record with the unique zip code 56604 as well as for three additional records. The same is employed on the record with the ”Profession” teacher which makes it unique. As a result, each quasi-identifier tuple appears in at least two records, making each record indistinguishable from at least one other record. Therefore these two records form an equivalence class.

Age	Zip Code	Profession	Alignment
*	*	Engineer	PartyX
34	56602	*	PartyY
*	56601	*	PartyX
*	*	Engineer	PartyZ
*	56603	*	PartyZ
*	*	Student	PartyY
*	*	Student	PartyX
*	56601	*	PartyY
*	56603	*	PartyZ
34	56602	*	PartyX

Table 2: Suppression

1.3 By employing generalization on any combination of columns/rows, make the released table 2-anonymous (keep the data in the table as much as possible).

Table 3 presents the result of applying generalization to table 1 in order to achieve 2-anonymity. Generalization is about generalizing each record attribute to a broader category. For the "Age" attribute, three categories are used: $20 - 40$, ≥ 20, and ≥ 50. The "Zip Code" attribute is generalized to $5660*$. For the "Profession" attribute, the only generalization is applied to the professions of Teacher and Student, which are grouped under the broader category of Educator. As a result, table 3 is now 2-anonymous through generalization operation.

Age	Zip Code	Profession	Alignment
20-40	5660*	Engineer	PartyX
$\geq$20	5660*	Doctor	PartyY
$\geq$50	5660*	Educator	PartyX
20-40	5660*	Engineer	PartyZ
$\geq$20	5660*	Doctor	PartyZ
$\geq$50	5660*	Educator	PartyY
20-40	5660*	Student	PartyX
$\geq$20	5660*	Doctor	PartyY
20-40	5660*	Engineer	PartyZ
20-40	5660*	Student	PartyX

Table 3: Generalization

2 Q2.Differential Privacy

2.1 a. What type of problem it tries to solve and what it adds to traditional anonymization techniques?

Differential Privacy aims to solve the problem of revealing the identity of an individual who has participated in a statistical database. In other words, while insights are gained from the overall population, the information about each individual should remain private. It enables the sharing of useful statistical information from sensitive databases, without the risk of data leakage. Since the traditional anonymization techniques are done on the data server, on the company side, the users have to trust the company to remove the identifications. Moreover, the extent of the anonymization would not be enough, such as revealing the sensitive data with the help of an auxiliary information database.

2.2 b. How it works in general terms. You can use simple examples to explain the concepts (e.g., function sensitivity level).

The idea of Differential Privacy, in simple words, is to add noise drawn from the Laplacian distribution to the query result. As a result, an adversary query will end up in a noisy response, and the result is different from the true one. For example, if an adversary aims to find out if one specific individual is in the query response set, with the help of differential privacy, the result would be with some extent of noise, not the actual true one. The amount of noisiness can be controlled by the factor of the "Scale Parameter". This factor is based on the sensitivity and the ϵ parameter. The sensitivity is the measure of how much a query result can change by adding or removing a single element from the database. Therefore the balance between privacy and accuracy is based on the value of ϵ which is the privacy parameter.

2.3 c. Given Table 4 (adapted from the UCI Machine Learning Repo), the adversary is allowed to count the number of people with income more than X (through querying) but we want to do this in a privacy-preserving way, i.e., the result is returned with a noise.

age	workclass	education	occupation	race	sex	hours/week	native-country	income
39	State-gov	Bachelors	Adm-clerical	White	Male	40	United-States	80000
50	Self-emp-not-inc	Bachelors	Exec-managerial	White	Male	13	United-States	45000
38	Private	HS-grad	Handlers-cleaners	White	Male	40	United-States	50000
53	Private	11th	Handlers-cleaners	Black	Male	40	United-States	60000
28	Private	Bachelors	Prof-specialty	Black	Female	40	Cuba	70000
37	Private	Masters	Exec-managerial	White	Female	40	United-States	45000
49	Private	9th	Other-service	Black	Female	16	Jamaica	40000
52	Self-emp-not-inc	HS-grad	Exec-managerial	White	Male	45	United-States	100000
31	Private	Masters	Prof-specialty	White	Female	50	United-States	30000
42	Private	Bachelors	Exec-managerial	White	Male	40	United-States	90000
37	Private	Some-college	Exec-managerial	Black	Male	80	United-States	50000

Table 4: People and Their incomes

2.3.1 1.What is the level of (local) sensitivity of the function we are considering?

The sensitivity is the maximum possible change that can occur by adding or removing a single individual from the database. Since the query counts the number of individuals with an income greater than X, then adding or removing a single individual can change the count number by 1, as that individual may meet the condition of income more than X or fail. Therefore $S(q) = 1$.

2.3.2 2. Show the result of applying Laplace noise with the results of the count query, and summarize your observations.

if $X = 50K$, then 5 individuals have an income greater than 50k. Since sensitivity:

$$S(q) = \max_{D,D'} \|Q_D - Q_{D'}\|_1 = 1 \tag{1}$$

The Laplace Mechanism is:

$$\text{Lap}(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{2}$$

where,

- b scale parameter: $\frac{S(q)}{\epsilon} = \frac{1}{\epsilon}$,
- μ location parameter: often set to 0.

Therefore a noise point which is η, is drawn from this Laplace distribution and is added to the query result, Listing 1. In the following, different values of ϵ which is the privacy parameter, and the corresponding results are discussed:

- if $\epsilon = 1$:

  then we should use the $\text{Lap}(1)$.

  One random value from $\text{Lap}(1)$ is $\eta = -0.68$, this random noise value added to the true value: $5 - 0,68 = 4.32$, and after rounding the noisy value, the query result is 4. Therefore, the true result can not be revealed and the presence or absence of an individual is not certain.

- if $\epsilon = 2$:

  With a greater ϵ like 2, $\text{Lap}(\frac{\text{Sensitivity}}{\epsilon}) = \text{Lap}(0.5)$. A random sample from Laplacian distribution is $\text{Lap}(0.5) = 0.28$. By adding the random noise $5 + 0.28 = 5.28$ and rounding the result, the query result would be 5 which is the true value. Therefore, by larger ϵ we got less privacy and more accurate query results.

- if $\epsilon = 0.5$:

  With a greater ϵ like 0.5, the Laplacian distribution is: $\text{Lap}(\frac{\text{Sensitivity}}{\epsilon}) = Lap(2)$. A random sample noise from the distribution is $\text{Lap}(0.5) = 4.3$. After adding the random noise $5 + 4.3 = 9.3$ and rounding the value, the query result is 9. By this smaller value for ϵ, the result has more privacy and is less accurate.

Therefore, with smaller ϵ, the variance which is $2(\frac{\text{sensitivity}}{\epsilon})^2$ is more, and the Laplacian distribution can add more noise to the query result, which leads to stronger privacy. However, with a greater value for ϵ, the lower variance can add less noise, therefore less privacy and more accurate query results.

Listing 1: Laplace Mechanism

```
import numpy as np

def laplace_mechanism(true_count, sensitivity, epsilon):

    # the scale parameter for the Laplace distribution
    scale = sensitivity / epsilon

    # the location parameter
    mu = 0

    # Laplace noise
    laplace_noise = np.random.laplace(mu, scale)

    # Add the noise to the true count
    noisy_count = true_count + laplace_noise

    print(f"The true values is {true_count}, The added laplace noise is {laplace_noise}, And the noisy count is
        {round(noisy_count)}")
```

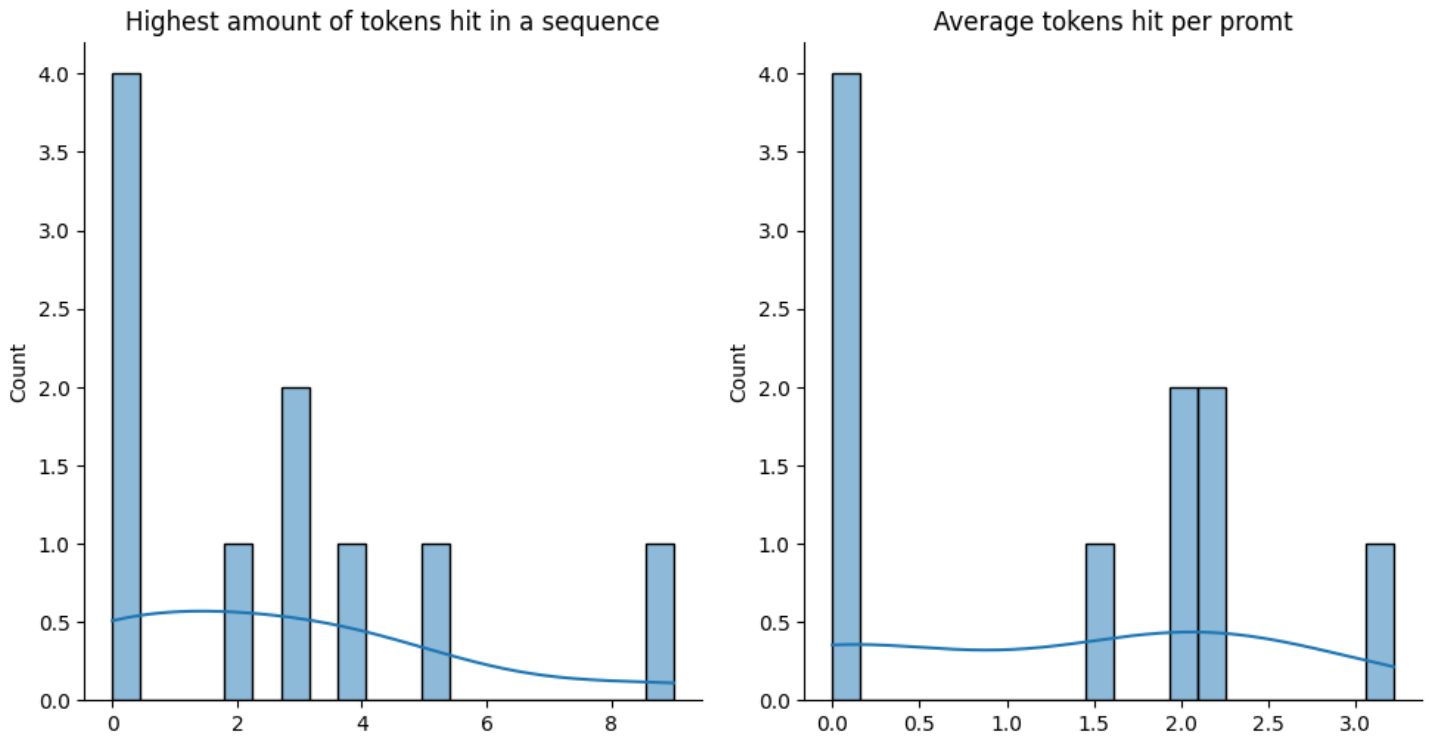

Figure 1: Graphs for random token data extraction attack

3 Q.3 Q.4 Targeted Data Extraction Attack Against LLMs

3.1 Did the model reveal memorized sequences during the targeted attack?

The two types of attacks, targeted prompts and random prompts show how much of the data can be extracted when having prior information or not.

We have seen that for targeted prompt attacks of 30 token length, the model is able to produce exact matching sequences of 30 tokens long following those 30 prompt tokens. This shows that a specific enough attack can retrieve an equal amount of data once for every about 500 tries (where every 10th prompt gives a result when each prompt is asked to produce 50 outputs). In the worst case given a targeted sequence attack of 30 tokens long, the model can give an output of about 5 token size exact sequence matches usually able to finish a sentence correctly.

Looking at the random token attacks,1 the model is still able to output sequences of tokens of up to 9 tokens long with an average prediction of 1.3 tokens long (single token matches are counted as no match as each token does exist at least once). This shows that the model is more likely to predict 2 or more tokens that exist in the dataset in a sequence rather than not predicting any matching sequence.

3.2 What type of samples were more prone to the data extraction attacks?

It can be observed from the targeted data extraction attack that the more tokens given to the model the more it is able to produce accurately. It can also be seen from the random data extraction attack that the more rare tokens are the more chances the model will remember and output the pieces of data with those rare tokens in it. Both of the above observations can be explained by the amount of context given to a specific model via its input; the model is trying to pinpoint from its training data which of the input data matches its training data the best thus large sequences and rare characters are given the model more specific pointers allowing the model to be more sure for its output.

3.3 What are the privacy risks of training models on sensitive data and how memorization can lead to privacy breaches?

Training a model with sensitive data could lead to the data being exposed to third parties especially when large models are tasked to learn from few data. This would make the models memorize the sensitive data rather than generalize which would allow attackers to

gain access to the data. This data could leak at any moment even if the model is trained on more data down the line. Furthermore, even if the sensitive data are not used to train the model, data that could connect the points to the sensitive data are still at risk of being leaked through the connected data. This leakage can happen through a number of different attacks. Some of those are the targeted and random data extraction attacks tested in this assignment, but also model inversion attacks and model inference attacks where information even though not exactly the same can be inferred from the input and output of the model respectively.