CSD Project 2

Kevin Gallagher

Wednesday 15th May, 2024; 17:05

1 Option 1: Data Privacy

Privacy is a human right that is gaining more and more attention internationally. Recent laws such as the GDPR and the CCPA require that companies provide privacy for their customers. However, these laws and most companies tend to define data protection as de-identification. In this project we're going to study the limits of de-identification by performing a linkage attack, then we're going to explore two alternatives, k-anonymity and differential privacy.

This project can be done either individually, or in groups of 2. In order to perform this project, you will need to download the following dataset:

https://raw.githubusercontent.com/kcg295/CSD-Project-2/main/DP/Data/sti_data.csv

This project is broken into three tasks, all of which will require turning in code as part of the eventual submission.

Before completing these tasks, take the dataset and experiment with it. Try to learn whatever you can without any external datasets.

Before describing the tasks, there is one important thing to keep in mind. Given that I don't have a Windows or Mac machine, **the programs you submit must be able to work on a Linux machine.** You may develop your program on whatever platform you wish, but please test it on a Linux machine (virtual or not) prior to submitting.

1.1 Linkage attacks - 5 values

After exploring the data, please download the following auxiliary datasets:

- https://raw.githubusercontent.com/kcg295/CSD-Project-2/main/DP/Data/marriage_data.csv
- https://raw.githubusercontent.com/kcg295/CSD-Project-2/main/DP/Data/earnings_data.csv

You must then write code that performs a **linkage attack** using these auxiliary datasets. In other words, you should look at the overlapping columns of the three datasets and attempt to find matches to de-identify the individuals in the original dataset.

For this section you must provide the source code for a program that can be run to perform the linkage attack. This program must take the three datasets as command line arguments.

In order to ensure that I can identify the program that performs the attack, please include the name of the source code file(s) and instructions on how to compile (if applicable) and run the program in your repository's README.

1.2 k-anonymity - 5 values

After seeing the alarming success of the linkage attack, we should now understand the necessity of protecting user data. One of the ways to do this is using something called k-anonymity. k-anonymity is actually a property of a database, rather than a technique. If you recall from class, a database is k-anonymous if each member of the dataset is part of a group of k members which have all of the same quasi-identifiers. The way we usually achieve k-anonymity is by generalizing the data in the dataset until this property is achieved.

For this task you will need to implement two programs: one that tests for k-anonymity and one that generalizes a dataset given a target k. Using these programs, you must test for k-anonymity for all k between 2 and 5 (inclusive), and generalize the dataset until the property is met. For each of these steps, you must note the results in your repository's README file in a section called "K-Anonymity".

For example, let's consider $k = 2$. For this value of k we must check to see if the dataset is already k-anonymous. To do so we would run our k-anonymity checker program on the dataset. If it returns true, please make a note of this on your repository's README. For example, you can say "The dataset was already 2-anonymous without the need to run the generalization program."

If our k-anonymity checker returns false, however, we must try to generalize the dataset to become k-anonymous. To do this we would run our generalizer program, which will give us a new, more generalized dataset. We would then run our k-anonymity checker on this new dataset to ensure the generalization worked. This process should also be logged in the repository's REAMDE. For example, you can say "The dataset was not 2-anonymous, and we needed to run the generalization program. After the program we checked again and the dataset was 2-anonymous."

For each time you generalize the dataset, please write a sentence or two containing observations about the utility of the resulting data. How useful does the data still seem after generalization? Why? After noting these things down, you must then repeat the process for $k = 3$, then $k = 4$, then $k = 5$.

For this section of the assignment you must provide the source code for a program that can be run to generalize the data until we reach k-anonymity for a given k, as well as a program that checks for k-anonymity for a given k. Like the last section, your repository README must identify the source code file(s) and provide instructions on how to compile (if applicable) and run the program. In addition, your REAMDE should contain a section called "K-Anonymity" that discusses the process of running the k-anonymity checker for all k from 2 to 5 (inclusive) and discusses the utility of the data after each run of the generalizer program.

1.3 Differential Privacy - 10 values

As we can see from the previous section, k-anonymity comes at a great cost – our dataset is basically useless now. However, there may be a way for us to balance utility and privacy better than k-anonymity. This is where differential privacy comes into play. Differential privacy protects the data by responding to queries on the dataset with noise injected. The amount of noise injected depends both on the security parameter (called the privacy budget) and the sensitivity of the query. For a full refreshed on differential privacy, I recommend going over the slides from the lecture as well as the following readings:

https://programming-dp.com/ch3.html
https://www.cis.upenn.edu/ aaroth/Papers/privacybook.pdf

In this section of the assignment you must implement a differentially private query system which will respond to queries on the original dataset with noisy responses. As a part of this, you will need to do the following:

1. Implement the query system.
2. Choose and argue for an adequate privacy budget. This can be done as comments in the code.
3. For each query calculate the query's sensitivity. For parameters with unbounded sensitivity, use clipping to ensure that we can create an artificially bounded sensitivity. In the portion of the code that performs clipping, leave a comment with a justification for why you chose those upper and lower bound values.
4. Using the sensitivity, decide how much noise to add to the results of the query.
5. When the privacy budget is expended, delete the database and respond to all further queries with NULL.

Queries will be passed to your program on the command line in SQL format. In order to ensure differential privacy, it may be necessary to convert data types of the dataset. For example, a boolean value cannot have noise added to it. It may be necessary to convert

the boolean to a floating point number (1.0 for true and 0.0 for false) and then add the noise.

For this section you must provide the source code for the program described above. Like in the previous sections, this code must work on a Linux machine, and your repository must identify the source code file(s) and provide instructions on how to compile (if applicable) and run the program in the repository's README file.

1.4 Submission

To submit this project, you must submit the link to your repository via email to Kevin Gallagher at k.gallagher@fct.unl.pt before the 7th of June, 2024 at 23:59. The subject of the email must contain the string "[CSD Project 2 Option 1 Submission]" and then the student numbers of the members of your group.

You repository must contain at least three programs and a README file. The first program corresponds to the de-identification portion of this assignment, the second program corresponds to the k-anonymity portion of this assignment, and the final program corresponds to the differential privacy portion of this assignment. To reiterate, all of these programs must work on a Linux machine.

The README file of this project must contain the names and student numbers of the members of the group, and must identify the files for each portion of the project and provide instructions on how to compile (if applicable) and run each portion of the project. Your code must be legible and un-obfuscated.

Good luck and have fun!