CSC 591: Privacy (Fall 2019)
Home Assignment #1
Assigned: Friday, Sept. 06, 2019, Due: Tuesday, Sept. 17, 2019

Instruction: No collaboration is permitted on this assignment.

Q1. Additional files required for this homework are available at XXX. In this question, we are going to work through how deanonymization works on databases. We will work with a subset of the original Netflix database. You are given a set of 15 movies in the folder movies. The identifiers for these movies are [03124, 06315, 07242, 16944, 17113, 10935, 11977, 03276, 14199, 08191, 06004, 01292, 15267, 03768, 02137]. The movies folder contains csv files for each movie. Each line of the csv file has three entries: a user id, the date of rating, and the rating provided.

Learning Objectives: The problems in the part are based on the paper 'Provable Deanonymization of Large Datasets with Sparse Dimensions' [1]. We will refer to the paper through all problems in this part. We will perform an attack along the lines of the original Netflix-IMDB deanonymization attack. In particular, we will learn how to identify a user by utilizing noisy and incomplete auxiliary information.

Starter Code: You may use the provided starter code (in Python) for this homework. The script reads each file from the movies folder and populates the database *db*. *db* is a Python dictionary. Dictionaries consist of pairs (called items) of keys and their corresponding values. To brush up your knowledge of the Python dictionary data-structure, please view this tutorial[2]. Each element of *db* is the tuple <*user-id, movie-dict*>. *movie-dict* is also a dictionary representing the user's ratings, with each item being the tuple <*movie-id, rating*>. It is not compulsory that you use the starter code provided. You can use any programming language of your choice (among C/C++/Java/Python).

Starter code for this problem is provided in *link.py*. You are given the auxiliary information for one user. The auxiliary information contains noisy ratings given by the user for 12 of the 15 movies. You can think of these being perturbed ratings given by a user on IMDB. This auxiliary information is provided in Table 1 and in the variable aux in *link.py*. You, as the attacker, want to identify the user id for whom the auxiliary information is provided.

Movie	Rating	Movie	Rating	Movie	Rating	Movie	Rating
14199	4.2	17113	4.2	06315	4.0	01292	3.3
11977	4.2	15267	4.2	08191	3.8	16944	4.2
07242	3.9	06004	3.9	03768	3.5	03124	3.5

Table 1: Auxiliary information for the target user

(a) Using Definition 4 of the paper, complete the function compute *weights()* and compute the weights of each movie. Tabulate the weights obtained for each movie. This should be a table with 15 movie-ids and their corresponding weights. [10]

$$w(i) = \frac{1}{\log(|supp(i)|}, |supp(i)| = no. of\ non\,null\ entries\ (i.e.,\ frequency)\ in\ column\,'i'\ (i.e.,\ i - th\,movie)$$

(b) How many users are present in the database? Using Definition 7 in the paper, complete the function *score()* and compute the scores of the auxiliary information with respect to every user's ratings in the database. What is the highest score? What is the second highest score? [15+5]

$$score(aux, r) = \sum_{i\,\epsilon\,supp(aux)} w(i) * \frac{T(aux(i), r(i))}{|supp(aux)|}$$

Where $|supp(aux)| = no. of\ non\ null\ attributes\ (movie\ ratings)\ in\ \ aux$

$$T\big(aux(i), r(i)\big) = 1 - \frac{|aux(i) - r(i)|}{p(i)},$$

[1] http://www.andrew.cmu.edu/user/divyasha/dss-post12.pdf

Where *p(i)* is the maximum possible difference between values of column *i* (i.e., *max* **range** possible even considering values from *aux).* The value of each column is scaled by p, so that the value for T (., .) lies in the interval [0, 1]

(c) What is the user-id of the user with the highest score? Write out the ratings of this user from the database, and verify if they are similar to the ratings in the auxiliary information. [4+6]

(d) Assume the eccentricity metric is γM, where $M = \sum_{i \, \epsilon \, supp(aux)} \frac{w(i)}{|supp(aux)|}$ is the scaled sum of weights of attributes in aux. Say, γ = 0.1 then what is the value of the eccentricity threshold? What is the difference between the highest and second highest score? Is it greater than the eccentricity metric? [4+4+2]

Q2. For the following questions consider table below-

ID	ZIP code	Age	Salary	Disease
1	47677	29	3K	Gastric ulcer
2	47602	22	4K	gastritis
3	47678	27	5K	Stomach cancer
4	47905	43	6K	gastritis
5	47605	30	7K	flu
6	47906	47	8K	bronchitis
7	47674	36	9K	bronchitis
8	47607	32	10K	pneumonia
9	47909	52	11K	Stomach cancer

(a) What are the quasi-identifiers and sensitive attributes in this table? [5]

(b) Compose a 3-anonymous, 3-diverse table. Show intermediate steps for deriving the final solution and also draw the generalization lattice (i.e., using incognito algorithm) for your solution. [15+10]

(c) Compute the t-closeness of your solution with respect to salary. Does your solution resolve the 'similarity attack' for the sensitive attribute 'disease'? If not, compute an alternative solution. [10+10]

Submission:

You have to submit three files:

1. Merge all the written parts into a single pdf file <your unity id>_HW1.pdf.
2. Rename the program file (.c/.cpp/.java/.py) you used for as <your unity id>_HW1.extension.

Zip all files into <your unity id>_HW1.zip and submit the zip file on Moodle.

1. http://www.andrew.cmu.edu/user/divyasha/dss-post12.pdf