KOÇ UNIVERSITY

COMP 430/530: Data Privacy and Security – Fall 2021

Homework Assignment #1

INTRODUCTION AND SETUP

In this assignment, you will implement k-anonymization algorithms in Python and compare their performance by executing your algorithms on a real dataset.

We have provided you with a simplified version of the **Adult** dataset, which is a commonly used dataset in the literature (original version can be found at: http://archive.ics.uci.edu/ml/datasets/Adult). Each row of **Adult** corresponds to one individual. Each column (attribute) contains information such as the individual's gender, education level, marital status, etc. The final column (***income***) is a binary attribute that tells whether the individual's yearly income is ≤ 50K or > 50K dollars. Throughout this assignment, you will treat every attribute other than ***income*** as a Quasi Identifier; and treat ***income*** as the Sensitive Attribute.

Here is a screenshot of the **Adult** dataset that is provided to you. Notice that the dataset is in **csv** (comma separated values) format and the first row contains attribute names.

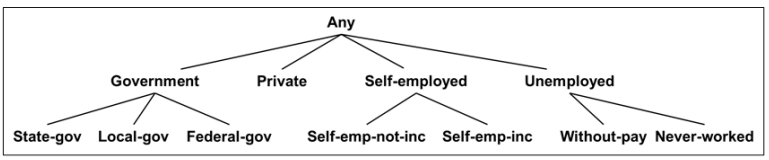
```
age,workclass,education,marital-status,occupation,relationship,gender,native-country,income
25,Private,11th,Never-married,Machine-op-inspct,Own-child,Male,United-States,<=50K
38,Private,HS-grad,Married-civ-spouse,Farming-fishing,Husband,Male,United-States,<=50K
28,Local-gov,Assoc-acdm,Married-civ-spouse,Protective-serv,Husband,Male,United-States,>50K
44,Private,Some-college,Married-civ-spouse,Machine-op-inspct,Husband,Male,United-States,>50K
34,Private,10th,Never-married,Other-service,Not-in-family,Male,United-States,<=50K
63,Self-emp-not-inc,Prof-school,Married-civ-spouse,Prof-specialty,Husband,Male,United-States,>50K
24,Private,Some-college,Never-married,Other-service,Unmarried,Female,United-States,<=50K
55,Private,7th-8th,Married-civ-spouse,Craft-repair,Husband,Male,United-States,<=50K
65,Private,HS-grad,Married-civ-spouse,Machine-op-inspct,Husband,Male,United-States,>50K
```

In addition to the **Adult** dataset, you are also given a folder named **DGHs**. Each file in the DGHs folder contains the domain generalization hierarchy of one of the QI attributes, e.g., **age.txt** contains the DGH for the **age** attribute, **education.txt** contains the DGH for the **education** attribute, and so forth.

When you study the contents of the DGH files, you will see that they follow a pattern such that the tabs indicate parent-child relationships in a DGH. For example, here is how **workclass.txt** represents the visual DGH of the **workclass** attribute:

```
Any
	Government
		State-gov
		Local-gov
		Federal-gov
	Private
	Self-employed
		Self-emp-not-inc
		Self-emp-inc
	Unemployed
		Without-pay
		Never-worked
```

DGH in workclass.txt



Visual representation of DGH for workclass

Your first goal should be to read datasets and DGHs in these formats into your program's memory. It is important to note that the **Adult** dataset and its DGHs are provided to you as samples, to help you in development and testing. Your code should not be specific to these inputs – **it should work for any dataset and any set of DGHs!** Hence, do not hardcode any paths, filenames, DGHs, etc. We may test your code with arbitrary datasets and DGHs, and your code should still work.

That being said, you can assume that while the contents of the files may change, their format will remain the same. For example:

- The dataset will always be a csv file and its first row will contain the names of attributes
- For a dataset containing N columns, 1 of those columns will be the SA and the rest will be QIs
- All DGHs will be given to you in one folder under different files, one file for each DGH
- All necessary DGHs will be present and all DGHs will be complete (i.e., no need to check if there is a missing DGH file or if the DGH covers all possible values for an attribute – it does).

The homework assignment consists of several parts. In each part, we give you the names and parameters of the functions you need to implement. **Do not modify function names or parameters (e.g., do not add or remove parameters)!** When grading, we will call exactly these functions with different inputs – if you modify function names or parameters, we won't be able to call your functions the way we are expecting to, and you will lose points.

In addition to the functions you are asked to implement, you are welcome to implement as many additional or helper functions as you like. Try to make your code modular, organized and easy-to-follow. This may help us in giving partial credit to partially correct implementations.

PART 1: COSTS OF ANONYMIZATION [20 pts]

In the lectures, we covered two metrics to measure costs of anonymization: Distortion Metric (MD) and Loss Metric (LM). In this part, you will implement these cost metrics.

cost_MD(*raw_dataset_file*, *anonymized_dataset_file*, *DGH_folder*):

- *raw_dataset_file* is a string containing the path of the raw dataset file, e.g., *"adult-hw1.csv".*
- *anonymized_dataset_file* is the path of the anonymized dataset, e.g., *"adult-anonymized.csv".*
- *DGH_folder* is the directory containing all DGH files.

This function should read the raw dataset and anonymized dataset from the corresponding files and calculate the cost of anonymization using the Distortion Metric. You can assume that the order of the rows and columns are the same in the raw and anonymized datasets. That is: (i) both datasets contain the same attributes in the same order, (ii) the N'th row in the anonymized dataset is the generalized version of the N'th row in the raw dataset. For example:

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Flu
13053	23	American	Flu
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Flu
14850	59	American	Flu

Raw dataset

Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Flu
130**	<30	*	Flu
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Flu
1485*	>40	*	Flu

Anonymized dataset

Notice that in this example, it also happens to be that consecutive records belong to the same equivalence class. This need not always be the case. For example, in the files given to you, rows 1-3-8 could constitute one equivalence class, rows 2-4-5 could constitute another equivalence class, etc.

The return value of the **cost_MD** function should be the MD cost.

cost_LM(*raw_dataset_file*, *anonymized_dataset_file*, *DGH_folder*):

This function follows the same parameters and assumptions as **cost_MD**. But instead of calculating the MD cost, it calculates and returns the Loss Metric (LM) cost. When calculating the LM cost, assume that the weights of all attributes in a dataset are identical, i.e., for a dataset with N attributes, $w_1 = w_2 = \ldots = w_N = 1/N$.

PART 2: RANDOM ANONYMIZER [20 pts]

In this part, you will implement a naive algorithm for k-anonymizing a raw dataset. The algorithm works as follows. Given a dataset ***D*** and the k-anonymity parameter ***k***, the algorithm randomly divides the records in ***D*** into clusters of size ***k***. (If the size of ***D*** is not a perfect multiple of ***k***, then there will be one exceptional cluster that contains between ***k*** and ***2k*** records, so that all records end up belonging to a cluster that contains at least ***k*** records.) Then, the algorithm converts each cluster of records into a k-anonymous equivalence class through generalizations. While doing so, the algorithm should **perform the minimum amount of generalization necessary to achieve k-anonymity for that equivalence class**, i.e., no redundant or over-generalizations!

Implement the random anonymizer under the following function.

random_anonymizer (*raw_dataset_file*, *DGH_folder, k, output_file*):

- *raw_dataset_file* is a string containing the path of the raw dataset file, e.g., *"adult-hw1.csv"*.
- *DGH_folder* is the directory containing all DGH files.
- *k* is the k-anonymity parameter.
- The function has no return value, but it should write the anonymized dataset to a file. The name of this file is passed in a parameter called *output_file*, e.g., *output_file = "anon.csv"*. Order of rows and columns in the *output_file* should be identical to that in *raw_dataset_file,* i.e., N'th row in the anonymized dataset should be the generalized version of the N'th row in the raw dataset.

PART 3: CLUSTERING-BASED ANONYMIZER [20 pts]

Now, let us implement a smarter k-anonymization algorithm based on the idea of clustering similar records together. Before describing the anonymization algorithm, we define a distance metric **dist** that the algorithm will use to measure the "distance" between two records. According to **dist**, the distance between two records is the total MD cost of hypothetically placing the two records in one equivalence class (EC) with the minimum amount of generalization necessary. For example, consider the two records on the left-hand side:

Job	Sex	Age	Disease
Engineer	Male	35	Hepatitis
Lawyer	Male	38	HIV

The distance between these two records is **4** because here is the hypothetical EC that would result from placing these two records together:

Job	Sex	Age	Disease
Professional	Male	[35-40)	Hepatitis
Professional	Male	[35-40)	HIV

As you can see, the Job attribute in both rows have been generalized one level (MD cost: +2) and the Age attribute in both rows have been generalized one level (MD cost: +2), yielding a total distance of 4.

In contrast, the distance between the following two records on the left-hand side is **2**, because the EC that would be constructed with these two records (shown on the right-hand side) has MD cost = 2 due to generalizations in the Age attribute:

Job	Sex	Age	Disease
Engineer	Male	35	Hepatitis
Engineer	Male	38	HIV

Job	Sex	Age	Disease
Engineer	Male	[35-40)	Hepatitis
Engineer	Male	[35-40)	HIV

Given a dataset ***D***, the distance metric **dist** explained above, and the k-anonymity value ***k***, the anonymization algorithm you need to implement is explained in the following figure.

Algorithm 1: Clustering-based anonymizer

Inputs: Dataset D, k-anonymity value k

1 Initially all records in D are "unmarked"
2 **while** *there exist at least* k *unmarked records in* D **do**
3 Pick the first (topmost) unmarked record from D, call this record rec
4 Find the $k-1$ unmarked records from D that are closest to rec according to metric $dist$
5 Create a k-anonymous EC from these records
6 Label all of the used records as "marked"
7 **end**
8 **if** $j > 0$ *unmarked records remain in* D **then**
9 Merge these records with the records in the EC that was constructed last
10 Create a $(k+j)$-anonymous EC from these records
11 **end**

On line 4, you may need to tie-break between multiple records that have the same distance to *rec*. For tie-breaking, you can assume always the topmost records (occurring earlier than others) in the dataset are chosen. By design of the algorithm and the tie-breaking strategy, this anonymizer is deterministic, i.e., it should yield the same result every time it is called with the same inputs.

clustering_anonymizer (*raw_dataset_file*, *DGH_folder, k, output_file*):

- Implement the above algorithm under this function.
- The parameters of this function are identical to those of **random_anonymizer** (Part 2). Similar to **random_anonymizer, clustering_anonymizer** should also preserve the order of the rows when writing the anonymized dataset to *output_file*.

PART 4: TOP-DOWN ANONYMIZER [20 pts]

Recall the "top-down" anonymization approach from our lectures. In the top-down approach, all records in the dataset are initially assumed to be maximally generalized at the root node of the specialization tree. Then, each node of the tree is recursively split into its children via specializations.

The split criteria, i.e., which specialization to perform at each node, is a key factor. If no specializations can be performed at a certain node because all specializations lead to violations of k-anonymity or there is no possible specialization left to perform according to DGHs, then that node stops splitting.

In this part, you will implement this top-down anonymization approach with the following split criteria. Say that you are considering splitting node ***n*** of the specialization tree, where D_n denotes the subset of records at node ***n*** (note: D_n is a subset of dataset ***D***). Furthermore, let $LM(D_n)$ denote the LM cost of D_n. At node ***n***, among the specializations that are possible and do not lead to a violation of k-anonymity, you should pick the specialization that maximizes LM cost improvement. That is, you should pick the specialization ***s**** such that:

$$s^* = \underset{s}{\operatorname{argmax}} \left\{ LM(D_n) - LM(D_{n \to s}) \right\}$$

where $LM(D_{n->s})$ captures what the LM cost of the resulting records would be if the specialization ***s*** was applied to D_n.

Important: Each node in the specialization tree makes its own decision about which specialization to perform. Nodes in the same tree depth need not apply the same specialization.

topdown_anonymizer (*raw_dataset_file*, *DGH_folder, k, output_file*):

- Implement your top-down anonymizer under this function.
- The parameters of this function are identical to those functions in Parts 2 and 3. Again, preserve the order of the rows when writing the anonymized dataset to *output_file*.

PART 5: MINI REPORT [20 pts]

Submit a **max** one-page report (hard limit!) containing your experiments and analysis of the different anonymization algorithms you implemented in Parts 2, 3 and 4.

First, run the 3 anonymizers (random, clustering-based, top-down) on the Adult dataset for different values of k: k = 5, 10, 20, 40, 80, 160, 320. In each run, measure the anonymizer's ***time cost*** (how long does it take to execute), ***LM cost***, and ***MD cost***. You may want to repeat each experiment a few times and average the results in order to improve their statistical significance.

Next, report your experiment results in tables and/or graphs. You should choose which tables or graphs are most suitable for your report. (Hint: **LM cost vs k**, **time vs k, MD cost vs k**).

Finally, briefly discuss your observations and take-aways. For example, what trade-offs exist? Which anonymizer is fastest? Which one has the lowest utility loss? Which anonymizer would you prefer under what settings? Do the results fit your expectations? What did you learn from this assignment?

When grading your report, in addition to correctness, we will pay attention to how you constructed your tables/graphs, their readability and quality, as well as the quality of your analysis and discussion.

BONUS [5 pts]

Perform an analysis similar to the above with 1-2 additional metrics (in addition to time, LM, MD costs). You may consult the lectures, research papers or the Web for anonymization-related metrics. At the end of your report, create a "Bonus" section (**max** half page) that contains:

- Formal definition of your metric(s)
- Your experiment results reported via tables or graphs
- A brief discussion and interpretation of your results

SUBMISSION

When you are finished, submit your assignment via Blackboard as follows:

- Move all of your Python files and your report into a folder named `**your KUNet ID**`.
- **Compress this folder into a single zip file**. (Don't use folder compression methods other than zip, e.g., those that are only available on Mac or Linux.)
- Upload your zip file to Blackboard.

Some reminders:

- After submitting, download your submission and double-check that: (i) your files are not corrupted, (ii) your submission contains all the files you intended to submit, including all of your source code and your report. If we cannot run your code because some of your code files are missing, we cannot give you credit!
- This homework is an **individual assignment**. All work needs to be your own. Submissions will be checked for plagiarism.
- If you borrow material from outside resources, make sure to **cite them properly**.
- **Your report should be readable on any computer. Ideally, it should be a pdf file.** We cannot grade reports that are only readable on a Mac.
- Submit only through **Blackboard**. Do not e-mail your assignment to the instructor or the TAs. Make sure to submit ahead of time to avoid Blackboard-related problems or delays.
- **Do not submit any data files** (such as the Adult dataset, DGHs, or your anonymized datasets).
- Do not change the names or parameters of the functions we will grade.
- If your code does not run (e.g., syntax errors) or does not terminate in a reasonable amount of time, you may receive 0 for the corresponding part.

GOOD LUCK!