Exercise: Privacy-preserving Synthetic Data Generation

(based on Responsible Data Science course by Julia Stoyanovich)

Objectives

In this exercise, you will use the open-source DataSynthesizer library to apply differential privacy mechanism to an input dataset, produce a new “synthetic” dataset that balances utility while meeting strong privacy guarantees.

Since we will not have time in a one-day class to write much code from scratch, you will run a jupyter notebook already provided to you, and answer questions about the output. You should try to get the code running, but don’t get hung up on environment issues: you can use the output provided in the notebook to answer the questions.

After completing this exercise, you will:

1. have explored the interaction between the complexity of the learned model (a summary of the real dataset) and the accuracy of results of statistical queries on the derived synthetic dataset, under differential privacy (goal 1)
2. understand the variability of results of statistical queries under differential privacy, by generating multiple synthetic datasets under the same settings (model complexity and privacy budget), and observing how result accuracy varies (goal 2)
3. explore the trade-off between privacy and utility, by generating and querying synthetic datasets under different privacy budgets, and observing the accuracy of the results (goal 3)
4. learn several useful methods for comparing probability distributions (goals 2 and 3)
5. have an environment for generating synthetic data from any tabular dataset, for future use and reference. (all goals)

DataSynthesizer: DataSynthesizer operates in one of three modes:

1. Random: The synthetic dataset is generated randomly from the type information and range of values in the dataset. (This seems quite safe, but is potentially not differentially private. Why not?)
2. Independent Attribute: Each attribute is modeled separately with a differentially private histogram.
3. Correlated Attribute: A differentially private Bayesian model is derived from the data, then sampled to generate synthetic data.

Setting up:

```
$ git clone git@github.com:billhowe/urbanalyticssummerschool.git
$ cd urbanalyticssummerschool/part1/exercise1
$ git clone https://github.com/DataResponsibly/DataSynthesizer.git
$ jupyter notebook exercise1_code.ipynb
```

(Note: You can complete the assignment without getting the code running, but please try to do so --- you will be better able to use the code in your own projects.)

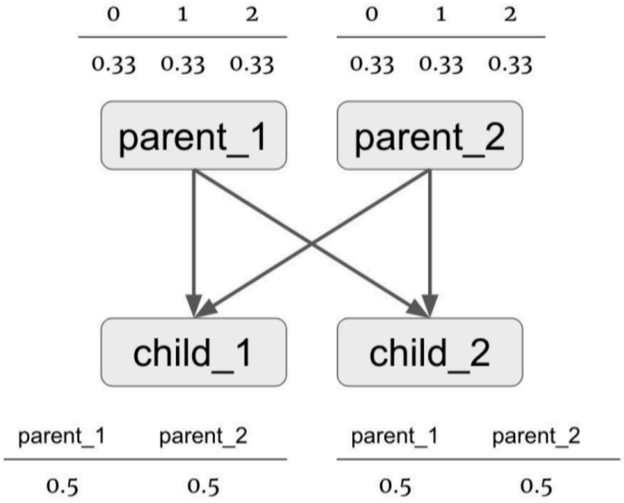
Datasets: You will take on the role of a data owner, who owns two sensitive datasets, called hw_compas and hw_fake, and is preparing to release differentially private synthetic versions of these datasets using the Data Synthesizer library.

The first dataset, hw_compas, is a subset of the dataset released by ProPublica as part of their COMPAS investigation. The hw_compas dataset has attributes age, sex, score, and race, with

the following domains of values: age is an integer between 18 and 96, sex is one of ‘Male’ or ‘Female’, score is an integer between -1 and 10, race is one of 'Other', 'Caucasian', 'African-American', 'Hispanic', 'Asian', 'Native American'.

The second dataset, hw_fake, is a synthetically generated dataset. We call this dataset “fake” rather than “synthetic” because you will be using it as input to the Data Synthesizer. We will use the term “synthetic” to refer to privacy-preserving datasets that are produced as the output of the Data Synthesizer.

We generated the hw_fake dataset by sampling from the following Bayesian network:



In this Bayesian network, parent_1, parent_2, child_1, and child_2 are random variables. Each of these variables takes on one of three values {0, 1, 2}.

- Variables parent_1 and parent_2 take on each of the possible values with an equal probability. Values are assigned to these random variables independently.
- Variables child_1 and child_2 take on the value of one of their parents. Which parent’s value the child takes on is chosen with an equal probability.

Getting Started: Review cells 3, 4, 5, and 6. These cells generate synthetic datasets using each mode of datasynthesizer for both sample datasets.

Goal 1

Review the cells in the notebook under Goal 1.

Q1: This cell simply computes descriptive statistics for the hw_compas dataset. **Briefly explain any substantial differences between the methods.**

Q2: These cells compare the relative performance (accuracy) of random mode (A) and of independent attribute mode (B) using plots of the distributions of values of age and sex attributes in hw_compas and in synthetic datasets generated under settings A and B. **Compare the histograms visually, and comment on the relative accuracy between the methods.**

This section also computes cumulative measures that quantify the difference between the probability distributions over age and sex in hw_compas vs. in privacy-preserving synthetic data. To do so, we use the Two-sample Kolmogorov-Smirnov test (KS test) for the numerical attribute and Kullback-Leibler divergence (KL-divergence) for the categorical attribute, using provided functions ks_test and kl_test. **Discuss the relative difference in performance under A and B under these measures.**

Q3 (hw_fake only): these cells compare the accuracy of the correlated attribute mode with k=1 (C) and with k=2 (D). We display the pairwise mutual information matrix using heatmaps, showing mutual information between all pairs of attributes, for the original hw_fake and for the two synthetic datasets (generated under C and D). **Briefly discuss how well or how poorly mutual information is preserved in synthetic data.**

Goal 2 (hw_compas only)

We wish to study the variability in accuracy of answers to Q1 under goal 1 for A, B, and C for attribute age. We fix epsilon = 0.1, generate 10 synthetic datasets by specifying different seeds, and plot accuracy as a box-and-whiskers plot. **Consider which mode gives more accurate results and why. In which cases do we see more or less variability?**

Goal 3 (both datasets)

Consider how well statistical properties of the data are preserved in as a function of the privacy budget. To improve robustness, the experiment aggregates 10 different synthetic datasets (with different seeds) for each value of epsilon, for each data generation setting (B, C, and D).

We compute the following metrics and visualize the results:

- KL-divergence over the attribute race in hw_compas, varying epsilon from 0.01 to 0.1 in increments of 0.01, generating synthetic datasets under B, C, and D.
- The difference in pairwise mutual information, aggregated (summed up) over all pairs of attributes, for both hw_compas and hw_fake, computed as follows:

Let D be the sensitive dataset and D' is its privacy-preserving synthetic counterpart.
Let m_ij represent the mutual information between attributes and i and j derived from D, and m'_ij represent the mutual information between the same two attributes, i and j, derived from some D'.

We compute the sum of the absolute value of the difference between m_ij and m'_ij, over all pairs i,j, with i<j.

We run these experiments for the following epsilon values: 0.0001. 0.001, 0.01, 0.1, 1, 10, and 100, generating synthetic datasets under B, C and D.

We then generate 3 plots, one for each data generation method (i.e., one plot for B, one for C, and one for D). The y-axis in all cases starts at 0. All plots have the same range of y-axis values, so that the values are comparable across experiments.

Discuss the trend for both metrics.

What to turn in:
Enter your answers in this google form:
https://forms.gle/Am5j6Nk5NU1zKZiF7