Differential Privacy Temporal Map Challenge (DeID2)

Sample Privacy Write-up

Title: Baseline NMD-Privatizer
Team: KRC

The first sprint in the Temporal Map Data Challenge uses the Baltimore Police Data. The 2019 data is provided as publicly available data for use during the algorithm design and development phase, but the final scoring will occur over a private data set on the same schema. The required output data is (non-zero, integer) counts of event types aggregated by each neighborhood and month.

Main Idea:
This section has a short, high-level description of the algorithm and privatization strategy.

The baseline NMD-Privatizer builds a histogram counting the number of events in each possible combination of Neighborhood (N), Month (M), and event Description (D). The algorithm then privatizes these counts by applying Laplace mechanism.

Pre-processing:
*This section covers pre-processing steps that have a sensitivity cost of zero. This means they either **(a)** don't interact with the ground truth input data at all (ex: using the input data dictionary/schema/etc to initialize an empty histogram) or **(b)** process each individual's records separately, without reference to any other individuals' records.*

First, an empty histogram data structure is created with bins for every combination of Neighborhood (N), Month (M), and event Description (D). To do this, we use the schema provided in parameters.json. Because this step doesn't interact with the input data, it has a sensitivity cost of 0.

Privatization and Privacy Proof:
This section covers the privatization process, including all algorithm steps that have a non-zero sensitivity cost. Every algorithm step must clearly account for its sensitivity cost and the privacy proof must demonstrate that sufficient privatization noise has been added to cover the given sensitivity by the completion of the privatization process.

The algorithm goes through the input data and for each event record, increments the NMD histogram bin for the correct neighborhood, month and event description. When this step is completed, we have a full histogram of NMD counts for the input dataset.

Because we're working with temporal data in which a single person can be associated with multiple records, we have to consider *max_records_per_individual* when computing the sensitivity of this histogram.

Recall that the sensitivity of a function F is defined as:

$$\Delta F = max \| F(D_1) - F(D_2) \|_1$$

Where D_1 and D_2 are two data-sets that differ by one individual, and $\|.\|_1$ is the l_1 norm. Intuitively, the l_1 norm is the sum of all differences across all outputs of the algorithm step. For a histogram, in our case, it's the total sum of absolute changes in bin counts, across all bins in the histogram, that occurs when we add or remove one person from the data set. Because one person can contribute to at most *max_records_per_individual* event records (which may be spread out across various histogram bins), adding a person or removing them from the dataset will cause changes to bin counts that sum up to at most *max_records_per_individual*. So the sensitivity of building our histogram is *max_records_per_individual*.

Because this histogram construction is the only time the NMD-baseline interacts with the ground truth input data, this is the only step of our algorithm with a non-zero sensitivity cost, and thus the total sensitivity of our algorithm is *max_records_per_individual.*

Differential privacy is achieved by adding Laplacian noise with scale = *max_records_per_individual / epsilon* to every bin in the NMD histogram (see more about the Laplace Mechanism here: 3.2 The Laplace mechanism).

Post-processing:

Post-processing steps must act only on noisy, privatized data; they cannot interact with the raw input data in any way. This ensures that post-processing steps also incur a sensitivity cost of zero.

The submission rules require the final privatized NMD counts to be positive integers. So, as post-processing, we take our privatized counts and set all negative values equal to 0, and round all values to integers.