PrivacyHistos Privacy Proof and Code Guide

Description & Benefits

PrivacyHistos is a set of generic open source functions to allow anyone to easily combine features and privatize histograms using any dataset. The benefits include the following:

1) Provide a simple and easily understood method for privatization.
2) Provide reasonable accuracy.
3) Reduce the amount of code needed for privatization.
4) Increase the speed of privatizing data.
5) Reduce coding errors

Specifications

Input

The input consists of four items:

1) Formatted Ground Truth Data (ground_truth)
    a. No schema required
    b. All data must be encoded as positive integers with no Nans
    c. The column names must be appended as follows:
        i. _c – indicates a categorical feature
        ii. _n – indicates a numeric feature
        iii. _i – One, and only one, column must be the individual identifier
        iv. _x – indicates a feature to include but not used as a combination feature
2) Sensitivity & Epsilon
    a. Sensitivity = (number of histograms * sample size) + population queries
        i. Number of histograms – the total number of histograms which may include other histograms not using the combined column approach.
        ii. Sample size – the number of individual records to use for building the histograms.
        iii. Population queries – the total number of population queries used in the overall approach.
    b. Epsilon – the epsilon value
3) Combination Dictionary (combo_dict)
    a. Format - {combo_name: column_list}
    b. combo_name – name of the the new combined feature
    c. column_list – a list of columns to combine (categorical and/or numeric)
    d. Example - {‘cp’: [‘company_c’, ‘payment_c’]}
    e. Explanation – Create a column named ‘cp’ combining categorical fields ‘company_c’ and ‘payment_c’
    f. Requirements – You cannot combine categorical columns with numeric columns

4) Number Dictionary (num_dict)
    a. Format - {column_name: [top_value, increment, max_range]}
    b. column_name – the name of the numeric feature
    c. top_value – the top value to use with the increment
    d. increment – the amount in each bin
    e. max_range – the maximum value of the numeric range
    f. Example – {‘fare_n’: [50, 5, 100]}
    g. Explanation – Create 11 numeric bins for numeric feature ‘fare_n’ with bins of 5 from 0-50 (10 bins of 5) with numbers over 50 in a bin of 50-100 (1 bin of 50).

Functions

There are five public functions:

1) **check_input(ground_truth, combo_dict, num_dict)**
   Tests the formatted ground truth data, the combination dictionary and the number dictionary for proper formatting and values.
    a. ground_truth – the formatted ground truth dataframe
    b. combo_dict – the combination dictionary
    c. num_dict – the numeric dictionary
    d. returns 1 if True

2) **pre_process(ground_truth, combo_dict, num_dict)**
   Pre-processes the ground truth data by combining the columns per the combination dictionary and number dictionary
    a. ground_truth – the formatted ground truth dataframe
    b. combo_dict – the combination dictionary
    c. num_dict – the numeric dictionary
    d. returns a new dataframe with combined columns (df)
    e. returns a numeric decoder dictionary (num_decode)
    f. returns a column decoder dictionary (col_decode)

3) **histo_test(df, combo_dict)**
   Counts the number of bins created for each combined column in the pre-processed dataframe. A useful tool for determining the optimal feature combinations.
    a. df – the pre-processed dataframe
    b. combo_dict – the combination dictionary
    c. prints a count of bins for each combined column

4) **create_private_histo(df, column, sample_size, sensitivity, epsilon)**
   Creates privatized histograms of the specified combined column using a specified sample size, sensitivity and epsilon.
    a. df – the pre-processed dataframe
    b. column – the combined column
    c. sample – enter 1 to use sampling

d. sample_size – the size of the sample
e. sensitivity – the sensitivity
f. epsilon – the epsilon
g. returns population (bins) as a list
h. returns privatized weights as a list

5) **col_decoder(num_dict, num_decode, col_decode, value, column)**
Decodes the specified column into a its privatized value.
a. num_dict – the numeric dictionary
b. num_decode – the numeric decoder dictionary
c. col_decode – the column decoder dictionary
d. value – a value to decode
e. column – the column to decode
f. returns decoded privatized value

Privacy Proof - Sensitivity

The *max_records_per_individual* is the number of individuals used to calculate the weights when computing the sensitivity of queries.

The sensitivity of function F is defined as: **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. For the histograms the l_1 norm is the total sum, or the absolute changes in bin counts, across all bins in the histogram, that occurs when we add or remove one individual from the data set.

A *population query* represents the number of queries counting a particular quantity (e.g. number of total records).

The *epsilon* value represents the privacy loss parameter and can be viewed as a measure of the additional privacy risk an individual could incur beyond the risk incurred in the opt-out scenario.

The formula for calculating the sensitivity for building the simulated data using histograms is:

((max_records_individual x number of histograms) + population query) / epsilon

Code Guide – privacy.py

check_input – lines 16-90
Tests the formatted ground truth data, the combination dictionary and the number dictionary for proper formatting and values. As a pre-processing data testing step no privatization is performed.

pre_process – lines 225-229
Pre-processes the ground truth data by combining the columns per the combination dictionary and number dictionary. As a pre-process step no privatization is performed.

histo_test – lines 291-294
Counts the number of bins created for each combined column in the pre-processed dataframe. A useful tool for determining the optimal feature combinations. As a reporting feature no privatization is performed.

create_private_histo – lines 308-318
Creates privatized histograms of the specified combined column using a specified sample size, sensitivity and epsilon.

The bins and are created in lines 309-314 and passed to the **weight** function along with the pre-processed data (df), combined column (col), and privacy parameters (sample_size, sensitivity and epsilon).

The **weight** function in lines in lines 258-278 creates the privatized weights applying noise using the Laplace Mechanism on line 273.

col_decoder – lines 332-356
Decodes the specified column into a its privatized value. As a post-processing step no privatization is performed.

Privatization Example

Using the Detroit example included in the repo, the sensitivity is calculated as follows:

sample size = 1 (maximum records per individual)
population queries = 1 (1 count of total incidents)
epsilon = 1.0
number of histograms = 4 (4 combined columns)

((max_records_individual x number of histograms) + population query) / epsilon

Sensitivity = ((1 x 4) + 1)/1.0 = 5