Differential Privacy for Data Analysis on Healthcare Data

Data Warehousing Team – Authors: Mia Pham & Frank Dinh

Example analysis can be found at: https://github.com/thuytruc1121/Differential-Privacy-on-NHANES.git

Overview

Healthcare data contains highly sensitive personal information. While analysing such data is crucial for improving public health outcomes, privacy risks must be carefully managed. Differential Privacy (DP) offers a mathematically rigorous method to ensure individual data protection while enabling useful statistical analysis.

This document introduces the concept of differential privacy, explains its importance in healthcare, and provides practical tools for implementing DP in data workflows using R and Python.

1. What is Differential Privacy?

Differential Privacy is a framework that ensures strong privacy guarantees by making the results of any analysis nearly indistinguishable whether or not an individual's data is included.

Simplified Definition:
A randomized algorithm is differentially private if, for any two datasets that differ in a single individual, the probability of any output is almost the same.

Mathematical Formulation:
A mechanism M satisfies (ε, δ)-differential privacy if for all neighboring datasets D_1 and D_2, and for all outputs S:

$$P[M(D_1) \in S] \leq e\text{^}\varepsilon * P[M(D_2) \in S] + \delta$$

- M: DP mechanism
- D_1, D_2: Neighboring datasets
- ε: Privacy budget (smaller = stronger privacy)
- δ: Probability of privacy breach

2. Why is Differential Privacy Important for Healthcare Data?

Healthcare datasets often contain personal, identifiable, and sensitive information, such as:
- Electronic Health Records (EHRs)

- Genomic data
- Insurance claims
- Clinical trial results

Real-World Use Case: Redback Healthcare Analytics Project
In the Redback project, healthcare data is analysed to support planning, monitor KPIs, and improve patient outcomes. This includes analysing hospital admissions, discharge trends, and outpatient services. However, using individual-level data raises serious privacy concerns.

Benefits of Applying DP:
- Enables valuable statistical analysis without compromising privacy
- Reduces legal and ethical risks for data custodians
- Builds public trust and encourages data sharing
- Supports decentralized and federated learning models

3. Key Differential Privacy Mechanisms

3.1 Laplace Mechanism

Adds noise from the Laplace distribution to numeric outputs like counts or means.
Example: If the true average patient wait time is 32 minutes, the released value may be 31.4 or 32.7 due to added noise.

3.2 Exponential Mechanism

Suitable for non-numeric outputs. It chooses an output based on a utility function, balancing utility and privacy.
Example: Selecting the best hospital discharge strategy based on utility scores, where higher-utility options are more likely to be selected.

4. Tools and Packages for Differential Privacy

4.1 R: diffpriv Package

- Website: CRAN - diffpriv
- Supports Laplace and Exponential mechanisms
- Good for prototyping and education

Example in R:

```
library(diffpriv)
mech <- DPMechLaplace(target = function(X) mean(X), sensitivity = 1)
mech <- releaseResponse(mech, X = c(120, 130, 150))
print(mech$response)
```

4.2 Python: diffprivlib (Recommended)

- Website: https://pypi.org/project/diffprivlib/
- Developed by IBM, supports integration with numpy and pandas

- Well-maintained and compatible with Python 3.8–3.11

Installation: pip install diffprivlib

Example:

```
import numpy as np
from diffprivlib.tools import mean

ages = np.array([29, 34, 42, 50, 31, 28, 40])
dp_mean = mean(ages, epsilon=1.0, bounds=(0, 100))
print(f"Differentially Private Mean: {dp_mean:.2f}")
```

4.3 Python: PyDP (Google's DP Library – Use with Caution)

- Website: https://pydp.readthedocs.io/
- Best used in Google Colab due to compatibility issues

Installation: !pip install python-pydp

Example:

```
from pydp.algorithms.laplacian import BoundedMean
dp_mean = BoundedMean(epsilon=1.0, lower_bound=0, upper_bound=100)
print(dp_mean.result([29, 34, 42, 50, 31, 28, 40]))
```

5. Walkthrough for Using diffprivlib

Step 1: Install and Import

```
pip install diffprivlib

import pandas as pd
from diffprivlib.tools import mean, median, var
```

Step 2: Prepare Data

```
data = pd.Series([120, 130, 128, 125, 132, 110, 122])
data_clipped = data.clip(lower=80, upper=180).to_numpy()
```

Step 3: Apply Private Statistics

```
dp_mean = mean(data_clipped, epsilon=0.5, bounds=(80, 180))
print("DP Mean:", dp_mean)
dp_var = var(data_clipped, epsilon=0.5, bounds=(80, 180))
print("DP Variance:", dp_var)
```

Step 4: Plot Results

```
import matplotlib.pyplot as plt

plt.hist(data_clipped, bins=10, alpha=0.6, label='Original Data')
plt.axvline(dp_mean, color='red', linestyle='--', label='DP Mean')
plt.legend()
plt.title("Differential Privacy with diffprivlib")
plt.show()
```

6. Google Differential Privacy Library

- GitHub: https://github.com/google/differential-privacy
- Implements in C++, Java, and Python
- Suitable for large-scale use
- Includes utilities for noise calibration, aggregation, and metrics

Summary

Differential Privacy provides essential protection for individuals while enabling responsible and ethical healthcare data analysis. Tools like diffpriv, diffprivlib, and PyDP make it accessible for researchers and data scientists.

To get started:
- Begin with the Laplace and Exponential mechanisms
- Experiment with different values of ε and δ
- Progress toward advanced applications like local differential privacy and federated analytics

Projects like the Redback Healthcare Analytics Project demonstrate the practical value of DP in real-world data initiatives.