KTH Stockholm
Department of Numerical Analysis

Privacy-Preserving Machine Learning

lorem ipsum upsum

Sijun John Tu

Master Thesis Report

Supervisors: Anders Szepessy (KTH) and Shahid Raza (RISE)

Abstract

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

December 21, 2023

Sijun John Tu

Contents

Notation

Mathematical conventions and notation used in this thesis:

$\mathbb{R}$	the real numbers

Additionally, we introduce the following conventions to describe various elements from different mathematical objects to make the notations and their meaning as consistent as possible:

a, b, c	scalar values
A, B, C	matrices

List of Figures

1 Introduction

2 Theoretical background on Differential Privacy

In this chapter we briefly describe and derive the most important results from Cynthia Dwork's work on differential privacy that was first introduced in ??. This summary heavily relies on her writings in her as well as lecture notes from ????.

2.1 Defining differential privacy

Differential privacy (DP) should be understood as an agreement between the data holder and the data subject: the latter should not be "affected, adversely or otherwise, by allowing [her] data to be used in any study or analysis, no matter what other studies, data sets or information sources are available". This addresses the paradox of learning something useful about a population while learning nothing about the individuals

Example 2.1.1 (Randomised response)**.** citation needed ??? proposes the following random answering procedure: In a study where participants are asked to answer with "Yes" or "No" whether they have engaged in an illegal or embarrassing activity A, they should:

1. Flip a coin
2. If the coin shows tails, then the participant should respond truthfully.
3. If the coin shows head, then the participant should flip the coin a second time and answer "Yes" if the second coin shows head and "no" otherwise.

This procedure ensures participants' privacy by "plausible deniability"; each participant's answer has non-zero probability of being truthful or not. By understanding the probabilities of the noise generation process, the data analyst can estimate the true number of "yes" and "no" answers. To this end, let p b the true percentage of "yes" answers, N the total number of participants, n_{true} the true number of "yes" responses and $\hat{n}_{obs}$ the observed number of "yes" responses. We assume a fair coin with equal probability of showing heads or tails. Then the expected number of "yes" answers after applying the described procedure is:

$$\mathbb{E}(''Yes'') = \frac{1}{4}n_{true} + \frac{1}{4}(N - n_{true}) + \frac{1}{2}n_{true} = \frac{1}{4}N + \frac{n_{true}}{2} \tag{1}$$

We can estimate this using the $\hat{n}_{obs} \approx \mathbb{E}(''Yes'') = \frac{1}{4}N + \frac{n_{true}}{2}$ and finally solving for n_{true} yields the estimate:

$$n_{\hat{true}} = 2\hat{n}_{obs} - \frac{1}{2}N \tag{2}$$

Definition 2.1.2 (Probability Simplex)**.** Given a discrete set B, the probability simplex over B is defined as the set

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|}, x_i \geq 0 \text{ and } \sum_i x_i = 1 \right\} \tag{3}$$

Definition 2.1.3 (Randomised Algorithm)**.** A randomized algorithm $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \to \Delta(B)$. On input $a \in A$ algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$

Definition 2.1.4 (Histogram representation of a data base)**.** Given a set $\mathcal{X}$, the universe of all possible records, the histrogram representation of a database x is the vector

$$x \in \mathbb{N}^{|\mathcal{X}|} \tag{4}$$

in which each entry x_i represents the number of elements in database x of type $i \in \mathcal{X}$.

Definition 2.1.5 (l_1-norm of a database in histogram representation)**.** The l_1-norm of a database is a measure of the size of the database and defined as:

$$||x||_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i| \tag{5}$$

This immediately gives rise to a notion of distance between two databases x and y, namely:

$$||x - y||_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i - y_i| \tag{6}$$

which basically counts the number of different entries.
Now we are ready to give the general definition of differential privacy:

Definition 2.1.6 ((ϵ, δ)-DP)**.** A randomised algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ)-differentially private if for all outcomes $S \subset ran\mathcal{M}$ and for all databases $x, y \in \mathbb{N}^{|\mathcal{X}|}$, such that $||x - y||_1$ (i. e. they only differ in one element) we have

$$\mathbb{P}(\mathcal{M}(x) \in S) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(y) \in S) + \delta \tag{7}$$

where the probability is taken over the randomness of $\mathcal{M}$. If $\delta = 0$, we say $\mathcal{M}$ is ϵ-differentially private.

why e^{ϵ}

Example 2.1.7 (Randomised response revisited)**.**

2.2 Important results for Differential Privacy

Theorem 2.2.1 (DP requires randomisation)**.** *Any non-trivial DP-mechanism requires randomisation.*

Proof. TBA □

Theorem 2.2.2 (Post-processing)**.** *Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomised algorithm that is (ϵ, δ)- DP. Further let $f : R \to R'$ an arbitrary function. Then $f \circ \mathcal{M}$ is also (ϵ, δ) -DP.*

Proof. First fix data sets $x, y \in \mathbb{N}^{|\mathcal{X}|}$, s. t. $||x - y||_1 \leq 1$ and outcome $S' \subseteq R'$. Define a set $S = \{r \in R : f(r) \in S'\}$. Then we have:

$$\begin{aligned}\mathbb{P}(f(\mathcal{M}(x)) \in S') &= \mathbb{P}(\mathcal{M}(x) \in S) \\ &\leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(y) \in S) + \delta \\ &= e^{\epsilon} \cdot \mathbb{P}(f(\mathcal{M}(y)) \in S') + \delta \end{aligned} \tag{8}$$

where the inequality follows from the $(\epsilon, \delta) - DP$ of $\mathcal{M}$. □

Theorem 2.2.3 (Group privacy)**.** *Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomised algorithm that is (ϵ, δ)- DP, then $\mathcal{M}$ is $(k\epsilon, ke^{k\epsilon}\delta)$- DP for groups of size k, i. e. it holds for databases $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $||x - y||_1 \leq k$ and for all $S \subseteq R$:*

$$\mathbb{P}(\mathcal{M}(x) \in S) \leq e^{k\epsilon} \cdot \mathbb{P}(\mathcal{M}(y) \in S) + k\delta \tag{9}$$

Proof. First fix data sets $x, y \in \mathbb{N}^{|\mathcal{X}|}$, s. t. $||x - y||_1 \leq k$ and outcome $S \subseteq R$. Now there exists a series of databases $z_0, ..., z_k$, such that $x = z_0$ and $y = z_k$ and $||z_{i+1} - z_i||_1 \leq 1$, i. e. we can find a series of databases that transforms x into y by removing or adding one record at a time. Then we have:

$$\begin{aligned}
\mathbb{P}(\mathcal{M}(x) \in S) &= \mathbb{P}(\mathcal{M}(z_0) \in S) \\
&\leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(z_1) \in S) + \delta \\
&\leq e^{\epsilon} \left(e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(z_2) \in S) + \delta\right) + \delta \\
&\leq ... \\
&= ke^{\epsilon} \cdot \mathbb{P}(f(\mathcal{M}(y)) \in S') + ke^{k\epsilon}\delta
\end{aligned} \tag{10}$$

□

Theorem 2.2.4 (Standard composition)**.** *Let $\mathcal{M}_1 : \mathbb{N}^{|\mathcal{X}|} \to R_1$ and $\mathcal{M}_2 : \mathbb{N}^{|\mathcal{X}|} \to R_2$ be two randomised algorithms that are (ϵ_1, δ_1)- and (ϵ_2, δ_2) DP, then their composition defined by $\mathcal{M}_{12} : \mathbb{N}^{|\mathcal{X}|} \to R_1 \times R_2$, $\mathcal{M}_{12}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ DP.*

Proof. TBA □

2.3 Example of DP-mechanism: Laplace mechanism

3 Time Series data generation

3.1 Overview

3.2 DP-MERF

DP-MERF [HAP20] is an efficient all purpose data generation algorithm that is based on minimising the so-called Maximum Mean Discrepancy between the real and the synthetic data distributions. The authors mainly verfied their results using tabular data like ????, but also image data, notably the MNIST ???CITE data set. It has not been used for time series data, but we will consider this data generation for generating time series data in this thesis, because accorrding to a recent survey [Hu+23], DP-MERF delivers the best all purpose data generation performance.

3.2.1 Maximum Mean Discrepancy

There are different ways to measure the "distance" between two distributions P and Q. On popular metric is the Maximum Mean Discrepancy (MMD) between P and Q, where the random variables are projected into another feature space and the expected values are compared to each other in this space.

Definition 3.2.1.1 (MMD)**.** Let $\phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a reproducing kernel hilbert space (RKHS) and P and Q some distributions over $\mathcal{X}$ and random variables $X \sim P$, $Y \sim Q$ given. Then the Maximum mean Discrepancy is defines as:

$$MMD(P, Q) = ||\mathbb{E}[\phi(X)] - \mathbb{E}[\phi(Y)]||_{\mathcal{H}} \tag{11}$$

Some "easy" features maps ϕ are for example:

Example 3.2.1.2. Let P and Q some distributions over $\mathcal{X}$ and random variables $X \sim P$, $Y \sim Q$ given.

- **Identity kernel**: $\mathcal{X} = \mathcal{H} = \mathbb{R}^d$ and $\phi(x) = x$, then we have:

$$\begin{aligned} MMD(P, Q) &= ||\mathbb{E}[\phi(X)] - \mathbb{E}[\phi(Y)]||_{\mathcal{H}} \\ &= ||\mathbb{E}[X] - \mathbb{E}[Y]||_{\mathbb{R}^d} \end{aligned} \tag{12}$$

  So we only compare the two distributions in terms of their means.

- **Quadratic kernel**: $\mathcal{X} = \mathbb{R}$ $\mathcal{H} = \mathbb{R}^2$ and $\phi(x) = (x, x^2)$, then we have:

$$\begin{aligned} MMD(P, Q) &= ||\mathbb{E}[\phi(X)] - \mathbb{E}[\phi(Y)]||_{\mathcal{H}} \\ &= ||\mathbb{E}[(X, X^2)] - \mathbb{E}[(Y, Y^2)]||_{\mathcal{H}} \\ &= || \begin{pmatrix} \mathbb{E}[X] \\ \mathbb{E}[X^2] \end{pmatrix} - \begin{pmatrix} \mathbb{E}[Y] \\ \mathbb{E}[Y^2] \end{pmatrix} ||_{\mathbb{R}^2} \\ &= \sqrt{(\mathbb{E}[X] - \mathbb{E}[Y])^2 - (\mathbb{E}[X^2] - \mathbb{E}[Y^2])^2} \end{aligned} \tag{13}$$

  So here we compare the two distributions in terms of their means and their variance (or first and second moments respectively).

- **Gaussian kernel** ????

Now instead of computing a possibly high or even infinite dimensional transformation ϕ one can use the well-known kernel trick ????REF. Let $k(x, y) =< \phi(x), \phi(y) >_{\mathcal{H}}$ be a kernel with corresponding reproducing kernel hilbert space $\mathcal{H}$, then the computation of the MMD simplifies to:

$$\begin{aligned}
MMD^2(P, Q) &= ||\mathbb{E}[\phi(X)] - \mathbb{E}[\phi(Y)]||^2_{\mathcal{H}} \\
&=< \mathbb{E}[\phi(X)], \mathbb{E}[\phi(X')] > - < \mathbb{E}[\phi(X)], \mathbb{E}[\phi(Y)] > - < \mathbb{E}[\phi(Y)], \mathbb{E}[\phi(X)] > \\
&\qquad\qquad + < \mathbb{E}[\phi(Y)], \mathbb{E}[\phi(Y')] > \\
&= \mathbb{E}[< \phi(X), \phi(X') >] - 2\mathbb{E}[< \phi(X), \phi(Y) >] + \mathbb{E}[< \phi(Y), \phi(Y') >] \\
&= \mathbb{E}[k(X, X')] - 2\mathbb{E}[k(X, Y)] + \mathbb{E}[k(Y, Y')] \qquad (14)
\end{aligned}$$

Where we introduced independent random variables $X, X' \sim P$, $Y, Y' \sim Q$.
Now given a training data set $X_m = \{x_i\}_{i=1}^m \sim P$ and a synthetic data set $X'_m = \{x_i\}_{i=1}^m \sim Q$ we can estimate their MMD^2 by estimating the expected value with a mean estimate:

$$\widehat{MMD}^2(X_m, X'_m) = \frac{1}{m^2} \sum_{i,j=1}^{m} k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j=1}^{m} k(x'_i, x'_j) - \frac{2}{m^2} \sum_{i,j=1}^{m} k(x_i, x'_j) \qquad (15)$$

Unfortunately, this will require $\mathcal{O}(m^2)$ computations which grows quadratically in the number of samples. This will be too big for a large training data set. As a remedy, the authors of [HAP20] propose to use Random Fourier Features based on a paper from 2007 [see RR07], to approximate the kernel k using its fourier transform and Monte-Carlo-Simulation. Thus,

$$k(x, y) \approx \hat{\Phi}(x)^T \hat{\Phi}(x') \qquad (16)$$

where $\hat{\Phi}(x) \in \mathbb{R}^D$ and $\hat{\Phi}_j(x) = \sqrt{\frac{2}{D}} cos(\omega_j^T x)$.

3.3 GAN based

4 Experiment

Appendix

References

[HAP20] Harder, F., Adamczewski, K., and Park, M. “Differentially Private Mean Embeddings with Random Features (DP-MERF) for Simple & Practical Synthetic Data Generation”. In: *CoRR* abs/2002.11603 (2020). arXiv: `2002.11603`. URL: `https://arxiv.org/abs/2002.11603`.

[Hu+23] Hu, Y., Wu, F., Li, Q., Long, Y., Garrido, G. M., Ge, C., Ding, B., Forsyth, D., Li, B., and Song, D. *SoK: Privacy-Preserving Data Synthesis*. 2023. arXiv: `2307.02106 [cs.CR]`.

[RR07] Rahimi, A. and Recht, B. “Random Features for Large-Scale Kernel Machines”. In: *Advances in Neural Information Processing Systems*. Ed. by Platt, J., Koller, D., Singer, Y., and Roweis, S. Vol. 20. Curran Associates, Inc., 2007. URL: `https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.