Differentially Private Small Dataset Release Using Random Projections

Scalable Algorithms for Data Analysis

K Praneeth Nayak S Jeevan

April 25, 2023

Contents

0.1 DPRP 2
0.2 DPRP OVERVIEW 2
0.3 PRIVACY GUARANTEES OF DPRP 4
0.4 ANALYSIS 7

0.1 DPRP

We first introduce DPRP - Differentially Private data release via Random Projections, our proposed method for releasing differentially private small datasets. Then we proceed to state and prove DPRP's formal privacy guarantees.

0.2 DPRP OVERVIEW

DPRP takes inspiration from non-private image compression and reconstruction techniques [12, 13] based on the low-rank approximation, and further extends the idea for the differentially private reconstruction of small tabular datasets. DPRP constitutes a model-free approach, whereby no parameter estimation of any sort is required, leading to minimal hyperparameter tuning and no iterative learning process. Due to its reconstruction based nature, DPRP works extremely well on small datasets (a performance bottleneck for current state-of-the-art). We present DPRP succinctly as Algorithm 1 and provide a line by line walkthrough for the readers.

Algorithm 1: DPRP: Differentially Private Reconstruction of Input Data

Input: Dataset:X; Privacy parameters: ϵ, δ; Privacy budget allocation: $b_1\%$ for random projection P, $1 - b_1\%$ for SVD(X_C); Number of dimensions for random projection P: k_1; Number of values from right singular vector to keep from SVD(X_C): k_2

Output: Differentially private dataset: X'

1 $R \sim \mathcal{N}(0, 1/\sqrt{k_1})^{d \times k_1}$

2 $P = XR$

3 $P' = P + M_1; M_1 \sim \mathcal{N}(0, \sigma_1^2)^{n \times k_1}$ // With budget $b_1\%$

4 $X_C = X^T X$

5 $\hat{V}'\hat{\Sigma}'\hat{V}'^T = \text{SVD}(X_C + M_2); M_2 \sim \mathcal{N}(0, \sigma_2^2)^{d \times d}$ // With budget $1 - b_1\%$

6 $V'_{k_2} = \hat{V}'[1, \cdots, k_2]$ // First k_2 columns

7 $X' = P'(\hat{V}'^T_{k_2} R)^+ \hat{V}'^T_{k_2}$

To start, the user provides the dataset $X^{n \times d}$ as an input to DPRP, along with the overall privacy budget, ϵ, δ; the allocation of the privacy budget, that is the share of the privacy budget for making the random projection P differentially private ($b_1\%$) and the share of the privacy budget for the differentially private SVD ($1 - b_1\%$); dimensionality of random projection, P, k_1; and the number of values from the right singular vector from $\hat{V}$ to use for the reconstruction, k_2.

Line 1 - 2 (Creating random projection): We start with creating the random projection, where we create a random matrix $R^{d \times k_1}$ with entries drawn from $\mathcal{N}\left(0, 1/\sqrt{k_1}\right)$ and create the projection $P^{n \times k_1} = XR$. We have to remember that up to this point, we have not made any differential privacy claims, so P still contains sensitive information from X. Line 3 (Differential privacy of P): To ensure differential privacy of P, we add a noise matrix M_1 ($P' = P + M_1$). Specifically, $M_1 \sim \mathcal{N}(0, \sigma_1^2)$ for some σ_1. Where σ_1 is chosen using Theorem 2.

Line 4-6 (Differential privacy of $SVD(X_C)$): For the reconstruction of X, we only need the right singular vector of decomposed X. But as X contains sensitive information, so will the right singular vector from decomposed X. Making the right singular vector differentially private is non-trivial. We do

not directly add noise to the right singular vector as it can lead to an overly noisy result and the right singular vector does not directly relate to the "per-user" principle of differential privacy. We follow a different approach [17], where we first calculate the covariance matrix $\left(X_C = X^T X\right)$, and then add noise to ensure the differential privacy of the covariance matrix $(X'_C = X_C + M_2)$, where $M_2 \sim \mathcal{N}\left(0, \sigma_2^2\right)$ and σ_2 is chosen according to Theorem 2. Then we perform the singular value decomposition on X'_C and choose the first k_2 values from the right singular vector $\left(V'_{k_2}\right)$.

Line 7 (Noisy reconstruction): Now, for the main part, we perform our noisy reconstruction of X. "+" refers to the Moore-Penrose pseudoinverse. It is noteworthy that only noisy P' and $\hat{V}'_{k_2}$ are required for the reconstruction, which we have earlier made differentially private, in addition to a random matrix R, which does not have any real data, leading to a differentially private reconstruction, X'.

We discuss some aspects of Algorithm 1 in Section 0.4. But, first, we provide the differential privacy guarantees of our reconstruction, as it remains to be shown that adding noise (M_1, M_2) to (P, X_C), and reconstructing X results in a differentially private output.

0.3 PRIVACY GUARANTEES OF DPRP

Before we state our main privacy guarantees, we start with two supporting Lemmas.

Lemma 2. [18] For two neighbouring datasets X and X' that only differ in one observation, i, with $\|X_i - X'_i\| \leq Z$, and a random Gaussian matrix P with entries drawn from $\mathcal{N}\left(0, \sigma_p^2\right)$, where $\sigma_p = 1/\sqrt{k_1}$. With probability at least $1 - \delta$, we have

$$\|XP - X'P\|_F \leq Z\sigma_p \sqrt{k_1 + 2\sqrt{k_1 \log(1/\delta)} + 2\log(1/\delta)}$$

Proof: [2]. Since X and X' only differ on one row i,

X :

$$\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ .. & .. & .. & \\ x_{i1} & x_{i2} & \ldots & x_{in} \\ .. & .. & .. & \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix}$$

and X' :

$$\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ .. & .. & .. & \\ x'_{i1} & x'_{i2} & \ldots & x'_{in} \\ .. & .. & .. & \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix}$$

we can write

$$(XP - X'P)_{mn} = 0, m \neq i$$

Consider an element in XP_{ij} it is dot product of X_i row and P_j column , So

$$\begin{aligned}(XP - X'P)_{ij} &=< X_i, P_j > - < X'_i, P_j > \\ &=< X_i - X'_i, P_j >\end{aligned}$$

Let $z = X_i - X'_i$. Now

$$P_{ij} \sim \mathcal{N}\left(0, \sigma_p^2\right)$$

$$< z, P_j >= \sum_i z_i P_{ij}$$

We apply scalar linear addition properties of normal distribution

$$< z, P_j > \sim \mathcal{N}\left(0, \|z\|^2 \sigma_p^2\right)$$

Here we are reducing the X matrix to k_1 dimensions.
Now if

$$Y = \mathcal{N}(0, 1)$$

then

$$< z, P_j > \sim \|z\| \sigma_p \mathcal{N}\left(0, 1\right)$$

Let $Y_j = \mathcal{N}(0, 1)$ and $\chi^2_{k_1}$ denote a chi-squared random variable with k_1 degrees of freedom. We can bound the matrix norm as

$$\begin{aligned}\|XP - X'P\|_F &= \sqrt{\sum_{j=1}^{k_1} < z, P_j >^2} \\ &= \sqrt{\sum_{j=1}^{k_1} \left(\|z\| \sigma_p Y_j\right)^2} \\ &= \|z\| \sigma_p \sqrt{\chi^2_{k_1}}\end{aligned}$$

Using Lemma 1 from [19], we can get the following tail bound on a random variable X, drawn from a χ^2 distribution with k_1 degrees of freedom

$$\Pr\left[X \geq k_1 + 2\sqrt{k_1 x} + 2x\right] \leq \exp(-x)$$

setting $x = \log(1/\delta)$ completes the proof.
Now , Put

$$X = \left(\|XP - X'P\|_F / \|z\| \sigma_p\right)^2$$

in above probabilistic equation we get

$$\|XP - X'P\|_F \leq Z \sigma_p \sqrt{k_1 + 2\sqrt{k_1 \log(1/\delta)} + 2\log(1/\delta)}$$

with probability $1-\delta$

Lemma 3.

The mechanism $M(D) = f(D) + G$, where G is a random Gaussian matrix with entries drawn from $\mathcal{N}(0, \sigma_1^2)$, satisfies (ϵ, δ) - differential privacy, if $\delta < \frac{1}{2}$, where $\sigma_1^2 = 2\Delta_2(f)^2(\log(1/2\delta) + \epsilon)/\epsilon^2$ and $\Delta_2(f)$ is the sensitivity

With the support of the two lemmas above, we are ready to state our main Theorem.

Theorem 2.

Algorithm 1 is (ϵ, δ) - differentially private, for $\epsilon > 0, 0 < \delta < 1/2$.

Proof. DPRP two components where we add noise (to the random projection P and the covariance matrix X_C) and finally adding noise to covariance matrix and taking svd component // We will prove differential privacy for two components seperatly and finally giving whole differential privacy .

So we will prove for first component .

we can see above that lemma 3 can help us to solve for component 1
we will use this lemma for random projection differential privacy proof.

Theorem A P' is (ϵ_1, δ_1)-differentially private if we add noise from $\mathcal{N}(0, \sigma_1^2)$; where
$\sigma_1 = Z\sigma_p\sqrt{k_1 + 2\sqrt{k_1 \log(2/\delta_1)} + 2\log(2/\delta_1)}\ \sqrt{2(\log(1/2\delta_1) + \epsilon_1)}/\epsilon_1$

Proof. Proof is from [18], summarized next for completeness. Replacing $\Delta_2(f)$ in Lemma 3 with RHS from Lemma 2, and with $\delta/2$, we get

$$\sigma_1 = Z\sigma_p\sqrt{k_1 + 2\sqrt{k_1 \log(2/\delta_1)} + 2\log(2/\delta_1)}\sqrt{2(\log(1/2\delta_1) + \epsilon_1)}/\epsilon_1$$

Here direct substitution is performed So no addition steps are mention. So we shoould add $\mathcal{N}(0, \sigma_1^2)$ noise to P to get differential privacy component.

Theorem $\mathbf{B}\hat{V}'$ is (ϵ_2, δ_2) - differentially private if we add noise to X_C from $\mathcal{N}\left(0, \mathcal{Z}^2\sqrt{2\ln 1.25/\delta_2}/\epsilon_2\right)$

Proof. For providing differential privacy for X_C and subsequently extending it to its singular value decomposition and hence $\hat{V}$, we follow the steps of [17], where we add Gaussian noise to each entry of X_C. Specifically,

$$X_C' = X_C + M_2$$

where M_2 is a $d \times d$ symmetric matrix whose upper triangular values are chosen from

$$\mathcal{N}\left(0, \mathcal{Z}^2\sqrt{2\ln 1.25/\delta_2}/\epsilon_2\right)$$

and lower triangular entries are copied from their upper triangular counterparts. Here $\mathcal{Z}$ is the L_2 sensitivity required for the Gaussian mechanism [3]. Differential privacy guarantees for the above follow directly from [17] and as differential privacy is closed under postprocessing [14], we can perform the decomposition on X_C' to get $\hat{V}'$ without any additional privacy loss.

Using sequential composition[14], we get the Algorithm 1 as (ϵ, δ) - differentially private, where $\epsilon = \epsilon_1 + \epsilon_2$ and $\delta = \delta_1 + \delta_2$.

0.4 ANALYSIS

We have performed DPRP on Indian liver dataset

we can achieve good results if we take K1 = 10 as it mostly nearly actual accuracy

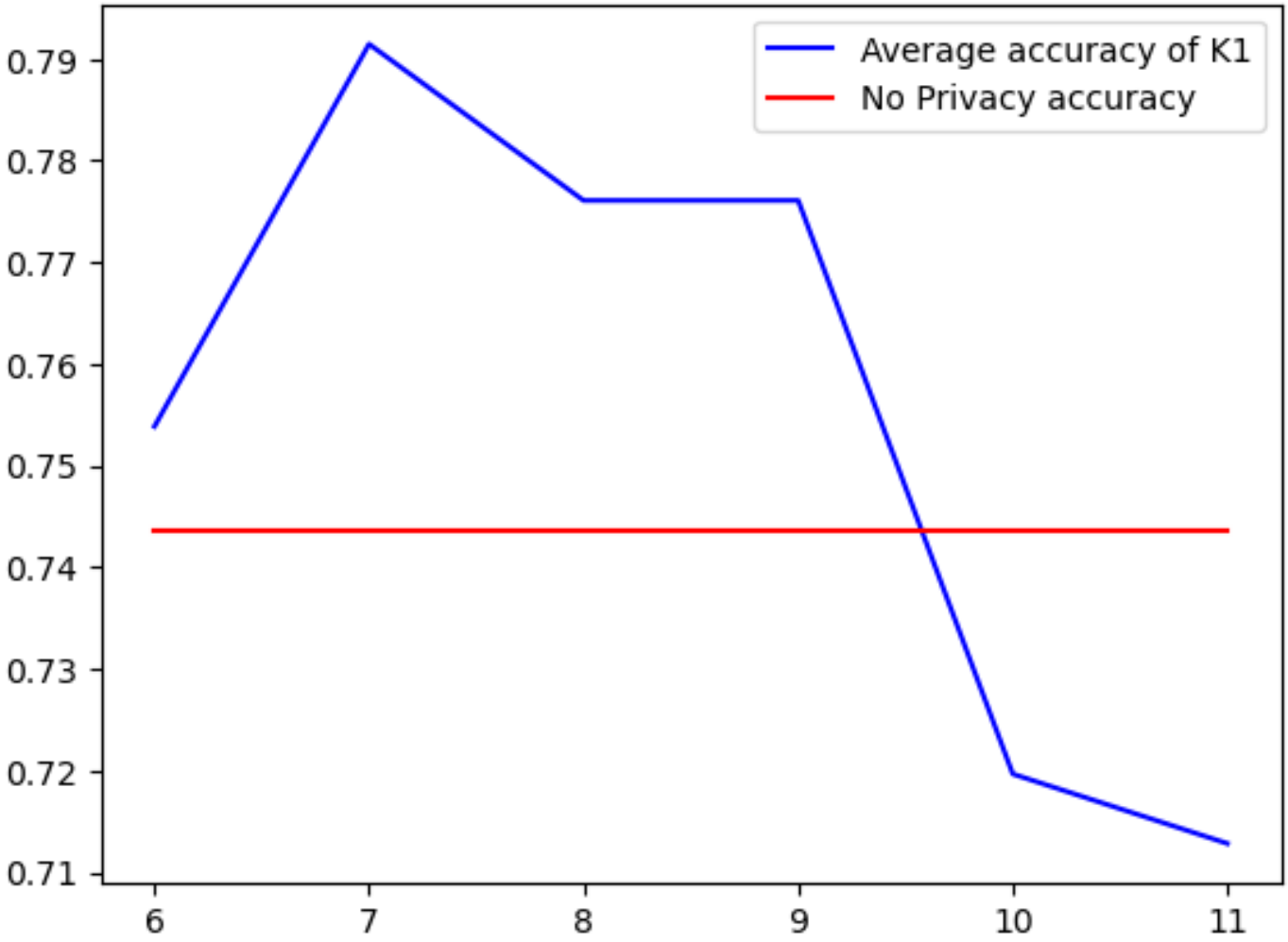

Figure 1: K1 vs accuracy On X' and X

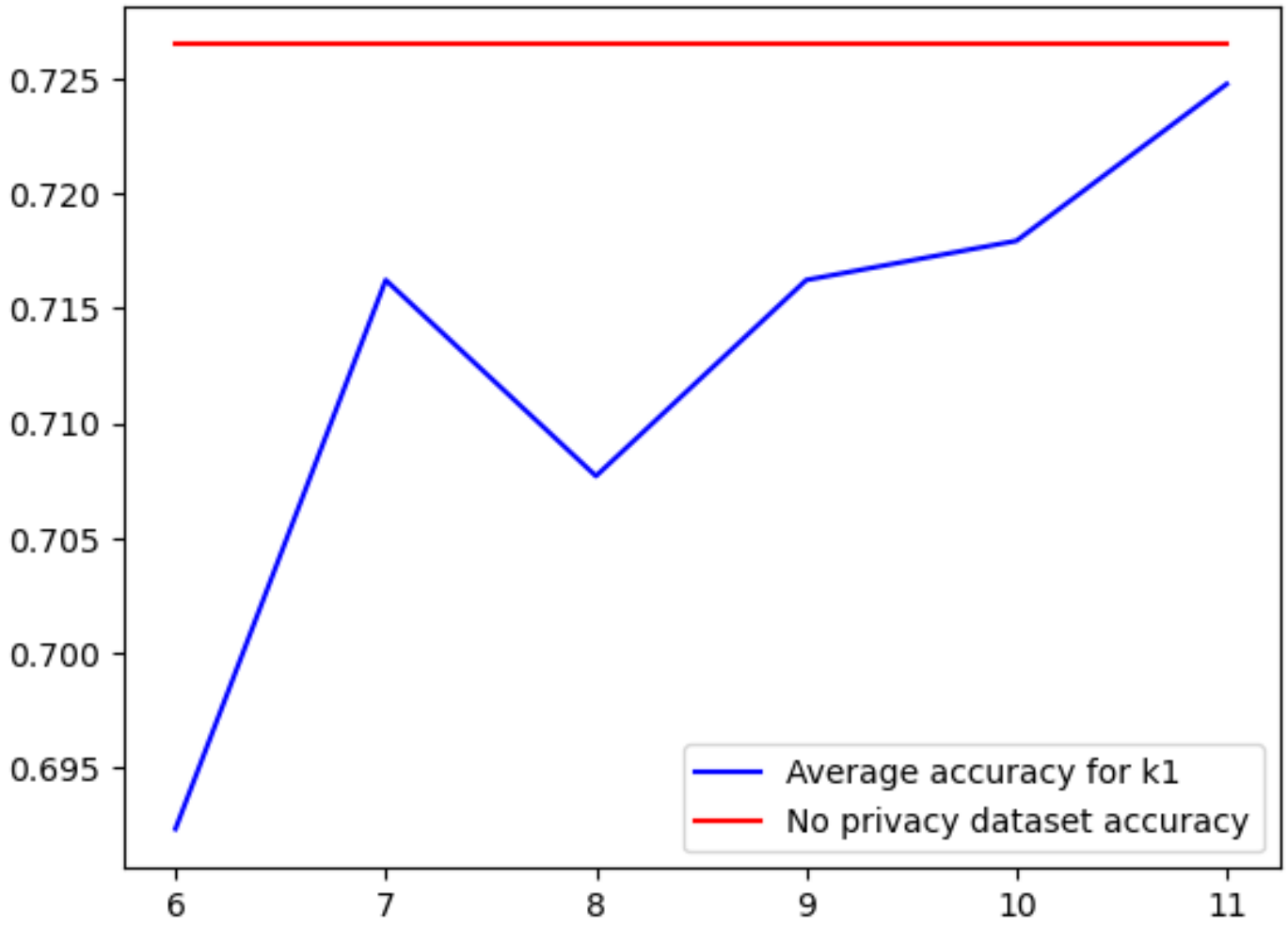

Figure 2: K1 vs accuracy , We have deleted one entry and it accuracy is as follow

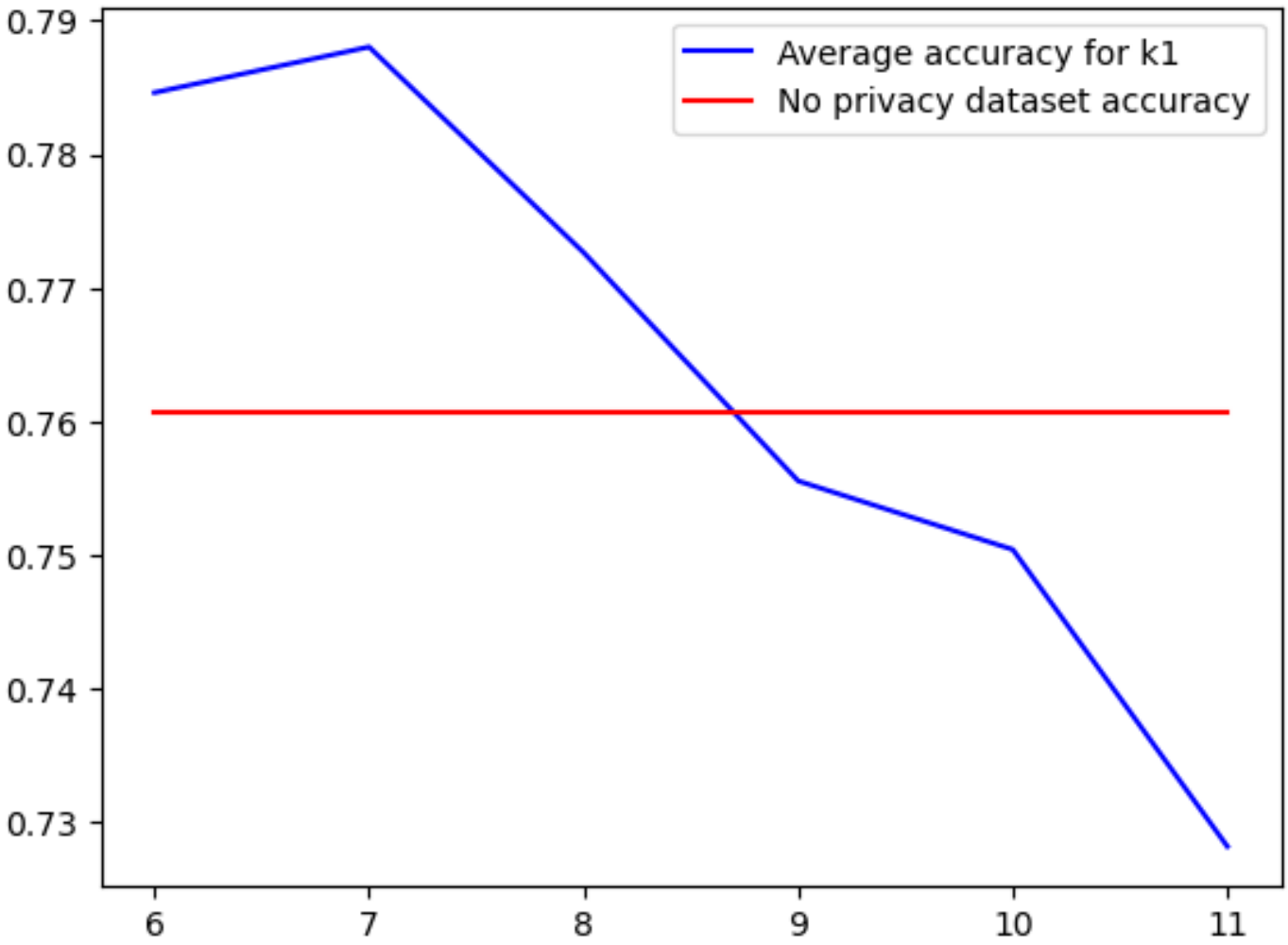

Figure 3: K1 vs accuracy , we have modified one entry and its accuracy is as follow