Local Differential Privacy for Sampling

Hisham Husain◦,‡ **Borja Balle**† **Zac Cranko**◦,‡ **Richard Nock**‡,◦

Abstract

Individual privacy is a major concern when analysing personal data. Differential privacy is a framework that make sure that the analysis does not reveal too much about an individual. But differential privacy requires a trusted party to analyse the data, with whom there's always some risk factor is associated. To tackle this problem, Local differential Privacy is a better approach where the data is randomised at the user level before sending to the untrusted party.

This is a relevant problem in many areas like data augmentation and data synthesis. in this paper, we have proposed a novel LDP method for sampling. Our approach is based on a boosting based density algorithm. The main idea behind this approach is to learn a sampler to generate synthetic data from distribution of each individual user's data while maintaining LDP. Our model provides theoretical guarantee on approximation errors and performs better than existing methods like DP Kernel Density and DP generative adversarial networks.

I. INTRODUCTION

In this paper, we develop a technique to utilize sensitive user data using local differential privacy protocols. We create locally private samplers and make them learn the individual's data distribution while maintain privacy. These samplers mirror the distribution of original data. This approach is applicable on any size of data as we use the distribution of data to generate samplers.

Our main contribution lies in developing a mollification boosting based algorithm that is not just simple but it can also handle any kind of data, even data that has continuous and unbounded domains. It is backed by guarantees of convergence rates within the classical boosting model which means that given the specific assumptions about the distribution in the mollification process, our algorithm will reliably converge.

Also if we make slightly stronger assumptions, we can show guaranteed approximations relative to the optimal distribution within the mollifier. If we relax the privacy constraints, we can show even better guaranteed convergence to the target distribution. Here we provide guarantee in terms of mode capture that is a significant challenge in generative approaches.

After talking about locally private sampling and mollifiers, we dive into our algorithm that learns about data density within a mollifier. We have also discussed how well our method approximates the data. Then, we'll look at other research related to our work and share the results of our experiments for discussion.

II. METHOD

To keep the data private, we have introduced ε private sampler A where ε is used to represent the level of privacy. Sampler A ensures that probability of getting a sample x from two slightly different samples is nearly same.

$$\frac{\Pr[A(P) = x]}{\Pr[A(P') = x]} \leq \exp(\varepsilon) \quad .$$

The sampler takes a dataset and gives a random sample that is similar to the original dataset while maintaining the privacy. These samplers can be generated by Laplacian mechanism or randomized response methods but samplers generated from these methods might not reflect the patterns completely. So we generate these samplers using "Mollification". It changes the dataset into a form that is both easier to work with can has maintained privacy.

To create private samplers, we use "mollifiers." These are sets of distributions that help protect privacy when sampling from a dataset. An ε-mollifier ensures that for any two distributions in the set, the probability of getting a particular sample is closely related.

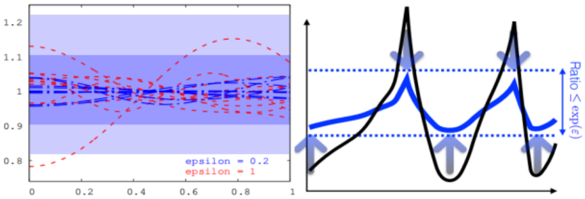


Figure 2: Left: example of mollifiers for two values of ε, $\varepsilon = 1$ (red curves) or $\varepsilon = 0.2$ (blue curves), with $\mathcal{X} = [0, 1]$. For that latter case, we also indicate in light blue the *necessary* range of values to satisfy (2), and in dark blue a *sufficient* range that allows to satisfy (2). Right: schematic depiction of how one can transform any set of finite densities in an ε-mollifier without losing the modes and keeping derivatives up to a positive constant scaling.

Mathematically, a set M is an ε-mollifier if:

$$Q(x) \quad \leq \quad \exp(\varepsilon) \cdot Q'(x), \forall Q, Q' \in \mathcal{M}, \forall x \in \mathcal{X}. \quad (2)$$

This means that the probability of getting a sample x from any distribution in M is close to the probability of getting the same sample from any other distribution in M while maintaining ε privacy.

Constructing Mollifiers:
Mollifiers plays an important role in privacy mechanisms, especially when we consider their convex hulls. Taking the convex hull of an ε-mollifier M = *{Q*1, . . ., *Qm}* produces a new ε-mollifier given by:

$$\mathsf{cvx}(\mathcal{M}) = \left\{\sum \alpha_i Q_i \;:\; \alpha_i \geq 0,\; \sum \alpha_i = 1\right\} \qquad (3)$$

This convex hull, denoted cvx(*M*) is again an ε-mollifier. A mollifier is considered convex if convex hull of M = M. Convex ε-mollifiers, denoted (MR), generated by an ε-LDP mechanism (R) on a finite set (X) can be expressed as:

$$\mathcal{M}_R = \{\mathsf{Law}(A_R(P)) \;:\; P \in \mathcal{D}(\mathcal{X})\} \;, \qquad (4)$$

Here, Law(*AR*(*P*)) is the distribution of the output of *AR*(*P*), which is a mixture of distributions *P* (*x*) · Law(*R*(*x*))

Another way to obtain mollifiers starting from a reference distribution Q_0 is to consider the set of all distributions which are close to Q_0. In particular, we define the ε-mollifier *relative* to Q_0, denoted $\mathcal{M}_{\varepsilon,Q_0}$, to be the set of all distributions Q such that

$$\sup_x \max\left\{\frac{Q_0(x)}{Q(x)}, \frac{Q(x)}{Q_0(x)}\right\} \leq \exp(\varepsilon/2) \;. \qquad (5)$$

To verify that this is indeed an ε-mollifier just note that for any $Q, Q' \in \mathcal{M}_{\varepsilon,Q_0}$ we have

$$\frac{Q(x)}{Q'(x)} = \frac{Q(x)}{Q_0(x)}\frac{Q_0(x)}{Q'(x)} \leq \exp(\varepsilon) \;. \qquad (6)$$

Whenever Q_0 is clear from the context we shall omit if from our notation.

The concept of private sampling through mollification involves finding a distribution *P*∗ inside a mollifier *M* that minimizes the KL divergence:

$$\hat{P} \in \operatorname*{argmin}_{Q \in \mathcal{M}} \mathrm{KL}(P, Q) \;. \qquad (7)$$

The mollification mechanism *A*M takes a dataset *P* and gives a sample from the mollified version *P*^. Its goal is to find the best match tot P within the mollifier. To do this, we solve an optimization problem, when the mollifier is convex, solving the optimization problem is easy. But for finite domains and relative mollifiers, we use a simple equation:

$$\hat{P}(x) = \min\left\{\max\left\{\frac{Q_0(x)}{e^{\varepsilon/2}}, \frac{P(x)}{C}\right\}, e^{\varepsilon/2} Q_0(x)\right\}, \;(8)$$

Even if we estimate *P* through sampling, this equation gives us good approximations for *P*^. It does not matter how rough these approximations are, *A*M keeps things private because they stay within the mollifier.

When dealing with infinitely large sets of data, it's a bit tricky to use a straightforward strategy. One way is to estimate probabilities using non-parametric methods, but those methods can be complex. Instead, we propose a modern approach which is inspired by generative modeling. In our method, we use boosted density estimation to smooth out the data.

Mollified Boosted Density Estimation (MBDE) Algorithm implemented in the paper use a weak learner and a boosting based algorithm.

```
Algorithm 1 MBDE(WL, $T$, $\varepsilon$, $Q_0$)
input: Weak learner WL, # iterations $T$, privacy parameter $\varepsilon$, initial distribution $Q_0$, private target $P$;
for $t = 1, \ldots, T$ do
    $\theta_t(\varepsilon) \leftarrow \left(\frac{\varepsilon}{\varepsilon + 4\log(2)}\right)^t$
    $c_t \leftarrow$ WL$(P, Q_t)$
    $Q_t \propto Q_{t-1} \cdot \exp(\theta_t(\varepsilon) \cdot c_t)$
end for
return: $Q_T$
```

1. Weak Learner: The basic learner, in this scenario is a Multi Layer Perceptron (MLP) classifier with a number of layers and neurons known as a "weak" learner because it only slightly outperforms random guessing, for the given task. Through training the weak learner strives to enhance its accuracy with each round.

2**. Boosting Algorithm:** a learning method combines weak learners to form a robust learner. The boosting algorithm sequentially trains instances of the learner. In each iteration emphasis is placed on examples misclassified by learners by assigning them greater importance through increased weights. The final model is an amalgamation of these learners, where each contributes based on its performance during training data analysis. Boosting seeks to decrease bias and variance thereby enhancing generalization performance compared to using one learner.

Definition 3 (WLA):

A weak learner satisfies the weak learning assumption (WLA) for given constants γP and γQ if, when given two distributions P and Q, it always returns a classifier c such that the expected value of c under P is greater than $c*$ times γP and the expected value of $-c$ under Q is greater than $c*$ times γQ , where $c*$ is the maximum absolute value of c.

MBDE is a private version of the DISCRIM algorithm. In MBDE algorithm we use a weak learner to distinguish between the target distribution P and the current guessed distribution Qt. We refine Qt iteratively for a fixed number of iterations T. Boosting starts with Q_0 as the initial distribution, Q_0 is often a simple non-informed distribution like a standard Gaussian. The classifier output is then combined into $Qt-1$ as part of the iterative process.

$$\begin{aligned} Q_t &= \frac{\exp(\theta_t(\varepsilon)c_t)Q_{t-1}}{\int \exp(\theta_t(\varepsilon)c_t)Q_{t-1}dx} \\ &= \exp\left(\langle\theta(\varepsilon), c\rangle - \varphi(\theta(\varepsilon))\right)Q_0, \qquad (9) \end{aligned}$$

where $\theta(\varepsilon) = (\theta_1(\varepsilon), \ldots, \theta_t(\varepsilon))$, $c = (c_1, \ldots, c_t)$ (from now on, c denotes the vector of all classifiers) and $\varphi(\theta(\varepsilon))$ is the log-normalizer given by

$$\varphi(\theta(\varepsilon)) = \log \int_{\mathcal{X}} \exp\left(\langle\theta(\varepsilon), c\rangle\right) dQ_0. \qquad (10)$$

This process repeats until $t = T$ and the proposed distribution is $Q_\varepsilon(x; P) \doteq Q_T$. We now show three formal results on MBDE.

MBDE is a private sampler Recall $\mathcal{M}_\varepsilon := \mathcal{M}_{\varepsilon,Q_0}$ is the set of densities whose range is in $\exp[-\varepsilon/2, \varepsilon/2]$ with respect to Q_0. Due to Lemma 2, it suffices to show that the output density Q_T of MBDE is in $\mathcal{M}_\varepsilon$.

Theorem 4 $Q_T \in \mathcal{M}_\varepsilon$.

Theorem 5 *For any $t \geq 1$, suppose* WL *satisfies at iteration t the WLA for γ_P^t, γ_Q^t. Then we have:*

$$KL(P, Q_t) \quad \leq \quad KL(P, Q_{t-1}) - \theta_t(\varepsilon) \cdot \Lambda_t, \qquad (11)$$

Theorem 6 *We have $\Delta(Q) \leq \varepsilon/2$, $\forall Q \in \mathcal{M}_\varepsilon$, and if* MBDE *is in the high boosting regime, then*

$$\Delta(Q_T) \quad \geq \quad \frac{\varepsilon}{2} \cdot \left\{\frac{\gamma_P + \gamma_Q}{2} \cdot (1 - \theta_T(\varepsilon))\right\}. \qquad (13)$$

As the privacy constraints decrease (ε increases), our MBDE method gets closer to the true underlying distribution (P). With higher boosting iterations, MBDE approaches the information- theoretic limit, which means it smooths out P, which ensures privacy. Additionally, if we assume that there's a maximum P beyond which P is included in the smoothed distributions, then MBDE provides accurate approximations for all privacy levels greater than P.

MBDE effectively captures the modes of the true distribution P, which is crucial for generative models. We define M$_B$(Q) as the total mass of a set B under distribution Q. Also, KL$_B$ is defined as Kullback-Leibler (KL) divergence between P and Q within B. Theorem 7 states that if KL divergence between P and the initial distribution Q$_0$ is sufficiently less than mass of B under P, then MBDE is able to capture a portion of B through Q$_T$.

$$\mathrm{M}_B(Q) \doteq \int_B dQ\ ,\ \mathrm{KL}_B(P, Q) \doteq \int_B \log\left(\frac{P}{Q}\right) dP,$$

Theorem 7 *Suppose* MBDE *stays in the high boosting. Then $\forall \alpha \in [0, 1]$, $\forall B \subseteq \mathcal{X}$, if*

$$\mathrm{M}_B(P) \quad \geq \quad \varepsilon \cdot \frac{h((2 - \gamma_P - \gamma_Q) \cdot T)}{h(\alpha) \cdot h(T)}, \qquad (14)$$

then $\mathrm{M}_B(Q_T) \geq (1 - \alpha)\mathrm{M}_B(P) - \mathrm{KL}_B(P, Q_0)$, where $h(x) \doteq \varepsilon + 2x$.

We cannot directly control the KL divergence between P and $Q0$. The KL divergence between P and Q0 reflects the quality of approximation of P from Q$_0$ within a certain region B, a small KL divergence relative to the mass of B ensures that we can capture a significant portion of B through QT .theorem 7 says that when a mode, especially a fat one, has a large mass over its region B, we can capture its considerable part by staying in the high boosting regime.

As the parameters γP and γQ approach 1, the condition on the total mass of B under P disappears with the number of boosting iterations T , which allow us to capture any fat region B whose mass is sufficiently large when compared to the KL divergence between P and Q_0.

Our method offers two key advantages over standard local differential privacy:

(i) We use a reference distribution $Q0$, which can be learned from public conversational data using a strong non-private algorithm. This allows us to create -mollifications centered at $Q0$ that include reasonable conversations with high utility.

(ii) Our method is non-interactive, meaning each user generates a private sample that's sent to the server for further processing.

III. EXPERIMENTS

In our experiments, we have used a neural network (NN) classifier as a weak learner which is trained with samples from the target distribution using Nesterov's accelerated gradient descent. We then define T as the number of boosting iterations and initialized Q_0 as a standard Gaussian.

$$\mathcal{X} \xrightarrow[\text{dense}]{\tanh} \mathbb{R}^{25} \xrightarrow[\text{dense}]{\tanh} \mathbb{R}^{25} \xrightarrow[\text{dense}]{\tanh} \mathbb{R}^{25} \xrightarrow[\text{dense}]{\text{sigmoid}} (0,1), \quad (15)$$

Our main opponent was a privacy-preserving approach called DPB. We settled for DPB because its technical aspects imply a privacy budget similar to ours in the context of local differential privacy. These methods both scale their privacy parameter to accommodate arbitrary dataset sizes while maintaining privacy, but do so differently.

The two key evaluation metrics employed to evaluate our method are mode coverage and negative log likelihood. Mode coverage is the evaluation of how well the method captures dense regions, by measuring what proportion of mass in the target distribution falls under regions where the model distribution reaches a threshold, for example 95% of the mass. Negative log likelihood is an overall loss measure that is calculated as expected value of negative logarithm of model distribution with respect to target distribution.

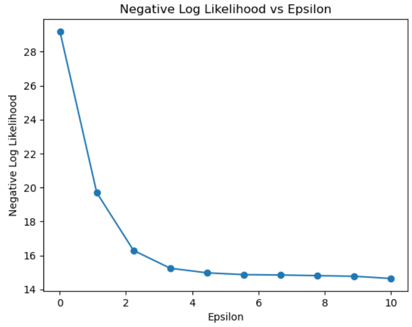
In our experiment we consider a mixture consisting of three 1D Gaussians with PDF :
$P\ (x) = 1/3\ (N\ (0.3, 0.01) + N\ (0.5, 0.1) + N\ (0.7, 0.1))$.
Finally, we will look at one-dimensional space containing Gaussian distributions with different means and same variances placed randomly over it. To analyze some parameters such as number of Gaussians or privacy parameter, experiments were done many times and mean values as well as standard deviations were calculated for each case.

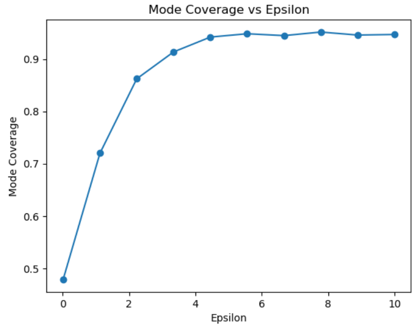
IV. RESULTS

The outcomes show that MBDE is more efficient than DPB in local differentially private density estimation. Even when ε assumes significantly lower values, MBDE achieves comparable outcomes to DPB. The smoothness and regularity of the patterns shown by the density models produced by MBDE can be attributed partly to the fact that Q0 was chosen as a standard Gaussian.

MBDE has been investigated across all domains, including random 1D Gaussians, in which case it consistently exhibits decreasing negative log likelihood (NLL) as ε increases. This means that MBDE does an excellent job of capturing all modes of the mixture leading to faster improvements for smaller ε values. In contrast, DPB's NLL curve plateaus indicating diminishing returns as ε increases. These deep classifiers are employed as sufficient statistics that explain why MBDE is superior to DPB.

In addition, much smaller standard deviations are seen in results for MBDE compared to those for DPB which suggest better and robust findings. For many experiments Mode Coverage generally increases with ε in the case of MBDE across all domains. This behavior is governed by choice of Q_0 serving as a representative part of multiple modes present there in it. As ε grows larger, this algorithm performs better showing significant improvement over Q_0. This trend is also observed in experiments with random 1D Gaussians, where small standard deviations lead to significant stability in MBDE solutions.





V. DISCUSSION AND CONCLUSION

In a nutshell, our paper proposes an innovative method for privately ascertaining densities at the regional level so as to create synthetic data. The article shows how mollifiers can help guarantee privacy, while still valuable in their own right. We applied boosting to show convergence guarantees and offer theoretical results on mode coverage and target-density approximation. Boosting framework helps reduce the complexity of the problem which is even more crucial with increasing dimensions of distribution.

Some additional assumptions that we could make in our approach include targeting sparse expected parameter or tuning Q0 with available information from public domain that could enhance convergences. Our experiments confirm the effectiveness of our method in capturing statistical properties of the true distribution thereby signifying its relevance in practical situations.