Local Differential Privacy for Sampling

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

Abstract

Differential privacy (DP) is a leading privacy protection focused by design on individual privacy. In the local model of DP, strong privacy is achieved by privatizing each user's individual data before sending it to an untrusted aggregator for analysis. While in recent years local DP has been adopted for practical deployments, most research in this area focuses on problems where each individual holds a single data record. In many problems of practical interest this assumption is unrealistic since nowadays most user-owned devices collect large quantities of data (e.g. pictures, text messages, time series). We propose to model this scenario by assuming each individual holds a distribution over the space of data records, and develop novel local DP methods to sample privately from these distributions. Our main contribution is a boosting-based density estimation algorithm for learning samplers that generate synthetic data while protecting the underlying distribution of each user with local DP. We give approximation guarantees quantifying how well these samplers approximate the true distribution. Experimental results against DP kernel density estimation and DP GANs displays the quality of our results.

1 Introduction

Over the past decade, differential privacy (DP) has evolved as the leading statistical protection model for individuals' data (Dwork and Roth, 2014). The basis of DP is that a mechanism is private whenever its output provides insufficient information to distinguish between two potential input datasets that differ on a single individual. In doing so, it guarantees plausible deniability regarding the presence of an individual in the input of the mechanism. Despite the popularity of DP, one shortcoming of the standard definition is the assumption of a *trusted* curator who has access to the full dataset of individuals. One way to get around this is to have individuals run their data through a DP mechanism at the local level before sending it for processing, ensuring that the curator only gets access to privatized data. This approach is called the *local model* of differential privacy (Raskhodnikova et al., 2008). It requires considerably weaker trust assumptions than the curator model, and was in fact the basis of the first large-scale deployments of DP by Apple (Differential privacy team, Apple, 2017) and Google (Erlingsson et al., 2014).

The interest in the local model has spurred research into local DP protocols for a number of practical tasks (see (Cormode et al., 2018) and references therein), as well as the search for intermediate privacy models achieving a compromise between the local and curator DP Bittau et al. (2017). However, while most of this research focuses, often implicitly, on the setting where each individual owns a *single* data record, a growing number of applications involve one individual contributing *multiple* data records. Examples include problems where the data evolves over time, as well as settings where locally each individual owns a whole dataset containing, e.g., pictures, text messages or historical device usage information.

In this paper we investigate a method to leverage sensitive user-level datasets in local DP protocols by constructing *locally private samplers* which can release synthetic data points from the distribution of the underlying dataset. Our framework accommodates local datasets of arbitrary sizes by modelling an individual's private data as a probability distribution – this is also applicable in situations where the dataset does not exist *per se* but an algorithm can sample from it by, e.g., interacting with the user. We formalize the problem by (1) introducing the notion of mollifier – a collection of valid distributions from which one can obtain samples with a desired privacy level; and, (2) cast the goal of learning a private sampler as the problem of computing the information-geometric projection of a private distribution onto a given mollifier – a process we call mollification. Our main contribution is an efficient approximate mollification algorithm based on recent advances in boosted density estimation (Cranko and Nock, 2019). In contrast with

◦ The Australian National University. ‡ Data61. † Currently at DeepMind.

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

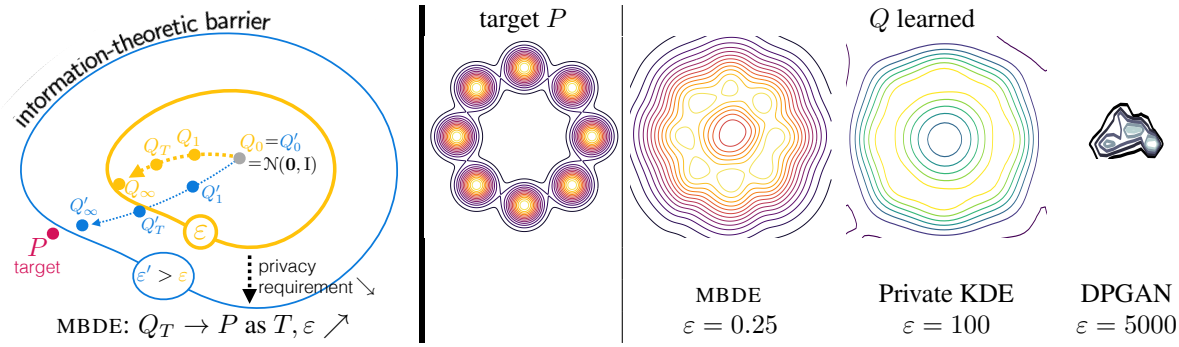


Figure 1: *Left*: Our method is guaranteed to get a Q_T that converges to P as the privacy constraint is relaxed and the number of boosting iterations increases (under a weak learning assumption). *Right*: Our method vs private KDE (Aldà and Rubinstein, 2017) and DPGAN (Xie et al., 2018) on a ring Gaussian mixture (see Section 5, $m = k = 10000$). Remark that the GAN is subject to mode collapse.

naive solutions we discuss below, our algorithm works on arbitrary data, including continuous unbounded domains. This algorithm comes with convergence rate guarantees in the classical boosting model, that is, under lightweight assumptions on the distribution iterates used in the mollification process. Under slightly stronger assumptions, we are able to show guaranteed approximation with respect to the *optimal* distribution in the mollifier. As the privacy constraint is relaxed, we get better approximation guarantees with respect to the target distribution itself. This is illustrated in Figure 1 (left). Last but not least, we provide guarantees in terms of capturing the modes of the target distribution, which is a prominent problem in generative approaches (Figure 1, right).

The rest of this paper is organized as follows. Section 2 introduces locally private sampling, mollifiers and their relationships. Section 3 introduces our algorithm that learns a density in a mollifier and shows several approximation properties in the boosting model. Section 4 summarizes related work, Section 5 presents and discusses experiments.

2 Private sampling and mollifiers

We now proceed to formalize the task of sampling from a private distribution in the local DP model. Then introduce the concept of mollification which solves this problem by first projecting the distribution into a carefully constructed set and releases a sample from the resulting projection.

Locally private sampling Suppose a user holds a private probability distribution $P \in \mathcal{D}(\mathcal{X})$ over some domain $\mathcal{X}$ and wants to release a sample from P while preserving the their privacy. We introduce a user-defined parameter, $\varepsilon > 0$, which represents a privacy budget – smaller ε correspond to a stronger privacy demand. An *ε-private sampler* is a randomized mapping $A : \mathcal{D}(\mathcal{X}) \to \mathcal{X}$ such that for any $x \in \mathcal{X}$ and any two distributions $P, P' \in \mathcal{D}(\mathcal{X})$ we have

$$\frac{\Pr[A(P) = x]}{\Pr[A(P') = x]} \leq \exp(\varepsilon) \ . \tag{1}$$

This is the same as saying that A is an *ε-locally differentially private* (LDP) mechanism[1] with inputs in $\mathcal{D}(\mathcal{X})$ and outputs in $\mathcal{X}$, which allows a user to release a privatized sample from their distribution P. Note that when the user has a dataset with records from $\mathcal{X}$ we can take P to be the empirical distribution over the sample.

A simple way to construct ε-private samplers given an ε-LDP mechanism $R : \mathcal{X} \to \mathcal{X}$ is a follows: take a sample $x_0 \sim P$ and then release the output of $R(x_0)$. This construction, which we denote by A_R, is appealing because it enables us to leverage any of the many local randomizers R that have been proposed in the literature, including, e.g., randomized response for discrete input spaces, and the Laplace mechanism with inputs on a bounded real interval. On the other hand, this generic construction is limited by the fact that A_R only accesses the private distribution P through a *single* sampling operation and has no information about the global shape of P.

Mollifiers To address this issue we propose to build private samplers by first projecting the distribution P onto a given mollifier and then releasing one sample from the projected distribution.

Definition 1 *Let $\mathcal{M} \subset \mathcal{D}(\mathcal{X})$ be a set of distributions*[2] *and $\varepsilon > 0$. We say $\mathcal{M}$ is an ε-**mollifier** iff*

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

[1] A randomized mechanism $R : \mathcal{Y} \to \mathcal{Z}$ is ε-LDP if $\Pr[R(y) = z] \leq e^\varepsilon \Pr[R(y') = z]$ for all y, y', z.

[2] For the sake of simplicity (and at the expense of slight abuses of language) we use the same notation for distributions and their densities with respect to some base measure throughout the paper.

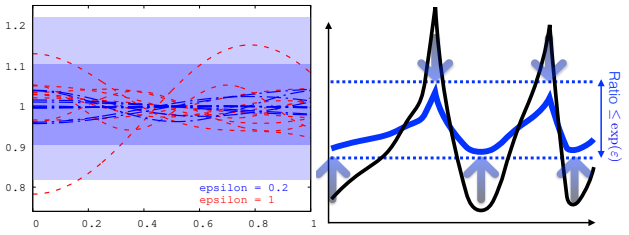


Figure 2: Left: example of mollifiers for two values of ε, $\varepsilon = 1$ (red curves) or $\varepsilon = 0.2$ (blue curves), with $\mathcal{X} = [0, 1]$. For that latter case, we also indicate in light blue the *necessary* range of values to satisfy (2), and in dark blue a *sufficient* range that 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.

For example, a singleton $\mathcal{M} = \{Q\}$ is a 0-mollifier. Intuitively, these mollifiers consist of distributions which are all close to each other with respect to the divergence used to define local DP. Figure 2 (left) features examples of mollifiers with densities supported in $\mathcal{X} = [0, 1]$. Two ranges indicated in blue depict necessary or sufficient conditions on the overall range of a set of densities to be a mollifier. For the necessary part, we note that any continuous density must have 1 in its range of values (otherwise its total mass cannot be unit), so if it belongs to an ε-mollifier, its maximal value cannot be $\geq e^{\varepsilon}$ and its minimal value cannot be $\leq e^{-\varepsilon}$. We end up with the range in light blue, in which any ε-mollifier has to fit. For the sufficiency part, we indicate in dark blue a possible range of values, $[e^{-\varepsilon/2}, e^{\varepsilon/2}]$, which gives a sufficient condition for the range of all elements in a set $\mathcal{M}$ for this set to be an ε-mollifier.

Mollifiers play a central role in the theory developed in this paper, and they might also be of independent interest in the field of differential privacy. Before we show how they relate to private samplers, we first discuss some properties.

Constructing mollifiers Taking the convex hull of a mollifier produces a new mollifier. That is, given an ε-mollifier[3] $\mathcal{M} = \{Q_1, \ldots, Q_m\}$, the convex hull

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

is again an ε-mollifier. We call $\mathsf{cvx}(\mathcal{M})$ the mollifier *generated* by $\mathcal{M}$. A mollifier is *convex* if $\mathsf{cvx}(\mathcal{M}) = \mathcal{M}$. Of particular interest are the convex ε-mollifiers generated by a ε-LDP mechanism R on some finite set $\mathcal{X}$, obtained as $\mathcal{M}_R := \mathsf{cvx}(\{R(x) \ : \ x \in \mathcal{X}\})$. This mollifier is in fact equivalent to the range of distributions of the naive sampler A_R, in the sense that

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

[3]Assumed finite for simplicity of exposition.

where $\mathsf{Law}(A_R(P))$ denotes the distribution of the output of $A_R(P)$ which can be written as the mixture $\mathsf{Law}(A_R(P)) = \sum_{x \in \mathcal{X}} P(x) \cdot \mathsf{Law}(R(x))$. This construction can be directly extended to bounded $\mathcal{X} \subset \mathbb{R}^d$, but for unbounded domains it is unclear how to proceed as most known LDP mechanisms R require bounded sensitivity.

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.

Unlike with finitely generated mollifiers, relative mollifiers are not easy to parametrize in closed form. This is due to the "non-parametric" nature of the definition of $\mathcal{M}_{\varepsilon,Q_0}$, as opposed to the parametric definition of $\mathsf{cvx}(\{Q_1, \ldots, Q_m\})$. However, from the point of view of the problem we consider in the sequel – namely, finding the closest projection of a distribution onto a given mollifier – we shall see that relative mollifiers are also computationally tractable. In particular, we show that finding such projections when $\mathcal{X}$ is finite can be done in closed-form, and that when $\mathcal{X}$ is infinite one can use boosting-based techniques to efficiently approximate the corresponding projection.

Private sampling via mollification We call *mollification* the process of taking a distribution P and finding a distribution $\hat{P}$ inside a given mollifier $\mathcal{M}$ that minimizes the KL divergence:

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

We pick the KL divergence for its popularity and the fact that it is the canonical divergence for broad sets of distributions (Amari and Nagaoka, 2000). The appeal of this construction stems from the following result, which says that a mechanism that releases samples from some distribution in a mollifier provides privacy.

Lemma 2 *Let $A : \mathcal{D}(\mathcal{X}) \to \mathcal{X}$ by a randomized mechanism such that, for any P, $A(P)$ releases a sample from some $Q \in \mathcal{M}$. If $\mathcal{M}$ is an ε-mollifier, then A is an ε-private sampler.*

Thus, the *mollification mechanism* $A_{\mathcal{M}}$ that on input P releases a sample from the mollification $\hat{P}$ is a private sampler which tries to maximize utility by finding the closest distribution to P in a given mollifier. In order to implement the mechanism $A_{\mathcal{M}}$ it is necessary to solve the optimization problem (7). Furthermore, one also requires that the resulting distribution admits an efficient sampling procedure. With respect to the first requirement, we note that the problem in (7) is convex whenever the mollifier $\mathcal{M}$ is convex. Thus, the mollification problem could be solved efficiently using (stochastic[4]) convex optimization methods as long as $\mathcal{M}$ has a tractable representation. However, here we take a different approach.

For the case where the domain $\mathcal{X}$ is finite, the optimum of (7) admits a simple closed-form whenever $\mathcal{M}$ is a relative mollifier. In particular, for $\mathcal{M}_{\varepsilon,Q_0}$ it is easy to solve the Karush-Kuhn-Tucker (KKT) optimality conditions for (7) to show that the optimum is given by

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

where C is a constant such that $\hat{P}$ sums to one. If P is only accessible through sampling, one can plug estimators for the probability of each element in $\mathcal{X}$ into the closed-form solution to obtain approximations to $\hat{P}$. An important observation is that no matter how bad this approximation is, the overall mechanism $A_{\mathcal{M}}$ remains private because the form of these closed-form solutions ensures the approximation is always inside the mollifier $\mathcal{M}_{\varepsilon,Q_0}$; this is a property that any private sampler using approximate mollification should satisfy.

When $\mathcal{X}$ is infinite this strategy is not immediately tractable, although one could try to obtain a non-parametric approximation to P and use it as a plug-in estimator in (8). Known properties of such estimators could be used to analyze the convergence of these non-parametric approximations, but the alternative approach we consider in this paper is more in line with modern methods in generative modelling. In particular, in Section 3 we provide a method for approximate mollification with relative mollifiers based on boosted density estimation. The boosting-based approach allows us to encode prior knowledge about the distributions P that we expect to encounter in practice in the choice of Q_0 and the architecture of the weak classifier trained at each iteration. This opens the door to using mollifiers learned from (non-private) data to improve the sample efficiency of private samplers; we leave this question for future research.

[4]Depending on whether we have access to P through a probability oracle for evaluating $P(x)$ or just through sampling.

Algorithm 1 MBDE(WL, T, ε, Q_0)

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

3 Mollification with approximation guarantees

The cornerstone of our approach to locally private sampling is an algorithm that (i) learns an explicit density in an ε-mollifier and (ii) with approximation guarantees with respect to the target P. We refer to the algorithm as MBDE, for Mollified Boosted Density Estimation; its pseudo-code is given in Algorithm 1.

To show convergence result on MBDE, we borrow the standard machinery from boosting, which includes classifiers $c : \mathcal{X} \to \mathbb{R}$ where $\text{sign}(c(x)) \in \{-1, 1\}$ denotes classes. For technical convenience we assume $c(x) \in [-\log 2, \log 2]$ and so the output of c is bounded. This is a common assumption in the boosting literature (Schapire and Singer, 1999). We also require a pivotal condition from boosting: the *weak learning* assumption.

Definition 3 (WLA) *Fix $\gamma_P, \gamma_Q \in (0, 1]$ two constants. We say that WeakLearner$(.,.)$ satisfies the **weak learning assumption** (WLA) for γ_P, γ_Q iff for any P, Q, WeakLearner(P, Q) returns a classifier c satisfying $\mathbb{E}_P[c] > c^* \cdot \gamma_P$ and $\mathbb{E}_Q[-c] > c^* \cdot \gamma_Q$, where $c^* = \sup_{x \in \mathcal{X}} |c(x)|$.*

Briefly stated, a weak learner can be thought of as an oracle taking as inputs two distributions P and Q and is required to always return a classifier c that weakly guesses the sampling from P vs Q. Remark that as the two inputs P and Q become "closer" in some sense to one another, it is harder to satisfy the WLA. However, this is not a problem as whenever this happens, we shall have successfully learned P through Q. The classical theory of boosting would just assume one constraint over a distribution M whose marginals over classes would be P and Q (Kearns, 1988), but our definition can in fact easily be shown to coincide with that of boosting (Cranko and Nock, 2019).

MBDE is a private refinement of the DISCRIM algorithm of (Cranko and Nock, 2019, Section 3). It uses a weak learner whose objective is to distinguish between the target P and the current guessed density Q_t — the index indicates the iterative nature of the algorithm. Q_t is progressively

refined using the weak learner's output classifier c_t, for a total number of user-fixed iterations T. We start boosting by setting Q_0 as the starting distribution, typically a simple non-informed (to be private) distribution such as a standard Gaussian (see also Figure 1, center). The classifier is then aggregated into Q_{t-1} as:

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

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

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

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

We observe that privacy comes with a price, as for example $\lim_{\varepsilon\to 0} \theta_t(\varepsilon) = 0$, so as we become more private, the updates on $Q_\bullet$ become less and less significant and we somehow flatten the learned density — such a phenomenon is not a particularity of our method as it would also be observed for standard DP mechanisms (Dwork and Roth, 2014).

Convergence guarantees for MBDE As explained in Section 2, it is not hard to fit a density in $\mathcal{M}_\varepsilon$ to make its sampling private. An important question is however what guarantees of approximation can we still have with respect to P, given that P may not be in $\mathcal{M}_\varepsilon$. We now give such guarantees to MBDE in the boosting framework, and we also show that the approximation is within close order to the best possible given the constraint to fit $Q_\bullet$ in $\mathcal{M}_\varepsilon$. We start with the former result, and for this objective include the iteration index t in the notations from Definition 3 since the actual weak learning guarantees may differ across iterations, even when they are still within the prescribed bounds.

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:*

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

where (letting $\Gamma(z) \doteq \log(4/(5-3z))$):

$$\Lambda_t = \begin{cases} c_t^*\gamma_P^t + \Gamma(\gamma_Q^t) & \text{if } \gamma_Q^t \in [1/3, 1] \text{ ("HBS")} \\ \gamma_P^t + \gamma_Q^t - \frac{c_t^* \cdot \theta_t(\varepsilon)}{2} & \text{if } \gamma_Q^t \in (0, 1/3) \text{ ("LBS")} \end{cases}. \quad (12)$$

Here, HBS means high boosting regime and LBS means low boosting regime.

Remark that in the *high* boosting regime, we are guaranteed that $\Lambda_t \geq 0$ so the bound on the KL divergence is guaranteed to decrease. This is a regime we are more likely to encounter during the first boosting iterations since Q_{t-1} and P are then easier to tell apart — we can thus expect a larger γ_Q^t. In the low boosting regime, the picture can be different since we need $\gamma_P^t + \gamma_Q^t \geq c_t^* \cdot \theta_t(\varepsilon)/2$ to make the bound not vacuous. Since $\theta_t(\varepsilon) \to_t 0$ exponentially fast and $c_t^* \leq \log 2$, a constant, the constraint for (12) to be non-vacuous vanishes and we can also expect the bound on the KL divergence to also decrease in the *low* boosting regime. We now check that the guarantees we get are close to the best possible in an information-theoretic sense. Let us define $\Delta(Q) \doteq \mathrm{KL}(P, Q_0) - \mathrm{KL}(P, Q)$. Intuitively, the farther P is from Q_0, the farther we should be able to get from Q_0 to approximate P, and so the larger should be $\Delta(Q)$. Notice that this would typically imply to be in the high boosting regime for MBDE. For the sake of simplicity, we consider γ_P, γ_Q to be the same throughout all iterations.

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*

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

Hence, as $\gamma_P \to 1$ and $\gamma_Q \to 1$, we have $\Delta(Q_T) \geq (\varepsilon/2) \cdot (1 - \theta_T(\varepsilon))$ and since $\theta_T(\varepsilon) \to 0$ as $T \to \infty$, MBDE indeed reaches (in the high boosting regime) the information-theoretic limit, which is the mollification of P. As ε increases (the privacy constraint is reduced), Theorem 6 shows that we are guaranteed to progressively come closer to P, and if we make the additional assumption that there exists $\varepsilon_P \ll \infty$ such that $P \in \mathcal{M}_{\varepsilon_P}$ – which appears to be quite reasonable given the definition in (5) –, then Theorem 6 delivers a direct approximability result for MBDE with respect to P for all privacy levels $\varepsilon \geq \varepsilon_P$. This is a new result compared to the privacy-free approximation bounds of P in (Cranko and Nock, 2019), but it requires to be in the high boosting regime.

MBDE captures the modes of P Mode capture is a prominent problem in the area of generative models (Tolstikhin et al., 2017). We have already seen that enforcing mollification can be done while keeping modes, but we

would like to show that MBDE is indeed efficient at building some Q_T with guarantees on mode capture. For this objective, we define for any set $B \subseteq \mathcal{X}$ and distribution Q,

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

respectively the total mass of B on Q and the KL divergence between P and Q restricted to B.

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* $h(x) \doteq \varepsilon + 2x$.

There is not much we can do to control $\mathrm{KL}_B(P,Q_0)$ as this term quantifies our luck in picking Q_0 to approximate P in B but if this restricted KL divergence is small compared to the mass of B, then we are guaranteed to capture a substantial part of it through Q_T. As a mode, in particular "fat", would tend to have large mass over its region B, Theorem 7 says that we can indeed hope to capture a significant part of it as long as we stay in the high boosting regime. As $\gamma_P \to 1$ and $\gamma_Q \to 1$, the condition on $\mathrm{M}_B(P)$ in (14) vanishes with T and we end up capturing any fat region B (and therefore, modes, assuming they represent "fatter" regions) whose mass is sufficiently large with respect to $\mathrm{KL}_B(P,Q_0)$.

With regards to a practical application, consider the problem of generating synthetic text data from conversational English. Each individual user holds their own distribution (own speech patterns and vocabulary) and the goal is to be able to model these distributions with privacy and approximation guarantees. We point out two implicit advantages of our method over standard local DP and federated learning methods: (i) Our method relies on the reference distribution Q_0, which in this application, one may use public conversational data to learn Q_0 using a strong non-private algorithm. In this case, the ε-mollifier centered at Q_0 will contain admissible conversations with relatively high utility, meaning that mollifications will still be reasonable. (ii) Our method is non-interactive: each user generates a privatized sample which is submitted to the server for post-processing.

To finish up this Section, recall that $\mathcal{M}_\varepsilon$ is also defined (in disguise) and analyzed in (Wang et al., 2015, Theorem 1) for posterior sampling. However, the convergence in (Wang et al., 2015, Section 3) does not dig into specific forms for the likelihood of densities chosen — as a result, it remains essentially in weak asymptotic form, and furthermore it is only applied to DP in the curator model. We exhibit particular choices for these mollifier densities, along with a specific training algorithm to learn them, that allow for significantly better approximation, quantitatively and qualitatively (mode capture) in the local DP setting.

4 Related work

A broad literature has been developed early for discrete distributions (Machanavajjhala et al., 2008) (and references therein). For a general Q not necessarily discrete, more sophisticated approaches have been tried, most of which exploit randomisation and the basic toolbox of differential privacy (Dwork and Roth, 2014, Section 3): given non-private $\tilde{Q}$, one compute the *sensitivity* s of the approach, then use a standard mechanism $M(\tilde{Q}, s)$ to compute a private Q. Such general approaches have been used for Q being the popular kernel density estimation (KDE, (Givens and Hoeting, 2013)) with variants (Aldà and Rubinstein, 2017; Hall et al., 2013; Rubinstein and Aldà, 2017).

On the algorithmic side, our work shares some ideas with DP methods based on the *multiplicate weights* technique (Hardt and Rothblum, 2010; Hardt et al., 2012; Ullman, 2015). These papers leverage ideas similar to boosting to solve problems like answering linear queries, solving convex minimization problems, or releasing synthetic data to accurately answer a pre-determined set of queries. None of these works, however, apply directly to the local DP model.

5 Experiments

Architectures We carried out experiments on a simulated setting inspired by (Aldà and Rubinstein, 2017), to compare MBDE (implemented following its description in Section 3) against differentially private KDE (Aldà and Rubinstein, 2017). As a weak learner for MBDE, we fit for each c_t a neural network (NN) classifier:

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

where $\mathcal{X} \in \{\mathbb{R}, \mathbb{R}^2\}$ depending on the experiment. At each iteration t of boosting, c_t is trained using 10000 samples from P and Q_{t-1} using Nesterov's accelerated gradient descent with $\eta = 0.01$ based on cross-entropy loss with 750 epochs. Random walk Metropolis-Hastings is used to sample from Q_{t-1} at each iteration. For the number of boosting iterations in MBDE, we pick $T = 3$. This is quite a small value but given the rate of decay of $\theta_t(\varepsilon)$ and the small dimensionality of the domain, we found it a good compromise for complexity vs accuracy. Finally, Q_0 is a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathrm{I}_d)$.

Contenders We know of no local differentially private sampling approach operating under conditions equivalent to ours, so our main contender is going to be a particular

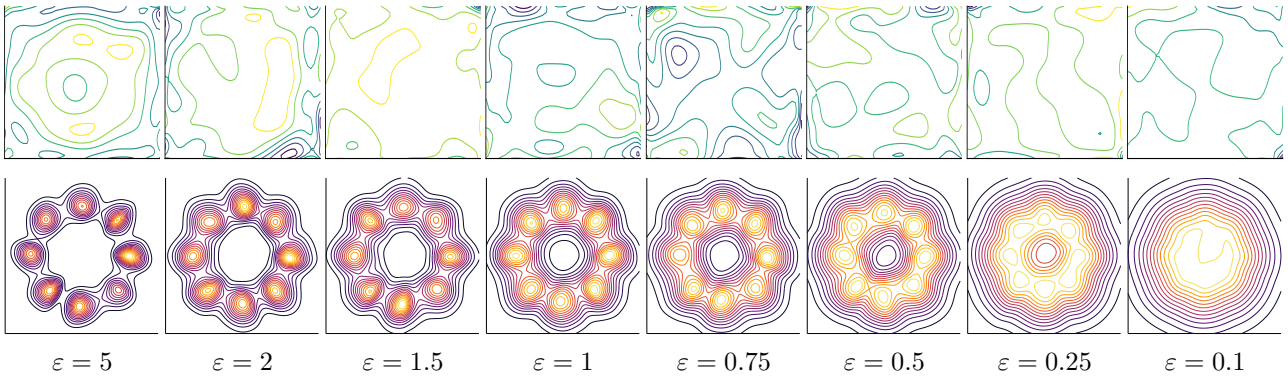


Figure 3: Gaussian ring: densities obtained for DPB (upper row) against MBDE (lower row)

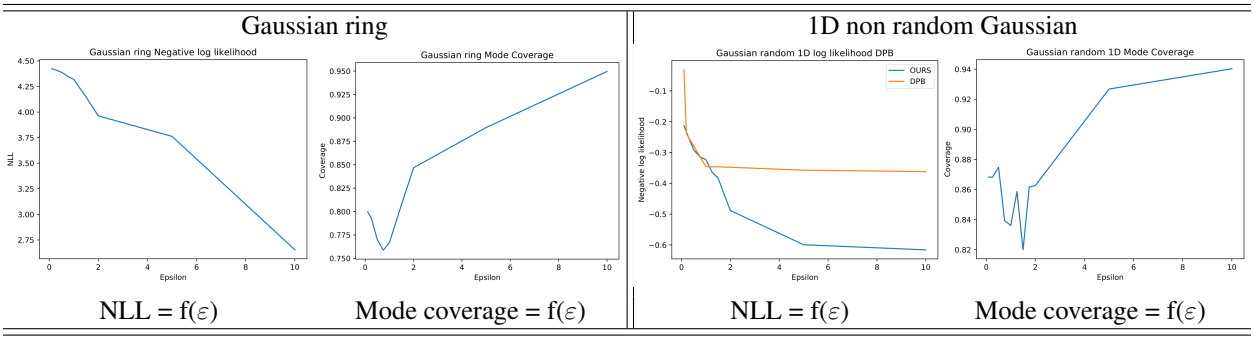


Figure 4: Metrics for MBDE (blue): NLL (lower is better) and mode coverage (higher is better). Orange: DPB (see text).

state of the art ε-differentially private approach which provides a private *density*, DPB (Aldà and Rubinstein, 2017). We choose this approach because digging in its technicalities reveal that *its* ***local differential privacy budget*** *would be roughly equivalent to ours, mutatis mutandis*. Here is why: this approach allows to sample a dataset of arbitrary size (say, k) while keeping the same privacy budget, *but* needs to be scaled to accomodate local differential privacy, while in our case, MBDE allows to obtain local differential privacy for one observation ($k = 1$), *but* its privacy budget needs to be scaled to accomodate for larger k. It turns out that in both approaches, the scaling of the privacy parameter to accomodate for arbitrary k and local differential privacy is roughly the same. In our case, the change is obvious: the privacy parameter ε is naturally scaled by k by the composition property of ε-LDP. In the case of (Aldà and Rubinstein, 2017), the requirement of local differential privacy multiplies the sensitivity[5] by k by the group privacy property.

We have also compared with a private GAN approach, which has the benefit to yield a simple sampler but involves a weaker privacy model (Xie et al., 2018) (DPGAN). For DPB, we use a bandwidth kernel and learn the bandwidth parameter via 10-fold cross-validation. For DPGAN, we train the WGAN base model using batch sizes of 128 and 10000 epochs, with $\delta = 10^{-1}$. We found that DPGAN is significantly outperformed by both DPB and MBDE, so to save space we have only included the experiment in Figure 1 (right). We observed that DPB does not always yield a positive measure. To ensure positivity, we shift and scale the output.

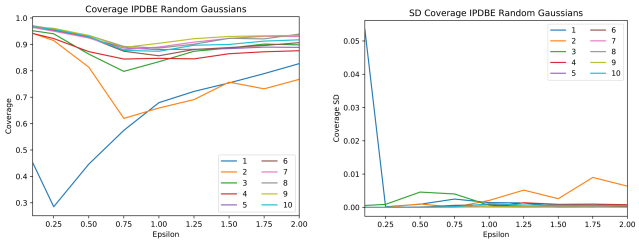
Metrics We consider two metrics, inspired by those we consider for our theoretical analysis and one investigated in (Tolstikhin et al., 2017) for mode capture. We first investigate the ability of our method to learn highly dense regions by computing *mode coverage*, which is defined to be $P(dQ < t)$ for t such that $Q(dQ < t) = 0.95$. Mode coverage essentially attempts to find high density regions of the model Q (based on t) and computes the mass of the target P under this region. Second, we compare the negative log likelihood, $-E_P[\log Q]$ as a general loss measure.

Domains We essentially consider three different problems. The first is the ring Gaussians problem now common to generative approaches (Goodfellow, 2016), in which 8 Gaussians have their modes regularly spaced on a

[5] Cf (Aldà and Rubinstein, 2017, Definition 4) for the sensitivity, (Aldà and Rubinstein, 2017, Section 6) for the key function $F_H(.,.)$ involved.

circle. The target P is shown in Figure 1. Second, we consider a mixture of three 1D gaussians with pdf $P(x) = \frac{1}{3}(\mathcal{N}(0.3, 0.01) + \mathcal{N}(0.5, 0.1) + \mathcal{N}(0.7, 0.1))$. For the final experiment, we consider a 1D domain and randomly place m gaussians with means centered in the interval $[0, 1]$ and variances 0.01. We vary $m = 1, \ldots, 10$, $\varepsilon \in (0, 2]$ and repeat the experiment four times to get means and standard deviations. More experiments can be found in the Appendix.

Results Figure 3 displays contour plots of the learned Q against DPB (Aldà and Rubinstein, 2017). Figure 4 provides metrics. We indicate the metric performance for DPB on one plot only since density estimates obtained for some of the other metrics could not allow for an accurate computation of metrics. The experiments bring the following observations: MBDE is significantly better at local differentially private density estimation than DPB if we look at the ring Gaussian problem. MBDE essentially obtains the same results as DPB for values of ε that are 400 times smaller as seen from Figure 1. We also remark that the density modelled are more smooth and regular for MBDE in this case. One might attribute the fact that our performance is much better on the ring Gaussians to the fact that our Q_0 is a standard Gaussian, located at the middle of the ring in this case, but experiments on random 2D Gaussians (see Appendix) display that our performances also remain better in other settings where Q_0 should represent a handicap. All domains, including the 1D random Gaussians experiments in Figure 1 (Appendix), display a consistent decreasing NLL for MBDE as ε increases, with sometimes very sharp decreases for $\varepsilon < 2$ (See also Appendix, Section 2). We attribute it to the fact that it is in this regime of the privacy parameter that MBDE captures all modes of the mixture. For larger values of ε, it justs fits better the modes already discovered. We also remark on the 1D Gaussians that DPB rapidly reaches a plateau of NLL which somehow show that there is little improvement as ε increases, for $\varepsilon \geq 1$. This is not the case for MBDE, which still manages some additional improvements for $\varepsilon > 5$ and significantly beats DPB. We attribute it to the flexibility of the sufficient statistics as (deep) classifiers in MBDE. The 1D random Gaussian problem (Figure 1 in Appendix) displays the same pattern for MBDE. We also observe that the standard deviation of MBDE is often 100 times *smaller* than for DPB, indicating not just better but also much more stable results. In the case of mode coverage, we observe for several experiments (*e.g.* ring Gaussians) that the mode coverage *decreases* until $\varepsilon \approx 1$, and then increases, on all domains, for MBDE. This, we believe is due to our choice of Q_0, which as a Gaussian, already captures with its mode a part of the existing modes. As ε increases however, MBDE performs better and obtains in general a significant improvement over Q_0. We also observe this phenomenon for the random 1D Gaussians (Figure 5) where the very small standard deviations (at least for $\varepsilon > .25$ or $m > 1$) display a significant stability for the solutions of MBDE.



Mean = f(ε) StDev = f(ε)

Figure 5: Mode coverage for MBDE on 1D random Gaussian.

6 Discussion and Conclusion

In this paper, we proposed a new method to learn densities that can be sampled from privately at the local level, paving the way for synthetic data generation. In order to prove privacy guarantees, we introduced the notion of mollifiers, which are of independent interest. Furthermore, we proved convergence guarantees of our method in the context of boosting along with additional formal results regarding capturing of modes and approximation of the target density. The use of the boosting framework allows to dampen the effects of a "curse of complexity" – *e.g.* when the dimension of the support of P increases –, as convergence primarily relies on weak guessing in sampling P vs sampling vs $Q_\cdot$. Additional assumptions, like sparsity in the expected parameters of the target or publicly available information allowing to tune Q_0, could boost further convergence. Finally, we conducted experiments, which advocate for our method, especially on the utility side of things when it comes to capturing statistical features of the true distribution.

Acknowledgments

We are indebted to Benjamin Rubinstein for providing us with the Private KDE code. We would also like to thank Arthur Street, Gianpaolo Gioiosa and anonymous reviewers for significant help in correcting and improving focus, clarity and presentation.

References

Aldà, F. and Rubinstein, B. (2017). The Bernstein mechanism: Function release under differential privacy. In *AAAI'17*.

Amari, S.-I. and Nagaoka, H. (2000). *Methods of Information Geometry*. Oxford University Press.

Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., and Seefeld, B. (2017). Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459. ACM.

Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. (2018). Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 1655–1658, New York, NY, USA. ACM.

Cranko, Z. and Nock, R. (2019). Boosted density estimation remastered. In *ICML'19*.

Differential privacy team, Apple (2017). Learning with differential privacy at scale.

Dwork, C. and Roth, A. (2014). The algorithmic foudations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407.

Erlingsson, U., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA. ACM.

Givens, G.-F. and Hoeting, J.-A. (2013). *Computational Statistics*. Wiley.

Goodfellow, I. (2016). Generative adversarial networks. NIPS'16 tutorials.

Hall, R., Rinaldo, A., and Wasserman, L.-A. (2013). Differential privacy for functions and functional data. *JMLR*, 14(1):703–727.

Hardt, M., Ligett, K., and McSherry, F. (2012). A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pages 2339–2347.

Hardt, M. and Rothblum, G. N. (2010). A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70. IEEE.

Kearns, M. (1988). Thoughts on hypothesis boosting. ML class project.

Machanavajjhala, A., Kifer, D., Abowd, J.-M., Gehrke, J., and Vilhuber, L. (2008). Privacy: Theory meets practice on the map. In *ICDE'08*, pages 277–286.

Raskhodnikova, S., Smith, A., Lee, H. K., Nissim, K., and Kasiviswanathan, S. P. (2008). What can we learn privately. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 531–540.

Rubinstein, B. and Aldà, F. (2017). Pain-free random differential privacy with sensivity sampling. In *34^{th} ICML*.

Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *MLJ*, 37:297–336.

Tolstikhin, I.-O., Gelly, S., Bousquet, O., Simon-Gabriel, C., and Schölkopf, B. (2017). Adagan: Boosting generative models. In *NIPS*30*, pages 5430–5439.

Ullman, J. (2015). Private multiplicative weights beyond linear queries. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 303–312. ACM.

Wang, Y.-X., Fienberg, S., and Smola, A.-J. (2015). Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *32^{nd} ICML*, pages 2493–2502.

Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. (2018). Differentially private generative adversarial network. *CoRR*, abs/1802.06739.