PrivInfer: A framework for differentially private Bayesian Machine Learning

Gilles Barthe[†], Gian Pietro Farina[*], Marco Gaboardi[*], Emilio Gallego Jesús Arias[+], Andrew D. Gordon[$], Justin Hsu[#], Pierre-Yves Strub[†]

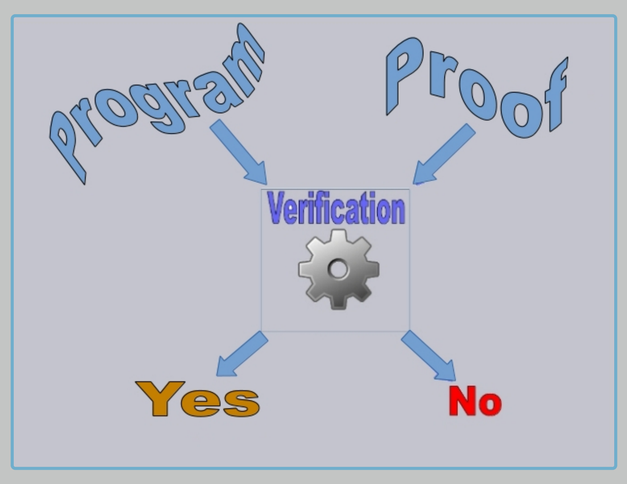
[†]Imdea Software, [*]SUNY University at Buffalo, [+]CRI Mines-ParisTech, [$]Microsoft Research, [#]University of Pennsylvania

What is PrivInfer?

PrivInfer is a framework which captures differentially private Bayesian machine learning algorithms, using techniques from programming languages literature. Its main components are:

- A probabilistic higher order functional progamming language to write probabilistic programs.
- A strong type system to **formally certify** probabilistic properties of programs such as differential privacy.

What is Formal Certification of Programs?



- With certification we can check that a *formal proof* of differential privacy of a *program* is indeed correct.
- In the case of **PrivInfer**:
  - ▷ proof $\sim$ type annotations.

Differentially private probabilistic inference

We are interested in giving a formal proof of programs that compute an *inference process* with the following parameters.

Inputs

- private data $\vec{x}$,
- public data $\vec{y}$,
- a set of parameter θ,
- a public prior $p(\theta)$.

Output

- a public posterior distribution $p(\theta \mid \vec{x}, \vec{y})$.

We want that such process to be (ϵ, δ)-differentially private.

A general point of view

Possible ways to achieve differential privacy in a Bayesian setting:

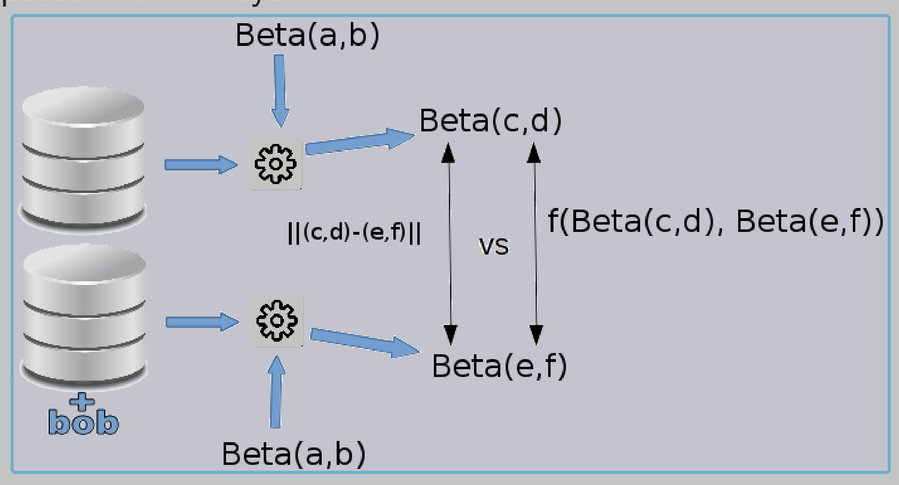
We can achieve differential privacy in different ways:

- By using specific families of priors
- By perturbing the inference algorithm
- By perturbing the input itself
- By perturbing the output of the inference

Output Perturbation: utility, sensitivity and metrics

In most works on Bayesian machine learning under differential privacy *sensitivity* is computed w.r.t to a metric on the parameters that specify the distribution (e.g. ℓ_1) but the *utility* is computed w.r.t a probabilistic distance.

It is natural to consider a situation where the sensitivity is computed with respect to a metric over probability measures:
PrivInfer for reasoning about programs where *utility* and *sensitivity* are computed in both ways.



Sensitivity with different metrics

What is f? f-divergences

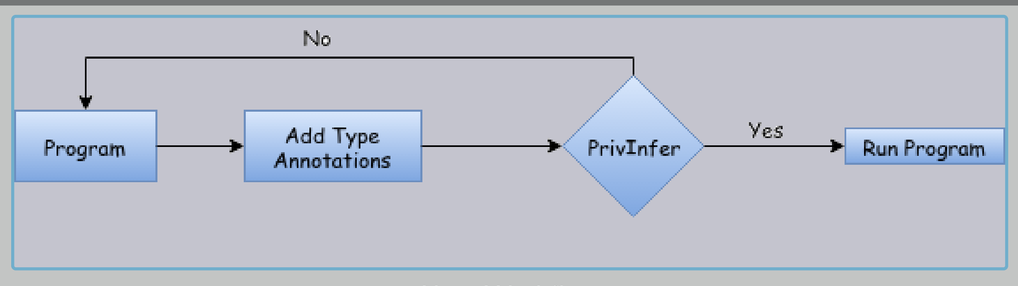
f-divergences is a rich class of metrics on probability distributions. Inspired by the definition of relative entropy, f-divergences are defined by a convex function f.

Crucially (ϵ, δ)-differential privacy is also an f-divergence.

f-diverg.	$f(x)$	Simplified form
HD(x)	$\frac{1}{2}(\sqrt{x}-1)^2$	$\sum_{a\in A}\frac{1}{2}\left(\sqrt{\mu_1(a)}-\sqrt{\mu_2(a)}\right)^2$
KL(x)	$x\ln(x)-x+1$	$\sum_{a\in A}\mu_1(a)\ln\left(\frac{\mu_1(a)}{\mu_2(a)}\right)$
ϵ-**D**(x)	$\max(x-e^\epsilon,0)$	$\sum_{a\in A}\max\left(\mu_1(a)-e^\epsilon\mu_2(a),0\right)$

Table 1: f-divergences for Hellinger distance (**HD**), KL divergence (**KL**), and ϵ-distance (ϵ-**D**)

Using PrivInfer



User Workflow

With **PrivInfer** we can formally verify programs where (ϵ, δ) differential privacy is achieved by

- Perturbing the input
- Perturbing the output

With **PrivInfer**, the user can also reason about other properties of his program which are defined in terms of f-divergences.