Towards differentially private probabilistic programming

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

[+]CRI Mines-ParisTech, [§]Harvard University, [†]Imdea Software, [$]Microsoft Research, [*]University of Dundee, [#]University of Pennsylvania

Objectives

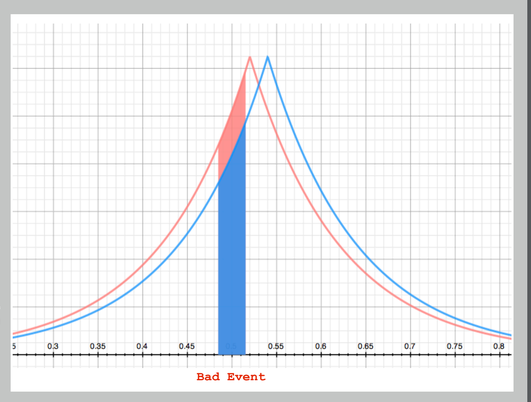
Design a tool for differentially private probabilistic programming featuring:

1. programming constructs to describe bayesian models and perform probabilistic inference,
2. programming constructs useful to ensure differential privacy,
3. type-checking as a method to ensure that the actual programs are differentially private.

Differential privacy

(ϵ, δ)-differential privacy [1] is a property of a mechanism $\mathbf{M : db} \to \mathcal{R}$ asserting that for any two databases $\mathbf{D}$ and $\mathbf{D'}$ differing in one row and for any subset $\mathbf{S} \subseteq \mathcal{R}$ of outputs:

$$\mathbf{Pr[M(D) \in S] \leq e^{\epsilon} \cdot Pr[M(D') \in S] + \delta}$$



- A standard way to ensure differential privacy is by adding some *statistical noise* to the result of a data analysis.
- Several mechanisms can ensure differential privacy and achieve at the same time a good *accuracy*.
- An important aspect of the theory of differential privacy is that it provides several *composition schemes* useful to compose different mechanisms.
- Robust to post-processing.

Differentially private probabilistic inference

We are interested in an *inference process* with the following parameters.

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

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

We want that such process satisfies

$$\mathbf{Pr[p}(\theta \mid \vec{x_1}, \vec{y}) \in \mathbf{S]} \leq \mathbf{e}^{\epsilon} \cdot \mathbf{Pr[p}(\theta \mid \vec{x_2}, \vec{y}) \in \mathbf{S]} + \delta.$$

for given values ϵ and δ, and for every two sets of private data $\vec{x_1}$ and $\vec{x_2}$ differing in one element.

Approaches for differentially private probabilistic inference

Adding noise on the input This corresponds to use some differentially private mechanism $\mathbf{M}$ to release $\vec{z} = \mathbf{M}(\vec{x})$ and then compute $\mathbf{p}(\theta \mid \vec{z}, \vec{y})$. This approach has been explored by Williams and McSherry [2].

Adding noise on the output This corresponds to use some differentially private mechanism $\mathbf{M}$ to release $\mathbf{M}(\mathbf{p}(\theta \mid \vec{x}, \vec{y}))$. The mechanism $\mathbf{M}$ should add noise proportional to the sensitivity of $\mathbf{p}(\theta \mid \vec{x}, \vec{y})$ with respect to $\vec{x}$:

$$\max_{\vec{x_1} \sim_1 \vec{x_2}} \mathbf{d}(\mathbf{p}(\theta \mid \vec{x_1}, \vec{y}), \mathbf{p}(\theta \mid \vec{x_2}, \vec{y}))$$

where $\mathbf{d}$ is a distance defined on the output of the mechanism.

Adding noise to the probabilistic inference process This corresponds to use some differentially private mechanism $\mathbf{M}$ to add noise in the inference of $\mathbf{p}(\theta \mid \vec{x}, \vec{y})$. This can be achieved by using an ad hoc algorithm adding noise when performing Bayesian update, or by designing differentially private versions of standard inference algorithms like variational methods or expectation propagation.

Noise-free differentially private Bayesian inference Under some conditions on the priors we can have that

$$\mathbf{Pr[p}(\theta \mid \vec{x_1}, \vec{y}) \in \mathbf{S]} \leq \mathbf{e}^{\epsilon} \cdot \mathbf{Pr[p}(\theta \mid \vec{x_2}, \vec{y}) \in \mathbf{S]} + \delta.$$

even without adding noise. This approach has been studied in [3].

An example in Infer.net Fun [4]

$$\begin{array}{l}
\textbf{Let DrugRehab } (\vec{z}) = \\
\quad \textbf{let v} = \textbf{[for i in 0..100} \to 1/101\textbf{]} \\
\quad \textbf{let d} = \textbf{random(Discrete(v))} \\
\quad \textbf{let p} = \textbf{d}/100 \\
\quad \textbf{For z in } \vec{\textbf{z}} \textbf{\{observe x} \Rightarrow \textbf{(x = Bernoulli(p)) in z\}} \\
\quad \textbf{return p}
\end{array}$$

$$\begin{array}{l}
\textbf{Let } \vec{\textbf{z}} = \textbf{Exp}_{(\epsilon,\textbf{s})}(\vec{\textbf{x}}) \textbf{ in} \\
\textbf{Let Discrete(v)} = \textbf{infer} < \textbf{@DrugRehab@} > (\vec{\textbf{z}})
\end{array}$$

Here $\mathbf{Exp}_{(\epsilon,\mathbf{s})}$ is the *exponential mechanism* with score function $\mathbf{s}$.

HOARe^2: Higher Order Approximate Relational Refinement types

To ensure differential privacy we use the tool **HOARe^2** [5]. This tool uses type-checking based on *relational refinement types* to ensure that two executions of the same program on two different inputs satisfy the differential privacy requirement.

Theorem (Ensuring differential privacy)

Given a probabilistic program $\mathbf{c}$*, if we can assign it the following type*

$$\mathcal{G} \vdash \mathbf{c :: \{x :: db \mid x_1 \sim_1 x_2\}} \to \mathfrak{M}_{\epsilon,\delta}\mathbf{\{y :: out \mid y_1 = y_2\}}$$

then $\mathbf{c}$ *represents an* (ϵ, δ)*-differentially private mechanism.*

Extending HOARe^2

- The language underlying **HOARe^2** is a probabilistic language *without observations*, using it for probabilistic inference requires some extension.
  - ▷ The semantics of **HOARe^2** has to be extended to accommodates sub-distributions and a filter semantics for *observe*.
  - ▷ the type system needs to be enriched with a typing rule for *observe*

$$\frac{\mathcal{G}, \mathbf{x :: \{x :: A \mid \phi\}} \vdash \mathbf{t :: \{b :: bool \mid b_1 = b_2\}} \quad \mathcal{G} \vdash \mathbf{u} :: \mathfrak{M}_{\epsilon,\delta}\mathbf{\{x :: A \mid \phi\}}}{\mathcal{G} \vdash \mathbf{observe\ x} \Rightarrow \mathbf{t\ in\ u} :: \mathfrak{M}_{\epsilon,\delta}\mathbf{\{x :: A \mid \phi \wedge t_1 = t_2 = true\}}}$$

Conclusion

- The aim of our work is designing a framework for differentially private Bayesian inference,
- we have studied here a way to ensure it by adding noise on the input,
- we believe that also all the other approaches are valuable and we plan to extend our framework for accommodating some of them in the future.
- we plan also to extend the framework to verify the accuracy of the probabilistic inference.

References

[1] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith.
Calibrating noise to sensitivity in private data analysis.
In *Theory of Cryptography, TCC*, volume 3876 of *LNCS*, pages 265–284. Springer, 2006.

[2] Oliver Williams and Frank McSherry.
Probabilistic inference and differential privacy.
In *Neural Information Processing Systems, NIPS*, pages 2451–2459, 2010.

[3] Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein.
Robust and private bayesian inference.
In *Algorithmic Learning Theory, ALT*, pages 291–305, 2014.

[4] Johannes Borgström, Andrew D. Gordon, Michael Greenberg, James Margetson, and Jurgen Van Gael.
Measure transformer semantics for bayesian machine learning.
In *European Symposium On Programming*, pages 77–96, 2011.

[5] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub.
Higher-order approximate relational refinement types for mechanism design and differential privacy.
In *Principles of Programming Languages, POPL*. ACM, 2015.