Property Testing for Differential Privacy

Anna C. Gilbert* and Audra McMillan#
Presented by Rishi Sonthalia*

*Department of Mathematics
University of Michigan

#Department of Computer Science
Boston University

October 2, 2018

Motivation

- Companies collect a lot of data, potentially violating privacy.

Motivation

- Companies collect a lot of data, potentially violating privacy.
- Differential privacy has developed as a methodology for producing meaningful data statistics while preserving privacy.

Motivation

- Companies collect a lot of data, potentially violating privacy.
- Differential privacy has developed as a methodology for producing meaningful data statistics while preserving privacy.
- DP has been gaining traction outside of theoretical research with several companies announcing large-scale deployment of DP mechanisms.

Motivation

- Companies collect a lot of data, potentially violating privacy.
- Differential privacy has developed as a methodology for producing meaningful data statistics while preserving privacy.
- DP has been gaining traction outside of theoretical research with several companies announcing large-scale deployment of DP mechanisms.
- The software behind the deployment is typically proprietary since it ostensibly provides commercial advantage.

Motivation

- Companies collect a lot of data, potentially violating privacy.
- Differential privacy has developed as a methodology for producing meaningful data statistics while preserving privacy.
- DP has been gaining traction outside of theoretical research with several companies announcing large-scale deployment of DP mechanisms.
- The software behind the deployment is typically proprietary since it ostensibly provides commercial advantage.
- With limited access to the software, can we verify the privacy guarantees of purportedly DP algorithms?

Motivation

- Companies collect a lot of data, potentially violating privacy.
- Differential privacy has developed as a methodology for producing meaningful data statistics while preserving privacy.
- DP has been gaining traction outside of theoretical research with several companies announcing large-scale deployment of DP mechanisms.
- The software behind the deployment is typically proprietary since it ostensibly provides commercial advantage.
- With limited access to the software, can we verify the privacy guarantees of purportedly DP algorithms?

Main Problem: Given black-box access to an algorithm claiming to perform a differentially private computation, can we test whether it is private?

(Informal) results

Main conclusion: Verifying differential privacy claims is HARD.

(Informal) results

Main conclusion: Verifying differential privacy claims is HARD.

- Verifying differential privacy is at least as hard as breaking privacy.

(Informal) results

Main conclusion: Verifying differential privacy claims is HARD.

- Verifying differential privacy is at least as hard as breaking privacy.

Let n be the size of the output space of the algorithm being tested.

- If the verifier is given no information about the algorithm then no verifier for any reasonable differential privacy definition can be run in time sublinear in n.

(Informal) results

Main conclusion: Verifying differential privacy claims is HARD.

- Verifying differential privacy is at least as hard as breaking privacy.

Let n be the size of the output space of the algorithm being tested.

- If the verifier is given no information about the algorithm then no verifier for any reasonable differential privacy definition can be run in time sublinear in n.
- Even if the verifier is given an (untrusted) full description of the algorithm being tested, only certain weak definitions of privacy can run in sublinear time.

(Informal) results

Main conclusion: Verifying differential privacy claims is HARD.

- Verifying differential privacy is at least as hard as breaking privacy.

Let n be the size of the output space of the algorithm being tested.

- If the verifier is given no information about the algorithm then no verifier for any reasonable differential privacy definition can be run in time sublinear in n.
- Even if the verifier is given an (untrusted) full description of the algorithm being tested, only certain weak definitions of privacy can run in sublinear time.

Differential Privacy

Two databases D and D' are neighbours if they differ on the data of a single individual.

Differential Privacy

Two databases D and D' are neighbours if they differ on the data of a single individual.

- A randomised algorithm $\mathcal{A}$ is ϵ-pure differentially private if for all neighbouring databases D, D' we have

$$\sup_K \frac{\Pr(\mathcal{A}(D) \in K)}{\Pr(\mathcal{A}(D') \in K)} \leq e^{\epsilon}.$$

Differential Privacy

Two databases D and D' are neighbours if they differ on the data of a single individual.

- A randomised algorithm $\mathcal{A}$ is ϵ-pure differentially private if for all neighbouring databases D, D' we have

$$\sup_K \frac{\Pr(\mathcal{A}(D) \in K)}{\Pr(\mathcal{A}(D') \in K)} \leq e^{\epsilon}.$$

- A randomised algorithm $\mathcal{A}$ is (ϵ, δ)-approximately differentially private if for all neighbouring databases D, D' and we have

$$\Pr(\mathcal{A}(D) \in K) \leq e^{\epsilon} \Pr(\mathcal{A}(D') \in K) + \delta.$$

The supremums are taken over all events K. The smaller ϵ and δ are, the "more private" the algorithm is. Think of $\epsilon \approx 0.1$ and $\delta \approx 10^{-6}$. Any outcome that occurs when the database is D is almost as likely to occur when the databases is D'.

Differential Privacy

Two databases D and D' are neighbours if they differ on the data of a single individual.

- A randomised algorithm $\mathcal{A}$ is ϵ-pure differentially private if for all neighbouring databases D, D' we have

$$\sup_K \frac{\Pr(\mathcal{A}(D) \in K)}{\Pr(\mathcal{A}(D') \in K)} \leq e^{\epsilon}.$$

- A randomised algorithm $\mathcal{A}$ is (ϵ, δ)-approximately differentially private if for all neighbouring databases D, D' and we have

$$\Pr(\mathcal{A}(D) \in K) \leq e^{\epsilon} \Pr(\mathcal{A}(D') \in K) + \delta.$$

The supremums are taken over all events K. The smaller ϵ and δ are, the "more private" the algorithm is. Think of $\epsilon \approx 0.1$ and $\delta \approx 10^{-6}$. Any outcome that occurs when the database is D is almost as likely to occur when the databases is D'.

Random Differential Privacy

If there is a distribution on the data universe then we can define a weaker notion of privacy. Let $\mathcal{D}$ be a distribution on the data universe.

- A randomised algorithm $\mathcal{A}$ is (ϵ, γ)-random pure differentially private if

$$\Pr\left[\sup_{K} \frac{\Pr(\mathcal{A}(D) \in K)}{\Pr(\mathcal{A}(D') \in K)} \leq e^{\epsilon}\right] \geq 1 - \gamma$$

- A randomised algorithm $\mathcal{A}$ is $(\epsilon, \delta, \gamma)$-random approximate differentially private if

$$\Pr\left[\forall K, \Pr(\mathcal{A}(D) \in K) \leq e^{\epsilon}\Pr(\mathcal{A}(D') \in K) + \delta\right] \geq 1 - \gamma$$

where the the outer probability is taken over pairs of neighbouring databases sampled from $\mathcal{D}^m$.

Random Differential Privacy

If there is a distribution on the data universe then we can define a weaker notion of privacy. Let $\mathcal{D}$ be a distribution on the data universe.

- A randomised algorithm $\mathcal{A}$ is (ϵ, γ)-random pure differentially private if

$$\Pr\left[\sup_{K} \frac{\Pr(\mathcal{A}(D) \in K)}{\Pr(\mathcal{A}(D') \in K)} \leq e^{\epsilon}\right] \geq 1 - \gamma$$

- A randomised algorithm $\mathcal{A}$ is $(\epsilon, \delta, \gamma)$-random approximate differentially private if

$$\Pr\left[\forall K, \Pr(\mathcal{A}(D) \in K) \leq e^{\epsilon}\Pr(\mathcal{A}(D') \in K) + \delta\right] \geq 1 - \gamma$$

where the the outer probability is taken over pairs of neighbouring databases sampled from $\mathcal{D}^m$.

Should the data analyst get an unlikely sample from the population, privacy may be violated. Since γ is the probability of choosing a "bad" sample, we would like it to be VERY small.

What form does a "verifier" take? Property Testing

Goal: Answer the question, *given a set of privacy parameters, is the algorithm at least this private?*

What form does a "verifier" take? Property Testing

Goal: Answer the question, *given a set of privacy parameters, is the algorithm at least this private?*

Access model: We are allowed to give the algorithm an input database D and observe an output $\mathcal{A}(D)$. We may also be given some side information about the algorithm.

What form does a "verifier" take? Property Testing

Goal: Answer the question, *given a set of privacy parameters, is the algorithm at least this private?*

Access model: We are allowed to give the algorithm an input database D and observe an output $\mathcal{A}(D)$. We may also be given some side information about the algorithm.

A property testing algorithm with query complexity q, proximity parameter α, privacy parameters Σ and side information S, makes q queries to the black-box and:

- (Completeness) ACCEPTS with probability at least $2/3$ if $\mathcal{A}$ is Σ-private and S is accurate.
- (Soundness) REJECTS with probability at least $2/3$ if $\mathcal{A}$ is α-far from being Σ-private.

What form does a "verifier" take? Property Testing

Goal: Answer the question, *given a set of privacy parameters, is the algorithm at least this private?*

Access model: We are allowed to give the algorithm an input database D and observe an output $\mathcal{A}(D)$. We may also be given some side information about the algorithm.

A property testing algorithm with query complexity q, proximity parameter α, privacy parameters Σ and side information S, makes q queries to the black-box and:

- (Completeness) ACCEPTS with probability at least $2/3$ if $\mathcal{A}$ is Σ-private and S is accurate.
- (Soundness) REJECTS with probability at least $2/3$ if $\mathcal{A}$ is α-far from being Σ-private.

Note that a valid verifier is allowed to reject simply because the side information it was given is inaccurate.

What does it mean to be *far* from private?

The definition of a property testing verifier required a notion of distance on the space of privacy parameters.

What does it mean to be *far* from private?

The definition of a property testing verifier required a notion of distance on the space of privacy parameters.

Privacy Notion	Σ	$\|\Sigma - \Sigma'\|$
pure DP	ϵ	$\lvert\epsilon - \epsilon'\rvert$
approximate DP	(ϵ, δ)	$\lvert\delta - \delta'\rvert$
random pure DP	(ϵ, γ)	$\min\{\lvert\epsilon - \epsilon'\rvert, \lambda\lvert\gamma - \gamma'\rvert\}$
random approximate DP	$(\epsilon, \delta, \gamma)$	$\min\{\lvert\delta_\epsilon - \delta'_\epsilon\rvert, \lambda\lvert\gamma - \gamma'\rvert\}$

The algorithm $\mathcal{A}$ is α-far from being Σ-private if $\min_{\Sigma'} \|\Sigma' - \Sigma\| > \alpha$, where the minimum is over all Σ' such that $\mathcal{A}$ is Σ'-private.

The scalar λ to penalise deviation in one parameter more than deviation in another parameter.

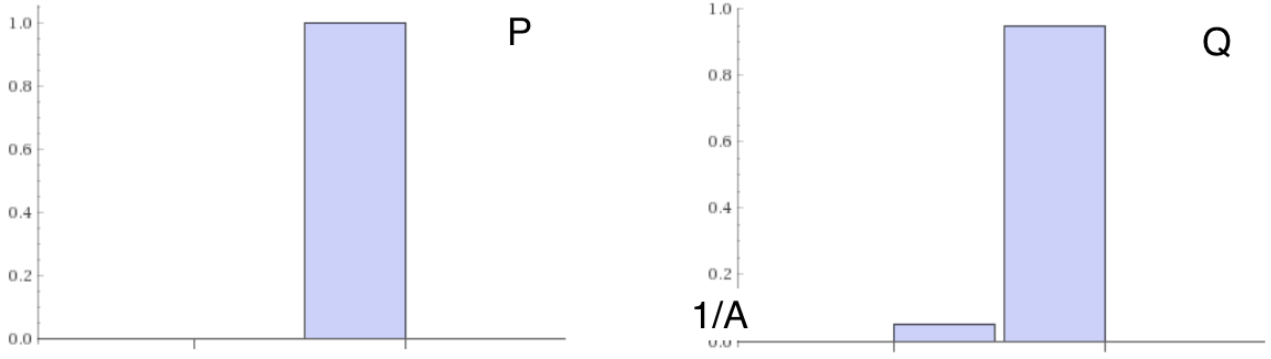
Why is verifying pure DP so hard?

Intuitively, the difficulty arises because two distributions can be close in statistical distance but still have high privacy parameters.

Why is verifying pure DP so hard?

Intuitively, the difficulty arises because two distributions can be close in statistical distance but still have high privacy parameters.

Consider the below two distributions.

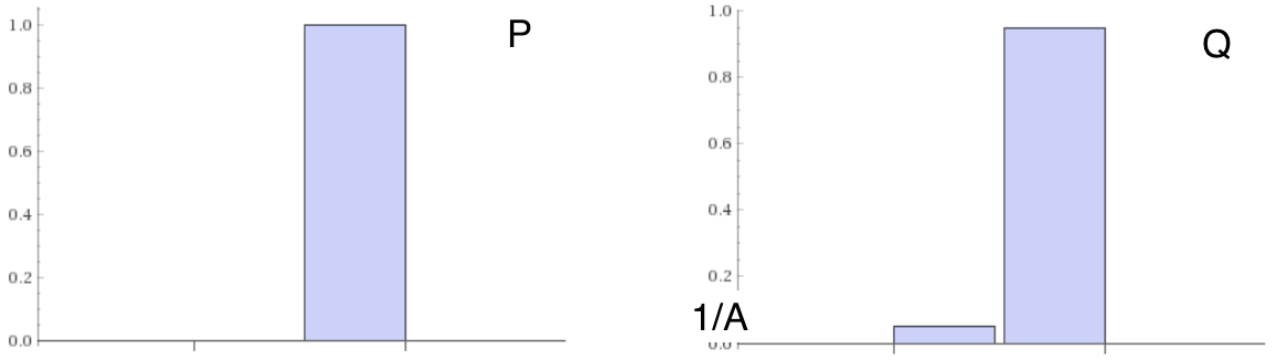


- Suppose we are told that for every database the output distribution is given by P. Such an algorithm is 0-pure DP.

Why is verifying pure DP so hard?

Intuitively, the difficulty arises because two distributions can be close in statistical distance but still have high privacy parameters.

Consider the below two distributions.

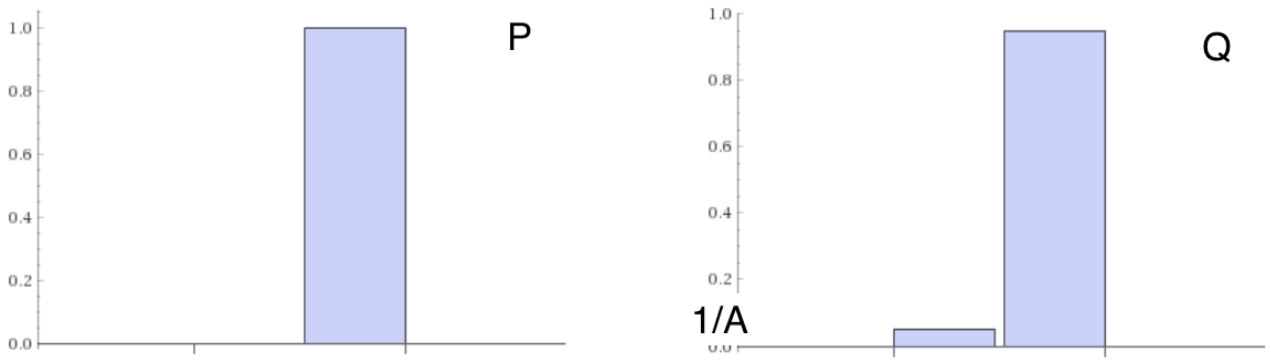


- Suppose we are told that for every database the output distribution is given by P. Such an algorithm is 0-pure DP.
- Imagine that actually the output distribution of $\mathcal{A}$ is sometimes P and sometimes Q. Then $\mathcal{A}$ is ∞-pure DP.

Why is verifying pure DP so hard?

Intuitively, the difficulty arises because two distributions can be close in statistical distance but still have high privacy parameters.

Consider the below two distributions.



- Suppose we are told that for every database the output distribution is given by P. Such an algorithm is 0-pure DP.
- Imagine that actually the output distribution of $\mathcal{A}$ is sometimes P and sometimes Q. Then $\mathcal{A}$ is ∞-pure DP.
- It takes at least A samples to distinguish between a 0-pure DP algorithm and a ∞-approximate DP algorithm. Take $A \to \infty$ to get that verification is impossible *even if we are given a complete, but untrusted description of the algorithm.*

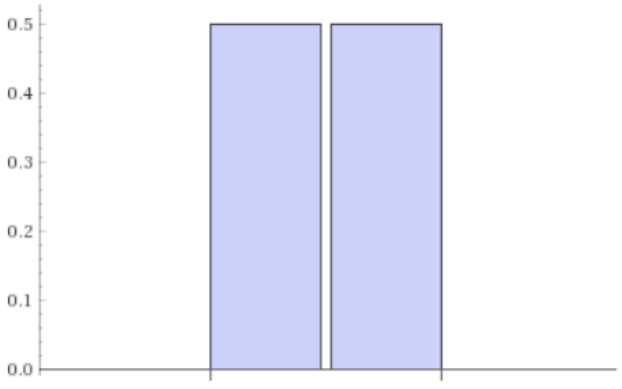
How do we get around this?

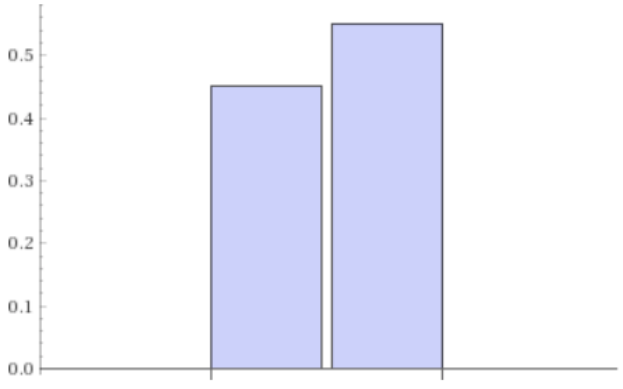
For approximate DP: we can push these "bad" events into the δ term. We still need at least $\theta\left(\frac{1}{\delta^2}\right)$ samples to *detect* the disclosive events.

How do we get around this?

For approximate DP: we can push these "bad" events into the δ term. We still need at least $\theta\left(\frac{1}{\delta^2}\right)$ samples to *detect* the disclosive events.

In the full information setting: For some distributions being close in statistical distance *does* imply the privacy parameters are small. Suppose we are told that the output distribution is a fair coin. Then being close in statistical distance does imply the privacy parameters are small since the quantity $\frac{\Pr(\mathcal{A}(D)=\pm)}{\Pr(\mathcal{A}'(D)=\pm)} = \frac{1/2 \pm 1/A}{1/2}$ can be controlled.

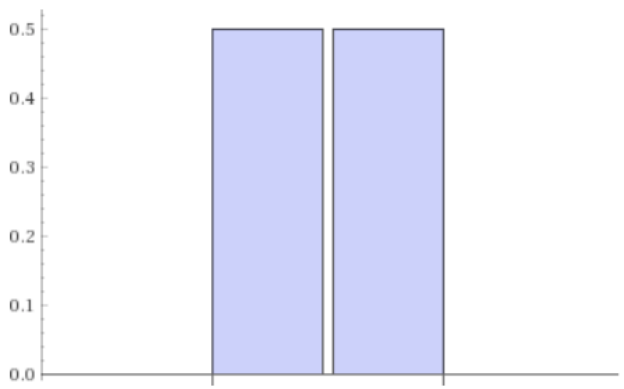


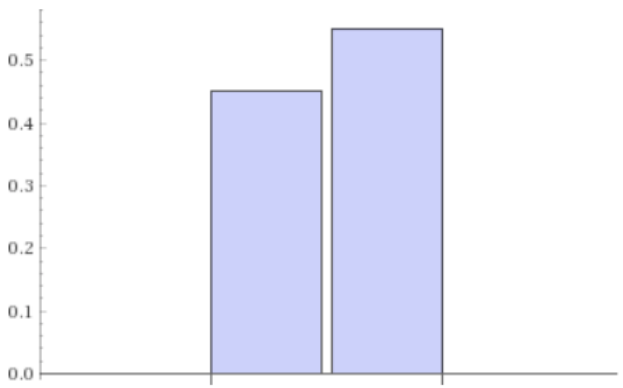


How do we get around this?

For approximate DP: we can push these "bad" events into the δ term. We still need at least $\theta\left(\frac{1}{\delta^2}\right)$ samples to *detect* the disclosive events.

In the full information setting: For some distributions being close in statistical distance *does* imply the privacy parameters are small. Suppose we are told that the output distribution is a fair coin. Then being close in statistical distance does imply the privacy parameters are small since the quantity $\frac{\Pr(\mathcal{A}(D)=\pm)}{\Pr(\mathcal{A}'(D)=\pm)} = \frac{1/2 \pm 1/A}{1/2}$ can be controlled.





We can extend this argument to whenever we know the minimum probability of any event, $\beta = \min_E \min_D \Pr(\mathcal{A}(D) \in E)$ is non-zero.

Bounds on Query Complexity of Privacy Verifiers.

The table contains per database query complexities. Multiply by the number of databases for the total query complexity of DP and by $\approx \frac{1}{\gamma}$ for the total query complexity of random DP.

No Information = the only known quantity is n

Full Information = a complete, untrusted description of the algorithm is given to the verifier.

	No Information	Full information
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$ $O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$ $O(\frac{n}{\alpha^2})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$

Bounds on Query Complexity of Privacy Verifiers.

The table contains per database query complexities. Multiply by the number of databases for the total query complexity of DP and by $\approx \frac{1}{\gamma}$ for the total query complexity of random DP.

$\alpha =$ proximity parameter of the property testing algorithm

	No Information	Full information
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$ $O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$ $O(\frac{n}{\alpha^2})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$

$n =$ size of output space of $\mathcal{A}$

$\beta = \min_E \min_D \Pr(\mathcal{A}(D) \in E)$ is the minimum probability of any event

Bounds on Query Complexity of Privacy Verifiers

	No Information	Full information	
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$	
		$O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$	$\times\frac{1}{\gamma}$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$	
	$O(\frac{n}{\alpha^2})$		

- Since $\beta < \frac{1}{n}$, verifying pure DP can not be done sublinear time even in the full information case.

Bounds on Query Complexity of Privacy Verifiers

	No Information	Full information	
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$	
		$O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$	$\times\frac{1}{\gamma}$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$	
	$O(\frac{n}{\alpha^2})$		

- Since $\beta < \frac{1}{n}$, verifying pure DP can not be done sublinear time even in the full information case.
- For verifying pure DP, $\alpha \approx \epsilon$, which is "small but not tiny" (say, 0.1) so $\frac{1}{\alpha^2}$ is feasible.

Bounds on Query Complexity of Privacy Verifiers

	No Information	Full information
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$
		$O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$
	$O(\frac{n}{\alpha^2})$	

$\times\frac{1}{\gamma}$

- Since $\beta < \frac{1}{n}$, verifying pure DP can not be done sublinear time even in the full information case.
- For verifying pure DP, $\alpha \approx \epsilon$, which is "small but not tiny" (say, 0.1) so $\frac{1}{\alpha^2}$ is feasible.
- For verifying approximate DP, $\alpha \approx \delta$. Now δ is, roughly, the probability of blatantly violating privacy so we want it to be extremely small (say, 10^{-8}) so $\frac{1}{\alpha^2}$ is infeasibly large.

Bounds on Query Complexity of Privacy Verifiers

	No Information	Full information
pure DP	Unverifiable	$\Omega\left(\frac{1}{\beta\alpha^2}\right)$
		$O\left(\frac{\ln n}{\alpha^2\beta^2}\right)$
approximate DP	$\Omega(\max\{n^{1-o(1)}, \frac{1}{\alpha^2}\})$	$O\left(\frac{\sqrt{n}}{\alpha^2}\right)$
	$O(\frac{n}{\alpha^2})$	

$\times\frac{1}{\gamma}$

- Since $\beta < \frac{1}{n}$, verifying pure DP can not be done sublinear time even in the full information case.
- For verifying pure DP, $\alpha \approx \epsilon$, which is "small but not tiny" (say, 0.1) so $\frac{1}{\alpha^2}$ is feasible.
- For verifying approximate DP, $\alpha \approx \delta$. Now δ is, roughly, the probability of blatantly violating privacy so we want it to be extremely small (say, 10^{-8}) so $\frac{1}{\alpha^2}$ is infeasibly large.
- γ is also the probability of blatantly violating privacy so $\frac{1}{\gamma}$ is also infeasibly large.

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

- Suppose an algorithm $\mathcal{A}$ that samples from P on half the databases and Q on the other half is α-far from ϵ- pure DP.

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

- Suppose an algorithm $\mathcal{A}$ that samples from P on half the databases and Q on the other half is α-far from ϵ- pure DP.
- A valid verification algorithm must distinguish between this algorithm and one that *always* samples from P.

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

- Suppose an algorithm $\mathcal{A}$ that samples from P on half the databases and Q on the other half is α-far from ϵ- pure DP.
- A valid verification algorithm must distinguish between this algorithm and one that *always* samples from P.
- Thus, it must be able to distinguish between the distributions P and Q.

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

- Suppose an algorithm $\mathcal{A}$ that samples from P on half the databases and Q on the other half is α-far from ϵ- pure DP.
- A valid verification algorithm must distinguish between this algorithm and one that *always* samples from P.
- Thus, it must be able to distinguish between the distributions P and Q.
- There must be a pair of neighbouring databases for which one outputs a sample from P and the other outputs a sample from Q. If the number of samples required to distinguish between P and Q is small then $\mathcal{A}$ must not satisfy strong privacy guarantees (since a small number of samples are needed to distinguish D and D'.)

Being hard to verify is an inherent property of privacy

The infeasibility of the lower bounds are not surprising when we think about what verification means for privacy.

- Suppose an algorithm $\mathcal{A}$ that samples from P on half the databases and Q on the other half is α-far from ϵ- pure DP.
- A valid verification algorithm must distinguish between this algorithm and one that *always* samples from P.
- Thus, it must be able to distinguish between the distributions P and Q.
- There must be a pair of neighbouring databases for which one outputs a sample from P and the other outputs a sample from Q. If the number of samples required to distinguish between P and Q is small then $\mathcal{A}$ must not satisfy strong privacy guarantees (since a small number of samples are needed to distinguish D and D'.)
- Thus, if the verification query complexity is small, it must only verify a weak privacy guarantee.

Take-home Messages

The lower bounds we obtained for verifying differential privacy are infeasible for the scale of parameters that are typically considered reasonable in the differential privacy literature, even when we suppose that the verifier has access to an (untrusted) description of the algorithm.

So, verifying differential privacy requires compromise by either the verifier or the algorithm owner. Either the verifier has to be satisfied with a weak privacy guarantee, or the algorithm owner has to compromise on side information or access to the algorithm.

Thank you!

Full version available at arXiv:1806.06427.