Lecture 8 Privacy Accounting (Part II) and a demo of *autodp*

Yu-Xiang Wang

COMPUTER SCIENCE
UC SANTA BARBARA
Computing. ReInvented.

Recap: last lecture

- Equivalent definitions of differential privacy
  - Via Hockey-stick divergence
  - Via Tradeoff function
- Analytical Gaussian mechanism
- Mechanism-specific analysis

Recap: HS Divergence as an equivalent definition of DP

Theorem: M is (ε,δ)-DP if and only if

$$\sup_{D \simeq D'} H_{e^\epsilon}(\mathcal{M}(D) \| \mathcal{M}(D')) \leq \delta.$$

And if and only if

$$\Pr_{o \sim \mathcal{M}(D)}[L_{P,Q} > \epsilon] - e^\epsilon \Pr_{o \sim \mathcal{M}(D')}[L_{Q,P} < -\epsilon] \leq \delta$$

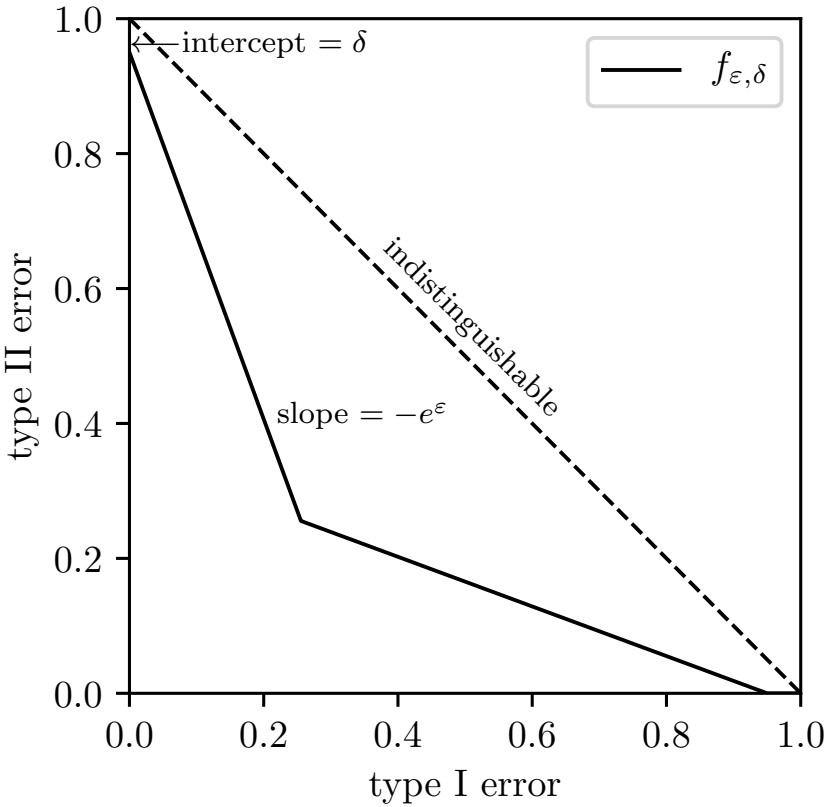
$$for\ all\ neighboring\ D, D'.$$

- Obtain exactly optimal Gaussian mechanism.

Recap: Tradeoff function characterization of DP

Theorem: M is (ε,δ)-DP if and only if the adversary's tradeoff region is lower bound by

$$f_{\varepsilon,\delta}(\alpha) = \max\left\{0, 1-\delta-\mathrm{e}^{\varepsilon}\alpha, \mathrm{e}^{-\varepsilon}(1-\delta-\alpha)\right\}$$



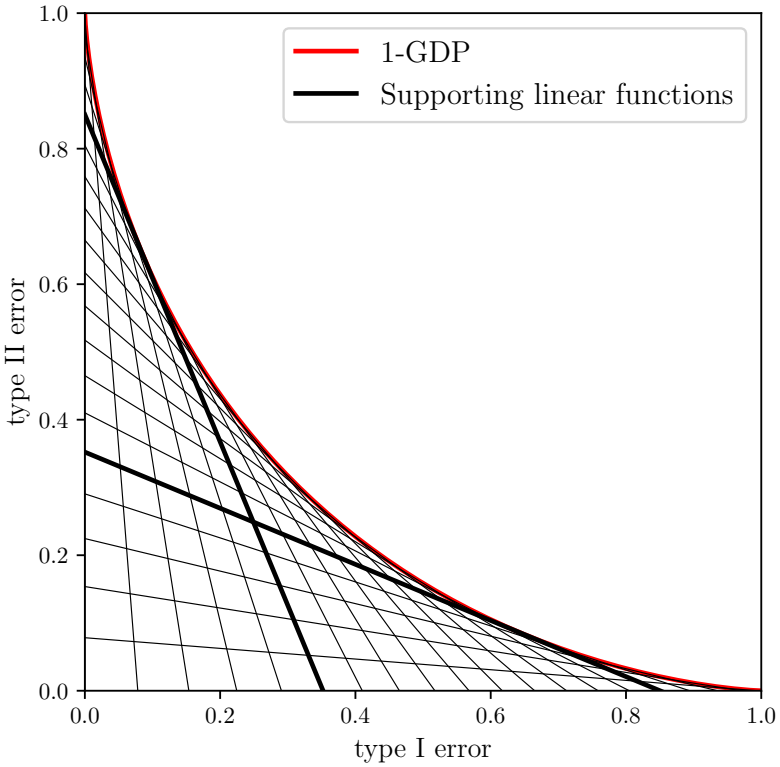
Recap: mechanism-specific analysis

- Describing a mechanism by two numbers (ε,δ) is somewhat crude, and lossy.
- More fine-grained description of a mechanism by a function
  - Privacy profile: $\delta(\varepsilon)$ --- smallest δ such that M satisfies (ε,δ) –DP. $\delta_{\mathcal{M}}(\epsilon) = \max_{x \simeq x'} H_\epsilon(\mathcal{M}(x') \| \mathcal{M}(x'))$
  - Tradeoff function $f_{\mathcal{M}} = \max_{x \simeq x'} T(\mathcal{M}(x), \mathcal{M}(x'))$
  - Renyi DP $\epsilon_{\mathcal{M}}(\alpha) = \max_{x \simeq x'} D_\alpha(\mathcal{M}(x) \| \mathcal{M}(x'))$
  - PLD: Distribution function of PLRV

Recap: Duality between tradeoff function and privacy profile

Proposition 2.12 (Primal to Dual). *For a symmetric trade-off function f, a mechanism is f-DP if and only if it is $\big(\varepsilon, \delta(\varepsilon)\big)$-DP for all $\varepsilon \geqslant 0$ with $\delta(\varepsilon) = 1 + f^*(-\mathrm{e}^{\varepsilon})$.*

- Example: Gaussian mechanism



This lecture

- Composition of mechanism-specific representations
  - RDP accountant
  - Fourier accountant
- Unified treatment via a dominating privacy loss random variable
  - And its characteristic function
- autodp: How you would represent DP mechanism and compute privacy loss

Readings

- “Optimal accounting of Differential Privacy Via Characteristic Function” by Zhu, Dong and W. https://arxiv.org/abs/2106.08567

- Autodp tutorial / a Jupyter notebook: https://github.com/yuxiangw/autodp/blob/master/tutorials/tutorial_new_api.ipynb

Composition of Mechanism Specific Representations

- If we have functional representations $f_1, f_2, \ldots, f_k$ of a mechanism M_1, M_2 ,…, M_k

- What is a functional representation of their composition?

Composition of Mechanism Specific Representations

1. RDP function: Just add up!
   - But… not tight

2. Tradeoff function:
   - Somewhat complex.
   - A central limit theorem exists

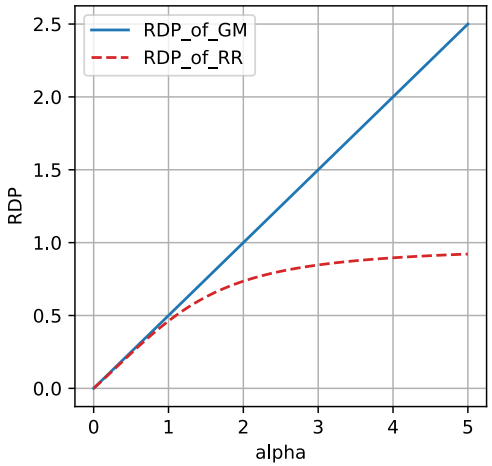
3. Privacy profile:
   - Convolution of the distribution of PLRVs for each pair of neighboring datasets

Renyi DP accountant a.k.a. analytical moments accountant

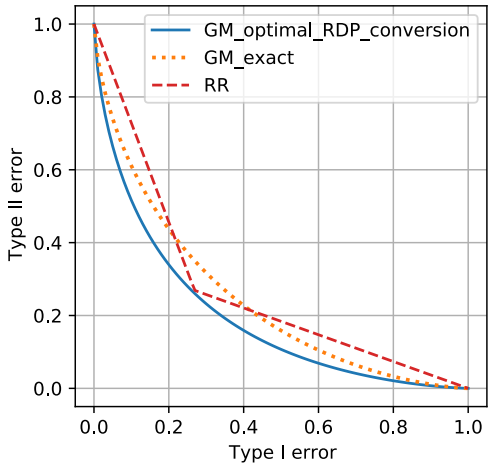
- Tracking RDP for all order as a symbolic functions.
- Numerical calculations for (ε, δ)-DP guarantees.
  - Could use the "tail bound" lemma
  - But more advanced conversion formula / algorithm exists.

(Abadi et al. 2016; Mironov, 2017; W., Balle, Kasiviswanathn, 2019)

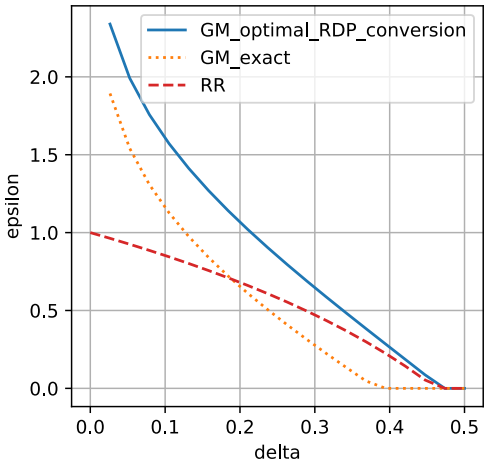
The conversion from RDP to approximate DP is lossy.



(a) RDP of RR and GM



(b) f-DP of RR and GM



(c) (ϵ, δ)-DP of RR and GM

Figure 1: The figure illustrates the RDP and f-DP of a Gaussian mechanism with (normalized) $\sigma = 1$, and a randomized response mechanism with $p = \frac{e}{1+e}$. Pane (a) shows the RDP function of RR and GM, clearly, RR also satisfies the same RDP of the Gaussian mechanism for all α. Pane (b) in the middle compares the f-DP of the two mechanisms, as well as the f-DP implied by the optimal conversion from RDP. Pane (c) shows the privacy profile of the two mechanisms, together with Pane (a), it demonstrates that the optimal f-DP and (ϵ, δ)-DP of GM cannot be achieved by a conversion from RDP.

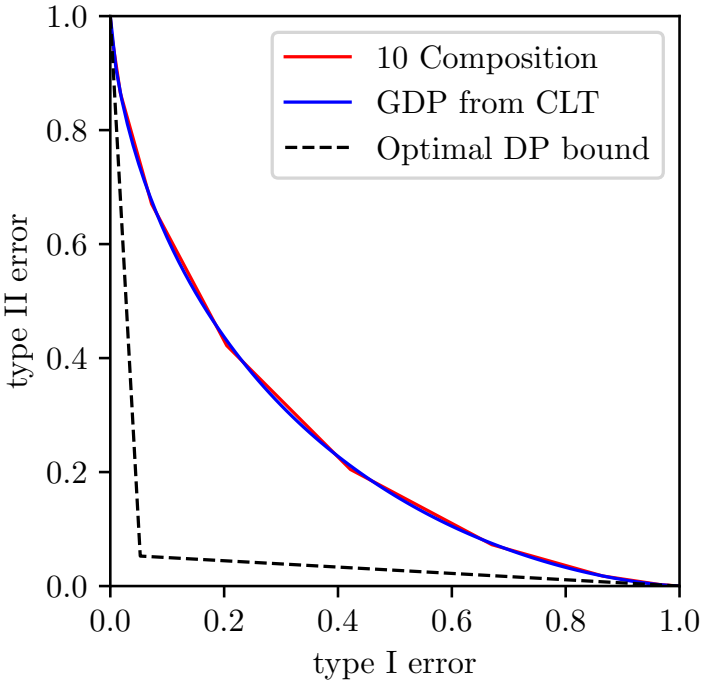
Composition of Tradeoff functions

- The composed mechanism

$$M : X \to Y_1 \times \cdots \times Y_n \ \textit{is} \ f_1 \otimes \cdots \otimes f_n\text{-}DP.$$

where $f \otimes g := T(P \times P', Q \times Q')$.

- A central limit theorem
  - Theorem 3.4 in Dong et al. 2019.



Composition of Privacy Profiles

- Somewhat tricky.
  - In principle, one can take all neighboring pairs of datasets, compose M_1, …, M_k by adding up the PLRVs.
  - Requires us to know the worst-pair of datasets to make it efficient.
  - Even then, how do we know the composition of worst-case pair for each is a worst-cases pair for the composition?

Dominating pair distributions

- We say that P,Q is a dominating pair of Mechanism M if for all α>0 $$\sup_{D \simeq D'} H_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq H_\alpha(P \| Q).$$
- When equal sign holds for all α>0, then it is a tight dominating pair.

Theorem 1: tight dominating pair always exists.

Theorem 2: Dominating pairs compose adaptively.

Theorem 3: (P,Q) is dominating if and only if M satisfies f-DP with f = T[P,Q].

(Zhu, Dong and W. https://arxiv.org/abs/2106.08567)

Two satisfying consequences

- Advanced composition for (ε,δ)-DP
  - One particular mechanism that attains the privacy-profile / tradeoff function pointwise.

Outcome	P_U	P_V
0	$\frac{e^{\varepsilon}(1-\delta)}{e^{\varepsilon}+1}$	$\frac{1-\delta}{e^{\varepsilon}+1}$
1	$\frac{1-\delta}{e^{\varepsilon}+1}$	$\frac{e^{\varepsilon}(1-\delta)}{e^{\varepsilon}+1}$
"I am U"	δ	0
"I am V"	0	δ

Leaky Randomized Response is a dominating pair for all (ε,δ)-DP mechanisms.

- Composition of Gaussian mechanism
  - Simply adding the PLRV or individual GMs

Advanced Composition Theorem for (ε,δ)-DP (proof for the δ>0 case)

- Composition of a sequence of k arbitrary (ε,δ)-DP mechanisms is dominated by the composition of k leaky randomized response.

Composition of Analytical Gaussian Mechanisms

- The adaptive composition for a sequence of Gaussian mechanisms with noise $\sigma_1, \sigma_2, \ldots$ and global L2 sensitivity $\Delta_1, \Delta_2, \ldots$ satisfies (ε,$\delta(\varepsilon)$)- DP with $\delta(\varepsilon) = \delta_{\mathcal{M}}(\varepsilon)$ where $\mathcal{M}$ is a Gaussian mechanism with noise multiplier $\sigma/\Delta = \left(\sum_i (\Delta_i/\sigma_i)^2\right)^{-1/2}$.

PLD formalism and Fourier accountant

- Discretizing the density function of the PLRV
- Fast Fourier Transform
- Multiply them together and take inverse FFT.
- Bound the approximation error

(Sommer et al. 2019; Koskela et al, 2020; Gopi et al, 2020)

Characteristic function representation of the dominating pairs

$$\Pr_{o\sim\mathcal{M}(D)}[L_{P,Q} > \epsilon] - e^{\epsilon}\Pr_{o\sim\mathcal{M}(D')}[L_{Q,P} < -\epsilon] \leq \delta$$

$$for\ all\ neighboring\ D, D'.$$

For a dominating pair P,Q, it suffices to represent the two PLRVs by their characteristic functions.

$$\phi_{\mathcal{M}}(\alpha) := \mathbb{E}_P[e^{i\alpha\log(p/q)}],\ \phi'_{\mathcal{M}}(\alpha) := \mathbb{E}_Q[e^{i\alpha\log(q/p)}]$$

Theorem 19 (Levy). *Let ϕ be the ch.f. of the distribution function F and $a < b$, then*

$$F(b) - F(a) = \lim_{T\to\infty}\frac{1}{2\pi}\int_{-T}^{T}\frac{\mathrm{e}^{-ita} - \mathrm{e}^{-itb}}{it}\cdot\phi(t)\,\mathrm{d}t.$$

The characteristic functions of the dominating pairs compose naturally just like RDP.

$$\phi_{\mathcal{M}}(\alpha) := \mathbb{E}_P[e^{i\alpha \log(p/q)}],\ \phi'_{\mathcal{M}}(\alpha) := \mathbb{E}_Q[e^{i\alpha \log(q/p)}]$$

- Take complex log

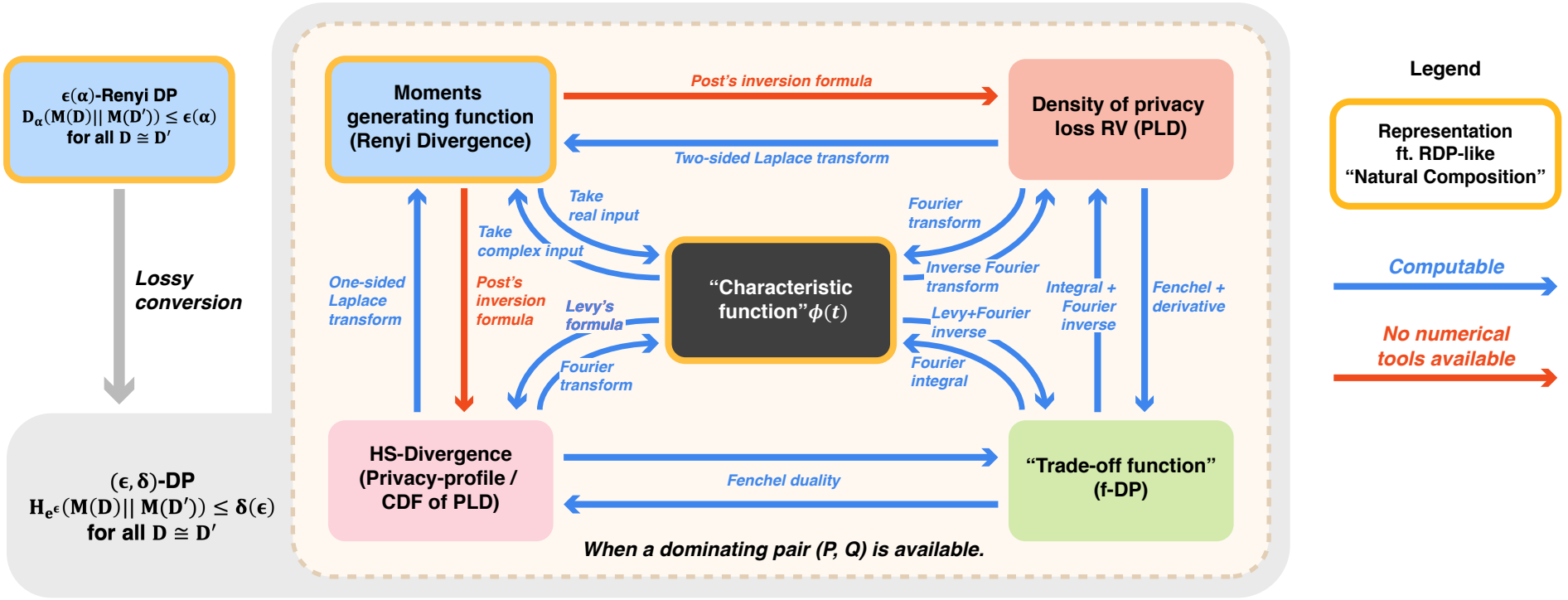
- Add up the magnitude, add up he phase.

Examples of ϕ-function for common mechanisms

Mechanism	Dominating Pair	ϕ function
Randomized Response	$P : \Pr_P[0] = p; Q : \Pr_Q[1] = p$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = pe^{\alpha i \log(\frac{p}{1-p})} + (1-p)e^{\alpha i \log(\frac{1-p}{p})}$
Laplace Mechanism	$P : p(x) = \frac{1}{2\lambda}e^{-\lvert x\rvert/\lambda}; Q : q(x) = \frac{1}{2\lambda}e^{-\lvert x-1\rvert/\lambda}$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = \frac{1}{2}\left(e^{\frac{\alpha i}{\lambda}} + e^{\frac{-\alpha i-1}{\lambda}} + \frac{1}{2\alpha i+1}(e^{\frac{\alpha i}{\lambda}} - e^{\frac{-\alpha i-1}{\lambda}})\right)$
Gaussian Mechanism	$P : \mathcal{N}(1,\sigma^2); Q : \mathcal{N}(0,\sigma^2)$	$\phi_{\mathcal{M}}(\alpha) = \phi'_{\mathcal{M}}(\alpha) = e^{\frac{-1}{2\sigma^2}(\alpha^2 - i\alpha)}$

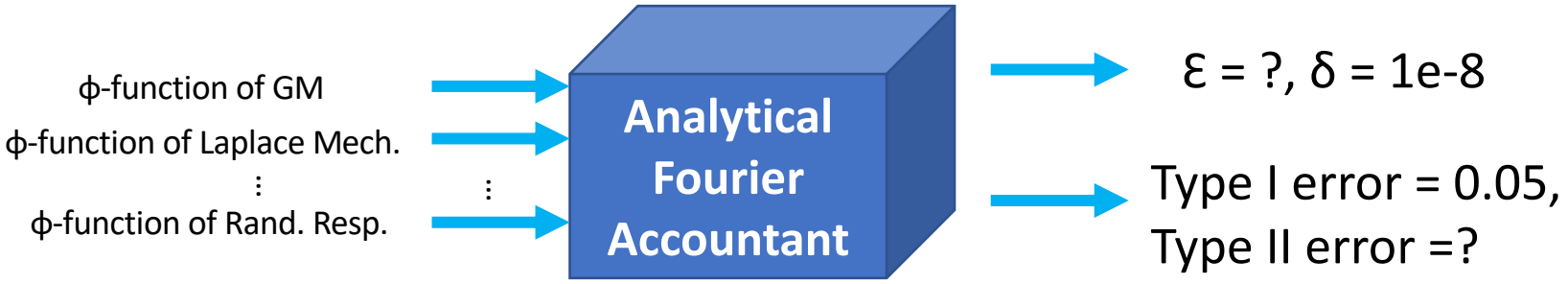
- Others that we know:
  - PureDP mechanisms are dominated by randomized response
  - ApproxDP mechanisms are dominated by leaky randomized response.
  - Exponential mechanism is dominated by two logistic distributions.
  - and so on …
- Research: expanding the list

Connection between ϕ-function and other representations



Zhu, Dong and W. (2020) https://arxiv.org/abs/2106.08567

Analytical Fourier accountant



- Composition: simply add up the log of phi functions
- Conversion to approx. DP via Levy's formula
- Conversion to tradeoff function via duality.

Zhu, Dong and W. (2020) https://arxiv.org/abs/2106.08567

Checkpoint: Mechanism specific analysis and privacy accounting

	Functional view	Pros	Cons
Renyi DP [Mironov, 2017]	$D_\alpha(P\|\|Q) \leq \epsilon(\alpha), \forall \alpha \geq 1$	Natural composition	lossy conversion to (ϵ, δ)-DP.
Privacy profile [Balle and Wang, 2018]	$\mathbb{E}_q[(\frac{p}{q} - e^\epsilon)_+] \leq \delta(e^\epsilon), \forall \epsilon \geq 0$	Interpretable.	messy composition.
f-DP[Dong et al., 2021]	Trade-off function f	Interpretable, CLT	messy composition.
PLD [Sommer et al., 2019, Koskela et al., 2020]	Probability density of $\log(p/q)$	Natural composition via FFT	Limited applicability.

Table 1: Modern functional views of DP guarantees and their pros and cons.

- Renyi DP is qualitatively different from approximate DP. Composition is quite natural with RDP.

- The composition of privacy-profiles and tradeoff functions are equivalent and somewhat messy.
  - The key to get it to work is to find a dominating pair
  - Using ϕ-function representation, we get the natural composition of RDP, and the tightness of privacy-profile / tradeoff functions.

Remainder of the lecture

- Autodp demo
- Introduction to differentially private machine learning

The main driving force behind tighter privacy accounting is the **mechanism-specific analysis**.

	Functional view	Pros	Cons
Renyi DP [Mironov, 2017]	$D_\alpha(P\|\|Q) \leq \epsilon(\alpha), \forall \alpha \geq 1$	Natural composition	lossy conversion to (ϵ, δ)-DP.
Privacy profile [Balle and Wang, 2018]	$\mathbb{E}_q[(\frac{p}{q} - e^\epsilon)_+] \leq \delta(e^\epsilon), \forall \epsilon \geq 0$	Interpretable.	messy composition.
f-DP[Dong et al., 2021]	Trade-off function f	Interpretable, CLT	messy composition.
PLD [Sommer et al., 2019, Koskela et al., 2020]	Probability density of $\log(p/q)$	Natural composition via FFT	Limited applicability.

Table 1: Modern functional views of DP guarantees and their pros and cons.

*** Somewhat complex for non-experts.**
*** Each has their own strengths / limitations.**
*** Limited availability for algorithmic components.**

But DP is designed to be modular! Many complex mechanisms are created using simple building blocks. There are various primitives and design tools that can be used, e.g., amplification by sampling / by shuffling / by post-processing.

autodp: automating differential privacy computation (for both laypersons and experts)

- Users describe their randomized algorithm to autodp

- autodp focuses on computing privacy losses

Open source project:
https://github.com/yuxiangw/autodp

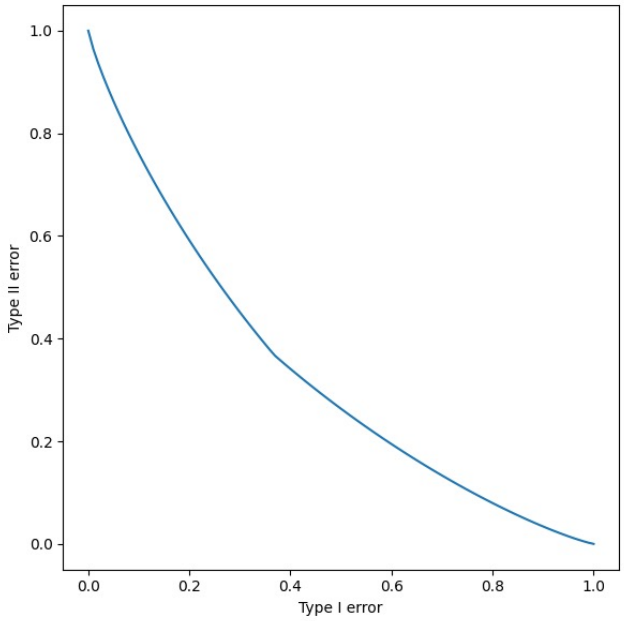
pip install autodp

Example code

```
from autodp.mechanism_zoo import ExactGaussianMechanism, PureDP_Mechanism
from autodp.transformer_zoo import Composition
import matplotlib.pyplot as plt

sigma1 = 5.0
sigma2 = 8.0

gm1 = ExactGaussianMechanism(sigma1,name='GM1')
gm2 = ExactGaussianMechanism(sigma2,name='GM2')
SVT = PureDP_Mechanism(eps=0.1,name='SVT')

# run gm1 for 3 rounds
# run gm2 for 5 times
# run SVT for once

# compose them with the transformation: compose.
compose = Composition()
composed_mech = compose([gm1, gm2, SVT], [3, 5, 1])
```

```
# Query for eps given delta
delta1 = 1e-6
eps1 = composed_mech.get_approxDP(delta1)

delta2 = 1e-4
eps2 = composed_mech.get_approxDP(delta2)


# Get name of the composed object, a structured description of the mechanism generated automatically
print('Mechanism name is \"', composed_mech.name,'\"')
print('Parameters are: ',composed_mech.params)
print('epsilon(delta) = ', eps1, ', at delta = ', delta1)
print('epsilon(delta) = ', eps2, ', at delta = ', delta2)


# Get hypothesis testing interpretation so we can directly plot it
fpr_list, fnr_list = composed_mech.plot_fDP()

plt.figure(figsize = (6,6))
plt.plot(fpr_list,fnr_list)
plt.xlabel('Type I error')
plt.ylabel('Type II error')
plt.show()
```

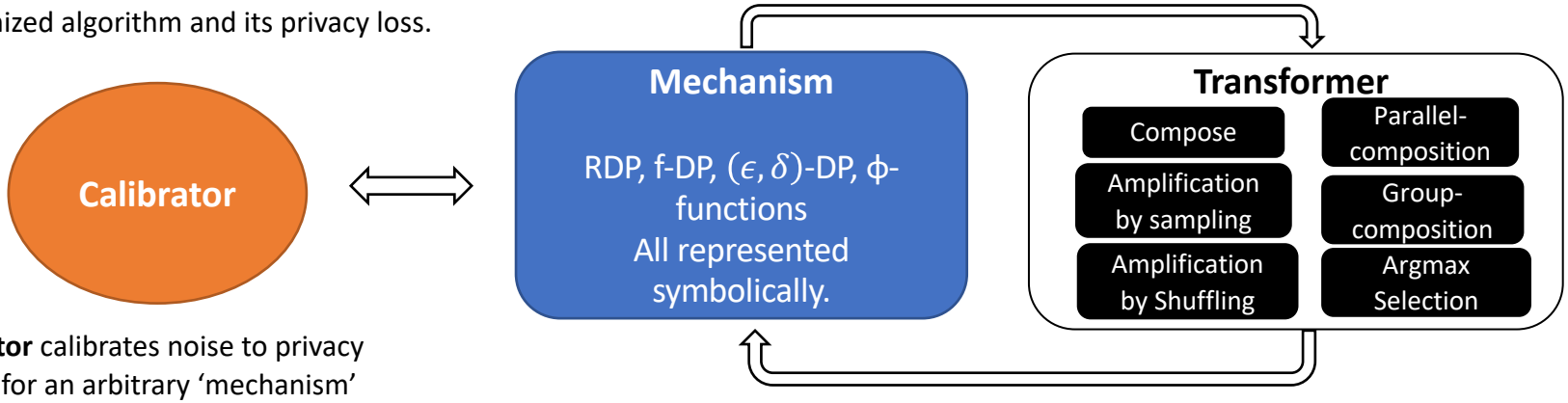
```
stdout:

Mechanism name is " Compose:{GM1: 3, GM2: 5, SVT: 1} "
Parameters are:  {'GM1:sigma': 5.0, 'GM2:sigma': 8.0, 'SVT:eps': 0.1}
epsilon(delta) =  2.18001192542518 , at delta =  1e-06
epsilon(delta) =  1.689983703842748 , at delta =  0.0001
```



Main classes of autodp

Mechanism is the base class that describes a randomized algorithm and its privacy loss.

Transformers manipulate functions (e.g., RDP) to create new **Mechanism**s.



Calibrator calibrates noise to privacy budget for an arbitrary 'mechanism'

Where did the '**accountant**' go?

The 'accountant' represents a composed mechanism that can be updated by adding new mechanisms. This can be represented by keep updating a 'Mechanism' with 'Compose'.

Mechanism class

- Keeps track of the functions that describe the privacy property of a mechanism
  - RDP function
  - Approx-DP privacy profile
  - f-DP
  - approx-RDP
- Comes with “name” and a dictionary of “parameters”
- Specific mechanisms inherits the base Mechanism class.
  - e.g. pureDP_mechanism, gaussian_mechanism …
  - When dedicated computation is available, they can be implemented easily.
- One can declare a Mechanism with any kind of descriptions
  - e.g., it propagates from RDP to others.

Transformer class

- ‘callable’ objects
  - Input: Mechanism(s), other parameters
  - Output: another Mechanism

- For example:
  - **Composition** inherits the Transformer class
  - Takes a list of mechanisms, and coefficients
  - Output the **composed mechanism**

- Subsample is a Transformer class:
  - Takes a mechanism, subsample rate
  - Output: sampled_mechanism

- The transcript of the computation is logged and parameters concatenated.
  - e.g., when using subsampling before applying Laplace mechanism, the computation is logged as (Laplace \circ Subsample) and the parameters are now {‘b’:2.0, ‘prob’:0.01}.

Calibrator class

- A calibrator aims at finding parameter configurations of a mechanism such that a pre-defined privacy budget is obtained.

- It takes a mechanism class (which contains a “constructor” that takes dictionary of parameters) and privacy budgets: eps, delta

- Optionally, a calibrator could take a utility function to maximize.

Key module of subroutines: Converter

- e.g. converter.rdp_to_approxdp
  - Take an RDP function, output an epsilon(delta)
- e.g., converter.rdp_to_fdp
  - Take an RDP function, output f(FPR)
- e.g. converter.puredp_to_RDP
- These are used extensively in Mechanism and Transformer

Template modules

- RDP_bank, fDP_bank, DP_bank
  - Templates of specialized implementations of various calculations of different mechanisms
- mechanism_zoo, transformer_zoo, calibrator_zoo
  - Implementation of commonly-used mechanisms (inherits the base Mechanism class)
  - Draw specialized implementations from RDP, fDP, DP banks
- Contributions will be very easily accepted into these banks and zoos.

Demo: Creating a mechanism from scratch and calibrating the noise

- Example: Private Multiplicative Weight with SVT + Gaussian mechanism.

(From Lecture 4)

Online query release with differential privacy

1. True data $p = x/n$, initial synthetic data $\tilde{p}_1 = 1/|\mathcal{X}|$
2. Adversary selects an online sequence of queries
   - If $|q^T \tilde{p}_t - q^T p| \geq \alpha$ ← Use AboveThresh for this
     1. Output $q^T p$ ← Use Laplace mechanism
     2. Set the loss vector to be $\ell_t := \text{sign}(q^T \tilde{p}_t - q^T p) \cdot q$
     3. Update $\tilde{p}_{t+1} = \text{Normalize}\big(\tilde{p}_t \cdot \exp(-\eta \ell_t)\big)$
     4. Increment t, i.e., t = t + 1
   - Else: output $q^T \tilde{p}_t$

- See Jupyter notebook.

Next lecture

- Differentially private machine learning