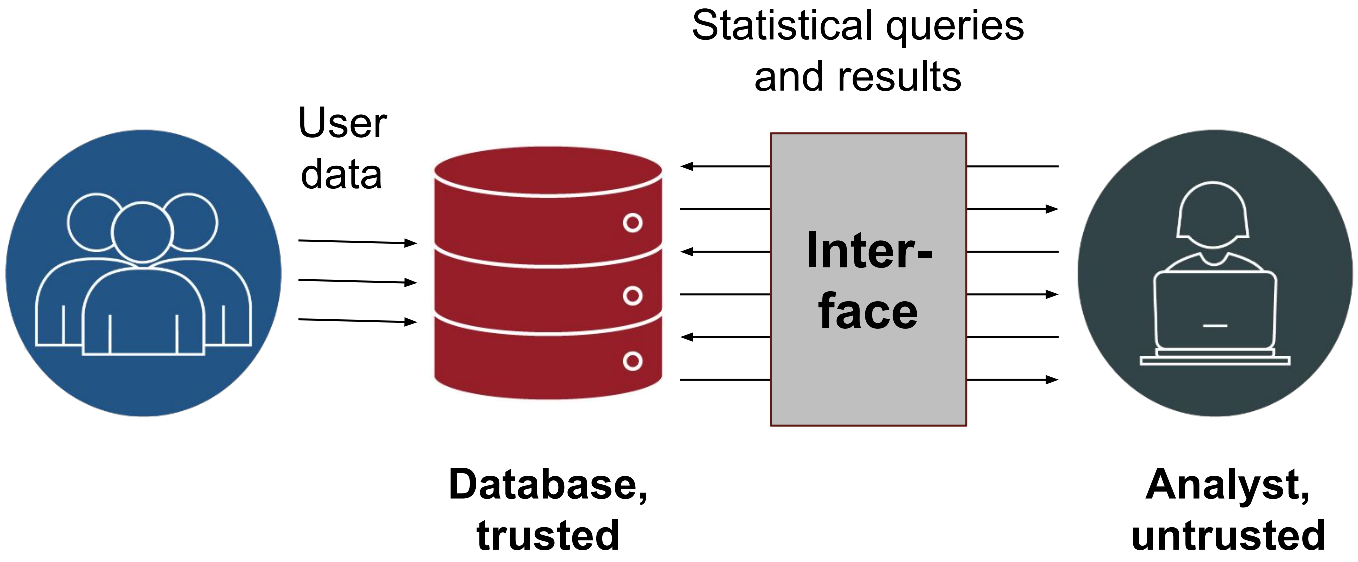
Privacy-Preserving Systems
(a.k.a., Private Systems)

CU Graduate Seminar

Instructor: Roxana Geambasu

Differential Privacy

Problem: Privacy-preserving statistical analyses



***Goal:** develop an interface that lets the analyst compute statistical queries without increasing the **privacy exposure** of individuals in the database to the analyst*

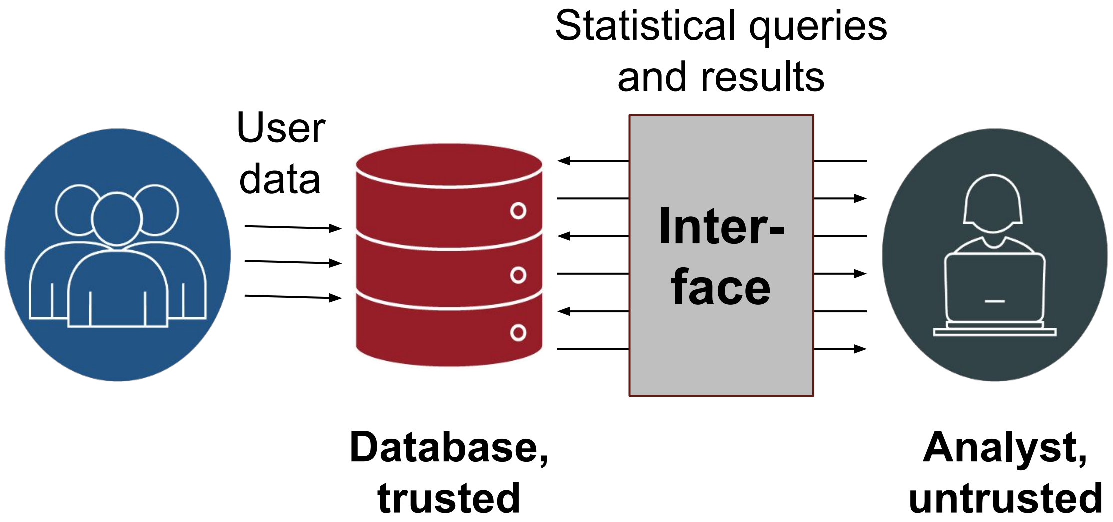
Recap: Take-aways from last time

1. Even without "PII," it's possible to learn sensitive info about individuals from data releases, especially with **side information** or **multiple queries** (a.k.a. **attacker context**).
2. It's difficult to determine what's "okay to release," because vulnerability to attack depends on **data distribution** and **attacker context.**
3. Ad-hoc solutions (incl. anonymization, k-anonymity, aggregates-only) are unreliable, because they too depend on **data distribution** and **attacker context.**
4. Today: **differential privacy**, a rigorous privacy technology to establish "what's okay to release" that does NOT depend on these!

Defining Privacy in Statistical Analyses

Requirements

- We need the interface to enforce a **rigorous definition of privacy**
- Requirements:
  1. **Resilient to side information**
  2. **Persist under repeated queries** (aka, closed under composition)
  3. **Independent of data distribution**



Strawman Definition 1

[Dalenius-77] *Access to the results of the analysis should not enable one to learn anything about any individual that one would not learn without access to the results.*

- Can be formalized, meets our requirements, and maps well onto ***semantic security*** (crypto's standard for message secrecy)
- Problem: not achievable for the statistical analysis [Dwork+10] because learning about populations implies learning about individuals

Strawman Definition 2

Definition: *Access to the results of the analysis should not enable one to learn anything about any individual in the dataset that one would not learn* ***if the individual was not in the dataset****.*

Problem: still not achievable, because the result of the analysis cannot be independent of *all* individuals in the dataset

Strawman Definition 2 (cont.)

Definition: *Access to the results of the analysis should not enable one to learn* ***anything*** *about* ***any individual*** *in the dataset that one would not learn if the individual was not in the dataset.*

Problem: still not achievable, because the result of the analysis cannot be independent of *all* individuals in the dataset

Maybe weaken one of the bolded terms above?

Strawman Definition 3

Definition: *Access to the results of the analysis should not enable one to learn* ***anything*** *about* **some (most?) individuals** *in the dataset that one would not learn if the individual was not in the dataset.*

- Can be achieved (e.g., by sampling a few individuals from the dataset and performing the computation only on data of sampled individuals)
- For the lucky individuals we didn’t sample, this definition meets at least the second requirement, but not for the unlucky others

Strawman Definition 4

Definition: *Access to the results of the analysis should not enable one to learn* **anything new confidently** *about* ***any individual*** *in the dataset that one would not learn if the individual was not in the dataset.*

Strawman Definition 4 (cont.)

Definition: *Access to the results of the analysis should not enable one to learn* **anything new confidently** *about* ***any individual*** *in the dataset that one would not learn if the individual was not in the dataset.*

- But what do **“anything new”** and **“confidently”** mean?

Strawman Definition 4 (cont.)

Definition: *Access to the results of the analysis should not enable one to learn* **anything new confidently** *about* ***any individual*** *in the dataset that one would not learn if the individual was not in the dataset.*

- But what do **“anything new”** and **“confidently”** mean?
- **Differential privacy formalizes these.**

References Cited

[Delanius-77] Dalenius. *Towards a methodology for statistical disclosure control.* Statistik Tidskrift 15, pp. 429–444, 1977.

Questions

- How would you define ***"anything new"*** and ***"confidently"*** to construct a privacy definition suitable for statistical analyses?

Proposed definition:

*Access to the results of the analysis should not enable one to learn **anything new confidently** about any individual in the dataset that one would not learn if the individual was not in the dataset.*

Defining Privacy in Statistical Analyses

The End

Differential Privacy

Differential Privacy (DP)

Lets us reason about **how much** statistical information, and **how accurate**, is **“safe to share”** in the face of an adversary with arbitrary side information and infinite computational power

Definition: A randomized query $f : X \to Y$ is $\varepsilon-$ DP if for any pair of databases $x, x' \in X$ differing in one entry and for any output set $S \subset Y$:

$$\Pr(f(x) \in S) \le e^{\varepsilon} \Pr(f(x') \in S).$$

The probabilities are taken over the randomness in f.

(Dwork, 2006)

Bayesian Interpretation

- Consider a prior distribution (X, X') on neighboring databases, modeling an adversary's *prior belief* on a real database, X, and a database X' that would have been obtained if a particular individual had not participated
- Given an output *y* from a ε-DP randomized function *f*, the adversary will have a *posterior belief* on the database: $X|_{f(X)=y}$.
- The definition of DP implies that this posterior distribution is close (in statistical distance, SD) to the posterior that would have been obtained if the function *f* had been run on *X'* instead of *X* (Vadhan, 2016)

$$SD(X|_{f(X)=y}, X|_{f(X')=y}) \leq 2\varepsilon.$$

Bayesian Interpretation (cont.)

- In particular, if the adversary's prior included all the information about X except for the *i*'th row (the data of individual *i*), then the adversary's posterior on row *i* would have been close to their prior on row *i*
- In that sense, the adversary does not learn **"<u>anything new</u>"** about any individual (i.e., that they couldn't have learned from the rest of the database)

Strawman Definition 4

Access to the results of the analysis should not enable one to learn **<u>anything new</u> confidently** *about any individual in the dataset that one would not learn if the individual was not in the dataset.*

Hypothesis Testing Interpretation

- Consider an adversary who wants to test, based on the output of a DP query, the null hypothesis that an individual, *i*, has contributed their data to a database *x*.
- The definition of DP has been proven to be equivalent to requiring that *any hypothesis test* has either *low significance* (it has high false-positive rate, FPR), or *low power* (it has high false-negative rate, FNR) (Wasserman, 2010)

$$FPR(f,x,x',S) + e^{\varepsilon} FNR(f,x,x',S) \geq 1 \quad \text{and}$$

$$e^{\varepsilon} FPR(f,x,x',S) + FNR(f,x,x',S) \geq 1 \quad (\forall x,x'.x \sim x', \forall S \subset Y)$$

Statistical Testing Interpretation

- The parameter **epsilon** controls the trade-off between significance vs. power of any hypothesis test.
- In that sense, the adversary cannot test **"<u>confidently</u>"** whether a particular individual was in the dataset that was used to produce a particular output.

Strawman Definition 4

Access to the results of the analysis should not enable one to learn **anything new <u>confidently</u>** *about any individual in the dataset that one would not learn if the individual was not in the dataset.*

Lay Interpretation and Cautions

- Whatever an adversary learns about you, they did not learn it **because** of the use of your data in the DP query; they could have learned the same thing even if you hadn’t contributed your data
- This does not mean that the adversary cannot learn **anything** about you from the output of a DP query!
- Example
  - Adversary learns that smoking correlates with lung cancer.
  - Adversary knows that X smokes.
  - He can deduce that X is more likely to get cancer than a nonsmoker.
  - ***But*** this deduction was not **caused** by X’s participation in the study.

Example Use of DP

Company X's
Terms and Conditions

[…]

You agree to allow Company X to use your data to compute and share results from statistical analyses under 0.1-user-level differential privacy.

✓ YES ✗ NO

Clicking ✓ YES vs. ✗ NO does not increase the privacy risk for any individual by more than 11%.

- If the risk of anyone learning a particular aspect about you was low, it remains low despite Company X's use of your data.

Cited References

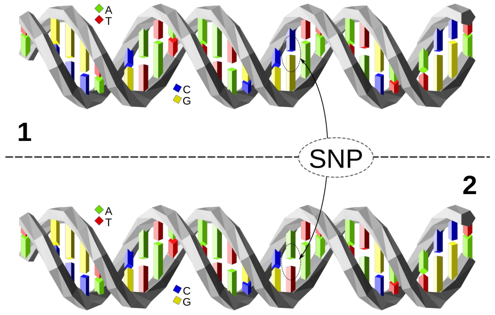
(Dwork, 2006) Dwork, McSherry, Nissim, & Smith. (2006). *Calibrating noise to sensitivity in private data analysis.* TCC.

(Vadhan, 2016) Vadhan. *The complexity of differential privacy.* https://privacytools.seas.harvard.edu/files/privacytools/files/complexityprivacy_1.pdf.

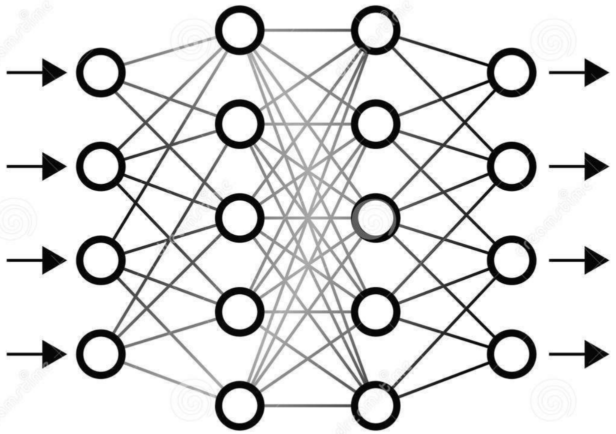
(Wasserman, 2010) Wasserman & Zhou. (2010). A statistical framework for differential privacy. *Journal of the American Statistical Association*.

Questions

- Considering DP’s interpretations, reason about how DP prevents the privacy attacks against aggregates that we discussed in previously:
  - Membership inference attacks
  - Data reconstruction attacks
  - ML memorization-based attacks



United States® Census Bureau



Differential Privacy

The End

DP Parameters and Properties

The Epsilon Parameter

- All interpretations we gave assume **“low” epsilon**
- Intuitively, epsilon bounds the privacy loss of any individual through the output of the statistical analysis
- Smaller values mean better privacy but hurt accuracy
  - For example, for many statistical analyses, epsilon < 1/*n* doesn’t make sense because DP error accumulates as *o*(1/*n*)
- Rule of thumb: epsilon = 0.1 for simple statistics, though for ML, epsilon = 1 or 10 are needed

DP Variant: (ε,δ)-DP

Definition: A randomized query $f: X \to Y$ is (ε,δ)- DP if for any pair of databases $x, x' \in X$ differing in one entry and for any output set $S \subset Y$:

$$\Pr(f(x) \in S) \leq e^{\varepsilon} \Pr(f(x') \in S) + \delta.$$

The probabilities are taken over the randomness in f.

- Adds an additive **delta** term to the original definition
- Enables randomization mechanisms that improve the privacy–accuracy trade-off
- Roughly interpreted as *"ε − DP with probability at least (1 − δ)"*

The Delta Parameter

- Aim for values of delta that are **less than the inverse of a super-linear polynomial in database size (*n*).**
- Delta = 1/*n* is dangerous: permits preserving “privacy” by publishing the complete records of a small number of database participants!
- Rule of thumb: delta = $1/n^2$ is generally acceptable.

Properties

- Property 1: closure under post-processing
- Property 2: closure under composition
- Property 3: independent of data distribution

Property 1: Closure Under Post-Processing

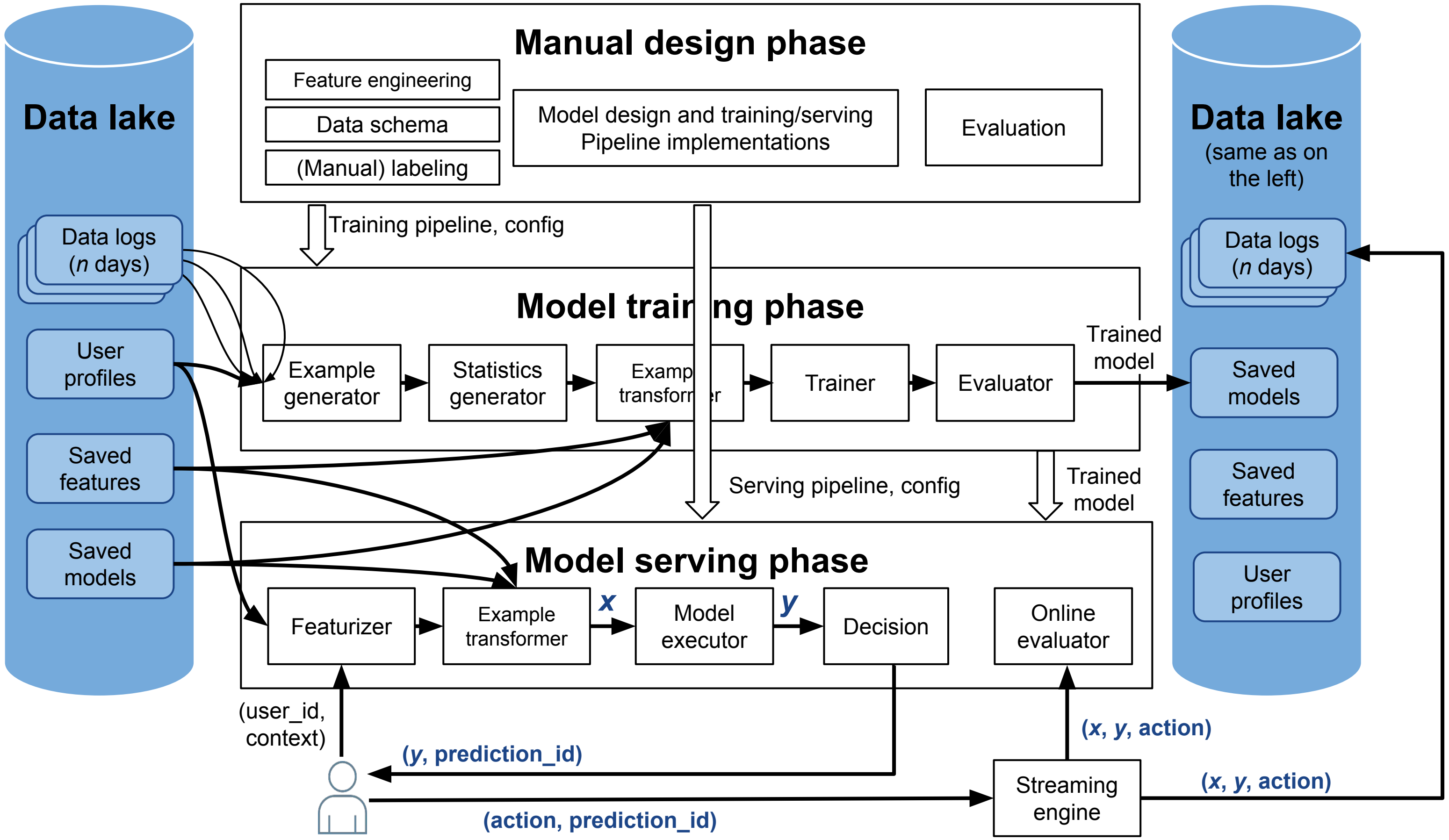
If f is (ε,δ)-DP then for any g we have: $g \circ f$ is (ε,δ)-DP.

- Trivially implies **resilience to arbitrary side information,**
  - **This was our first requirement**
- Can allow **safe**, **unlimited reuse of the outputs** from DP computations, such as models trained with DP training procedures
- Question: How could this property be useful in addressing data exposure risks in modern ML ecosystems?

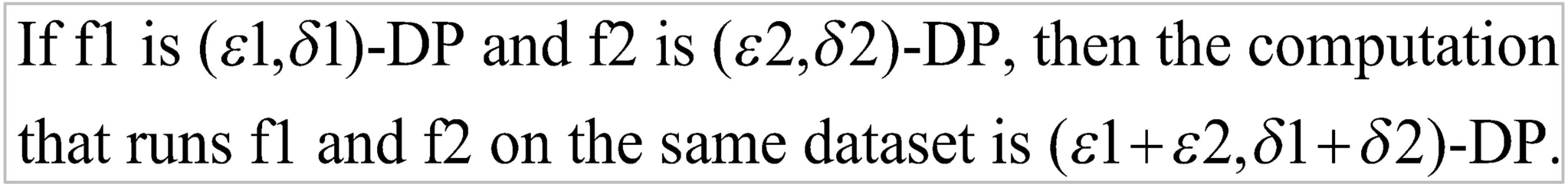
Usefulness

Thanks to post-processing, DP could be used to:

- Safely share models and features across teams
- Safely retain models and features beyond the raw data's expiration date
- What else?



Property 2: Closure under Composition

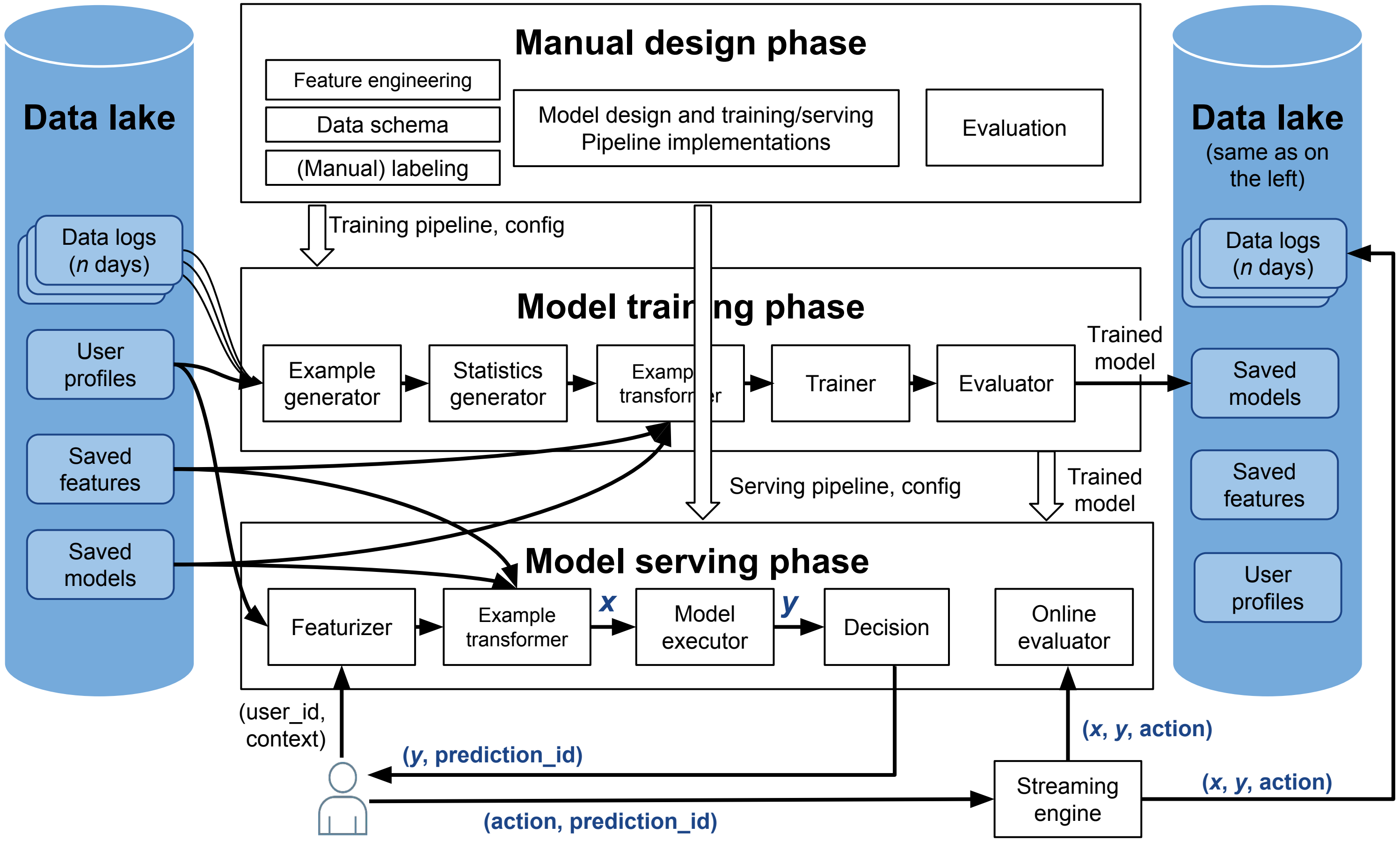


If f1 is $(\varepsilon 1,\delta 1)$-DP and f2 is $(\varepsilon 2,\delta 2)$-DP, then the computation that runs f1 and f2 on the same dataset is $(\varepsilon 1+\varepsilon 2,\delta 1+\delta 2)$-DP.

- **This was our third requirement**
- Bounds how much new information an analyst can learn about any individual in the database across multiple queries
- Encodes that the more things you learn from a dataset, the more you also learn about individuals
- Tighter composition formulas exist, in which individuals’ privacy loss degrades as square root of the number of composed DP computations instead of linearly as above

Usefulness

- ML ecosystems release many models/statistics learned from the same users' data, so it is important to **bound the cumulative privacy risk** resulting from all releases
- Composition enables **modular development** of complex systems from small parts, such as basic DP mechanisms and algorithms



Data lake

Data logs (*n* days)

User profiles

Saved features

Saved models

Manual design phase

Feature engineering

Data schema

(Manual) labeling

Model design and training/serving Pipeline implementations

Evaluation

Training pipeline, config

Model training phase

Example generator

Statistics generator

Example transformer

Trainer

Evaluator

Trained model

Serving pipeline, config

Trained model

Model serving phase

Featurizer

Example transformer

x

Model executor

y

Decision

Online evaluator

(user_id, context)

(*y*, prediction_id)

(action, prediction_id)

Streaming engine

(*x*, *y*, action)

(*x*, *y*, action)

Data lake

(same as on the left)

Data logs (*n* days)

Saved models

Saved features

User profiles

Property 3: Independent of data distribution

- DP is a property of the computation on a dataset, not of the dataset!
  - Unlike k-anonymity or other anonymization techniques

Definition: A randomized query $f : X \rightarrow Y$ is (ε,δ)- DP if for any pair of databases $x, x' \in X$ differing in one entry and for any output set $S \subset Y$:

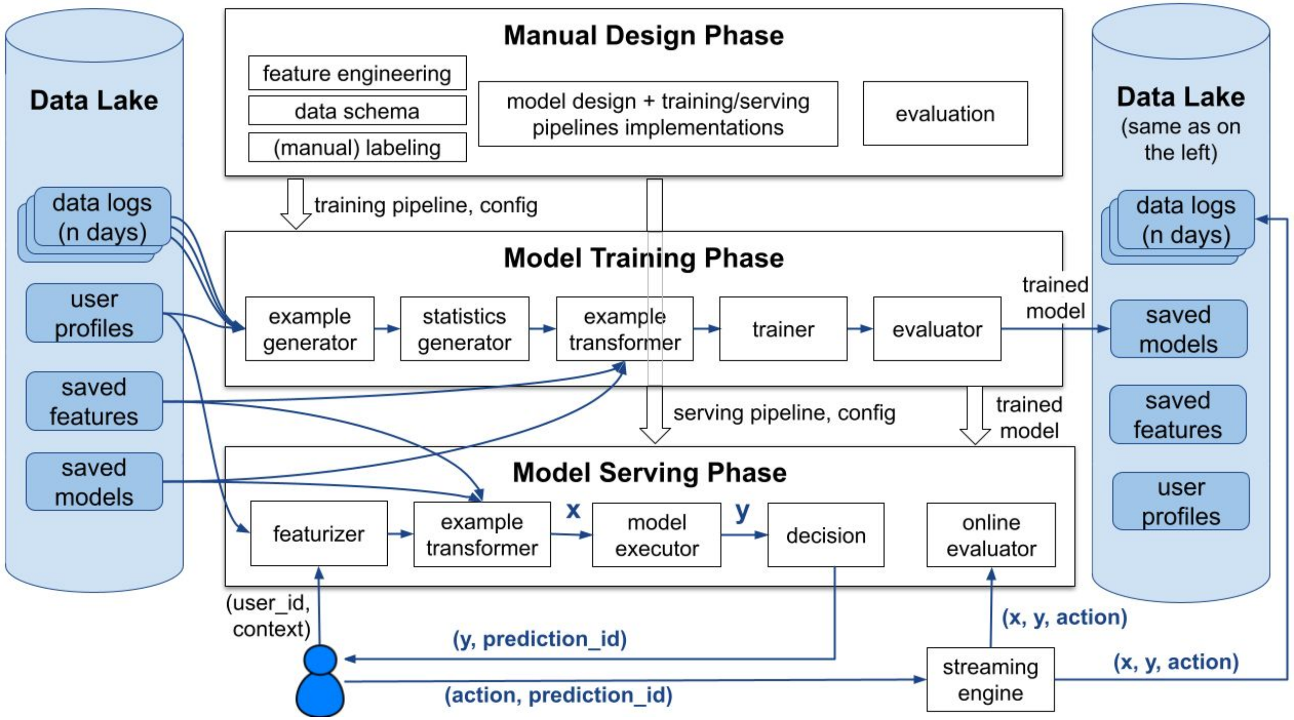
$$\Pr(f(x) \in S) \leq e^{\varepsilon} \Pr(f(x') \in S) + \delta.$$

The probabilities are taken over the randomness in f.

- **This was our third requirement**
  - But is has implications to utility, b/c it's a worst-case definition that must hold under worst-case dataset! Often noted as a disadvantage for this reason, but it's also a big advantage to not have to worry about dataset properties… It's a tradeoff, and DP, like crypto, makes worst-case assumptions…

Questions

- In what other ways can differential privacy, based on its properties, be leveraged to mitigate the data exposure risks in ML ecosystems that we previously discussed?



DP Parameters and Properties

The End

Basic DP Mechanisms

Making Statistical Analyses DP

Basic approach to making a function DP

1. Decompose it into sub-functions for which you either have a DP implementation, or that fit into a class of functions for which you can apply one of the several **basic DP mechanisms.**
2. Use the **composition** and **post-processing** closure properties to determine the guarantee achieved by the overall function that combines the sub-functions.

Basic DP Mechanisms

- Multiple DP mechanisms exist, each supporting different classes of functions.
  1. Laplace mechanism
  2. Gaussian mechanism
  3. Exponential mechanism
  4. Smooth sensitivity mechanism
  5. Test and release mechanism
- We’ll focus on the first two.

Laplace and Gaussian Mechanisms

- Suitable for some numeric functions: $f^{np}: X^n \rightarrow \mathrm{R}^k$.*
  - Counting, averaging, computing histograms, contingency tables, etc.
  - Also doing one step of gradient descent, which is an average over some real values (more on that later)
- Approach: perturb each output dimension with independent draws from a calibrated Laplace/Gaussian distribution

*I denote the non-private function as f^{np} to distinguish it from the DP version, which I've thus far been denoting as *f*.

Laplace Mechanism

(Dwork, 2006)

L1-Sensitivity

- Define **L1-sensitivity** of a function $f : X^n \rightarrow \mathrm{R}^k$ as:

$$\Delta_f = \max_{x,x':d(x,x')\leq 1} \| f(x) - f(x') \|_1 \qquad \left(\| f(x) - f(x') \|_1 = \sum_{i=1}^{k} | f_i(x) - f_i(x') | \right)$$

- That is, how much can one entry affect the value of the function?

L1-Sensitivity

- Define **L1-sensitivity** of a function $f : X^n \to \mathrm{R}^k$ as:

$$\Delta_f = \max_{x,x':d(x,x')\le 1} \| f(x) - f(x') \|_1$$

$$\left(\| f(x) - f(x') \|_1 = \sum_{i=1}^{k} | f_i(x) - f_i(x') | \right)$$

- That is, how much can one entry affect the value of the function?
  - "How many people in a room have brown eyes?": Sensitivity = ?
  - "How many have brown eyes, how many have blue eyes, how many have green eyes, and how many have red eyes?": Sensitivity = ?
  - "How many have brown eyes and how many are taller than six feet?": Sensitivity = ?

L1-Sensitivity

- Define **L1-sensitivity** of a function $f : X^n \rightarrow \mathrm{R}^k$ as:

$$\Delta_f = \max_{x,x':d(x,x')\leq 1} \| f(x) - f(x') \|_1$$

$$\left(\| f(x) - f(x') \|_1 = \sum_{i=1}^{k} | f_i(x) - f_i(x') | \right)$$

- That is, how much can one entry affect the value of the function?
  - “How many people in a room have brown eyes?”: Sensitivity = **1**
  - “How many have brown eyes, how many have blue eyes, how many have green eyes, and how many have red eyes?”: Sensitivity = **1**
  - “How many have brown eyes and how many are taller than six feet?”: Sensitivity = **2**

Laplace Distribution

- **The Laplace distribution,** *Lap(b)*, is the probability distribution with p.d.f.

$$\Pr[x] = \frac{1}{2b} e^{-\frac{|x|}{b}}$$

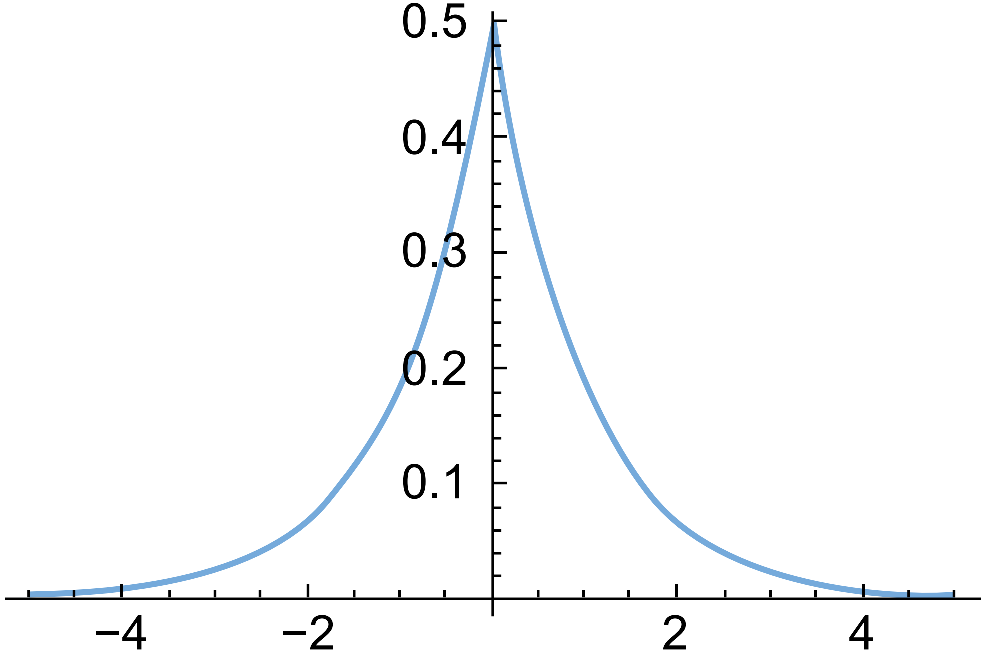
- That is, a symmetric, double-exponential distribution

$$z \sim Lap(b):$$

$$E[|z|] = b$$

$$\Pr[|z| \geq tb] = e^{-t}$$

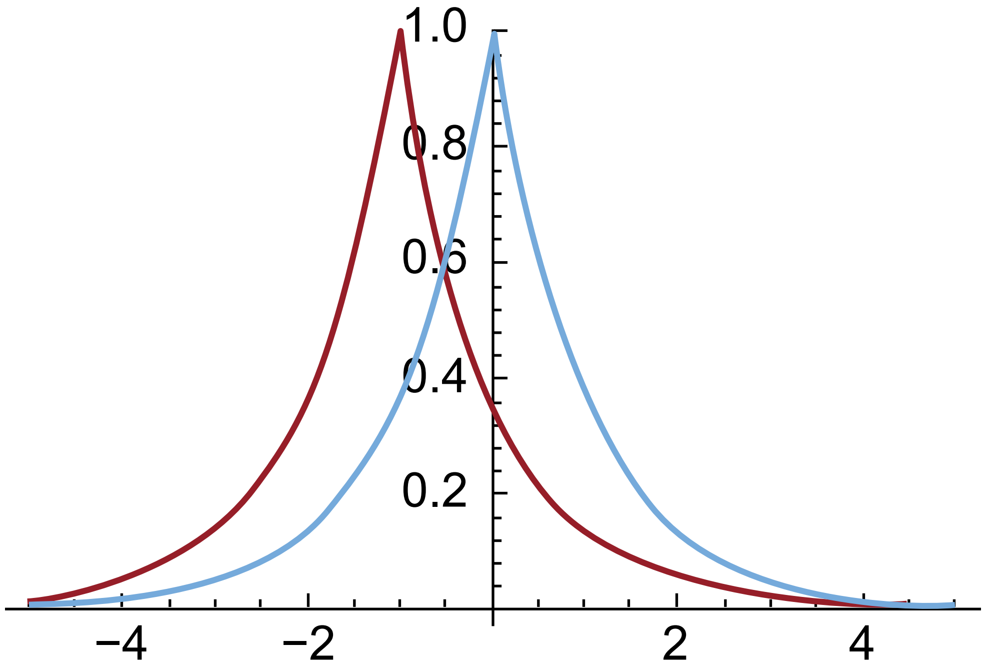
$$stdev(Lap(b)) = b\sqrt{2}$$



Laplace Mechanism for DP

Laplace($x, f^{np}: X^n \rightarrow R^k, \varepsilon$)

1. Let $\Delta_{f^{np}}$ be the ℓ_1, sensitivity of f^{np}
2. For $i = 1$ to k: Let $z_i \sim \mathrm{Lap}(\frac{\Delta_{f^{np}}}{\varepsilon})$
3. Output $f^{np}(x) + (z_1, \ldots, z_k)$



Theorem: Laplace(.) satisfies (ϵ, 0) − DP.

Proof: See Aaron Roth's lecture notes.

Privacy-Accuracy Trade-Off

- The noise distribution depends on 1/epsilon and L1-sensitivity of the computation, **not** on the database or its size!
- This means that a DP computation based on the Laplace mechanism will be more accurate when sensitivity is small and epsilon is large.

Privacy-Accuracy Trade-Off

- The noise distribution depends on 1/epsilon and L1-sensitivity of the computation, **not** on the database or its size!
- This means that a DP computation based on the Laplace mechanism will be more accurate when sensitivity is small and epsilon is large.
- Example: “How many people in a room have brown eyes?” Sensitivity = **1**

$$z \sim Lap(\frac{1}{\varepsilon}):$$

$$E[|z|] = \frac{1}{\varepsilon}$$

$$\Pr[|z| \geq t\frac{1}{\varepsilon}] = e^{-t}$$

$$\xrightarrow{\varepsilon = 1}$$

$$E[|z|] = 1$$

$$\Pr[|z| \geq t] = e^{-t}$$

$$\Pr[|z| \geq 7] < 0.1\%$$

- For a small value of the function (small group of people), +/−7 matters.
- But for a large count (e.g., 1,000), +/−7 doesn’t matter!
- So, the larger the database, the less the noise will impact accuracy.

Gaussian Mechanism

(Dwork, 2006)

Gaussian Mechanism for DP

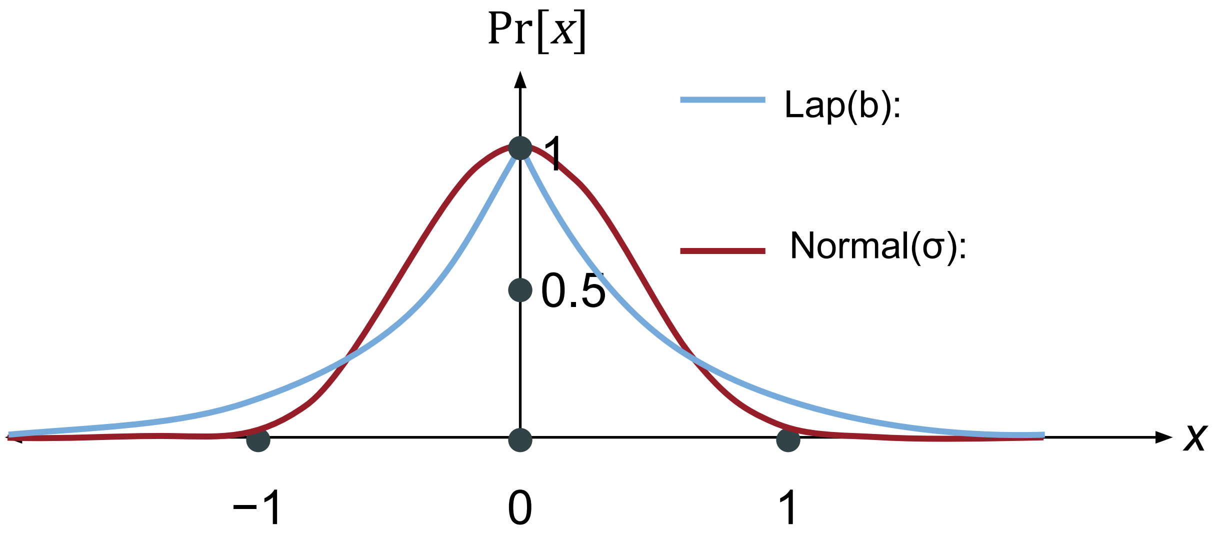
- Enforces **(ϵ, δ) − DP** by perturbing the output of a real-valued function with noise drawn from a normal (a.k.a. Gaussian) distribution with mean 0 and standard deviation dependent on the L2-sensitivity (denoted here as Δ_2) of the function:

$$\sigma = \frac{\Delta_2}{\varepsilon}\sqrt{2\ln\left(\frac{1.25}{\delta}\right)}$$

- L2-sensitivity is defined similarly to L1, but we take the maximum 2-norm (Euclidian distance) between *f*(*x*) and *f*(*x'*).

Laplace vs. Gaussian Mechanism

- The Laplace distribution (blue) has a better variance, but the tail of the Gaussian distribution (red) decreases faster.
- This is why Laplace gives pure DP but Gaussian needs the delta greater than 0 to account for its sharp cut of the tails.
- Thus, Laplace gives better semantic but you can get better utility from Gaussian.



When Laplace/Gaussian Don’t Work

- What if we have a nonnumeric function?
  - “What’s the most common eye color in the room?”
- What if the perturbed answer isn’t “almost as good as” the exact answer?
  - “Which price would bring the most money from a set of buyers?”

Exponential mechanism
(McSherry, 2007)

- What if L1/L2-sensitivity is large?
  - “What’s the median salary in a salary database?”

Smooth sensitivity
(Nissim, 2007)
(and other mechanisms)

Cited References (No Particular Order)

(Dwork, 2006) Dwork, McSherry, Nissim, & Smith. *Calibrating noise to sensitivity in private date analysis.* TCC, 2006.

(McSherry, 2007) McSherry & Talwar. *Mechanism design via differential privacy.* FOCS, 2007.

(Nissim, 2007) Nissim, Raskhodnikova, & Smith. *Smooth sensitivity and sampling in private data analysis.* STOC, 2007.

Questions

Determine L1-sensitivities of the following statistics:

- Number of families at zipcode 10027 and number of families in NYC (zipcode 10027 is in NYC).

  a. 1

  b. 2

  c. 3

$$\Delta_f = \max_{x,x':d(x,x')\leq 1} \| f(x) - f(x') \|_1$$

Questions

Determine L1-sensitivities of the following statistics:

- Counts of families per NYC zip code.

  a. 1

  b. 2

  c. number of zipcodes in NYC

$$\Delta_f = \max_{x, x' : d(x, x') \leq 1} \| f(x) - f(x') \|_1$$

Questions

Determine L1-sensitivities of the following statistics:

- Average age of N people in a group (assume age is 0-120).

  a. 1

  b. 2

  c. 1/N

  d. 120/N

  e. N

$$\Delta_f = \max_{x,x':d(x,x')\leq 1} \| f(x) - f(x') \|_1$$

Basic DP Mechanisms

The End

Composite DP Algorithms

DP Algorithms

- There are some DP libraries that implement basic algorithms for statistical data analysis.
  - Google's statistics library: basic statistics (Wilson, 2020)
  - IBM's library: *k*-means, regression, PCA (Holohan, 2019)
  - TensorFlow privacy, Pytorch Opacus: stochastic gradient descent (SGD) (McMahan, 2018)
- We'll discuss some algorithms from Google's statistics library.
  - But best to read their code!

Bounded Sum

- Code
- Goal: compute sum over a set of values bounded in range [*L*,*U*]
  - *L*, *U* assumed to be public knowledge or obtained through the "approximate bounds" DP algorithm (also implemented in Google's lib)
- Method
  - Laplace mechanism with sensitivity max(|*L*|,|*U*|)
  - If *L* and *U* need to be computed privately too, then use part of the privacy budget for the approximate bounds algorithm, and the remainder for the Laplace mechanism (per the composition property of DP)

Bounded Mean

- Code
- Goal: compute mean over a set of values bounded in range [*L*,*U*]
- One option
  - Compute DP sum: *S* (with half the budget)
  - Compute DP count: *N* (with half the budget)
  - Return *S*/*N*, but sum has sensitivity max(|*U*|,|*L*|)
- Observe
  - For fixed *N*, the mean has L1-sensitivity 1/*N* * max(|*U*|,|*L*|)
  - So, it's tempting to compute the mean and apply Laplace for that sensitivity
  - That assumes the count is public information but often it isn't

Bounded Mean (cont.)

- Google’s implementation rewrites the average in terms relative to the middle of the interval [*L*,*U*] $\left(middle = \frac{L+U}{2}\right)$
- That enables calculating the sum of all input values with lower sensitivity than it would take if doing noisy sum/noisy count: |*U* – *L*|/2

$$average = \frac{\sum x_i}{N} = middle + \frac{\boxed{\sum (x_i - middle)}}{\boxed{N}}$$

Function 1, sensitivity: |U-L|/2

Function 2, sensitivity: 1

$$noisy\_average = middle + \frac{\sum x_i - N \cdot middle + Lap(|U-L|/\varepsilon)}{N + Lap(2/\varepsilon)}$$

Bounded Variance

- Code
- Goal: compute variance over a set of values bounded in [*L*,*U*]
- Method
  - Variance can be written as [Mean of squares − Mean squared].

$$\text{var}iance = \frac{\sum (x_i - \mu)^2}{n} = \frac{\sum x_i^2}{N} - \left(\frac{\sum x_i}{N} \right)^2$$

  - DP variance can therefore be computed by applying the preceding DP **bounded mean** algorithm twice, each time with half of the budget.

Approximate Bounds

- Code
- Goal: establish approximate [*L*,*U*] range over a set of values
- Method (described for non-negatives)
  - Organize data into **logarithmic histogram bins** (e.g., [0,1], (1,2], (2,4], (4,8],…)
  - Each bin keeps a **DP count** of the number of values in its range. Bin *i* holds the DP count of values in range $\left(scale \cdot base^{i-1}, scale \cdot base^{i}\right]$.
  - Given a confidence parameter *c*, we choose a **threshold *t*** after which to declare a bin as non-empty with probability >=*c*. The formula for *t* is:

$$t = \frac{1}{\varepsilon}\log(1 - c^{\frac{1}{base-1}}) \quad \text{(Wilson, 2020)}$$

  - ***L*** = leftmost bin with DP count greater than *t*. ***U*** = rightmost bin with DP count greater than *t*

Approximate Bounds (cont.)

- Example: base = 2, num_bins = 4, inputs = {0, 0, 0, 0, 1, 3, 7, 8, 8, 8}
    - The bins and (non-DP) counts are [0, 1]: 5 ; (1, 2]: 0 ; (2, 4]: 1 ; (4, 8]: 4
    - If success_probability = .9 and epsilon = 1, we get threshold *t* = 3.5
    - Since the count of bin [4, 8] > *t*, we would return max := 2^3 = 8
    - Since the count of bin [0, 1] > *t*, we would return min := 0
    - With DP counts, the procedure gives approximate values of course
- The parameters of this scheme—num_bins, base—are not trivial to set without some external knowledge of the rough range you want to capture

Cited References

(Holohan, 2019) Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, & Killian Levacher. (2019). *Diffprivlib: The IBM differential privacy library.* arXiv:1907.02444.

(McMahan, 2018) H. Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, & Peter Kairouz. (2018). *A general approach to adding differential privacy to iterative training procedures.* arXiv:1812.06210.

(Wilson, 2020) Royce J. Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, & Bryant Gipson. (2020). *Differentially private SQL with bounded user contribution.* PETs.

Task for Home

- Inspect the code in Google’s statistics library corresponding to:
  - Bounded sum
  - Bounded mean
  - Bounded variance
  - Approximate bounds
  - And any other function you wish.

Composite DP Algorithms

The End

DP SGD Algorithm

Stochastic Gradient Descent (SGD)

- DP version was described in (Abadi, 2016) and implemented in several libraries, including Tensorflow Privacy and Pytorch Opacus
- Without privacy, SGD is:
  - Repeated
    - Sample a batch from input dataset
    - Calculating $\nabla\theta$ with respect to the loss function
    - $\theta := \theta + \nabla\theta$
- With privacy, at each iteration you clip and noise the gradients $\nabla\theta$ using the Gaussian mechanism

Differentially Private SGD (Outline)

Input: Examples $\{x_1, \dots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, xi)$. Parameters: learning rate η_t, noise scale σ, batch size L, gradient norm bound C

Initialize θ_0 randomly

```
for $t \in [T]$ do
    Take a random sample $L_t$ with sampling probability $L/N$
    Compute gradient
    For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, xi)$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|2}{C}\right)$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
```

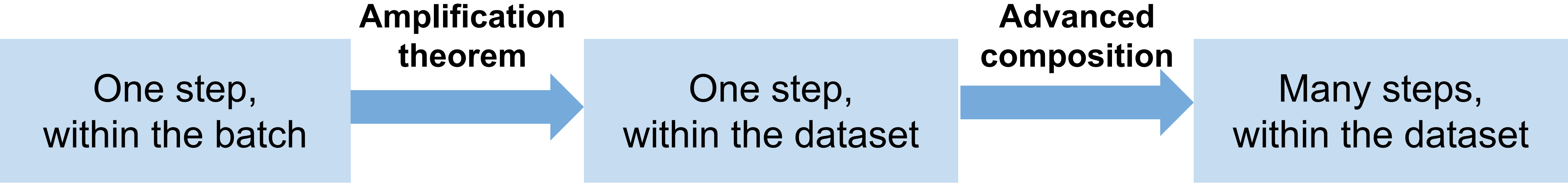
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method

Differentially Private SGD (Outline)

```
**Input:** Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, xi)$ . Parameters: learning rate $\eta_t$ , noise scale $\sigma$, batch size $L$, gradient norm bound $C$
**Initialize** $\theta_0$ randomly
**for** $t \in [T]$ **do**
  Take a random sample $L_t$ with sampling probability $L/N$
  **Compute gradient**
  For each $i \in L_t$ , compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, xi)$
  **Clip gradient**
  $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) \, / \, \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|2}{C}\right)$
  **Add noise**
  $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
  **Descent**
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
**Output** $\theta_T$ and compute the overall privacy cost $(\varepsilon, \delta)$ using a privacy accounting method
```

Privacy Analysis

- The preceding algorithm receives as arguments a noise scale factor σ, gradient norm *C*, batch size *L*, and number of iterations *T*.
- Because each step uses the Gaussian mechanism, the gradient at each step is (ε,δ) − DP with respect to the batch. ε,δ are determined from σ.
- Question: What's the DP guarantee after many steps for the gradient with regard to the dataset? Answer: Apply the **amplification theorem** and **composition**.



One step,
within the batch

Amplification theorem

One step,
within the dataset

Advanced composition

Many steps,
within the dataset

(ε,δ) − DP

(2qε, qδ) − DP

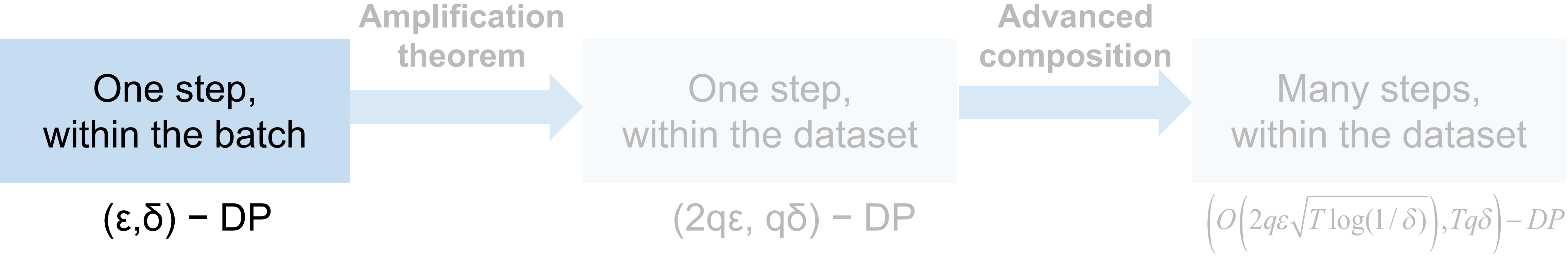
$\left(O\left(2q\varepsilon\sqrt{T\log(1/\delta)}\right), Tq\delta\right) - DP$

Determining ε,δ From σ

- To make a real-valued function DP with the Gaussian mechanism, we add noise from a normal distribution with standard deviation shown to the right.
- In the preceding algorithm, we added noise with standard deviation σ*C*, where $C = \Delta_2$.
- Fixing δ to something reasonable, we can now compute ε based on σ, as shown on the right.

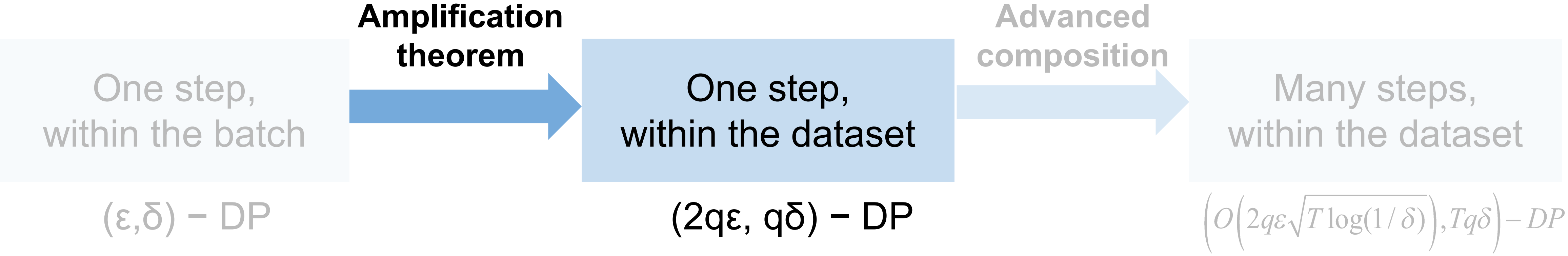
$$\frac{\Delta_2}{\varepsilon}\sqrt{2\ln\left(\frac{1.25}{\delta}\right)}$$

$$\varepsilon = \frac{1}{\sigma}\sqrt{2\ln\left(\frac{1.25}{\delta}\right)}$$


One step, within the batch
Amplification theorem
One step, within the dataset
Advanced composition
Many steps, within the dataset
(ε,δ) − DP
(2qε, qδ) − DP
$\left(O\left(2q\varepsilon\sqrt{T\log(1/\delta)}\right), Tq\delta\right) - DP$

Amplification Theorem

- *N*: data size; *L*: size of each batch
- Let *q* = *L*/*N*
- **Amplification theorem:** if gradient is (ε,δ) − DP within the batch, then it is (2qε, qδ) − DP within the dataset



One step, within the batch

(ε,δ) − DP

Amplification theorem

One step, within the dataset

(2qε, qδ) − DP

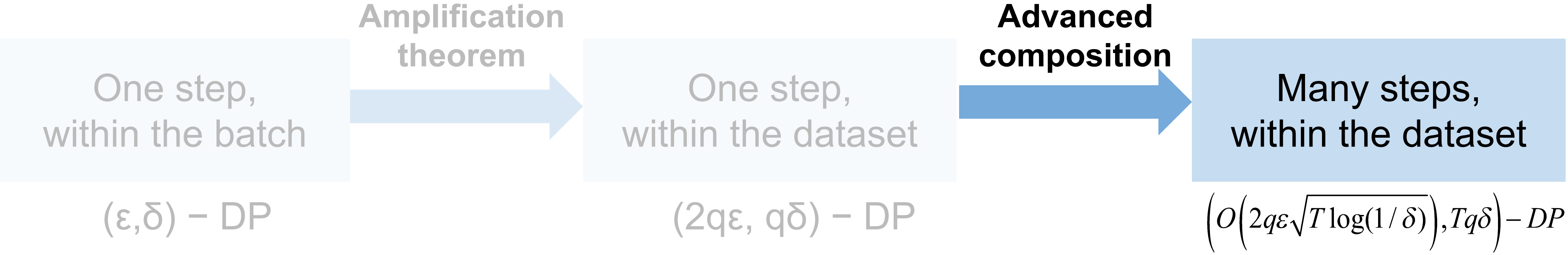
Advanced composition

Many steps, within the dataset

$\left(O\left(2q\varepsilon\sqrt{T\log(1/\delta)}\right),Tq\delta\right)-DP$

Advanced Composition

- Applying the same (ε,δ) − DP algorithm *T* times will give an $\left(O\left(\varepsilon\sqrt{T\log(1/\delta)}\right), T\delta\right)$ − DP algorithm.



One step, within the batch

(ε,δ) − DP

Amplification theorem

One step, within the dataset

(2qε, qδ) − DP

Advanced composition

Many steps, within the dataset

$\left(O\left(2q\varepsilon\sqrt{T\log(1/\delta)}\right), Tq\delta\right) - DP$

Cited References

(Abadi, 2016) Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, & Li Zhang. (2016). *Deep learning with differential privacy.* CCS.

DP SGD Algorithm

The End

Demo: DP Neural Network Training

Demo

- Demo link (accessible to Columbia Lionmail accounts)

Homework 2 Overview

(CA walks through HW2 notebook, posted on courseworks)