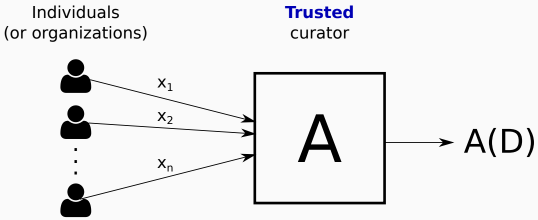
PRIVACY PRESERVING MACHINE LEARNING

LECTURE 6: BEYOND THE CENTRALIZED MODEL OF DIFFERENTIAL PRIVACY

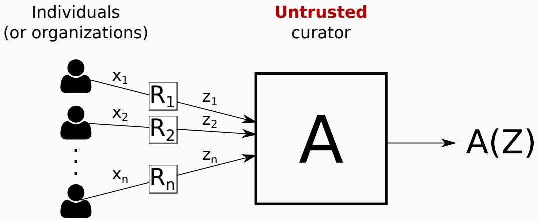
Aurélien Bellet (Inria)

Master 2 Data Science, University of Lille

Trusted curator model (also called global model or centralized model):
$\mathcal{A}$ is differentially private wrt dataset D



Untrusted curator model (also called local model or distributed model):
Each $\mathcal{R}_i$ is differentially private wrt record (or local dataset) x_i



TODAY'S LECTURE

1. Local Differential Privacy (LDP)
2. Intermediate trust models
3. Federated Learning
4. Wrapping up

Local Differential Privacy (LDP)

PRIVATELY ANSWERING TO A SURVEY

- Consider the following setup:
  - A researcher wants to conduct a survey of n individuals, which consists of a single yes/no question that the researcher asks each individual
  - The researcher is interested in the proportion of "yes" answers
  - However the subject matter is very sensitive or embarrassing, such as "did you have sex with a prostitute this month?" or "have you ever assaulted someone?"
- If the researcher was fully trusted to collect the true individual answers, we could use Laplace or Gaussian mechanisms to make the final result differentially private
- However, this is not the case here: we can expect that just asking the individuals to reply truthfully will induce important bias in the result of the survey
- **How can we provide privacy to the participants while getting an unbiased result?**

- We denote the truthful answer of individual i by $x_i \in \{0,1\}$ and the true proportion of "yes" by $Y = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Consider the following simple randomized approach: each participant answers truthfully ($z_i = x_i$) with probability p and falsely ($z_i = \neg x_i$) with probability $1-p$
- Let's do it! If you agree, we can use $p = 0.75$ (you can flip a coin two times, or just use a random number generator)
- The expected proportion of "yes" is given by $pY + (1-p)(1-Y)$, so we can recover an unbiased estimate $\hat{Y}$ of Y by computing:

$$\hat{Y} = \frac{\frac{1}{n}\sum_{i=1}^{n} z_i + p - 1}{2p-1}$$

- This approach, which dates back to [Warner, 1965], satisfies local differential privacy!

- As always, let $\mathcal{X}$ denote an abstract data domain
- A local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Z}$ is a randomized function which maps an input $x \in \mathcal{X}$ to an output $z \in \mathcal{Z}$

Definition (Local Differential Privacy [Kasiviswanathan et al., 2008, Duchi et al., 2013])

Let $\varepsilon > 0$ and $\delta \in (0, 1)$. A local randomizer algorithm $\mathcal{R}$ is (ε, δ)-locally differentially private (LDP) if for all $x, x' \in \mathcal{X}$ and any possible $z \in \mathcal{Z}$:

$$\Pr[\mathcal{R}(x) = z] \leq e^{\varepsilon} \Pr[\mathcal{R}(x') = z] + \delta.$$

- This is equivalent to (ε, δ)-DP for datasets of size 1!
- LDP is a much stronger model than central DP (no trusted curator)
- Indeed, LDP allows participants to have plausible deniability even if the curator is compromised: they can deny having value x on the basis of lack of evidence

K-ARY RANDOMIZED RESPONSE: ALGORITHM & PRIVACY GUARANTEES

- Assume a K-ary data domain $\mathcal{X} = \{v_1, \ldots, v_K\}$

Algorithm: K-**ary Randomized Response** $\mathcal{R}_{\mathrm{RR},K}(x, \varepsilon)$ [Kairouz et al., 2014]

1. Sample $b \sim \mathrm{Ber}(K/(e^{\varepsilon} + K - 1))$
2. If $b = 0$ output x, else output $y \sim \mathrm{Unif}(\mathcal{X})$

- $K-$RR will output the true value w.p. $\frac{e^{\varepsilon}-1}{e^{\varepsilon}+K-1}$, or a random value w.p. $\frac{K}{e^{\varepsilon}+K-1}$
- This can be seen as a generalization of the simple binary version that we used earlier

Theorem (DP guarantees for K-RR mechanism)

Let $\varepsilon > 0$. The K-ary randomized response mechanism $\mathcal{R}_{RR,K}(\cdot, \varepsilon)$ satisfies ε-LDP.

Proof.

- For any $x, x' \in \mathcal{X}$ and $z \in \mathcal{Z}$, we want to show that $\frac{\Pr[\mathcal{R}_{\mathrm{RR},K}(x)=z]}{\Pr[\mathcal{R}_{\mathrm{RR},K}(x')=z]} \leq e^{\varepsilon}$
- If $x \neq z \wedge x' \neq z$ or $x = x' = z$, then clearly $\Pr[\mathcal{R}_{\mathrm{RR},K}(x) = z] = \Pr[\mathcal{R}_{\mathrm{RR},K}(x') = z]$
- We thus focus on the case $x = z$ and $x' \neq z$. We have:

$$\Pr[\mathcal{R}_{\mathrm{RR},K}(x) = z] = \frac{e^{\varepsilon} - 1}{e^{\varepsilon} + K - 1} + \frac{K}{K(e^{\varepsilon} + K - 1)} = \frac{e^{\varepsilon}}{e^{\varepsilon} + K - 1}$$

$$\Pr[\mathcal{R}_{\mathrm{RR},K}(x') = z] = \frac{1}{e^{\varepsilon} + K - 1}$$

- Taking the ratio gives us the desired result

$\square$

- Let $h = (h_1, \dots, h_K)$ denote the histogram of the private data: $h_k = \frac{1}{n}\sum_{i=1}^n \mathbb{I}[x_i = v_k]$
- Letting $p = \frac{e^\varepsilon - 1}{e^\varepsilon + K - 1}$, K-RR allows us to obtain an unbiased estimate $\hat{h}$ of h by setting

$$\hat{h}_k = \frac{\left(\frac{1}{n}\sum_{i=1}^n \mathbb{I}[z_i = v_k]\right) - \frac{1-p}{K}}{p} = \frac{\left(\frac{1}{n}\sum_{i=1}^n \mathbb{I}[z_i = v_k]\right)(e^\varepsilon + K - 1) - 1}{e^\varepsilon - 1}$$

Theorem (ℓ_2 error of K-ary randomized response)

Let $\varepsilon > 0$. The histogram $\hat{h}$ obtained using the K-ary randomized response mechanism satisfies for any $k \in \{1, \dots, K\}$:

$$\mathbb{E}[(\hat{h}_k - h_k)^2] = \frac{K - 2 + e^\varepsilon}{n(e^\varepsilon - 1)^2}.$$

- Proof: exercise

- Let f be a public function from $\mathcal{X}$ to a bounded numeric range (say $f : \mathcal{X} \rightarrow [0, 1]$)
- We want to compute an averaging query $\bar{f} = \frac{1}{n} \sum_{i=1}^{n} f(x_i)$
- How to do this in the LDP setting?
- We can readily use the Laplace and Gaussian mechanisms!
- Indeed, seeing each input as a dataset of size 1, the query $f(x)$ sensitivity is 1:

$$\Delta_1(f) = \max_{x,x'} |f(x) - f(x')| = 1, \quad \text{and similarly } \Delta_2(f) = 1$$

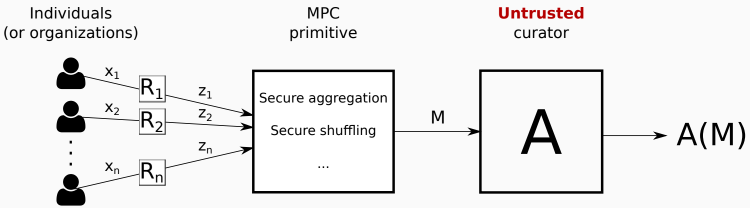
- For instance, with the Laplace mechanism, we get an estimate of $\bar{f}$ with variance $2/n\varepsilon^2$

- As one can expect, there is a large utility gap between the central and the local model of DP: it is typically a factor of $O(1/\sqrt{n})$ in ℓ_1 error (or $O(1/n)$ in ℓ_2 error)
- Example 1: histograms
    - In the local model, we have seen that $\mathbb{E}[(\hat{h}_k - h_k)^2] = O(1/n)$
    - In the central model, we can compute the exact $h_k = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[x_i = v_k]$ and add Laplace noise calibrated to its ℓ_1 sensitivity $1/n$, hence we get $\mathbb{E}[(\hat{h}_k - h_k)^2] = O(1/n^2)$
- Example 2: averaging queries
    - In the local model, we have seen that we get a variance of $O(1/n)$
    - In the central model, we can compute the exact $\bar{f}$ and add Laplace noise calibrated to its ℓ_1 sensitivity $\Delta_1(\bar{f}) = 1/n$, hence we get a variance of $O(1/n^2)$
- This gap is known to be unavoidable for some queries like averaging [Chan et al., 2012]
- This restricts the usefulness of LDP to applications where n is very large

Intermediate trust models

- The gap between local and central DP is due to the lack of a trusted curator
- If the participants could simulate the trusted curator without anyone learning anything more than the final result, we would obtain the best of both worlds!
- Designing such protocols is precisely the focus of secure multi-party computation (MPC), a subfield of cryptography
- It seems too good to be true. What is the catch?
- First, the guarantees of MPC only hold against computationally-bounded adversaries: this gives rise to the relaxed notion of computational DP [Mironov et al., 2009]
- Second, general-purpose MPC is computationally intractable, so we need to restrict our attention to MPC primitives that are sufficiently efficient

USEFUL MPC PRIMITIVES

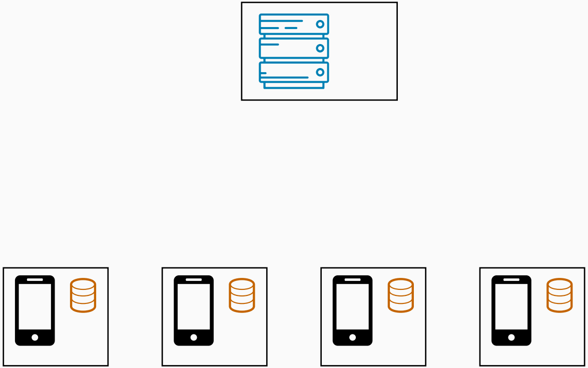


- Secure aggregation takes as input a value z_i for each participant i and outputs $\sum_{i=1}^{n} z_i$
  - Very natural to use in averaging/sum queries
  - State-of-the-art protocols [Bonawitz et al., 2017] have communication cost of $O(n^2)$
- Secure shuffling takes as input a value z_i for each participant i and outputs a random permutation of the inputs (i.e., makes communications anonymous)
  - Generic privacy amplification results [Erlingsson et al., 2019, Balle et al., 2019]
  - Practical implementations are costly (e.g., layers of servers + non-collusion assumptions)

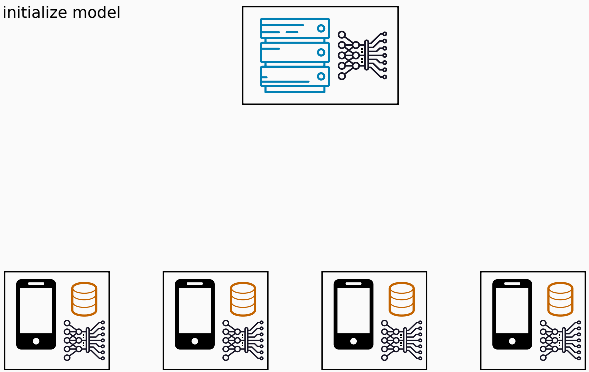
Federated Learning

A BROAD DEFINITION OF FEDERATED LEARNING

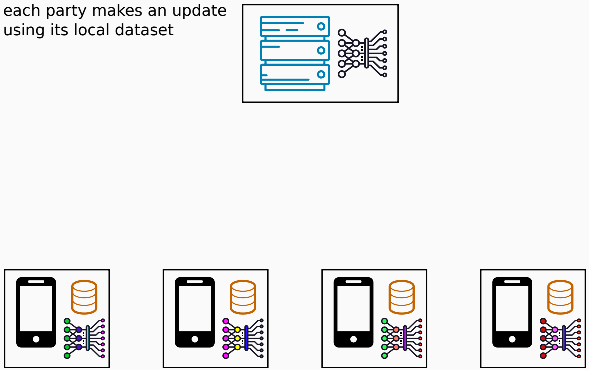
- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized



- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized

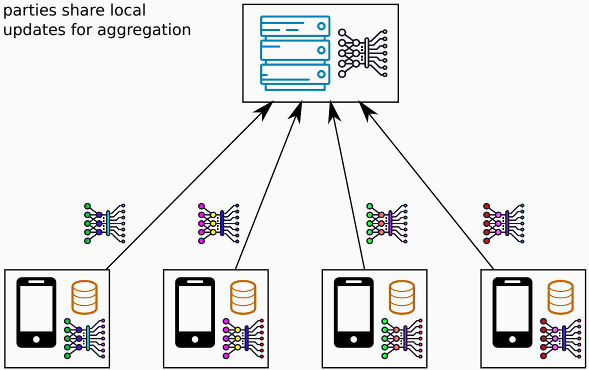


- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized

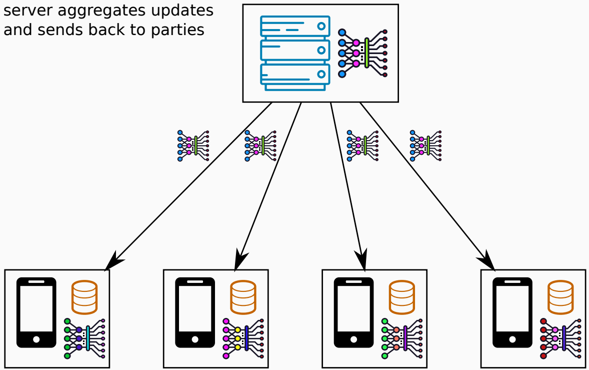


A BROAD DEFINITION OF FEDERATED LEARNING

- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized

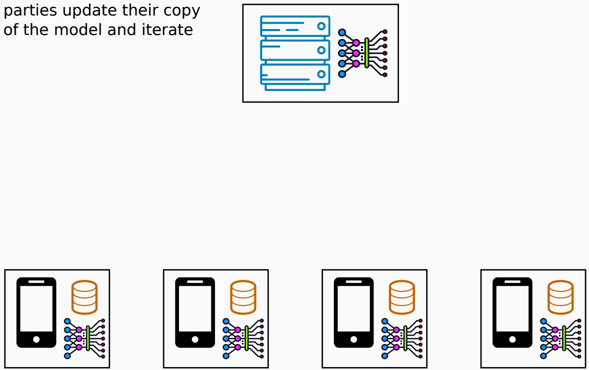


- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized



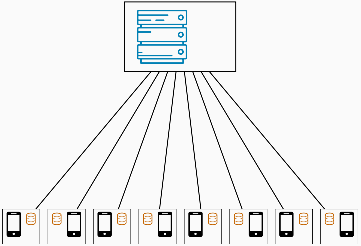
A BROAD DEFINITION OF FEDERATED LEARNING

- **Federated Learning (FL)** [Kairouz et al., 2021] aims to collaboratively train a ML model while keeping the data decentralized



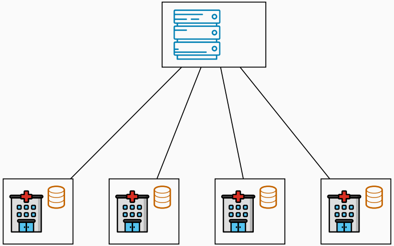
- We would like the final model to be as good as the centralized solution (ideally), or at least better than what each party can learn on its own

Cross-device FL



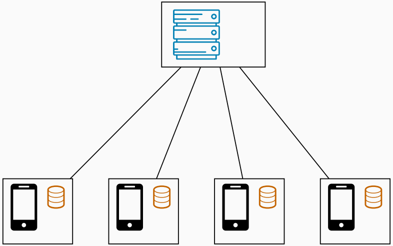
- Massive number of parties (up to 10^{10})
- Small dataset per party (could be size 1)
- Limited availability and reliability
- Some parties may be malicious

Cross-silo FL



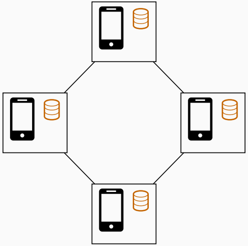
- 2-100 parties
- Medium to large dataset per party
- Reliable parties, almost always available
- Parties are typically honest

Server-orchestrated FL



- Server-client communication
- Global coordination, global aggregation
- Server is a single point of failure and may become a bottleneck

Fully decentralized FL



- Device-to-device communication
- No global coordination, local aggregation
- Naturally scales to a large number of devices

EMPIRICAL RISK MINIMIZATION IN FL

- We consider a set of n parties (clients)
- Each party i holds a dataset $\mathcal{D}_i$ of m_i points
- Let $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_n$ be the joint dataset and $m = \sum_i m_i$ the total number of points
- We denote by $\theta \in \mathbb{R}^p$ the model parameters
- We want to solve ERM problems of the form $\min_{\theta \in \mathbb{R}^p} F(\theta; \mathcal{D})$ where:

$$F(\theta; \mathcal{D}) = \sum_{i=1}^{n} \frac{m_i}{m} F_i(\theta; \mathcal{D}_i) \quad \text{and} \quad F_i(\theta; \mathcal{D}_i) = \sum_{(x,y) \in \mathcal{D}_i} L(\theta; x; y),$$

where $L(\theta; x, y)$ is the loss function

FEDAVG (AKA LOCAL SGD) [McMahan et al., 2017]

Algorithm FedAvg (server-side)

initialize θ
for each round $t = 0, 1, \ldots$ **do**
  for each client i in parallel **do**
    $\theta_i \leftarrow \text{ClientUpdate}(i, \theta)$
  end for
  $\theta \leftarrow \sum_{i=1}^{n} \frac{m_i}{m} \theta_i$
end for

Algorithm ClientUpdate(i, θ)

Parameters: batch size B, number of local steps E, learning rate η

for each local step $1, \ldots, E$ **do**
  $\mathcal{B} \leftarrow$ mini-batch of B examples from $\mathcal{D}_i$
  $\theta \leftarrow \theta - \frac{m_i}{B} \eta \sum_{(x,y) \in \mathcal{B}} \nabla L(\theta; x, y)$
end for
send θ to server

- For $E = 1$, it is equivalent to classic parallel SGD
- For $E > 1$: each client performs multiple local SGD steps before communicating

- A simple approach is to use local gradient perturbation to make each client update DP with respect to its local dataset
- In particular, when $E = 1$ we recover DP-SGD but the gradient used to update has increased variance (because noise is added locally before aggregation)
- Secure aggregation or other DP aggregation schemes [Sabater et al., 2020] can be readily used to recover the utility of centralized DP-SGD
- This is also the case with secure shuffling [Girgis et al., 2020]

Wrapping up

TAKE-AWAYS OF THE COURSE

1. Any personal information can be sensitive, and anonymization is hard
2. Privacy should be a property of the analysis, not of a particular output
3. Differential privacy provides a robust mathematical definition of privacy
4. Simple DP primitives can be used as basis to design complex algorithms
5. In ML, this leads to approaches based on output, objective and gradient perturbation
6. When there is no trusted curator, DP can be deployed locally at the participants' level
7. This can be used to train models while keeping data decentralized and confidential

- Privacy-preserving ML and federated learning are booming topics in the core ML community but also in applied fields and in the industry
- They are my main current research interests and key topics for the Inria Magnet team
- If you liked these topics, there may be opportunities for you (Master internships, PhD positions, engineer positions)
- Get in touch with me if you're interested!

[Balle et al., 2019] Balle, B., Bell, J., Gascón, A., and Nissim, K. (2019).
The Privacy Blanket of the Shuffle Model.
In *CRYPTO*.

[Bonawitz et al., 2017] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017).
Practical Secure Aggregation for Privacy-Preserving Machine Learning.
In *CCS*.

[Chan et al., 2012] Chan, T.-H. H., Shi, E., and Song, D. (2012).
Optimal Lower Bound for Differentially Private Multi-party Aggregation.
In *ESA*.

[Duchi et al., 2013] Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013).
Local Privacy and Statistical Minimax Rates.
In *FOCS*.

[Erlingsson et al., 2019] Erlingsson, U., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. (2019).
Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity.
In *SODA*.

[Girgis et al., 2020] Girgis, A. M., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. (2020).
Shuffled Model of Federated Learning: Privacy, Communication and Accuracy Trade-offs.
Technical report, arXiv:2008.07180.

[Kairouz et al., 2021] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021).
Advances and Open Problems in Federated Learning.
Foundations and Trends® in Machine Learning, 14(1–2):1–210.

[Kairouz et al., 2014] Kairouz, P., Oh, S., and Viswanath, P. (2014).
Extremal mechanisms for local differential privacy.
In *NIPS*.

[Kasiviswanathan et al., 2008] Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. D. (2008).
What Can We Learn Privately?
In *FOCS*.

[McMahan et al., 2017] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017).
Communication-efficient learning of deep networks from decentralized data.
In *AISTATS*.

[Mironov et al., 2009] Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. P. (2009).
Computational Differential Privacy.
In *CRYPTO*.

[Sabater et al., 2020] Sabater, C., Bellet, A., and Ramon, J. (2020).
Distributed Differentially Private Averaging with Improved Utility and Robustness to Malicious Parties.
Technical report, arXiv:2006.07218.

[Warner, 1965] Warner, S. L. (1965).
Randomised response: a survey technique for eliminating evasive answer bias.
Journal of the American Statistical Association, 60(309):63–69.