CONTRIBUTIONS TO DECENTRALIZED AND PRIVACY-PRESERVING MACHINE LEARNING

HABILITATION THESIS (HDR) DEFENSE

Aurélien Bellet (Inria Magnet / CRIStAL / Université de Lille)

November 30, 2021

Design ML algorithms that
take into account
societal and ethical issues

Design ML algorithms that
take into account
societal and ethical issues

Make ML algorithms accessible to citizens, so they can collectively define their own usage

Design ML algorithms that **take into account societal and ethical issues**	**Make ML algorithms accessible to citizens**, so they can collectively define their own usage

- **Decentralized ML:** learn collaboratively while keeping control of your data
- **Privacy-preserving ML:** ensure ML does not leak your sensitive data
- **Fair ML**: ensure ML model does not discriminate or is not overly biased
- **Speech privacy:** use voice interfaces without being personally identifiable
- **Transparent & reproducible ML**
- **Open source development**

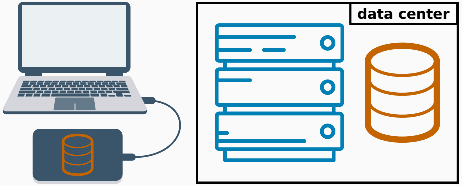
Design ML algorithms that **take into account societal and ethical issues**	**Make ML algorithms accessible to citizens**, so they can collectively define their own usage

- **Decentralized ML:** learn collaboratively while keeping control of your data
- **Privacy-preserving ML:** ensure ML does not leak your sensitive data
- **Fair ML**: ensure ML model does not discriminate or is not overly biased
- **Speech privacy:** use voice interfaces without being personally identifiable
- **Transparent & reproducible ML**
- **Open source development**

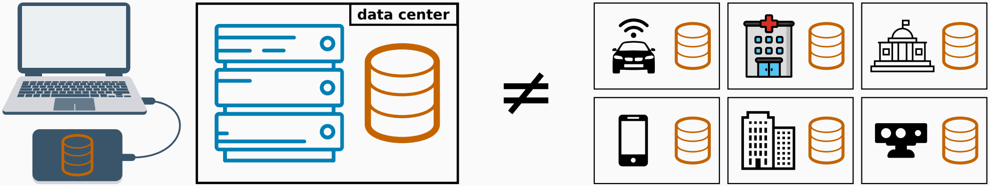
What is Decentralized and Privacy-Preserving Machine Learning?

- The standard setting in ML considers a centralized dataset processed in a tightly integrated system

- The standard setting in ML considers a centralized dataset processed in a tightly integrated system



- The standard setting in ML considers a centralized dataset processed in a tightly integrated system
- But in the real world data is often decentralized across many parties



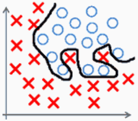
WHY CAN'T WE JUST CENTRALIZE THE DATA?

1. Sending the data may be too costly
    - Self-driving cars are expected to generate several TBs of data a day 
    - Some wireless devices have limited bandwidth/power 

WHY CAN'T WE JUST CENTRALIZE THE DATA?

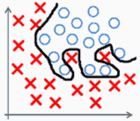
1. Sending the data may be too costly
    - Self-driving cars are expected to generate several TBs of data a day
    - Some wireless devices have limited bandwidth/power
2. **Data may be considered too sensitive to be shared**
    - We see a growing public awareness and regulations on data privacy
    - Keeping control of data can give a competitive advantage in business and research

1. The local dataset may be too small
    - Sub-par predictive performance (e.g., due to overfitting)
    - Non-statistically significant results (e.g., medical studies)

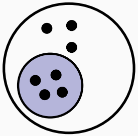


HOW ABOUT EACH PARTY LEARNING ON ITS OWN?

1. The local dataset may be too small
   - Sub-par predictive performance (e.g., due to overfitting)
   - Non-statistically significant results (e.g., medical studies)



2. The local dataset may be biased
   - Not representative of the target distribution



Decentralized learning (also called **federated learning**)
aims to collaboratively train ML models
while keeping data decentralized

$\longrightarrow$ shared exploitation of the data rather than sharing the data itself

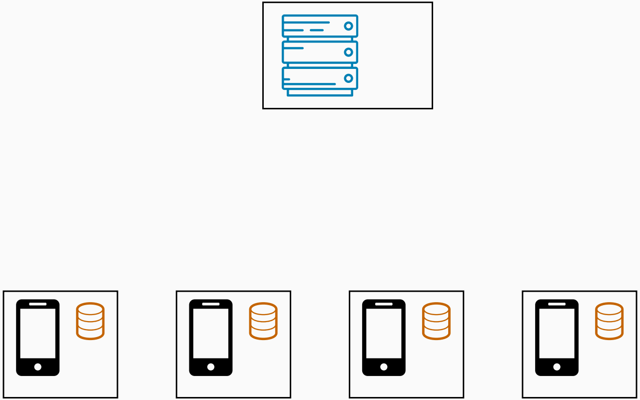
Decentralized learning (also called **federated learning**)
aims to collaboratively train ML models
while keeping data decentralized

→ shared exploitation of the data rather than sharing the data itself

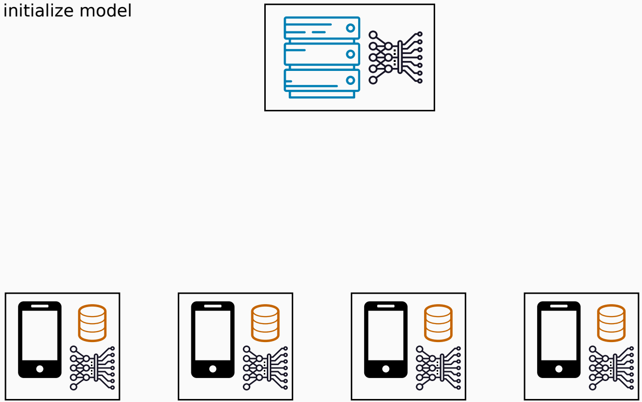
- When I started working on this in 2015-2016, it was a newly emerging topic
- It is now in a booming phase[1]

[1] https://www.forbes.com/sites/robtoews/2020/10/12/the-next-generation-of-artificial-intelligence/

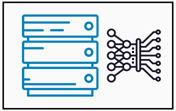
DECENTRALIZED LEARNING: TYPICAL PROCESS

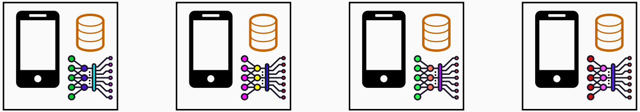


DECENTRALIZED LEARNING: TYPICAL PROCESS

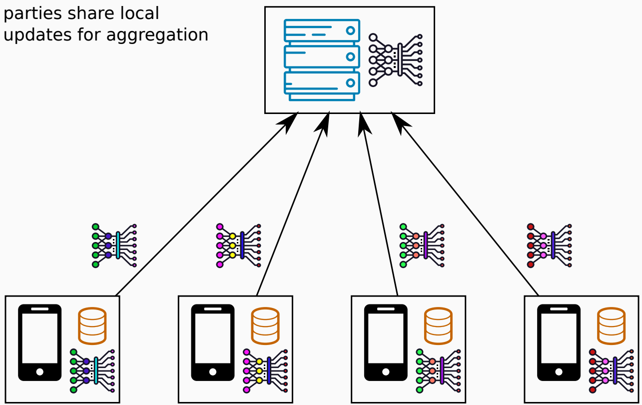


each party makes an update
using its local dataset



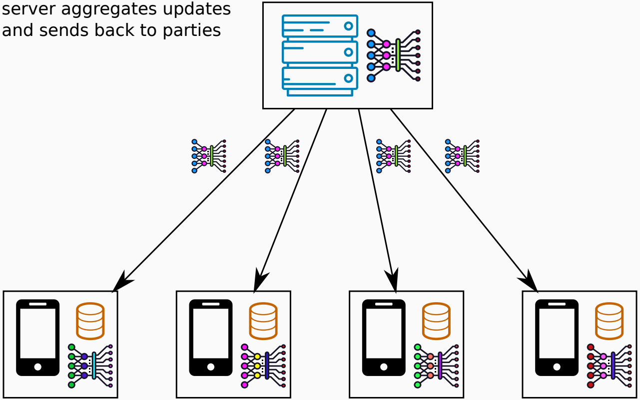


DECENTRALIZED LEARNING: TYPICAL PROCESS



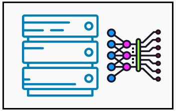
DECENTRALIZED LEARNING: TYPICAL PROCESS

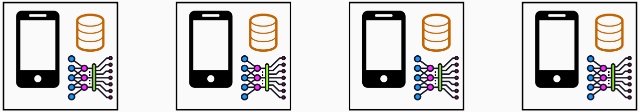
server aggregates updates
and sends back to parties



DECENTRALIZED LEARNING: TYPICAL PROCESS

parties update their copy
of the model and iterate



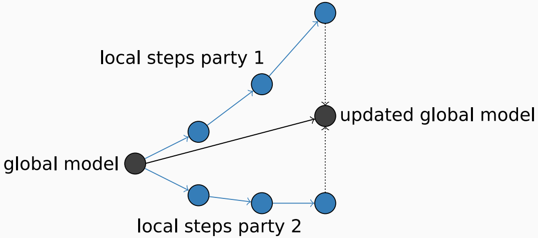


CHALLENGE 1: DEALING WITH DATA HETEROGENEITY

- Decentralized learning comes with many challenges, distinct from those of classic distributed ML on a cluster (see our collaborative survey [Kairouz et al., 2021])

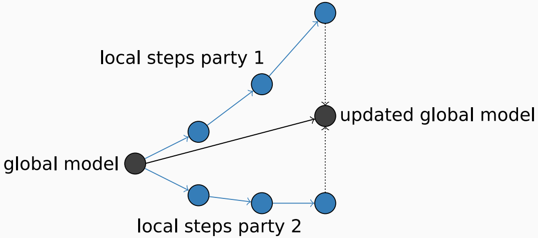
CHALLENGE 1: DEALING WITH DATA HETEROGENEITY

- Decentralized learning comes with many challenges, distinct from those of classic distributed ML on a cluster (see our collaborative survey [Kairouz et al., 2021])
- Local datasets are often highly heterogeneous, because they reflect the usage and production patterns specific to each party



CHALLENGE 1: DEALING WITH DATA HETEROGENEITY

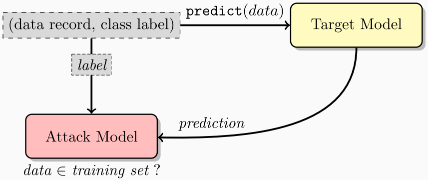
- Decentralized learning comes with many challenges, distinct from those of classic distributed ML on a cluster (see our collaborative survey [Kairouz et al., 2021])
- Local datasets are often highly heterogeneous, because they reflect the usage and production patterns specific to each party



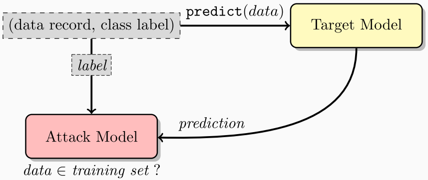
- **Challenges:** design low-communication decentralized algorithms that scale to many parties and learn models that are useful to all users

- Not sharing data is insufficient to obtain robust privacy guarantees

- Not sharing data is insufficient to obtain robust privacy guarantees
- Information about training individual training points can be extracted from a trained model [Shokri et al., 2017, Paige et al., 2020]



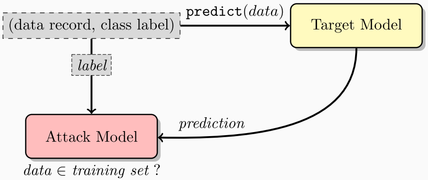
- Not sharing data is insufficient to obtain robust privacy guarantees
- Information about training individual training points can be extracted from a trained model [Shokri et al., 2017, Paige et al., 2020]



- Decentralized learning offers an additional attack surface because the server and/or other parties observe model updates (not only the final model)

CHALLENGE 2: PROTECTING PRIVACY

- Not sharing data is insufficient to obtain robust privacy guarantees
- Information about training individual training points can be extracted from a trained model [Shokri et al., 2017, Paige et al., 2020]



- Decentralized learning offers an additional attack surface because the server and/or other parties observe model updates (not only the final model)
- **Challenges:** design decentralized learning algorithms with rigorous privacy guarantees while minimizing the impact on the utility of the resulting models

FOCUS ON TWO SETS OF CONTRIBUTIONS

1. Decentralized Learning of Personalized Models
2. Better Privacy-Utility Trade-offs for Decentralized Learning

Decentralized Learning of Personalized Models

WARM-UP: LEARN A GLOBAL MODEL FOR EVERYONE

- A set of n users who behave honestly (i.e., follow the protocol)
- Each user u holds a dataset $\mathcal{D}_u$ of m_u data points, and we let $m = \sum_u m_u$
- Models with parameters θ (e.g., weights of a linear classifier or neural network)
- A standard objective is to learn a global model by solving a problem of the form

$$\arg\min_{\theta} \sum_{u=1}^{n} \frac{m_u}{m} F_u(\theta; \mathcal{D}_u)$$

PROPOSED FORMULATION: LEARN PERSONALIZED MODELS

- We propose to learn personalized models $\Theta = (\theta_1, \ldots, \theta_n)$ and a similarity graph represented by pairwise weights $w = (w_{u,v})_{u<v}$ by solving

$$\arg\min_{\Theta, w \geq 0} \sum_{u=1}^{n} \qquad \frac{m_u}{m} F_u(\theta_u; \mathcal{D}_u)$$

- Trade-off between learning accurate models on local data

PROPOSED FORMULATION: LEARN PERSONALIZED MODELS

- We propose to learn personalized models $\Theta = (\theta_1, \ldots, \theta_n)$ and a similarity graph represented by pairwise weights $w = (w_{u,v})_{u<v}$ by solving

$$\underset{\Theta, w \geq 0}{\arg\min} \sum_{u=1}^{n} d_u(w) \frac{m_u}{m} F_u(\theta_u; \mathcal{D}_u) + \frac{\lambda_1}{2} \sum_{1 \leq u < v \leq n} w_{u,v} \|\theta_u - \theta_v\|^2$$

- Trade-off between learning accurate models on local data and learning similar models for similar users (the degree $d_u(w) = \sum_{v \neq u} w_{u,v}$ is a normalizing factor)

PROPOSED FORMULATION: LEARN PERSONALIZED MODELS

- We propose to learn personalized models $\Theta = (\theta_1, \dots, \theta_n)$ and a similarity graph represented by pairwise weights $w = (w_{u,v})_{u<v}$ by solving

$$\underset{\Theta, w \geq 0}{\arg\min} \sum_{u=1}^{n} d_u(w) \frac{m_u}{m} F_u(\theta_u; \mathcal{D}_u) + \frac{\lambda_1}{2} \sum_{1 \leq u < v \leq n} w_{u,v} \|\theta_u - \theta_v\|^2$$

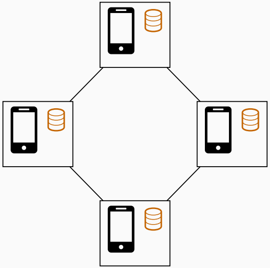
- Trade-off between learning accurate models on local data and learning similar models for similar users (the degree $d_u(w) = \sum_{v \neq u} w_{u,v}$ is a normalizing factor)
- Captures flexible relationships: hyperparameter $\lambda_1 \geq 0$ interpolates between learning purely local models and a shared model per connected component

PROPOSED FORMULATION: LEARN PERSONALIZED MODELS

- We propose to learn personalized models $\Theta = (\theta_1, \dots, \theta_n)$ and a similarity graph represented by pairwise weights $w = (w_{u,v})_{u<v}$ by solving

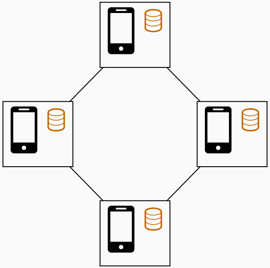
$$\arg\min_{\Theta, w \geq 0} \sum_{u=1}^{n} d_u(w) \frac{m_u}{m} F_u(\theta_u; \mathcal{D}_u) + \frac{\lambda_1}{2} \sum_{1 \leq u < v \leq n} w_{u,v} \|\theta_u - \theta_v\|^2 + \lambda_2 g(w),$$

- Trade-off between learning accurate models on local data and learning similar models for similar users (the degree $d_u(w) = \sum_{v \neq u} w_{u,v}$ is a normalizing factor)
- Captures flexible relationships: hyperparameter $\lambda_1 \geq 0$ interpolates between learning purely local models and a shared model per connected component
- Graph regularizer $g(w)$: avoid trivial graph, encourage sparsity



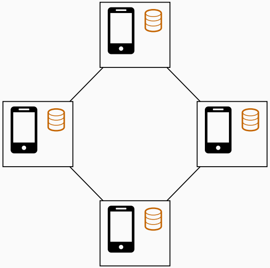
- We remove the need for a central server: instead, each user communicates with a small number of neighbors in a network graph

FULLY DECENTRALIZED SETTING



- We remove the need for a central server: instead, each user communicates with a small number of neighbors in a network graph
- We consider an asynchronous time model: users become active asynchronously and in parallel at random times

FULLY DECENTRALIZED SETTING



- We remove the need for a central server: instead, each user communicates with a small number of neighbors in a network graph
- We consider an asynchronous time model: users become active asynchronously and in parallel at random times

$\rightarrow$ Naturally scales to many users (as long as network graph is sparse)

- We will solve the problem by alternating optimization over Θ and w

LEARNING THE MODELS GIVEN THE GRAPH

- We will solve the problem by alternating optimization over Θ and w
- For fixed graph w, we design an algorithm to optimize the models Θ where each user u communicates only with its neighborhood in w: $\mathcal{N}(u) = \{v : w_{u,v} > 0\}$

- We will solve the problem by alternating optimization over Θ and w
- For fixed graph w, we design an algorithm to optimize the models Θ where each user u communicates only with its neighborhood in w: $\mathcal{N}(u) = \{v : w_{u,v} > 0\}$
- At step $t \geq 0$, a random user u becomes active:
  1. user u combines a weighted average of neighbors' models and a local gradient step:

$$\theta_u(t+1) = (1-\alpha)\theta_u(t) + \alpha\Big(\sum_{v\in\mathcal{N}(u)} \frac{w_{u,v}}{d_u(w)}\theta_v(t) - \frac{m_u}{\lambda_1 m}\nabla F_u(\theta_u(t); \mathcal{D}_u)\Big)$$

  2. user u sends its updated model $\theta_k(t+1)$ to its neighborhood $\mathcal{N}(k)$

Reminder of the objective: $\sum d_u(w)\frac{m_u}{m}F_u(\theta_u;\mathcal{D}_u) + \frac{\lambda_1}{2}\sum w_{u,v}\|\theta_u - \theta_v\|^2 + \lambda_2 g(w)$

- We avoid having isolated users and control the graph sparsity with the regularizer:

$$g(w) = -1^T \log(d(w)) + \lambda_3\|w\|^2$$

Reminder of the objective: $\sum d_u(w)\frac{m_u}{m}F_u(\theta_u;\mathcal{D}_u) + \frac{\lambda_1}{2}\sum w_{u,v}\|\theta_u - \theta_v\|^2 + \lambda_2 g(w)$

- We avoid having isolated users and control the graph sparsity with the regularizer:

$$g(w) = -1^T \log(d(w)) + \lambda_3\|w\|^2$$

- For fixed models Θ, we design an algorithm to optimize the graph w where users contact a small number of peers via decentralized peer sampling [Jelasity et al., 2007]

LEARNING THE GRAPH GIVEN THE MODELS

Reminder of the objective: $\sum d_u(w)\frac{m_u}{m}F_u(\theta_u;\mathcal{D}_u) + \frac{\lambda_1}{2}\sum w_{u,v}\|\theta_u - \theta_v\|^2 + \lambda_2 g(w)$

- We avoid having isolated users and control the graph sparsity with the regularizer:

$$g(w) = -1^T \log(d(w)) + \lambda_3\|w\|^2$$

- For fixed models Θ, we design an algorithm to optimize the graph w where users contact a small number of peers via decentralized peer sampling [Jelasity et al., 2007]

- At step $t \geq 0$, a random user u becomes active:
  1. Use peer sampling to contact a set $\mathcal{V}$ of ρ users, request their model and degree
  2. Update the weights with users in $\mathcal{V}$ via a gradient update
  3. Send each user $v \in \mathcal{V}$ the updated weight $w(t+1)_{u,v}$

Theorem (Convergence rates, informal [Bellet et al., 2018, Zantedeschi et al., 2020])

Let $J(\Theta, w)$ be the joint objective.

Theorem (Convergence rates, informal [Bellet et al., 2018, Zantedeschi et al., 2020])

Let $J(\Theta, w)$ be the joint objective.

1. *For fixed w, let $M(\Theta) = J(\Theta, w)$. There exists $\kappa > 0$ such that for any $T > 0$:*

$$\mathbb{E}\left[M(\Theta(T)) - M^\star\right] \leq \left(1 - \frac{\kappa}{n}\right)^T \left(M(\Theta(0)) - M^\star\right).$$

Theorem (Convergence rates, informal [Bellet et al., 2018, Zantedeschi et al., 2020])

Let $J(\Theta, w)$ be the joint objective.

1. *For fixed w, let $M(\Theta) = J(\Theta, w)$. There exists $\kappa > 0$ such that for any $T > 0$:*

$$\mathbb{E}\left[M(\Theta(T)) - M^\star\right] \leq \left(1 - \frac{\kappa}{n}\right)^T \left(M(\Theta(0)) - M^\star\right).$$

2. *For fixed Θ, let $G(w) = J(\Theta, w)$. There exists $\kappa' > 0$ such that for any $T > 0$:*

$$\mathbb{E}[G(w(T)) - G^*] \leq \left(1 - \frac{\rho\kappa}{n(n-1)}\right)^T (G(w(0)) - G^*).$$

Theorem (Convergence rates, informal [Bellet et al., 2018, Zantedeschi et al., 2020])

Let $J(\Theta, w)$ be the joint objective.

1. *For fixed w, let $M(\Theta) = J(\Theta, w)$. There exists $\kappa > 0$ such that for any $T > 0$:*

$$\mathbb{E}\left[M(\Theta(T)) - M^\star\right] \leq \left(1 - \frac{\kappa}{n}\right)^T \left(M(\Theta(0)) - M^\star\right).$$

2. *For fixed Θ, let $G(w) = J(\Theta, w)$. There exists $\kappa' > 0$ such that for any $T > 0$:*

$$\mathbb{E}[G(w(T)) - G^*] \leq \left(1 - \frac{\rho\kappa}{n(n-1)}\right)^T (G(w(0)) - G^*).$$

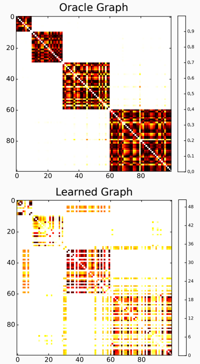
3. *The alternating optimization of Θ and w converges to a local minimum of J.*

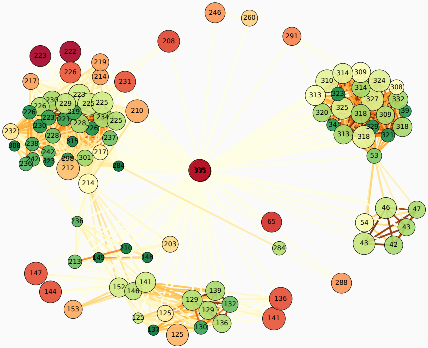
EMPIRICAL RESULTS

- On heterogeneous data, our approach typically outperforms both global and purely local models

EMPIRICAL RESULTS

- On heterogeneous data, our approach typically outperforms both global and purely local models
- Our formulation can learn complex relationships between users





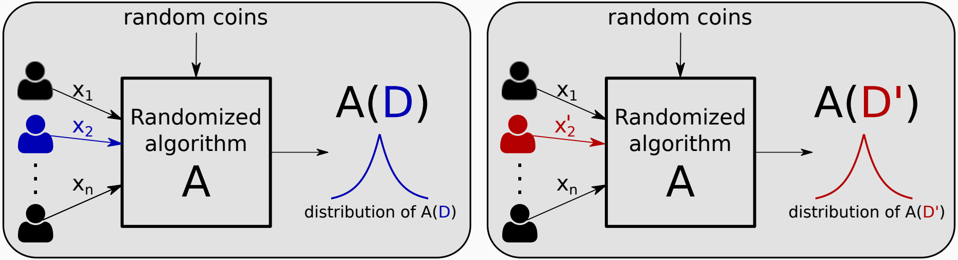
We proposed to learn personalized models in a fully decentralized setting:

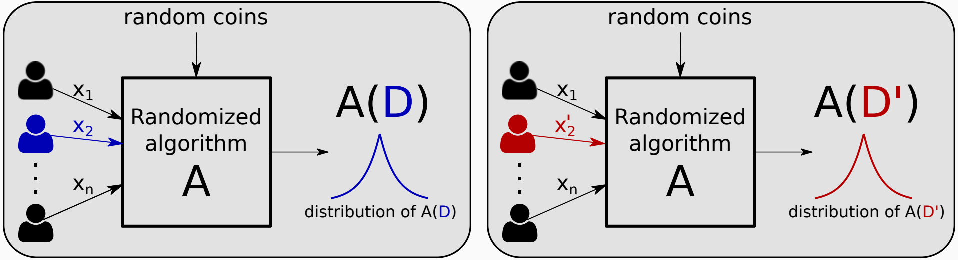
- We modeled relationships between users by a sparse similarity graph
- We leveraged this graph to learn better personalized models for each user
- We jointly optimized the models and the graph

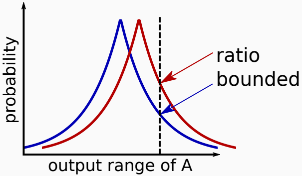
$\rightarrow$ the first method for personalized decentralized learning: this has become a standard approach to deal with heterogeneous data

Better Privacy-Utility Trade-offs for Decentralized Learning

PRIVACY NOTION: DIFFERENTIAL PRIVACY







Definition ([Dwork et al., 2006], informal)

$\mathcal{A}$ is (ϵ, δ)-DP if for all neighboring datasets $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$ and $\mathcal{D}' = \{x_1, x'_2, x_3, \dots, x_n\}$ and all possible sets of outputs S:

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

- **Central DP**: a trusted curator collects raw data and runs a DP algorithm $\mathcal{A}$ on it $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result

TRUST MODELS: CENTRAL DP VERSUS LOCAL DP

- **Central DP**: a trusted curator collects raw data and runs a DP algorithm $\mathcal{A}$ on it $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- **Local DP**: there is no trusted curator so each user must locally randomize its contributions $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all users

- **Central DP**: a trusted curator collects raw data and runs a DP algorithm $\mathcal{A}$ on it $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- **Local DP**: there is no trusted curator so each user must locally randomize its contributions $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all users
- Local DP is a suitable model for decentralized learning without trusted parties but, for a fixed (ϵ, δ)-DP guarantee, its utility cost is typically $\sqrt{n}$ larger

- **Central DP**: a trusted curator collects raw data and runs a DP algorithm $\mathcal{A}$ on it $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- **Local DP**: there is no trusted curator so each user must locally randomize its contributions $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all users
- Local DP is a suitable model for decentralized learning without trusted parties but, for a fixed (ϵ, δ)-DP guarantee, its utility cost is typically $\sqrt{n}$ larger

$\rightarrow$ study intermediate models allowing better utility without relying on trusted parties

A KEY FUNCTIONALITY: DP AGGREGATION

- In most decentralized algorithms with a server, interaction is needed only to aggregate local updates $\rightarrow$ this is the step we need to make private

- In most decentralized algorithms with a server, interaction is needed only to aggregate local updates $\rightarrow$ this is the step we need to make private
- **Differentially private aggregation:** given a private value $\theta_u \in [0,1]$ for each user u, we want to accurately estimate $\theta^{avg} = \frac{1}{n}\sum_u \theta_u$ under an (ϵ, δ)-DP constraint

A KEY FUNCTIONALITY: DP AGGREGATION

- In most decentralized algorithms with a server, interaction is needed only to aggregate local updates $\rightarrow$ this is the step we need to make private
- **Differentially private aggregation:** given a private value $\theta_u \in [0, 1]$ for each user u, we want to accurately estimate $\theta^{avg} = \frac{1}{n} \sum_u \theta_u$ under an (ϵ, δ)-DP constraint
- Central DP: trusted server computes θ^{avg} and adds Gaussian noise

A KEY FUNCTIONALITY: DP AGGREGATION

- In most decentralized algorithms with a server, interaction is needed only to aggregate local updates $\rightarrow$ this is the step we need to make private
- **Differentially private aggregation:** given a private value $\theta_u \in [0, 1]$ for each user u, we want to accurately estimate $\theta^{avg} = \frac{1}{n}\sum_u \theta_u$ under an (ϵ, δ)-DP constraint
- Central DP: trusted server computes θ^{avg} and adds Gaussian noise
- Local DP: each user u adds (more) Gaussian noise to θ_u before sharing it

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each user u generates independent Gaussian noise η_u

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each user u generates independent Gaussian noise η_u
Each user u selects a random set of k other users

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each user u generates independent Gaussian noise η_u
Each user u selects a random set of k other users
for all selected pairs of users $u \sim v$ **do**
    Users u and v securely exchange pairwise-canceling Gaussian noise $\Delta_{u,v} = -\Delta_{v,u}$

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each user u generates independent Gaussian noise η_u
Each user u selects a random set of k other users
for all selected pairs of users $u \sim v$ **do**
  Users u and v securely exchange pairwise-canceling Gaussian noise $\Delta_{u,v} = -\Delta_{v,u}$
Each user u sends $\hat{\theta}_u = \theta_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u$ to the server

- Assume that pairs of users are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each user u generates independent Gaussian noise η_u
Each user u selects a random set of k other users
for all selected pairs of users $u \sim v$ **do**
  Users u and v securely exchange pairwise-canceling Gaussian noise $\Delta_{u,v} = -\Delta_{v,u}$
Each user u sends $\hat{\theta}_u = \theta_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u$ to the server

- **Estimate of the average:** $\hat{\theta}^{avg} = \frac{1}{n} \sum_u \hat{\theta}_u = \theta^{avg} + \frac{1}{n} \sum_u \eta_u$

- **Adversary**: coalition of the server with a proportion $1-\tau$ of the users

- **Adversary**: coalition of the server with a proportion $1 - \tau$ of the users

Theorem (Privacy of GOPA [Sabater et al., 2020], informal)

- *Let each user select $k = O(\log(\tau n)/\tau)$ other users*
- *Set the independent noise variance so as to satisfy (ϵ, δ')-DP in the central model*
- *For large enough pairwise noise variance, GOPA is (ϵ, δ)-DP with $\delta = O(\delta')$.*

- **Adversary**: coalition of the server with a proportion $1 - \tau$ of the users

Theorem (Privacy of GOPA [Sabater et al., 2020], informal)

- *Let each user select $k = O(\log(\tau n)/\tau)$ other users*
- *Set the independent noise variance so as to satisfy (ϵ, δ')-DP in the central model*
- *For large enough pairwise noise variance, GOPA is (ϵ, δ)-DP with $\delta = O(\delta')$.*

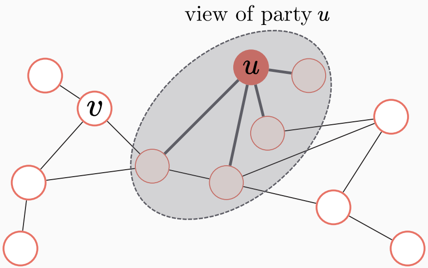
- Same utility as central DP with only logarithmic number of messages per user

HOW ABOUT FULLY DECENTRALIZED ALGORITHMS?

- In fully decentralized learning, there is no global aggregation step

HOW ABOUT FULLY DECENTRALIZED ALGORITHMS?

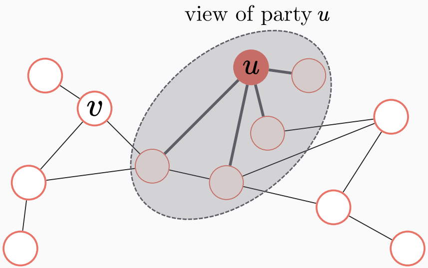
- In fully decentralized learning, there is no global aggregation step



- But there is no server observing all messages, and each user u has a limited view

HOW ABOUT FULLY DECENTRALIZED ALGORITHMS?

- In fully decentralized learning, there is no global aggregation step



- But there is no server observing all messages, and each user u has a limited view
- **Question:** can this be used to prove stronger differential privacy guarantees?
- Motivated by previous work on private rumor spreading [Bellet et al., 2020]

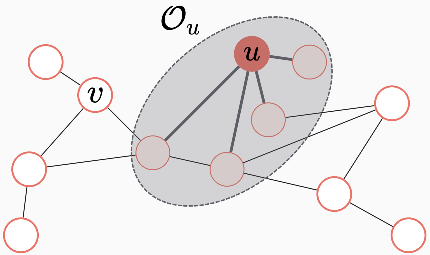
- Let $\mathcal{O}_u$ be the set of messages sent and received by user u

- Let $\mathcal{O}_u$ be the set of messages sent and received by user u

Definition (Network DP [Cyffers and Bellet, 2020])

An algorithm $\mathcal{A}$ satisfies (ϵ, δ)-network DP if for all pairs of distinct users $u, v \in \{1, \dots, n\}$ and all pairs of datasets $\mathcal{D}, \mathcal{D}'$ that differ only in the local dataset of user v, we have:

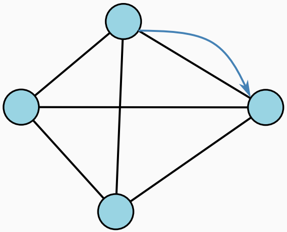
$$\Pr[\mathcal{O}_u(\mathcal{A}(\mathcal{D}))] \leq e^{\epsilon} \Pr[\mathcal{O}_u(\mathcal{A}(\mathcal{D}'))] + \delta.$$



- This is a relaxation of local DP: if $\mathcal{O}_u$ contains the full transcript of messages, then network DP boils down to local DP

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{u=1}^{n} F_u(\theta; \mathcal{D}_u)$ and a complete graph

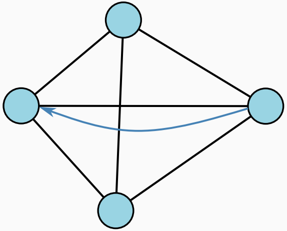
- Consider the standard objective $F(\theta;\mathcal{D}) = \frac{1}{n}\sum_{u=1}^{n} F_u(\theta;\mathcal{D}_u)$ and a complete graph
- We consider a decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

Initialize model θ
for $t = 1$ to T **do**
  Current user updates θ by a gradient update with Gaussian noise
  Current user sends θ to a random user
return θ

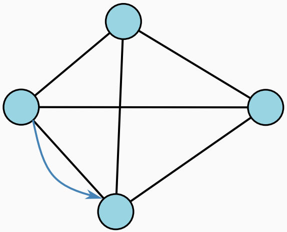
- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{u=1}^{n} F_u(\theta; \mathcal{D}_u)$ and a complete graph
- We consider a decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

Initialize model θ
for $t = 1$ to T **do**
  Current user updates θ by a gradient update with Gaussian noise
  Current user sends θ to a random user
return θ

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{u=1}^{n} F_u(\theta; \mathcal{D}_u)$ and a complete graph
- We consider a decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

Initialize model θ
for $t = 1$ to T **do**
  Current user updates θ by a gradient update with Gaussian noise
  Current user sends θ to a random user
return θ

Theorem ([Cyffers and Bellet, 2020], informal)

To achieve a fixed (ϵ, δ)-DP guarantee with the previous algorithm, the standard deviation of the noise is $O(\sqrt{n}/\ln n)$ smaller under network DP than under local DP.

Theorem ([Cyffers and Bellet, 2020], informal)

To achieve a fixed (ϵ, δ)-DP guarantee with the previous algorithm, the standard deviation of the noise is $O(\sqrt{n}/\ln n)$ smaller under network DP than under local DP.

- Accounting for the limited view in fully decentralized algorithms amplifies privacy guarantees by a factor of $O(\ln n/\sqrt{n})$, nearly recovering the utility of central DP

Theorem ([Cyffers and Bellet, 2020], informal)

To achieve a fixed (ϵ, δ)-DP guarantee with the previous algorithm, the standard deviation of the noise is $O(\sqrt{n}/\ln n)$ smaller under network DP than under local DP.

- Accounting for the limited view in fully decentralized algorithms amplifies privacy guarantees by a factor of $O(\ln n/\sqrt{n})$, nearly recovering the utility of central DP
- The proof leverages recent results on privacy amplification by iteration [Feldman et al., 2018] and exploits the randomness of the path taken by the model

Theorem ([Cyffers and Bellet, 2020], informal)

To achieve a fixed (ϵ, δ)-DP guarantee with the previous algorithm, the standard deviation of the noise is $O(\sqrt{n}/\ln n)$ smaller under network DP than under local DP.

- Accounting for the limited view in fully decentralized algorithms amplifies privacy guarantees by a factor of $O(\ln n/\sqrt{n})$, nearly recovering the utility of central DP
- The proof leverages recent results on privacy amplification by iteration [Feldman et al., 2018] and exploits the randomness of the path taken by the model
- We show some robustness to collusion (albeit with smaller privacy amplification)

We proposed decentralized methods that nearly match the utility of central DP:

1. We designed a aggregation protocol for **decentralized learning with a server**
   $\rightarrow$ avoids costs and implementation issues of secure computation-based solutions

We proposed decentralized methods that nearly match the utility of central DP:

1. We designed a aggregation protocol for **decentralized learning with a server**

   $\rightarrow$ avoids costs and implementation issues of secure computation-based solutions

2. We showed how to exploit the limited view of users in **fully decentralized algorithms**

   $\rightarrow$ the first work to show that full decentralization can amplify privacy guarantees, providing a new motivation for such algorithms beyond scalability

Putting Decentralized Learning to Practice

CHALLENGE 3: REAL DEPLOYMENTS

- Technological challenges: develop general-purpose software libraries which can be easily deployed in production systems

CHALLENGE 3: REAL DEPLOYMENTS

- Technological challenges: develop general-purpose software libraries which can be easily deployed in production systems
- Regulatory/legal challenges: when should model updates be considered as personal data? how to ensure compliance with current regulations (e.g., GDPR)?

CHALLENGE 3: REAL DEPLOYMENTS

- Technological challenges: develop general-purpose software libraries which can be easily deployed in production systems
- Regulatory/legal challenges: when should model updates be considered as personal data? how to ensure compliance with current regulations (e.g., GDPR)?
- Convincing stakeholders: what are the key merits of decentralized learning for a given application? how to make it as transparent as possible to the end-users?

- We are currently exploring these questions with Lille University Hospital in the context of my project FLAMED

- We are currently exploring these questions with Lille University Hospital in the context of my project FLAMED
- We have started developing our own code base and will soon deploy a proof-of-concept across 4 French hospitals

- We are currently exploring these questions with Lille University Hospital in the context of my project FLAMED
- We have started developing our own code base and will soon deploy a proof-of-concept across 4 French hospitals
- Deployments on concrete medical studies with real data by the end of the year

- We are currently exploring these questions with Lille University Hospital in the context of my project FLAMED
- We have started developing our own code base and will soon deploy a proof-of-concept across 4 French hospitals
- Deployments on concrete medical studies with real data by the end of the year
- We have some official support from CNIL (the French Data Protection Authority) on legal aspects (such as writing DPIAs)[2]

[2] https://www.cnil.fr/fr/bac-sable-donnees-personnelles-la-cnil-accompagne-12-projets-dans-le-domaine-de-la-sante-numerique

Future Research

Related topics

"Improve DP guarantees at no cost in utility by exploiting the way information is exchanged in fully decentralized ML"

(4-year grant funded by the French National Research Agency, started in 2021)

Three complementary research directions:

1. (Broadening the scope of) privacy amplification by decentralization
2. Secure multi-party computation meets decentralized algorithms
3. Data-adaptive decentralized communication

Show that fully decentralized algorithms amplify privacy in a variety of settings

- General and time-evolving topologies to balance privacy, scalability and robustness
- Algorithms allowing more parallel computation
- Lower bounds on the error achievable under network DP
- Further relaxations, e.g. when each user may trust a few peers in the network

$\rightarrow$ PhD of Edwige Cyffers

Combine secure multi-party computation (MPC) and decentralized algorithms

- Decentralized algorithms that use MPC primitives in local steps
- Trade-offs between computation, communication and privacy ruled by the number of parties involved in local steps

→ Postdoc (to hire) + collaborations with MPC experts like Adrià Gascón

Design data-adaptive topologies for faster convergence under heterogeneous data

- Optimization of the topology under statistical assumptions on data heterogeneity
- General types of heterogeneity, extending our work on label skew [Bellet et al., 2021]
- Dynamic adaptation of the topology while learning

→ Postdoc of Batiste Le Bars + collaboration with computing systems team at EPFL

An Inria-wide project on decentralized learning
$\rightarrow$ Coordinated by G. Neglia and myself, to start in 2022

- Foster collaborations between Inria teams on this topic
- Multidisciplinary: ML, optimization, privacy & security, networks, systems...

Future Research

Broader topics

- Achieving better privacy-utility trade-offs in private optimization may be possible by making additional assumptions on the problem structure
- We have recently started considering finer coordinate-wise regularity assumptions [Mangold et al., 2021]
- Assumptions about the structure of the optimal solution (such as sparsity) are promising directions to tackle high-dimensionality

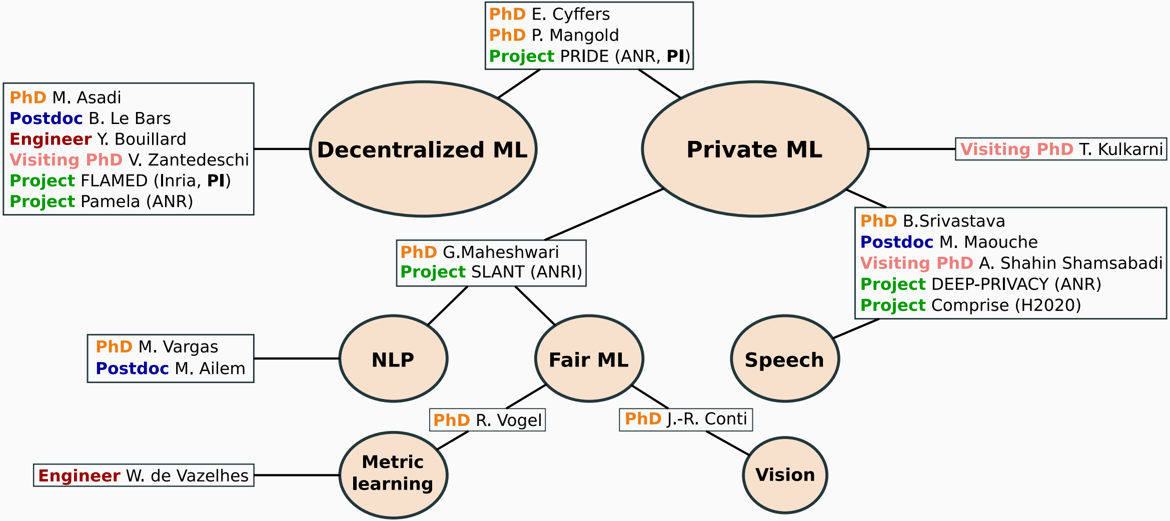
$\longrightarrow$ PhD of Paul Mangold

- Rich signals like speech, images, and text embed various types of information
- We typically want to protect specific modalities (e.g., personal attributes of the writer) while fully retaining others (e.g., the meaning of the text)
- Formal notions like DP are necessary to get clear guarantees, but need to be relaxed and combined with techniques from representation learning and signal processing

$\rightarrow$ PhD of Gaurav Maheshwari (text), PhD of Jean-Rémy Conti (images)

THANK YOU FOR YOUR ATTENTION!

SUMMARY OF PROJECTS AND SUPERVISION ACTIVITIES



REFERENCES I

[Bellet et al., 2020] Bellet, A., Guerraoui, R., and Hendrikx, H. (2020).
Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols.
In *DISC*.

[Bellet et al., 2018] Bellet, A., Guerraoui, R., Taziki, M., and Tommasi, M. (2018).
Personalized and Private Peer-to-Peer Machine Learning.
In *AISTATS*.

[Bellet et al., 2021] Bellet, A., Kermarrec, A.-M., and Lavoie, E. (2021).
D-Cliques: Compensating for Data Heterogeneity with Topology in Decentralized Federated Learning.
Technical report, arXiv:2104.07365.

[Cyffers and Bellet, 2020] Cyffers, E. and Bellet, A. (2020).
Privacy Amplification by Decentralization.
Technical report, arXiv:2012.05326.

[Dwork et al., 2006] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
Calibrating noise to sensitivity in private data analysis.
In *Theory of Cryptography (TCC)*.

[Feldman et al., 2018] Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. (2018).
Privacy Amplification by Iteration.
In *FOCS*.

[Jelasity et al., 2007] Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., and van Steen, M. (2007).
Gossip-based peer sampling.
ACM Trans. Comput. Syst., 25(3).

[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.

[Mangold et al., 2021] Mangold, P., Bellet, A., Salmon, J., and Tommasi, M. (2021).
Differentially Private Coordinate Descent for Composite Empirical Risk Minimization.
Technical report, arXiv:2110.11688.

[Paige et al., 2020] Paige, B., Bell, J., Bellet, A., Gascón, A., and Ezer, D. (2020).
Reconstructing Genotypes in Private Genomic Databases from Genetic Risk Scores.
In *International Conference on Research in Computational Molecular Biology RECOMB.*

[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.

[Shokri et al., 2017] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017).
Membership Inference Attacks Against Machine Learning Models.
In *IEEE Symposium on Security and Privacy (S&P).*

[Zantedeschi et al., 2020] Zantedeschi, V., Bellet, A., and Tommasi, M. (2020).
Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs.
In *AISTATS.*