Master MVA

Responsible Machine Learning

Nicolas Vayatis

Lecture on Privacy / technical aspects

What privacy means in data analysis

- Consider a database and a user who makes queries on the database and receives answers.
- Suppose information about Zorro can be found in the database.
- Protecting the privacy of Zorro means the user should not learn anything new about Zorro she does not already know.
- If the user may learn something about him then it should be some general characteristic of the whole population.

The flaws of privacy-preserving data analysis

But... what if the purpose of the user is to segment the population wrt to credit risk or health?

- Then, in order not to unveil the risk status of Zorro, the user should:
  - either not know Zorro belongs to the database!
  - or she should not have access to the features driving the classifier or risk score!
- Two strategies arise:
  - Anonymization
  - Summary statistics

Are those two strategies safe? Well...

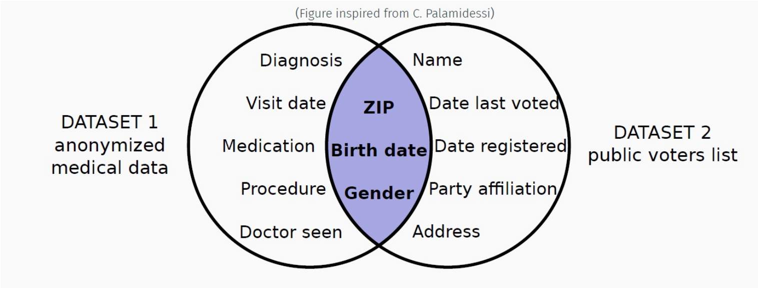
Reported cases of privacy leaks

- Data leakage in 2020 (at Q3)
  - 2,935 publicly reported breaches
  - 36 billion records exposed
  - Among which: Facebook, Instagram, Microsoft, TikTok, Google Cloud Server, etc.
- Data breaches with anonymized data by linkage between different but overlapping databases
  - AOL search data leak (2006)
  - Netflix prize (2007-2009)

Ref. Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In IEEE Symposium on Security and Privacy.

The limits of simple ideas

Anonymization is not safe due to linkage



Indeed: 87% of the US population can be identified based on ZIP/BD/Gender!

Anonymization is not safe due to linkage on nonsensitive data!

Let U and V be feature vectors of nonsensitive data, S a feature vector of sensitive data. Assume:

- Database #1 contains (private) data (ID, U, S)
- Database #1a contains (public) anonymized data (U, S)
- Database #2 contains public data (ID, U, V)

Then:

- If U is unique, then ID may be linked to S from DB #1a and DB #2
- The larger the dimension of U (and/or the smaller the sample size), the more likely U will be unique

Summary statistics are not safe!

Two types of threats:

1. Differential attacks by querying the data set

   Example: average performance of a group of people before and after a new member joins...

2. Membership inference attacks

   Contingency tables or test statistics can actually lead to recover the identity of an individual if the data set is not too large.

   Example: Intensive research in the field of Genome-wide association studies (GWAS) [Homer et al. (2008), Wang et al. (2009), Sei and Ohsuga (2021)]

Privacy in Machine Learning

Privacy at risk with Machine Learning

- ML algorithms are prone to membership inference and variants
  - An attack is made to determine whether a subject belongs to a training data set.
  - If successful, it becomes possible to infer individual information: e.g. participating to a clinical study can thus unveil the fact that the patient was treated in a certain hospital for a given disease.
- Being prone to membership inference attacks increases the risk for ML algorithms outcome to be classified as personal data under the GDPR.

  Shokri et al. (2017): Membership inference attacks against machine learning models
  Hu et al. (2021): Membership Inference Attacks on Machine Learning: A Survey

Privacy vs. Accuracy vs. Sample size

- If sample size is small, one cannot achieve both privacy and accuracy
- To achieve accuracy, need many features which will eventually identify the individual if the data set is small

N.B.: large/small sample size should be discussed wrt dimension

(Regularized) Empirical Risk Minimization

- Mother of global Machine Learning procedures: Optimization of a risk functional formed by the sum of a data-fitting term and a penalty (regularizer):

$$\widehat{f} \in \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i)) + \lambda \, \mathrm{pen}(f) \right\}$$

- In shallow learning: most algorithms boil down to an optimization problem with explicit penalty
- In the case of deep learning: no explicit regularization ($\mathrm{pen}(f) = 0$) but regularization operates through SGD and operators linking successive layers of computation

Private Empirical Risk Minimization
1. Data perturbation

- Same procedure, perturbed data:

$$\widehat{f}^D \in \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(\widetilde{Y}_i, f(\widetilde{X}_i)) + \lambda \ \mathrm{pen}(f) \right\}$$

- Example: *k-anonymity* (Sweeney, 2002)
  - Define a set of attributes as quasi-identifiers
  - Suppress/generalize attributes and/or add dummy records to make every record in the dataset indistinguishable from at least $k-1$ other records with respect to quasi-identifiers

k-anonymity example

Name	Birth date	Zip code	Gender	Diagnosis	...
Ewen Jordan	1993-09-15	13741	M	Asthma	...
Lea Yang	1999-11-07	13440	F	Type-1 diabetes	...
William Weld	1945-07-31	02110	M	Cancer	...
Clarice Mueller	1950-03-13	02061	F	Cancer	...

Name	Birth date	Zip code	Gender	Diagnosis	...
	1993-09-15	13741	M	Asthma	...
	1999-11-07	13440	F	Type-1 diabetes	...
	1945-07-31	02110	M	Cancer	...
	1950-03-13	02061	F	Cancer	...

	Quasi identifiers			Sensitive attribute	
Name	Age	Zip code	Gender	Diagnosis	...
	20-30	13***		Asthma	...
	20-30	13***		Type-1 diabetes	...
	70-80	02***		Cancer	...
	70-80	02***		Cancer	...

Question: pros/cons?

Private Empirical Risk Minimization
2. Output perturbation

- Same procedure, change decision rule: $\widehat{f}^{O} = T(\widehat{f})$ where

$$\widehat{f} \in \operatorname*{arg\,min}_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i)) + \lambda \ \mathrm{pen}(f) \right\}$$

- Example: Global Sensitivity Method also referred to as Laplace or Gaussian mechanisms (Dwork et al., 2006)

Global Sensitivity Method

- Assume D and D' are databases which differ by one record
- Let a function Q (query, statistic) based on D or after training based on D then the global sensitivity of Q is given by:

$$S(Q) = \sup_{D,D'} |Q(D) - Q(D')|$$

- Laplace Mechanism: consider the output given by:

$$Q(D) + Z \ , \quad \text{where} \quad Z \sim \frac{S(Q)}{\varepsilon} Lap(0,1)$$

Notation: $Lap(0,1)$ is a centered Laplace distribution with density $p(u) = (1/2)\exp(-|u|)$

Question: why Laplace?

Simple example: "private" mean

- Assume we have a single feature bounded in $[0, 1]$ in the database D of size n and $Q(D) = \bar{D}$
- Then the global sensitivity $S(Q)$ of Q equals $1/n$
- Then the Laplace mechanism offers an output perturbation of the form $Q(D) + Z$ where

$$Z \sim \frac{1}{n\varepsilon} \mathit{Laplace}(0, 1)$$

Other example: linear SVM case

- Consider the following inference principle:

$$\widehat{w} \in \operatorname*{arg\,min}_{w \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i w^T X_i) + \frac{\lambda}{2} \|w\|^2 \right\}$$

  with ℓ convex

- Pseudocode for private version

Algorithm 1 Private linear SVM with output perturbation

Input: training data $\{(X_i, Y_i) : i = 1, \ldots, n\}$, privacy parameter ε, amount of regularization λ
Solve raw optimization problem to get $\widehat{w}$
Draw $Z = z$ according to $\mathbb{P}\{Z = z\} \propto e^{-\varepsilon \|z\|}$
return Compute $\widetilde{w} = \widehat{w} + \frac{z}{n\lambda}$

Private Empirical Risk Minimization
3. Risk perturbation

- Same procedure, change risk criterion:

$$\widehat{f}^R \in \arg\min_{f \in \widetilde{\mathcal{F}}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i)) + \lambda \, \widetilde{\mathrm{pen}}(f) \right\}$$

- Example: Private SVM with finite feature maps (Rubinstein et al., 2009)

Private SVM - second version

- Main ingredients:
  - Random and finite feature map and induced kernel
  - Dual optimization solver
  - Laplace mechanism

- Pseudocode

Algorithm 2 Private linear SVM with objective perturbation

Input: training data $\{(X_i, Y_i) : i = 1, \dots, n\}$, convex loss ℓ, parameter ε, amount of regularization λ, finite feature map $\Phi : \mathbb{R}^d \to \mathbb{R}^F$ and induced kernel

Solve dual optimization problem to get $\widetilde{\alpha}$ based on induced kernel

Compute $\widetilde{w} = \sum_{i=1}^{n} \widetilde{\alpha}_i Y_i \Phi(X_i)$

Draw IID sample $Z = z$ from Laplace distribution $(0, \lambda)$

return Compute $\widetilde{w}^{\mathbb{R}} = \widetilde{w} + z$

Private Empirical Risk Minimization
4. Algorithm perturbation

- Same procedure, change algorithm:

$$\widehat{f}^A \in \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i)) + \lambda \operatorname{pen}(f) \right\}$$

- Example: Private SGD (Abadi et al. (2016), Song et al. (2013))

Non-private SGD

$$J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, (\mathbf{x}_i, y_i)) + \lambda R(\mathbf{w})$$

- select a random data point
- take a gradient step

$$
\begin{aligned}
&\mathbf{w}_0 = \mathbf{0} \\
&\text{For } t = 1, 2, \ldots, T \\
&\quad i_t \sim \mathsf{Unif}\{1, 2, \ldots, n\} \\
&\quad \mathbf{g}_t = \nabla \ell(\mathbf{w}_{t-1}, (\mathbf{x}_{i_t}, y_{i_t})) + \lambda \nabla R(\mathbf{w}_{t-1}) \\
&\quad \mathbf{w}_t = \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta_t \mathbf{g}_t) \\
&\hat{\mathbf{w}} = \mathbf{w}_T
\end{aligned}
$$

Private SGD with noise

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, (\mathbf{x}_i, y_i)) + \lambda R(\mathbf{w})$$

$\mathbf{w}_0 = \mathbf{0}$

For $t = 1, 2, \ldots, T$

$\quad i_t \sim \text{Unif}\{1, 2, \ldots, n\}$

$\quad \mathbf{z}_t \sim p_{(\varepsilon,\delta)}(\mathbf{z})$

$\quad \hat{\mathbf{g}}_t = \mathbf{z}_t + \nabla \ell(\mathbf{w}_{t-1}, (\mathbf{x}_{i_t}, y_{i_t})) + \lambda \nabla R(\mathbf{w}_{t-1})$

$\quad \mathbf{w}_t = \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta_t \hat{\mathbf{g}}_t)$

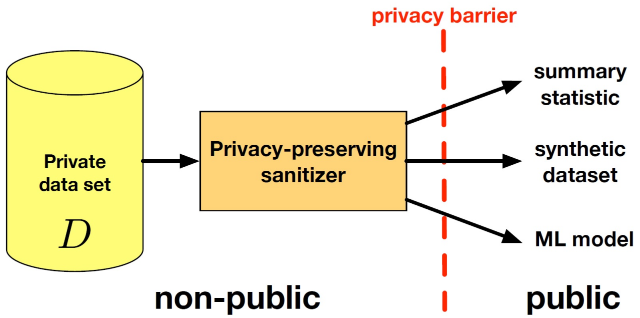
$\hat{\mathbf{w}} = \mathbf{w}_T$

- select random data point
- add noise to gradient

[SCS15]

Differential privacy

Property of Sanitizer



Aggregate information computable

Individual information protected
(robust to side-information)

Differential Privacy

Data +  → Algorithm → Outcome

Data +  → Algorithm → Outcome

Participation of a person does not change outcome

Definition of differential privacy
Dwork, McSherry, Nissim, Smith (2006)

- Consider $A(S)$ where A is a *randomized* algorithm operating on a data set S
- Let S' be a data set which differs from S by one data point.
- We consider that the randomized algorithm will satisfy differential privacy at level ε (privacy loss) if the following loglikelihood ratio is uniformly bounded over S; S' and B :

$$\sup_{B} \sup_{S,S'} \left| \log \left(\frac{P(A(S) \in B)}{P(A(S') \in B)} \right) \right| \leq \varepsilon$$

Check theorems for Private SVM

- We refer to Chaudhuri et al. (2011) or Rubinstein et al. (2009)
- Under some assumptions, differential privacy is guaranteed with some ε

Discussion and further topics related to privacy

Some names on differential privacy

- Cynthia Dwork (Harvard) - 2014 book on "The Algorithmic Foundations of Differential Privacy" (with Aaron Roth)
- Helen Nissenbaum (Cornell Tech)
- Catuscia Palamidessi (INRIA, France) - book and Master course about Foundations of Privacy
- Kamalika Chaudhuri* (UCSD) - NIPS 2017 tutorial
- Aurélien Bellet* (INRIA, France) - Master course on Privacy Preserving Machine Learning

*more ML flavor in their research

Check workskop series at the Simons Foundation on "Data Privacy: Foundations and Applications" - Jan. 15 – May 17, 2019

Typical expected guarantees of privacy-preserving methods (Dwork, 2014)

- future-proof (side information, post-processing)
- group privacy
- permanence through composition
- programmable

Further topics

- Regulatory - How to account for privacy?
- Implementation - Where to place sanitizers along a pipeline? How to deal with privacy during the data exploration stage? Can deep learning preserve the privacy of all its parameters and still generalize?
- Under constraints - How to optimize privacy budget along several stages ?

  preprocessing/ training/ cross-validation/ testing/ hyperparameter calibration
- Resilience to attacks

Distributed Machine Learning

Why looking for alternative to centralized learning?

- Latency (IoT, multiplication of data sources, sensors, etc.)
- Privacy
- Jurisdiction (data considered too sensitive to be merged)
- Knowledge sharing ("Winner-Takes-All" effect)

Distributed learning

Generally, the output is a parameter, a gradient or a prediction

- Goal: Estimate the output by optimizing computing power through distributed optimization
- Assumption 1: Data are collected at the **server** level
- Assumption 2: Data are **equally** split between nodes (machines)
- Final estimate: Aggregation of local estimates by the central server
- Main setups: *One-shot* or *Multi-round* (e.g. stochastic gradient descent)

Federated learning

- Goal: estimate a common output over multiple nodes (denominated devices or clients) without having access to data, enhancing privacy (Important warning: the output can leak information e.g. memorization property of large models)
- Assumption 1: Data are collected at the **node** level
- Assumption 2: Nodes do not communicate any observation data neither to the central server nor between them, but do transmit their estimate of the output
- Final estimate: Aggregation of local estimates by the central server

Challenges of federated learning

Federated optimization aims at handling data with the following properties:

- In the cross-device setting (nodes stand for devices/people): massively distributed counter to distributed learning assumptions or cross-silo setting (nodes stand for institutions/entities), the number of nodes, m, can be very large and can be much larger than the sample size per node.
- Non-*i.i.d.* (e.g. algorithm SCAFFOLD or personalization)
- Unbalanced *i.e.* sample size per node with considerable order of variations.
- In the cross-device setting: limited communications, with frequently unavailable nodes.

Algorithm 3 FedAvg [McMahan et al., 2017]

Initialize model parameter θ_0 and round $t = 0$
for each round $t = 0, T$ **do**
  randomly generate $\mathcal{S}_t$, a subset of all nodes of size $\lfloor Cm \rfloor$
  for each node $j \in \mathcal{S}_t$ **do**
    $\theta^j_{t+1} = \text{NodeUpdate}(j, \theta_t)$
  end for
  $\theta_{t+1} = \sum_j w_j \theta^j_{t+1}$ with w_j proportional to the sample size
  $t = t + 1$
end for
return θ_{t+1}

Algorithm 4 NodeUpdate(j, θ)

Require: η, f
$\mathcal{B}$ = data of client j splitting in batches of size B
for each epoch $e \in 1..E$ **do**
  for each batch $b \in \mathcal{B}$ **do**
    $\theta = \theta - \frac{\eta}{B} \nabla f(\theta, b)$
  end for
end for
return θ

Experimental results

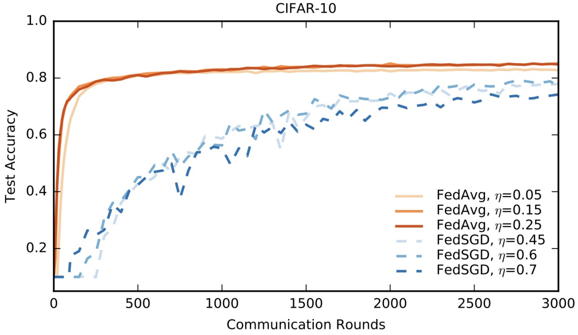


Figure 4: Test accuracy versus communication for the CIFAR10 experiments. `FedSGD` uses a learning-rate decay of 0.9934 per round; `FedAvg` uses $B = 50$, learning-rate decay of 0.99 per round, and $E = 5$.

References

Advances and Open Problems in Federated Learning

Peter Kairouz[7*] H. Brendan McMahan[7*] Brendan Avent[21] Aurélien Bellet[9]
Mehdi Bennis[19] Arjun Nitin Bhagoji[13] Kallista Bonawitz[7] Zachary Charles[7]
Graham Cormode[23] Rachel Cummings[6] Rafael G.L. D'Oliveira[14]
Hubert Eichner[7] Salim El Rouayheb[14] David Evans[22] Josh Gardner[24]
Zachary Garrett[7] Adrià Gascón[7] Badih Ghazi[7] Phillip B. Gibbons[2]
Marco Gruteser[7,14] Zaid Harchaoui[24] Chaoyang He[21] Lie He [4]
Zhouyuan Huo [20] Ben Hutchinson[7] Justin Hsu[25] Martin Jaggi[4] Tara Javidi[17]
Gauri Joshi[2] Mikhail Khodak[2] Jakub Konečný[7] Aleksandra Korolova[21]
Farinaz Koushanfar[17] Sanmi Koyejo[7,18] Tancrède Lepoint[7] Yang Liu[12]
Prateek Mittal[13] Mehryar Mohri[7] Richard Nock[1] Ayfer Özgür[15]
Rasmus Pagh[7,10] Hang Qi[7] Daniel Ramage[7] Ramesh Raskar[11]
Mariana Raykova[7] Dawn Song[16] Weikang Song[7] Sebastian U. Stich[4]
Ziteng Sun[3] Ananda Theertha Suresh[7] Florian Tramèr[15] Praneeth Vepakomma[11]
Jianyu Wang[2] Li Xiong[5] Zheng Xu[7] Qiang Yang[8] Felix X. Yu[7] Han Yu[12]
Sen Zhao[7]

[1]Australian National University, [2]Carnegie Mellon University, [3]Cornell University,
[4]École Polytechnique Fédérale de Lausanne, [5]Emory University, [6]Georgia Institute of Technology,
[7]Google Research, [8]Hong Kong University of Science and Technology, [9]INRIA, [10]IT University of Copenhagen,
[11]Massachusetts Institute of Technology, [12]Nanyang Technological University, [13]Princeton University,
[14]Rutgers University, [15]Stanford University, [16]University of California Berkeley,
[17] University of California San Diego, [18]University of Illinois Urbana-Champaign, [19]University of Oulu,
[20]University of Pittsburgh, [21]University of Southern California, [22]University of Virginia,
[23]University of Warwick, [24]University of Washington, [25]University of Wisconsin–Madison

https://federated-learning.org/