Membership Inference Attacks
Making the most out of the white-box setting

Nathan Bigaud

15/09/2022

Outline

Context and notations

Membership inference attacks

Our contributions

References

Internship overview

Setting	Five months internship at the Magnet team of INRIA Lille
Topic	How access to detailed information about a model could improve existing privacy attacks
Main contribution	A proof-of-concept adaptation of a state-of-the-art privacy attack

Motivation

Privacy attacks on machine learning models aim to infer information about individual data points used during training

Recent work has shown that commonly used models are vulnerable to major privacy breaches

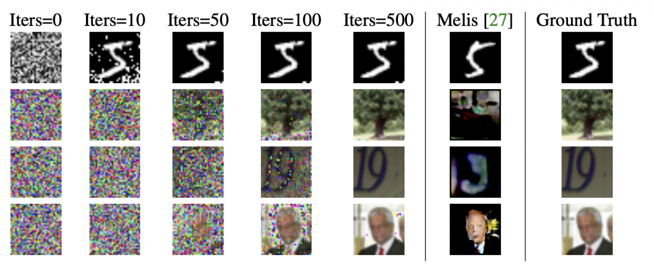


Figure 1 – Deep leakage results on images (Geiping et al. 2020)

Understanding of these attacks enables privacy auditing and defense design

Notations

We consider a classification problem :

- **A model** $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized with weights θ
- **An i.i.d. sample** D from some underlying distribution $\mathbb{D}$: $D = z_{1:n}$
- n **binary membership variables** associated with each z_i, noted m_i
- **A training algorithm** $\mathcal{T}$, $f_\theta \leftarrow \mathcal{T}(D)$

f_θ **is trained on a subset of** D, determined by $m_{1:n}$:

$$\theta_{t+1} \leftarrow \theta_t - \eta \sum_{i=1}^{n} \nabla_\theta \ell(z_i) * m_i$$

An adversary $\mathcal{A}$ tries to infer information on a point z_1 given some information about the model, denoted abstractly by $I_{z_1}(f_\theta)$

The membership inference game

We focus on membership inference : $\mathcal{A}$ tries to determine whether a point z_1 was part of the training set of f_θ.

- $\mathcal{A}$ **is given access** to D and $I_{z_1}(f_\theta)$
- **They try to estimate** $\mathcal{A}\left(I_{z_1}(f_\theta), z_1\right) := \mathbb{P}\left(m_1 = 1 \mid I_{z_1}(f_\theta), z_1\right)$

The information available to the adversary varies between attack settings :

- **In a black-box setting**, the adversary's observation is limited to the output of the model : $I_{z_1}(f_\theta) = f_\theta(z_1)$
- **In a white-box setting,** the attacker obtains full read access to the model and its training history : $I_{z_1}(f_\theta) = \{\theta_t\}_{1 \leq t \leq T}$

The typical MIA approach (1/3)

The typical approach to build a concrete attack for z_1 is to use $I_{z_1}(f_\theta)$ to estimate :

$$\begin{cases} \mathbb{Q}_{in}(z_1) := \{f \leftarrow \mathcal{T}(D \cup z_1)) \mid D \leftarrow \mathbb{D}\} \\ \mathbb{Q}_{out}(z_1) := \{f \leftarrow \mathcal{T}(D \backslash z_1)) \mid D \leftarrow \mathbb{D}\} \end{cases}$$

By :

$$\begin{cases} \hat{\mathbb{Q}}_{in}(z_1) := \{I_{z_1}(f) \mid f \leftarrow \mathcal{T}(D \cup z_1)), D \leftarrow \mathbb{D}\} \\ \hat{\mathbb{Q}}_{out}(z_1) := \{I_{z_1}(f) \mid f \leftarrow \mathcal{T}(D \backslash z_1)), D \leftarrow \mathbb{D}\} \end{cases}$$

The typical MIA approach (2/3)

$\mathcal{A}$ **uses information about models similar to** f as proxy to estimate $\mathbb{Q}_{in/out}(z_1)$:

- $\mathcal{A}$ **trains** N **models** $f^{1:N}$ with the same architecture as f
- **Each** f^i **is trained on a random subset of** D, determined by $m^i_{1:n}$
- **The adversary collects** $I_{z_1}(f^i)$ for each shadow model

The typical MIA approach (3/3)

Membership inference can be done implicitly or explicitly

- **Implicitly** : $\mathcal{A}$ trains a classifier on $I_{z_1}(f^{1:N})$ with labels $m_1^{1:N}$ to discriminate between $\mathbb{Q}_{in}(z_1)$ and $\mathbb{Q}_{out}(z_1)$ (e.g. Shokri et al. 2017)
- **Explicitly** : $\mathcal{A}$ uses a parametric model of the distributions $\mathbb{Q}_{in/out}(z_1)$ and a likelihood ratio test.

$$\Lambda(I_{z_1}(f_\theta); z_1) = \frac{\mathbb{P}\left(m_1 = 1 \mid \hat{\mathbb{Q}}_{\text{in}}(z_1)\right)}{\mathbb{P}\left(m_1 = 0 \mid \hat{\mathbb{Q}}_{\text{out}}(z_1)\right)}$$

The current best attack : LIRA (Carlini et al. 2022)

1. **Query** model loss $\ell(f(x), y)$
2. **Train** N "shadow models"
   $f^i_{\text{out}} \leftarrow \text{Train}(D \backslash z), f^i_{\text{in}} \leftarrow \text{Train}(D \cup z)$
3. **Compute** losses
   $L_{\text{out}} = \left\{\ell(f^i_{\text{out}}(x), y)\right\}_{1 \leq i \leq N}$
   $L_{\text{in}} = \left\{\ell(f^i_{\text{in}}(x), y)\right\}_{1 \leq i \leq N}$
4. **Fit** Gaussians to L_{out} and L_{in}
5. **Output** likelihood ratio :

$$\Lambda = \frac{\mathbb{P}\left(\ell(f(x), y) \mid \mathcal{N}(\mu_{\text{in}}, \sigma_{\text{in}})\right)}{\mathbb{P}\left(\ell(f(x), y) \mid \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}})\right)}$$

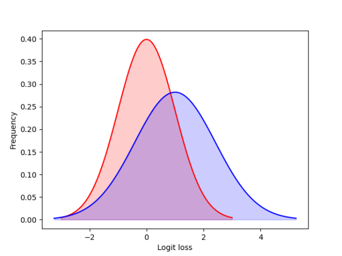


Figure 2 – Frequencies for L_{in} and L_{out}

Evaluating membership inference attacks

Carlini et al. 2022 argue attacks should be evaluated on worse-case metrics, focusing on the true positive rate at a low, fixed false positive rate.

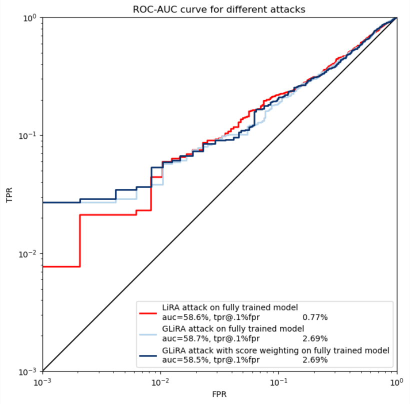


Figure 3 – Comparison of ROC-AUC curve and key metrics for LiRA and GLiRA on sparse dataset at 98% sparsity and logistic regression

Our goal : leverage the white-box setting to improve LiRA

In federated learning, clients owning private data and exchange gradients to train a central model

We attack gradients updates to focus on signals lost at the aggregation stage in the loss

(a) LiRA target distribution

(b) GLiRA target distribution

Figure 4 – Overview of idealized target distributions for LiRA vs GliRA

GLiRA : Pseudo code

1. **Query** gradient $\nabla(f(x), y)$
2. **Train** N "shadow models"

$$f^i_{\text{out}} \leftarrow \text{Train}(D \backslash z), f^i_{\text{in}} \leftarrow \text{Train}(D \cup z)$$

3. **Compute** gradients

$$\nabla_{\text{out}} = \left\{\nabla_\theta(f^i_{\text{out}}(x), y)\right\}_{1 \leq i \leq N}, \nabla_{\text{in}} = \left\{\nabla_\theta(f^i_{\text{in}}(x), y)\right\}_{1 \leq i \leq N}$$

4. **Fit** Gaussians to ∇_{out} and ∇_{in}
5. **Output** likelihood ratio [1] :

$$\Lambda = \frac{\mathbb{P}\left(\nabla_\theta(f(x), y) \mid \mathcal{N}\left(\mu_{\text{in}}, \sigma_{\text{in}}\right)\right)}{\mathbb{P}\left(\nabla_\theta(f(x), y) \mid \mathcal{N}\left(\mu_{\text{out}}, \sigma_{\text{out}}\right)\right)}$$

1. We additionally show a weighted version of that attack, using $w = |\mu_{in} - \mu_{out}|$

GLiRA : Proof of concept on D_{dirac} (1/3)

An "edge" case of loss-based attacks : datapoint with spurious features

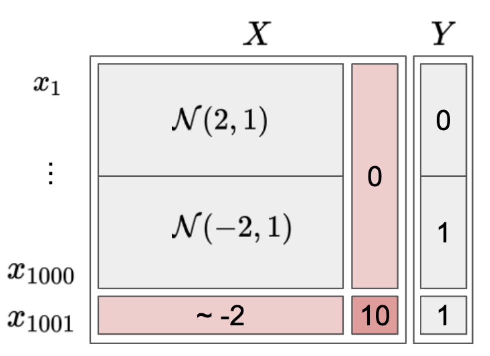


Figure 5 – Adding a target point to build D_{dirac}

GLiRA : Proof of concept on D_{dirac} (2/3)

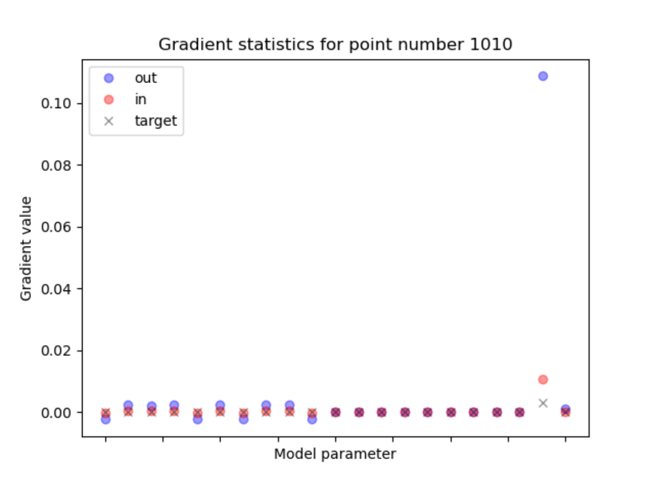


Figure 6 – Comparison of $\nabla_{in}, \nabla_{out}, \nabla_{obs}$ for point z_{1010}

GLiRA : Proof of concept on D_{dirac} (3/3)

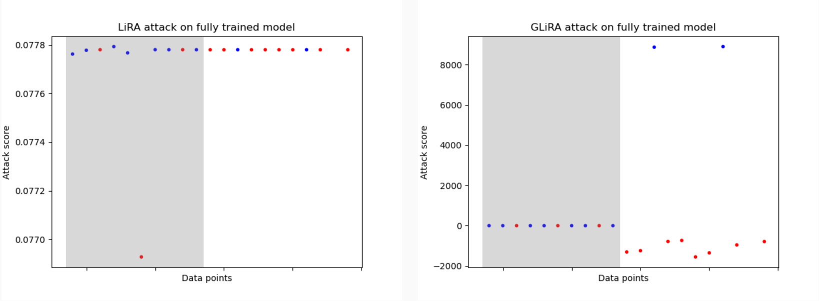


Figure 7 – Overview of scores for LiRA vs GliRA for 20 last points of D_{dirac}

	tpr@0.1%fpr	tpr@1.0%fpr
LiRA	0.12% ± 0.18%	1.09% ± 0.47%
GLiRA	1.05% ± 0.4%	**2.05%** ± 0.62%
GLiRA weighted	**1.1%** ± 0.47%	2.0% ± 0.74%

GliRA : Test on sparse dataset (1/3)

A less engineered sparse problem : D_{sparse}, sparse at 98%

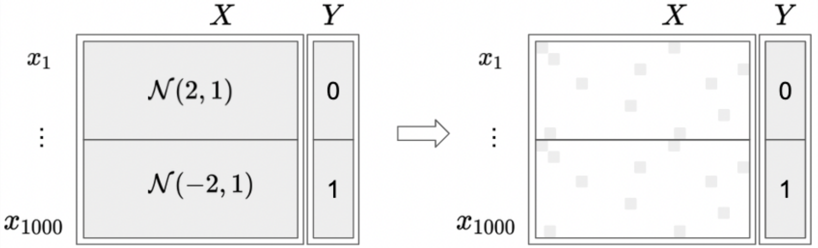


Figure 8 – Building D_{sparse}

GliRA : Test on sparse dataset (2/3)

Results at 99% sparsity on 150 runs of the attacks on different models show very unstable results and no significant gain so far

	tpr@0.1%fpr	tpr@1.0%fpr
LiRA	0.63% ± 0.6%	**3.06%** ± 1.16%
GLiRA	0.65% ± 0.55%	2.74% ± 1.07%
GLiRA weighted	**0.71%** ± 0.59%	2.93% ± 1.2%

	tpr@5.0%fpr	tpr@10.0%fpr
LiRA	**10.25%** ± 2.34%	**17.03%** ± 3.12%
GLiRA	9.34% ± 2.19%	16.05% ± 3.13%
GLiRA weighted	9.52% ± 2.17%	16.29% ± 3.05%

GliRA : Test on sparse dataset (3/3)

Results at 95% sparsity on 150 runs of the attacks on different models show very unstable results and no significant gain so far

	tpr@0.1%fpr	tpr@1.0%fpr
LiRA	**0.59%** ± 0.57%	2.91% ± 1.11%
GLiRA	0.53% ± 0.54%	2.74% ± 1.06%
GLiRA weighted	**0.59%** ± 0.54%	**2.96%** ± 1.02%

	tpr@5.0%fpr	tpr@10.0%fpr
LiRA	**11.29%** ± 1.93%	19.65% ± 2.59%
GLiRA	10.55% ± 1.86%	18.99% ± 2.64%
GLiRA weighted	11.21% ± 1.9%	**19.74%** ± 2.61%

Open questions

On the short-term :

- **Investigate high instability** and unexplained behaviours of our attacks
- **Investigate impact of problem setting**, e.g. sparsity, learning schedules, feature scale
- **Justify** the choice of attacking the gradient over other high-dimension signals
- **Explore exploiting prior knowledge** on feature distribution in sparse datasets (see Von Thenen, Ayday et Cicek 2019)

Longer term, extend analysis to more complex datasets and models, e.g. genomics data used by Homer et al. 2008, or MNIST as used in Carlini et al. 2022.

References i

Carlini, Nicholas et al. (2022). "Membership inference attacks from first principles". In : *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, p. 1897-1914.

Geiping, Jonas et al. (2020). "Inverting Gradients–How easy is it to break privacy in federated learning ?" In : *arXiv preprint arXiv :2003.14053*.

Homer, Nils et al. (2008). "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". In : *PLoS genetics* 4.8, e1000167.

Sablayrolles, Alexandre et al. (2019). "White-box vs black-box : Bayes optimal strategies for membership inference". In : *International Conference on Machine Learning*. PMLR, p. 5558-5567.

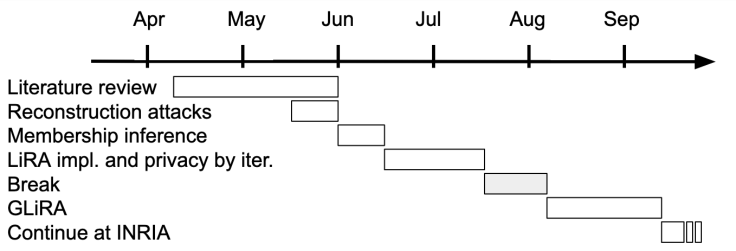
Shokri, Reza et al. (2017). "Membership inference attacks against machine learning models". In : *2017 IEEE symposium on security and privacy (SP)*. IEEE, p. 3-18.

Von Thenen, Nora, Erman Ayday et A Ercument Cicek (2019). "Re-identification of individuals in genomic data-sharing beacons via allele inference". In : *Bioinformatics* 35.3, p. 365-371.

Thank you for your attention!

Internship process and timeline

This internship was done over the course of five months at the Magnet team of INRIA Lille, under the supervision of Aurélien Bellet and Marc Tommasi.



Our work started on a very open theme, namely privacy threats in the federated learning (FL) setting, with an important exploratory aspect.

After iterating over several research themes, we focused on experimenting with a new white-box membership inference attack (MIA)

Federated learning is a privacy-focused learning setting

Federated Learning (FL) is a machine learning setting where N clients collaboratively train a model f_θ under the orchestration of a central server, while keeping the training data decentralized.

The main goal of this setting is to improve privacy, which is achieved by only exchanging minimal information on how to improve that central model - e.g., gradient updates.

If each N users own a part $\mathbf{z}_n = \{\mathbf{x}_n, \mathbf{y}_n\}$ of the data, a simple learning algorithm for the central server is :

$$\theta_{t+1} \leftarrow \underbrace{\theta_t - \eta \sum_{n=1}^{N}}_{\text{server}} \underbrace{\nabla_\theta \ell\left(f_{\theta_t}(\mathbf{x}_n), \mathbf{y}_n\right)}_{\text{users}}$$

We aimed at investigating how the FL settings opens new avenues for privacy attacks, as the granularity of information being shared between servers provides new ways to breach privacy.

"Black-box attacks are as good as white-box attacks" (Sablayrolles et al. 2019)

Sablayrolles et al. 2019 consider a case where $I(f_\theta) = \theta$ but make a key assumption : taking T as a temperature parameter, controlling the stochasticity of θ :

$$\mathbb{P}\left(\theta \mid z_1, \ldots, z_n, m_1, \ldots, m_n\right) \propto e^{-\frac{1}{T}\sum_{i=1}^{n} m_i \ell\left(\theta, z_i\right)}$$

From this assumption they derive the optimal strategy for membership inference using an important assumption on the distribution of the parameters. Attacking z_1, noting $\tau = \{z_2, \ldots, z_n, m_2, \ldots, m_n\}$, and $\lambda = \mathbb{P}_D\left(m_i = 1\right)$:

$$\mathcal{A}\left(\theta, z_1\right) = \mathbb{E}_\tau\left[\sigma\left(\log\left(\frac{\mathbb{P}\left(\theta \mid m_1 = 1, z_1, \tau\right)}{\mathbb{P}\left(\theta \mid m_1 = 0, z_1, \tau\right)}\right) + t_\lambda\right)\right] \text{ with } t_\lambda = \log\left(\frac{\lambda}{1-\lambda}\right)$$