Membership Inference

Membership Inference Attacks

Recall: data reconstruction attacks
"blatantly non-private" algorithms
↳ show that something is not private

Defeating reconstruction attacks $\not\Rightarrow$ privacy

A quick detour through crypto

Key recovery vs IND-CPA security
indistinguishability

↳ clearly insecure

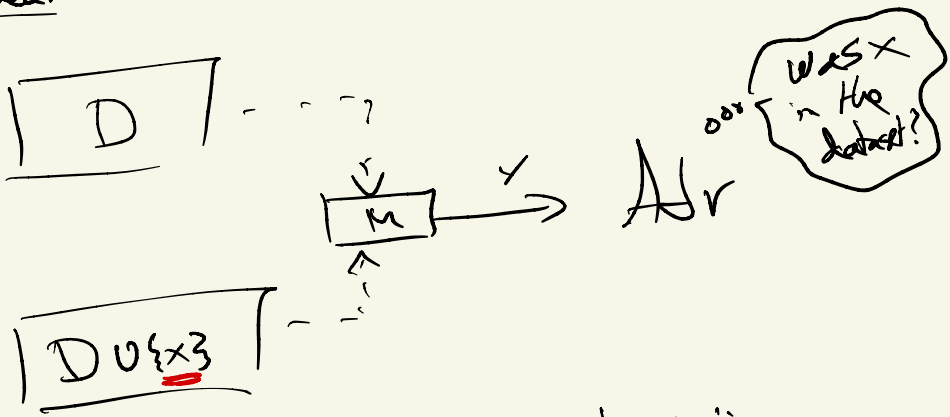
↳ easy to create schemes where key recovery is impossible but are clearly insecure

E.g. $Enc(k, m) := m$

or $:= DES(k_{1:n/2}, m)$

Differential privacy $\overset{\text{N}}{\text{N}}$ hardness of inferring membership

Experiment



- "Membership inference attack"
- "Tracing attacks"

A formal security game for MI attacks

Adversary — Challenger

D, x →

$D_0 = D$, $D_1 = D \cup \{x\}$

$b \xleftarrow{\$} \{0,1\}$

← $M(D_b)$

output b'

Adv wins if $b' = b$

Quantifying Adv's success

1) $Adv_{MI}(A) = 2\left|Pr[b' = b] - 1/2\right|$

in crypto this is fine because we want

$Adv(A) = negl(\lambda)$

With DP guarantees, we can never get $negl(\lambda)$ advantage (if we also want utility)

2) Confusion matrix:

$TPR = Pr[b' = 1 \mid b = 1]$

$FPR = Pr[b' = 1 \mid b = 0]$

$TNR = Pr[b' = 0 \mid b = 0]$

$FNR = Pr[b' = 0 \mid b = 1]$

Connection between DP and MI:

Theorem [KOV'15]. Let $\varepsilon > 0$, $\delta \in [0,1]$

A mechanism M is (ε, δ)-DP iff

$$e^{\varepsilon} \cdot FNR \geq TNR - \delta \quad \text{and}$$
$$e^{\varepsilon} \cdot FPR \geq TPR - \delta$$

for any adversary in the MI game

Observations:

DP protects against something much stronger than reconstruction (namely MI)

A mechanism M is (ε, δ)-DP iff

$$e^{\varepsilon} \cdot FNR \geq TNR - \delta \quad \text{and} \quad (*)$$
$$e^{\varepsilon} \cdot FPR \geq TPR - \delta$$

Proof 1) (ε, δ) DP $\Rightarrow$ $(*)$

For simplicity, assume that Adv is deterministic

Let S be the set of outputs of M for which Adv outputs $b' = 0$.

By DP: $\Pr[M(D_0) \in S] \leq e^{\varepsilon} \Pr[M(D_1) \in S] + \delta$

$\Pr[M(D_0) \in S]$ = TNR

$\Pr[M(D_1) \in S]$ = FNR

by symmetry, we can get the same for TPR, FPR

2) $(*) \Rightarrow (\varepsilon, \delta)$-DP

Assume that $\forall S \subseteq Y$, if the attacker outputs $b' = 0$ whenever M's output is in S, then $(*)$ holds

$$\Rightarrow \Pr[M(D_0) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D_1) \in S] + \delta$$

and if we assume that all sets $S \subseteq Y$ where A_{DV} outputs $b' = 1$ whenever M's output is in S,

$$\Pr[M(D_1) \in S] \leq e^{\varepsilon} \Pr[M(D_0) \in S] + \delta$$

$\Rightarrow$ This is exactly (ε, δ)-DP

Corollary A mechanism is (ε, δ)-DP iff for all adversaries in the MI game we have:

$$e^{\varepsilon} \geq \max\left(\frac{TNR - \delta}{FNR}, \frac{TPR - \delta}{FPR}\right)$$

example: if there exists an attacker with a TPR = 90% at a FPR of 1% then the mechanism is not ε-DP for $\varepsilon \leq \ln 90 \approx 4.5$

Interpreting / applying MI:

1) Actual attack (goal of Adv: infer whether someone participated in some "sensitive" statistical experiment)

this is a rather weak attack

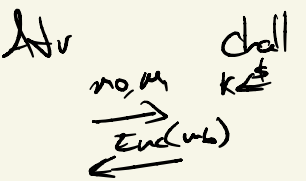
⇒ what we assume: Adv knows all the data about x

⇒ what we get: Adv learns one bit

2) lower-bounding / auditing DP

⇒ if we can find a MI attack, we know the system is not DP (for some ε budgets)

Analogy: Chosen plaintext attack



↳ in practice, we don't really care about an attacker learning b

↳ Why we care: if such an attack exists, the scheme is not secure

From MI to lower bounds on DP

Recall: lower bounds from reconstruction attacks

if Dinur Nissim attacks work $\Rightarrow$ scheme is not DP

We know: you can't answer n counting queries with $o(1)$ noise

↳ check your HW!

Now: a similar (but tighter) result for the Gaussian mechanism

what we know: if we add noise $\tau = \tilde{O}(\sqrt{k}/n)$ to each query, we can answer k counting queries with (ε, δ)-DP (Gaussian mechanism)

$= \{\vec{x}_1, \dots, \vec{x}_n\} \in \{0,1\}^{n \times k}$

Theorem: Let $M(D) = \frac{1}{n}\sum_{i=1}^{n} \vec{x}_i + N(0, \tau^2 \cdot I_{k \times k})$

Assume $\tau = o(\sqrt{k}/n)$, and $\delta = o(1)$

Then M is **not** (ε, δ)-DP for any $\varepsilon = O(1)$

Proof: we'll build a MI attack against M

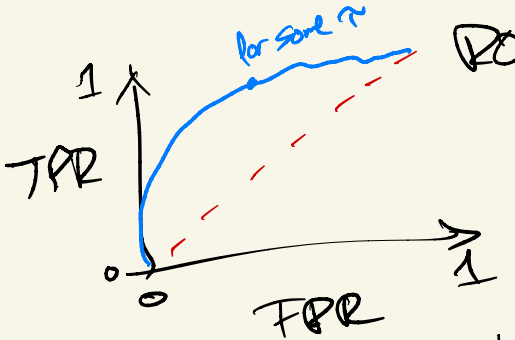
- Adv gets to pick D and $\vec{x}$

$$D_0 = \{\vec{0}, \cdots, \vec{0}\} \qquad \vec{x} = [1, 1, \cdots, 1]$$
$$D_1 = \{\vec{0}, \cdots, \vec{x}\}$$

- Adv sends $D, \vec{x}$ to the challenger and gets back $\vec{y} = M(D_b)$

$D_0 = D$ $\quad$ $D_1 = D \cup \{\vec{x}\}$

- Adv computes $z = \langle \vec{y}, \vec{x} \rangle$ and outputs $b' = 1$ iff $z \geq \tau$



Intuition: • if $b = 0$ (so $\vec{x}$ is not in the input)

then $\vec{y} = M(D_0) = N(0, \tau^2 I)$

$$\mathbb{E}[\langle \vec{y}, \vec{x} \rangle] = 0$$

• if $b = 1$, $\vec{y} = M(D_1) = \begin{bmatrix} 1/n \\ \vdots \\ 1/n \end{bmatrix} + N(0, \tau^2 I)$

$$\mathbb{E}[\langle \vec{y}, \vec{x} \rangle] = \frac{1 \times n}{n}$$

More formally:

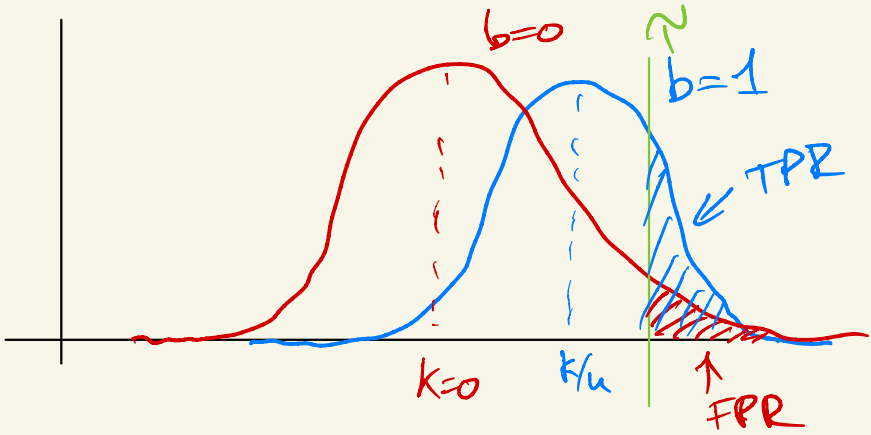
if $b=0$: $z = \begin{bmatrix} N(0,\sigma^2) \\ \vdots \\ N(0,\sigma^2) \end{bmatrix} \cdot [1, \dots, 1] = \sum_{i=1}^{k} N(0,\sigma^2)$

$= N(0, k\sigma^2)$

$\begin{bmatrix} x_{n+N} \\ \vdots \\ x_{n+N} \end{bmatrix} \cdot [1, \dots, 1]$

if $b=1$: $z = \langle \vec{y}, \vec{x} \rangle = N(0, k\sigma^2) + \langle \vec{x}_n, \vec{x} \rangle$

$= N\left(\frac{k}{n}, k\sigma^2\right)$



let's analyze the attack:

set $\tau = k/n$

- $TPR = \Pr[z \geq \tau \mid b=1] = 50\%$

by symmetry of Gaussian

- upper bound FPR

> Let $X \sim N(0, \sigma^2)$ and $t > 0$ then
> $$\Pr[X > t] \leq \exp\left(\frac{-t^2}{2\sigma^2}\right)$$

$$FPR = \Pr[z \geq \tau \mid b=0]$$
$$= \Pr[N(0, k\sigma^2) \geq k/n]$$
$$\leq \exp\left(-\frac{(k/n)^2}{2k\sigma^2}\right) \qquad \sigma = o\left(\frac{\sqrt{k}}{n}\right)$$

$$\exp\left(\underbrace{-\frac{k}{2n^2\sigma^2}}_{\omega(1)}\right) = o(1)$$

$$\exp\left(-\frac{k}{n^2\sigma^2}\right) \qquad \sigma^2 = o\left(\frac{\sqrt{k}}{n}\right)$$

$$= \exp\left(-\frac{k\, n^4}{n^2 k}\right) \qquad \text{e.g.} = \frac{\sqrt{k}}{n^2}$$

$$= \exp(-u^2) = o(1)$$

$$TPR = 50\%$$

$$FPR = o(1)$$

by Corollary: $\varepsilon \geq \ln\left(\frac{TPR - \delta}{FPR}\right)$

constant $= o(1)$; less than constant

$$= \omega(1)$$

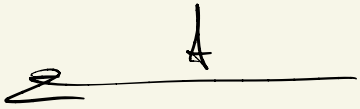
What we showed: to answer k counting queries with the Gaussian mech. and (ε, δ)-DP for $\varepsilon = o(1)$ and $\delta = o(1)$

we need $\sigma = \Omega(\sqrt{k}/n)$

and thus incur error $\Omega\left(\frac{\sqrt{k}}{n}\right)$ per query

Membership inference for machine learning

Adversary — **Challenger**



train a model f on a dataset D

guess whether x was in the training data

Deep neural networks

- $f_\theta : X \to [0,1]^C$ ← # of classes
  - θ: weights
  - X: input
- Denote $f_\theta(x)_y$ the probability of class y
- A feed-forward network:

$$f(x) = \sigma(f_K \circ \dots \circ f_2 \circ f_1(x))$$

σ: softmax; $f_K \circ \dots \circ f_2 \circ f_1(x) = z_\theta(x)$; K layers

$$z_\theta : X \to \mathbb{R}^C$$

$$\sigma(z) = \left[\frac{e^{z_1}}{\sum_i e^{z_i}}, \cdots, \frac{e^{z_c}}{\sum_i e^{z_i}} \right] \in [0,1]^c$$

softmax — logits

Training a neural network: gradient descent

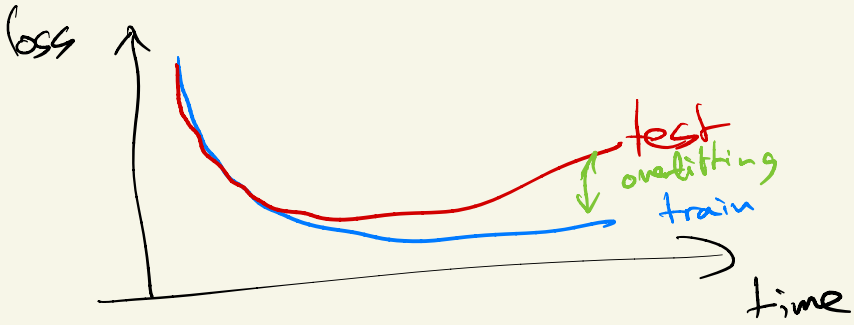
$$\theta_{t+1} \leftarrow \theta_t - \eta \sum_{(x,y) \in B} \nabla_\theta \ell(\theta_t, x, y)$$

loss, e.g. cross entropy — batch

Overfitting

$$\hat{L}_D(\theta) \leq L(\theta)$$

empirical loss — training set — true loss



How do to MI ?

$b \leftarrow \{0,1\}^{\$}$

$D_0 = D$

$D_1 = D \cup \{x\}$

θ ← $\theta \leftarrow \text{Train}(D_b)$

How to test whether θ comes from D_0 or D_1?

1) "White-box attack": look at the parameters θ and figure out if some info on x is encoded in there somehow

2) "Black-box attack": we'll just look at model outputs $f_\theta(x)$, in particular the loss $\ell(\theta, x, y)$

if $x \in D$ → we expect this loss to be $\approx \hat{L}_D(\theta, x, y)$

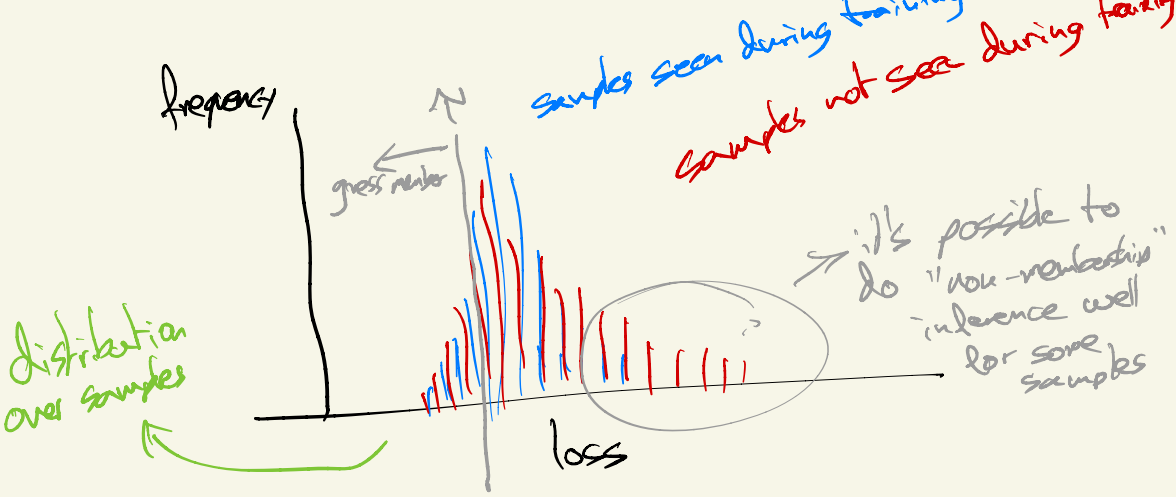
if $x \notin D$ → we expect this loss to be $\approx L_D(\theta, x, y)$

A first MI attacks: Global loss thresholding

$$A_{global}(\theta, x, y; \tau) = \begin{cases} 1 \text{ if } \ell(\theta, x, y) \le \tau \\ 0 \text{ otherwise} \end{cases}$$

θ: query access is enough

How well does this work? (Spoiler: not very well)



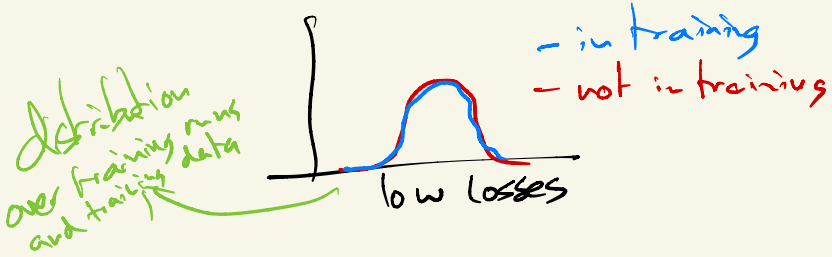
Evaluation

- Accuracy $= \Pr_{\text{over many } x\text{'s } \sim P}[b' = b] \approx 70\%$
- TPR to FPR ratio $\gg 1$ (for small FPR)

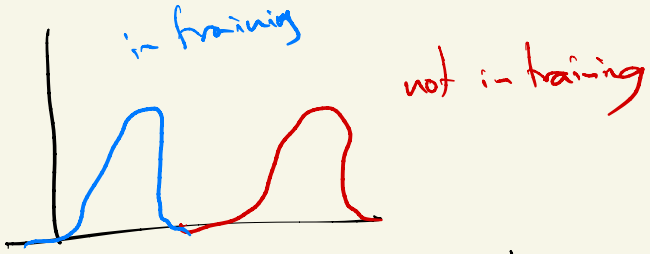
What we want: confident membership inference for some samples

Difficulty calibration

- inliers (a prototypical "cat" or "dog")



- outliers (a cat that looks like a dog)



"per-sample overfitting"

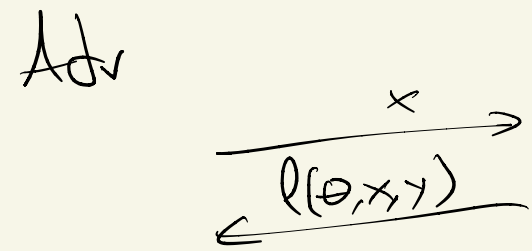
The people most at risk from privacy inference might be the ones who need privacy the most

A stronger attack: per-sample likelihood ratios

Likelihood ratio test

- suppose you have a null hypothesis H_0 ($x \notin D$) and an alternative hypothesis H_1 ($x \in D$) and you want to decide which is more likely given observed data
- likelihood ratio test: $\Lambda = \frac{\Pr[\text{data} \mid H_0]}{\Pr[\text{data} \mid H_1]}$
- Neyman-Pearson: the test "$\Lambda \geq \tau$" is the test that yields the highest TPR at any fixed FPR

How do we do a likelihood ratio test for a neural network?



we want to compute

$$\Pr_{\text{training randomness}}\left[\ell(\theta, x, y) \mid \theta \text{ was trained on } D_b\right]$$

we don't know how to compute this probability ...

We'll estimate it!

- train N models on datasets that either contain or don't contain x ("shadow models")
- $D \leftarrow P^n$ let $D_0 = D$, $D_1 = D \cup \{x\}$

$\theta \leftarrow$ train on D_0 ; $\theta \leftarrow$ train on D_1 $\pm N$

$$Loss_{IN} = \left\{ loss(\theta^{IN}, x, y) \;\middle|\; \theta^{IN} \xleftarrow{D \leftarrow P} Train(D \cup \{x\}) \right\}$$

$$Loss_{OUT} = \left\{ loss(\theta^{OUT}, x, y) \;\middle|\; \theta^{IN} \xleftarrow{D \leftarrow P} Train(D) \right\}$$

Last step: approximate $Pr[\ell(\theta, x, y) \mid H_b]$

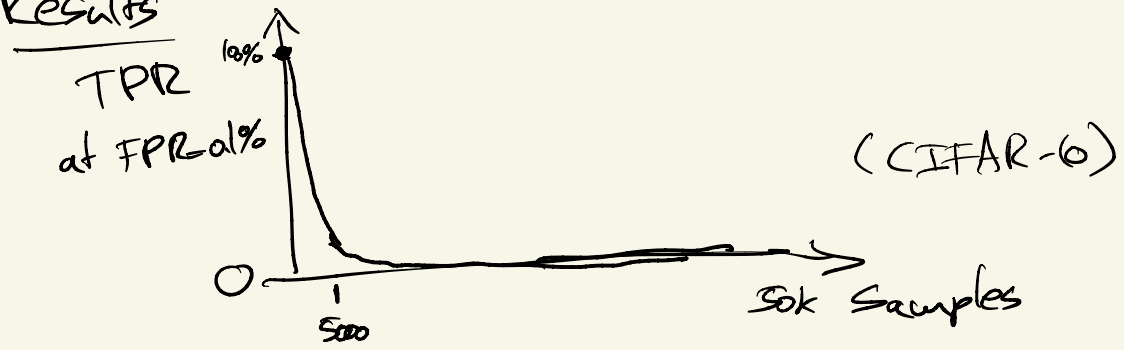
by $N(\mu_{IN}; \sigma^2_{IN})$

$N(\mu_{OUT}, \sigma^2_{OUT})$

Then: $\dfrac{Pr[\ell(\theta, x, y) \mid H_0]}{Pr[\ell(\theta, x, y) \mid H_1]}$ is the

ratio of two Gaussians

Results



Aside:

Idea: 1) Collect a dataset D

2) calculate the TPR at FPR=1% for all samples

3) remove all samples with TPR > 10% $\Rightarrow D'$

4) train a model on D'

5) Q: how vulnerable are the most vulnerable samples in D'?

A: >> 10% TPR at FPR=1%

What are MI attacks for machine learning useful for?

Auditing Privacy (lower bounds on DP)

Idea: Use Corollary 1 ($e^\varepsilon \geq \frac{TPR - \delta}{FPR}$ ...)

to lower bound privacy

Why?

1) Debugging DP-SGD implementations

Ex: . I claim $\varepsilon = 0.2$ for my implementation

. our MI attack → $\frac{TPR}{FPR} \approx e^{2.7} \approx 15$

by corollary 1, $\varepsilon \geq 2.7$ ↯

2) Understanding the "empirical privacy" of DP-SGD

Recall the analysis of DPSGD:

for one step of gradient descent, the analysis is tight if one gradient has norm $\geq C$ and is orthogonal to all other gradients.

ex: $D = \{\vec{0}, \ldots, \vec{0}\}$ $\vec{x} = \begin{bmatrix} 0 \\ \vdots \\ C \end{bmatrix}$

The full analysis is tight if:

1) each step is tight

2) the attacker gets to see all intermediate model updates $\theta_1, \ldots, \theta_{T-1}$

intuitively, the privacy of DP-SGD should be better than what the analysis says

We can estimate this empirically using MI attacks !! (DPSGD auditing)

(Roughly) the results: it seems that the DPSGD analysis is extremely conservative

(e.g. provable $\varepsilon \leq 20$, the best MI attack we have might refute $\varepsilon \geq 1$)

Q: - are our attacks bad?
- is the DPSGD analysis bad?