NOTICE

- Action items
  - 03/11: Final term-project presentation
    - 10 min presentation + 1-3 min Q&A (strict)
    - Presentation MUST cover:
      - 1-2 slide on your research motivation and goals
      - 1-2 slides on your hypotheses and experimental design
      - 3-4 slides on your most interesting results
      - 1 slides on your conclusion and implications
  - 03/18: Final exam (online, 24 hrs., unlimited trials)
  - 03/20: Final project report (online, template is on the class website)
  - 03/20: Late submissions for critiques and HW 1-4 (online)

NOTICE

- Critique stats
  - Ranks, based on “Overall Merits” and reviews “> 5” (25% of us voted)
  - Top-5 “research” papers that we liked the most
    - Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks
    - Poisoning the Unlabeled Dataset of Semi-Supervised Learning
    - Explaining and Harnessing Adversarial Examples
    - Extracting Training Data from Large Language Models
    - Certified Defenses for Data Poisoning Attacks
  - Top-5 “research” papers that we liked the least
    - Delving into Transferable Adversarial Examples and Black-box Attacks
    - Model Inversion that Exploit Confidence Information and Basic Countermeasures
    - Poisoning Attacks against Support Vector Machines
    - Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors
    - The Space of Transferable Adversarial Examples

CS 499/579: TRUSTWORTHY ML
DIFFERENTIAL PRIVACY

Sanghyun Hong
sanghyun.hong@oregonstate.edu

SAIL
Secure AI Systems Lab

How can we achieve private learning?

Deep learning with differential privacy, Abadi et al., ACM CCS 2015

DEFINITION OF MEMORIZATION

- Feldman and Zhang's
  - For a training algorithm A
  - Operating on a training set S
  - Quantify the label memorization as follows:

$$\mathtt{mem}(\mathcal{A}, S, i) := \Pr_{h \leftarrow \mathcal{A}(S)}[h(x_i) = y_i] - \Pr_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x_i) = y_i].$$

  - Problem: the estimation requires tons of training of a model on data

DEFINITION OF MEMORIZATION

- Feldman and Zhang's
  - For a training algorithm A
  - Operating on a training set S
  - New way to quantify the label memorization

$$\texttt{infl}(\mathcal{A}, S, i, j) := \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S)}[h(x'_j) = y'_j] - \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x'_j) = y'_j].$$

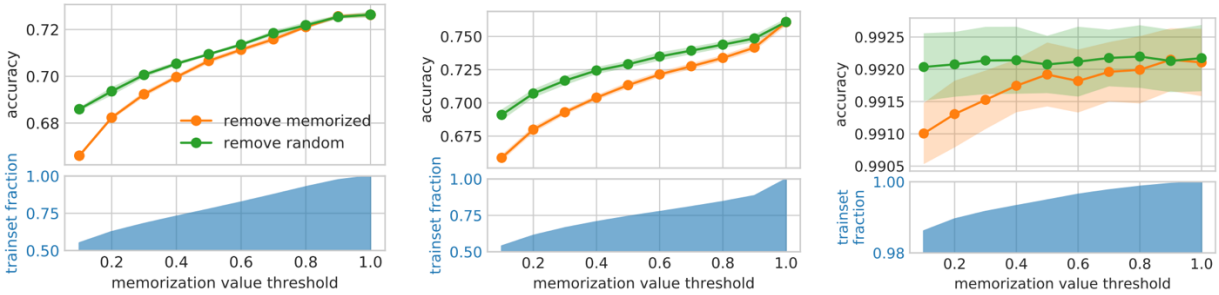
    - Use the test-set to measure the memorization
    - How much influence a single example on the test-set
    - Memorization is high, when the influence (acc. difference) is high

DEFINITION OF MEMORIZATION

- Feldman and Zhang's
  - New way to quantify the label memorization

$$\texttt{infl}(\mathcal{A}, S, i, j) := \Pr_{h \leftarrow \mathcal{A}(S)}[h(x'_j) = y'_j] - \Pr_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x'_j) = y'_j].$$

  - How much influence a single example on the test-set
  - Memorization is high, when the influence (acc. difference) is high



(a) ImageNet (b) CIFAR-100 (c) MNIST

Figure 2: Effect on the test set accuracy of removing examples with memorization value estimate above a given threshold and the same number of randomly chosen examples. Fraction of the training set remaining after the removal is in the bottom plots. Shaded area in the accuracy represents one standard deviation on 100 (CIFAR-100, MNIST) and 5 (ImageNet) trials.

DEFINITION OF AN ALGORITHM BEING PRIVATE

- A private model (an algorithm)
  - Feldman and Zhang's label memorization

$$\texttt{infl}(\mathcal{A}, S, i, j) := \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S)}[h(x'_j) = y'_j] - \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x'_j) = y'_j].$$

    - How much influence a single example on the test-set
    - Memorization is high, when the influence (acc. difference) is high

  - Property of a private model
    - Given any training instance, its influence on the test acc. is low

REVISITING DIFFERENTIAL PRIVACY

- ϵ-Differential Privacy
  - A randomized algorithm $M: D \rightarrow R$ with domain D and a range R satisfies ϵ-differential privacy if for any two adjacent inputs $d, d' \in D$ and any subset of outputs $S \subset R$ it holds

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S]$$

REVISITING DIFFERENTIAL PRIVACY

- ϵ-Differential Privacy
  - A randomized algorithm $M: D \rightarrow R$ with domain D and a range R satisfies ϵ-differential privacy if for any two adjacent inputs $d, d' \in D$ and any subset of outputs $S \subset R$ it holds

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S]$$

- (ϵ, δ)-Differential Privacy

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

  - δ: Represent some catastrophic failure cases [Link, Link]
  - $\delta < 1/|d|$, where $|d|$ is the number of samples in a database

REVISITING DIFFERENTIAL PRIVACY

- (ϵ, δ)-Differential Privacy **[Conceptually]**

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

  - You have two databases d, d' differ by one item
  - You make the same query M to each and have results $M(d)$ and $M(d')$
  - You ensure the distinguishability between the two under a measure ϵ
    - ϵ is large: those two are distinguishable, less private
    - ϵ is small: the two outputs are similar, more private
  - You also ensure the catastrophic failure probability under δ

REVISITING DIFFERENTIAL PRIVACY

- (ϵ, δ)-Differential Privacy

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

- Mechanism for (ϵ, δ)-DP: Gaussian noise

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, S_f^2 \cdot \sigma^2)$$

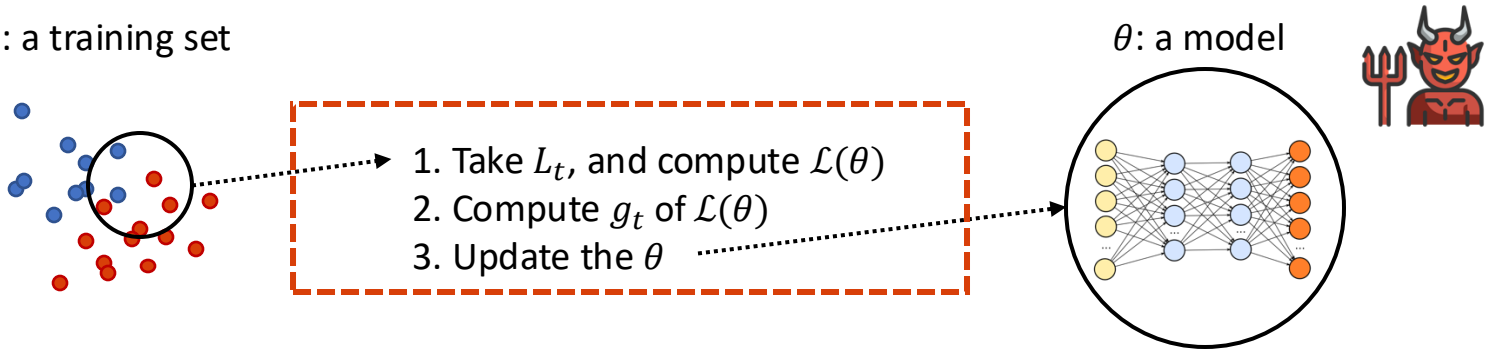
  - $M(d)$: (ϵ, δ)-DP query output on d
  - $f(d)$: non (ϵ, δ)-DP (original) query output on d
  - $N(0, S_f^2 \cdot \sigma^2)$: Gaussian normal distribution with mean 0 and the std. of $S_f^2 \cdot \sigma^2$

Post-hoc: Set the Goal ϵ and Calibrate the noise $S_f^2 \cdot \sigma^2$!

DIFFERENTIAL PRIVACY FOR MACHINE LEARNING

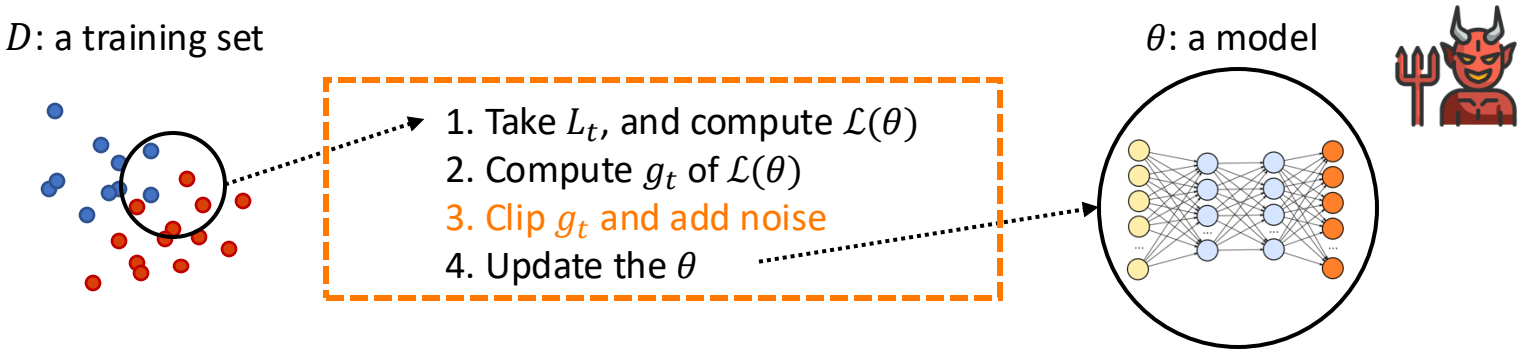
- Revisiting mini-batch stochastic gradient descent (SGD)
  1. At each step t, it takes a mini-batch L_t
  2. Computes the loss $\mathcal{L}(\theta)$ over the samples in L_t, w.r.t. the label y
  3. Computes the gradients g_t of $\mathcal{L}(\theta)$
  4. Update the model parameters θ towards the direction of reducing the loss

This Process Should Be (ϵ, δ)-DP!



MAKE EACH MINI-BATCH SGD STEP (ϵ, δ)-DP

- Mini-batch stochastic gradient descent (SGD)
  1. At each step t, it takes a mini-batch L_t
  2. Computes the loss $\mathcal{L}(\theta)$ over the samples in L_t, w.r.t. the label y
  3. Computes the gradients g_t of $\mathcal{L}(\theta)$
  4. Clip (scale) the gradients to $1/C$, where $C > 1$
  5. Add Gaussian random noise $N(0, \sigma^2 C^2 \mathbf{I})$ to g_t
  6. Update the model parameters θ towards the direction of reducing the loss



MAKE THE ENTIRE TRAINING PROCESS (ϵ, δ)-DP

- Mini-batch stochastic gradient descent (SGD)
  - SGD iteratively computes the (ϵ, δ)-DP step T times
  - **Problem:** how do we compute the total privacy leakage ϵ_{tot} over T iterations?

- Privacy accounting with moment accountant
  - **Key intuition:** DP has the **composition** property
    - Suppose the two mechanism M_1 and M_2 satisfies $(\varepsilon_1, \delta_1)$- and $(\varepsilon_2, \delta_2)$-DP the composition of those mechanisms $M_3 = M_2(M_1)$ satisfies $(\varepsilon_1+\varepsilon_2, \delta_1+\delta_2)$-DP
    - If each step t satisfies (ε, δ)-DP, the total SGD process satisfies $(\varepsilon T, \delta T)$-DP
  - **Moment accountant:** tracking the total privacy leakage εT over T iterations

PUTTING ALL TOGETHER

- DP-Stochastic Gradient Descent (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do** // we train a model θ with the privacy budget ε_{budget}

Take a random sample L_t with sampling probability L/N // iterate over *T* mini-batches

Compute gradient // compute the gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$ // clip the magnitude of the gradients

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$ // add Gaussian random noise to the gradients

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

$\varepsilon, \delta \leftarrow$ compute the privacy cost (leakage) so far // compute the privacy cost (leakage) up to *t* iterations

If $\varepsilon > \varepsilon_{buget}$: then break; // if the cost is over the budget, then stop training

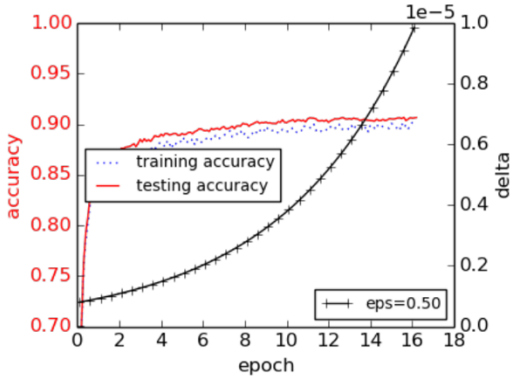
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

EVALUATION

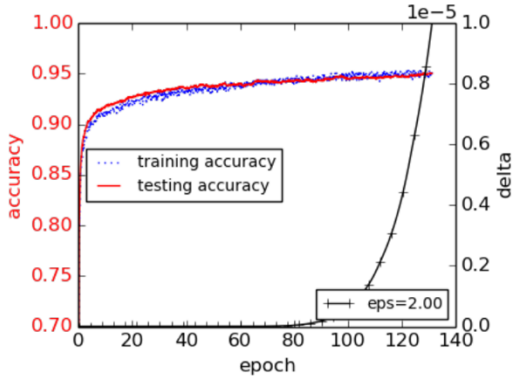
- Setup
  - Datasets: MNIST | CIFAR-10/100
  - Models:
    - MNIST: 2-layer feedforward NN on 60-dim. PCA projected inputs
    - CIFAR-10/100: A CNN with 2 conv. layers and 2 fully-connected layers
  - Metrics:
    - Classification accuracy
    - Privacy cost (ε_{budget})

EVALUATION

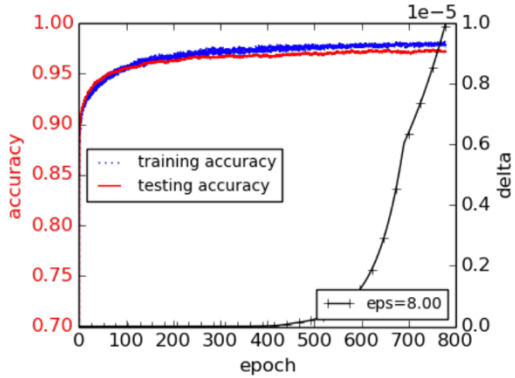
- Impact of Noise
  - Dataset, Models: MNIST, 2-layer feedforward NN
  - Setup: 60-dim PCA projected inputs | Clipping threshold ($\mathbf{C}$): 4 | Noise (σ): 8, 4, 2 (from the left)
  - **Summary:**
    - On MNIST, DP-SGD offers reasonable acc. under various privacy costs (**clean:** 98.3%)
    - The accuracy of private models decreases as we decrease the privacy cost



(1) Large noise



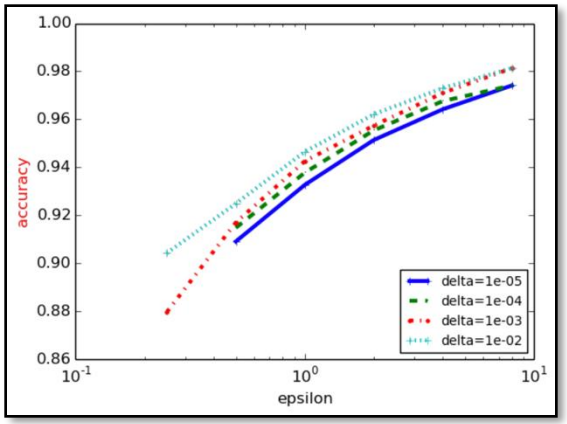
(2) Medium noise



(3) Small noise

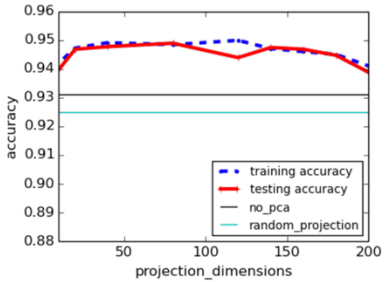
EVALUATION

- Impact of Noise
  - Dataset, Models: MNIST, 2-layer feedforward NN
  - Setup: 60-dim PCA projected inputs | Clipping threshold ($\mathbf{C}$): 4 | Noise (σ): 8, 4, 2 (from the left)
  - **Summary:**
    - On MNIST, DP-SGD offers reasonable acc. under various privacy costs (**clean:** 98.3%)
    - The accuracy of private models decreases as we decrease the privacy cost

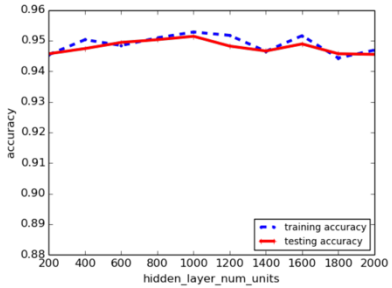


EVALUATION

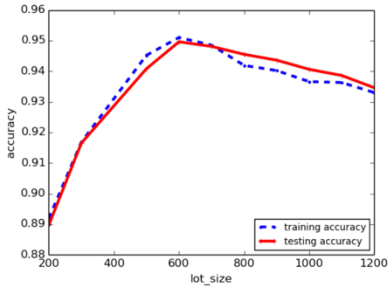
- Impact of Hyper-parameter Choices
  - Dataset, Models: MNIST, 2-layer feedforward NN
  - Setup: 60-dim PCA projected inputs



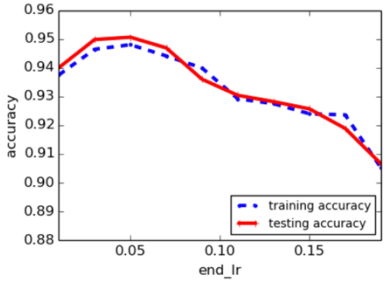
(1) variable projection dimensions



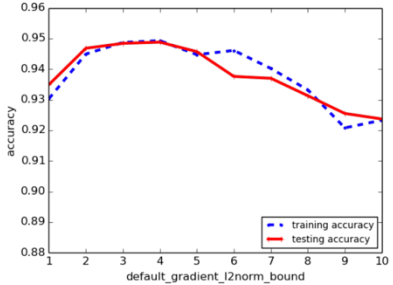
(2) variable hidden units



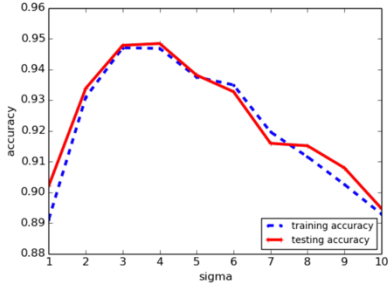
(3) variable lot size



(4) variable learning rate



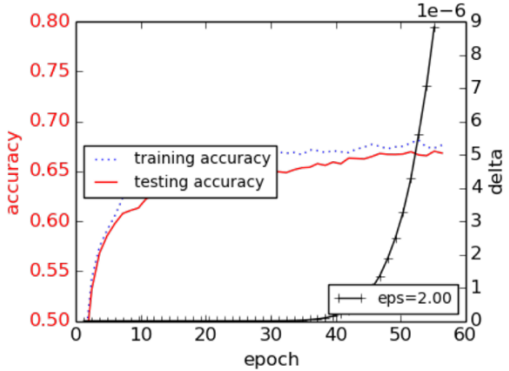
(5) variable gradient clipping norm



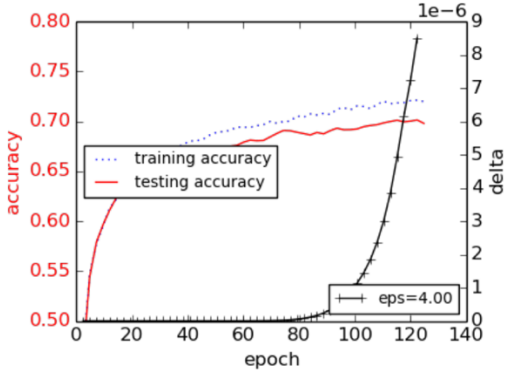
(6) variable noise level

EVALUATION

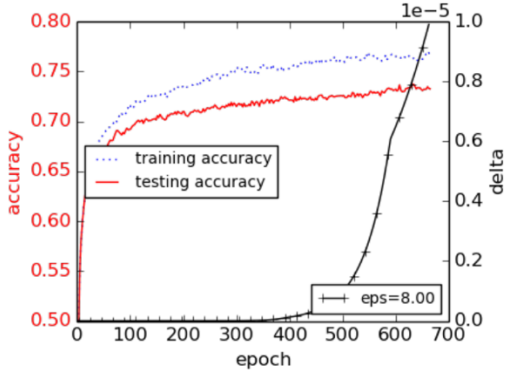
- Impact of Noise
  - Dataset, Models: CIFAR-10, CNN
  - Setup: Clipping threshold ($\mathbf{C}$): 3 | Noise (σ): 6
  - **Summary:**
    - On CIFAR-10, DP-SGD offers reasonable acc. under various privacy costs (**clean:** 80%)
    - The accuracy of private models decreases as we decrease the privacy cost



(1) $\varepsilon = 2$



(2) $\varepsilon = 4$



(3) $\varepsilon = 8$

WHAT DOES IT MEAN BY EPSILON = 2/4/6 IN CIFAR-10?

EVALUATING DIFFERENTIALLY PRIVATE MACHINE LEARNING IN PRACTICE, JAYARAMAN AND EVANS, USENIX SECURITY 2019

EMPIRICAL EVALUATIONS OF PRIVACY RISKS IN DP-MODELS

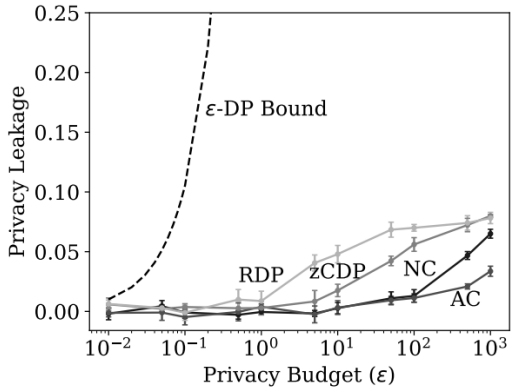
- Setup
  - **Datasets:** Purchase-100 | CIFAR-100 (on 50-dim PCA projected inputs)
  - **Models:** Logistic regressions | 2-layer feedforward NNs

  - **Privacy Attacks:**
    - Membership inference: Yeom *et al*. and Shokri *et al*.

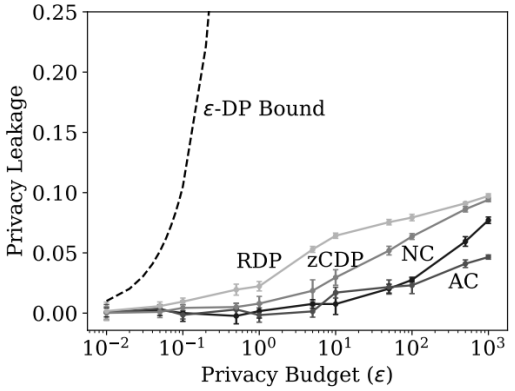
  - **DP-SGD:**
    - Set the clipping norm ($\mathbf{C}$) to 1
    - Set the prob. of catastrophic failures ($\boldsymbol{\delta}$) to $10^{-5} < 1/|N|$ (N~60k in MNIST and 50k in CIFAR)
    - Set the batch size to 200
    - Set the learning rate to 0.01 for Adam optimizer
    - Vary ε from 0.01 to 1000
    - Compare (ϵ, δ)-DP with other DP-mechanisms: AC, CDP, zCDP, and RDP
    - Run 5-times and measure the (TPR – FPR) and accuracy loss on average

EVALUATION ON CIFAR-100, LRS

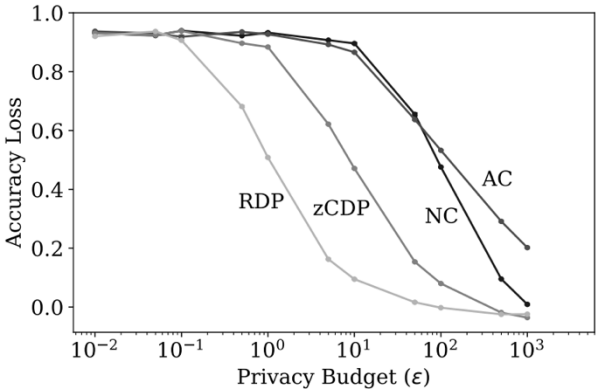
- Summary
  - Yeom *et al*. and Shokri *et al*. are weak privacy attacks
  - In other words, (ϵ, δ)-DP theoretically offers very strong privacy bounds
  - If a DP-mechanism offers stronger bound, the acc. of models decrease accordingly



(a) Shokri et al. membership inference



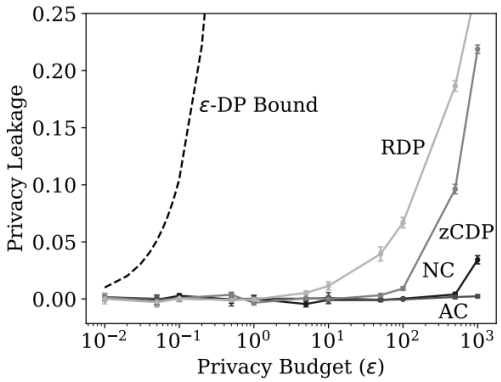
(b) Yeom et al. membership inference



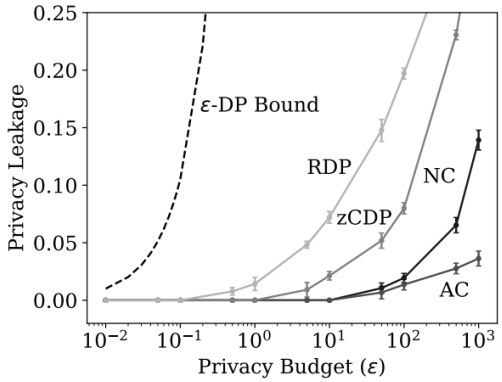
(b) Per-instance gradient clipping

EVALUATION ON CIFAR-100, LRS

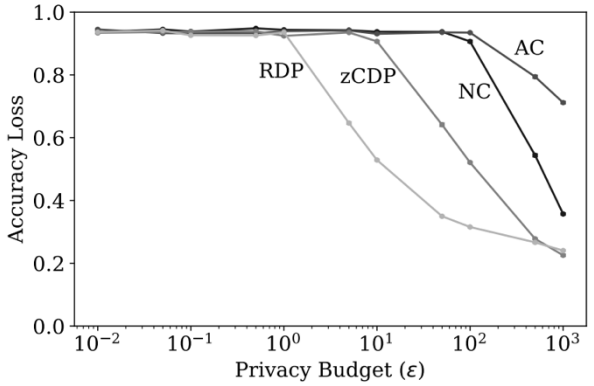
- Summary
  - Yeom *et al*. and Shokri *et al*. are weak privacy attacks
  - In other words, (ϵ, δ)-DP theoretically offers very strong privacy bounds
  - If a DP-mechanism offers stronger bound, the acc. of models decrease accordingly
  - Compared to LRs, NNs leak more in higher privacy budgets



(a) Shokri et al. membership inference



(b) Yeom et al. membership inference



(a) CIFAR-100

EVALUATION ON MI PREDICTIONS: LRs vs. NNs

- Summary
  - Yeom *et al*. and Shokri *et al*. are weak privacy attacks
  - In other words, (ϵ, δ)-DP theoretically offers very strong privacy bounds
  - If a DP-mechanism offers stronger bound, the acc. of models decrease accordingly
  - Compared to LRs, NNs leak more in higher privacy budgets
  - Predictions (TPRs and FPRs) are more consistent in LRs than NNs in CIFAR-100

Figure 3: Overlap of membership predictions across two runs of logistic regression with RDP at $\epsilon = 1000$ (CIFAR-100)

(a) Overlap of membership predictions across two runs

Thank You!

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/F23

SAIL
Secure AI Systems Lab