Rensselaer

why not change the world?®

Privacy Preservation and Evaluation in Machine Learning

Joseph Pedersen

Committee Members: Dr. Kristin Bennett, Dr. Isabelle Guyon, Dr. William Wallace, Dr. John Mitchell

DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINERRING | 7/8/2022

OUTLINE

- Illustrative Example
- Background
- Contributions
- Differentially Private Wasserstein GAN with gradient penalty
- Privacy Evaluation: Metric based and Attack based
- Novel Defense Strategy: Optimally Soft Labels
- Conclusion and Future Work

- The Netflix Challenge

- $1 Million to build a better movie recommender
- Only released user movie ratings with dates, with no other identifying information.
- What could go wrong?

- The Netflix Challenge

USER 17493258	Home Alone 7/13/2021 3 stars	Rambo 7/17/2021 5 stars	Naughty 7/22/2021 4 stars	Frozen 7/28/2021 2 stars	▪ ▪ ▪
USER 39225681	Madagascar 7/11/2021 4 stars	Rocky 7/15/2021 2 stars	Unbreakable 7/20/2021 1 stars	Bambi 7/26/2021 3 stars	▪ ▪ ▪
USER 98261030	The Sandlot 7/15/2021 4 stars	Alien 7/18/2021 3 stars	Rocky 7/21/2021 3 stars	Madagascar 7/25/2021 5 stars	▪ ▪ ▪

- The Netflix Challenge + IMDb (linkage)

USER 17493258	Home Alone 7/13/2021 3 stars	Rambo 7/17/2021 5 stars	Naughty 7/22/2021 4 stars	Frozen 7/28/2021 2 stars	▪ ▪ ▪
John Doe	Home Alone 7/14/2021 3 stars	Rambo 7/17/2021 5 stars		Frozen 7/29/2021 2 stars	▪ ▪ ▪

"in-the-closet lesbian mother sued Netflix for privacy invasion"

SOURCE: Ryan Singel. Netflix cancels recommendation contest after privacy lawsuit. Retrieved from https://www.wired.com/2010/03/netflix-cancels-contest/ on May 18, 2020.

- How can we try to avoid privacy loss?
  - Release synthetic data instead of real data
  - Use provably private release (e.g. differential privacy)
  - Metric based estimate of privacy before release
  - Estimate robustness against privacy attack

- Synthetic Data – ideally from same distribution

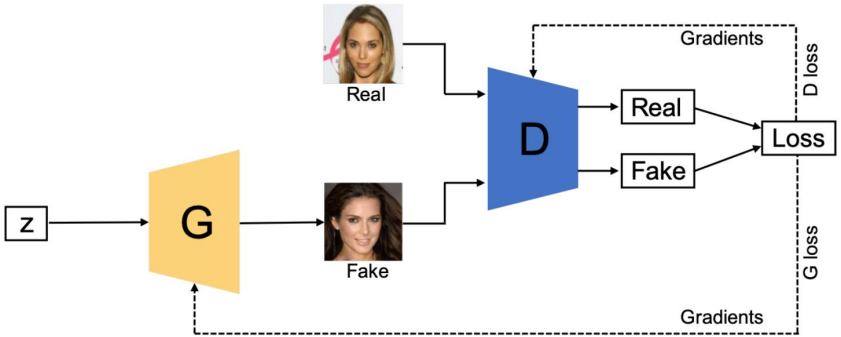


IMAGE SOURCE: Zhengwei Wang, Qi She, Tomas E. Ward. Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. arXiv:1906.01529v6 [cs.LG] 29 Dec 2020

Synthetic Data is NOT Automatically Private

- Fake ≠ Private
- Generators can overfit their training data
- Weights of neural network can memorize training data

SOURCE: Devansh Arpit and Stanisław Jastrzebski and Nicolas Ballas and David Krueger and Emmanuel Bengio and Maxinder S. Kanwal and Tegan Maharaj and Asja Fischer and Aaron Courville and Yoshua Bengio and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. arXiv:1706.05394v2 [stat.ML] 1 Jul 2017

- One row in data has bounded impact on output

Definition 2.2.0.3. *(Differential Privacy) A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if $\forall \mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and $\forall x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:[11]*

Adjacent databases

Multiplicative factor

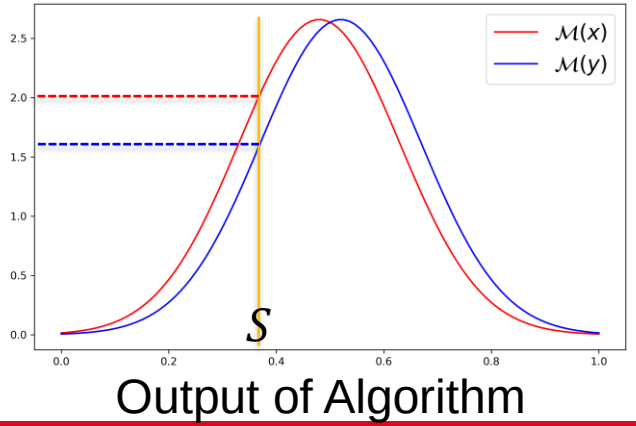
$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta \quad (2.2)$$

Privacy loss hyper-parameter

If $\mathcal{M}$ is $(\varepsilon, 0)$-differentially private, we may refer to it as ε-differentially private.[11]

$$\varepsilon \approx \log \frac{2}{1.6}$$

PDF of output for two adjacent inputs



Output of Algorithm

SOURCE: Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4):211–407, 2014.

- To make DP, add noise based on sensitivity
  - Noise can degrade output quality
  - Better bound on sensitivity → less noise needed

Definition 2.2.0.4. *(ℓ_1-sensitivity) The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:[11]*

$$\Delta f = \max_{\substack{x,y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1 \tag{2.3}$$

SOURCE: Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4):211–407, 2014.

- DP deep learning – add noise to gradients

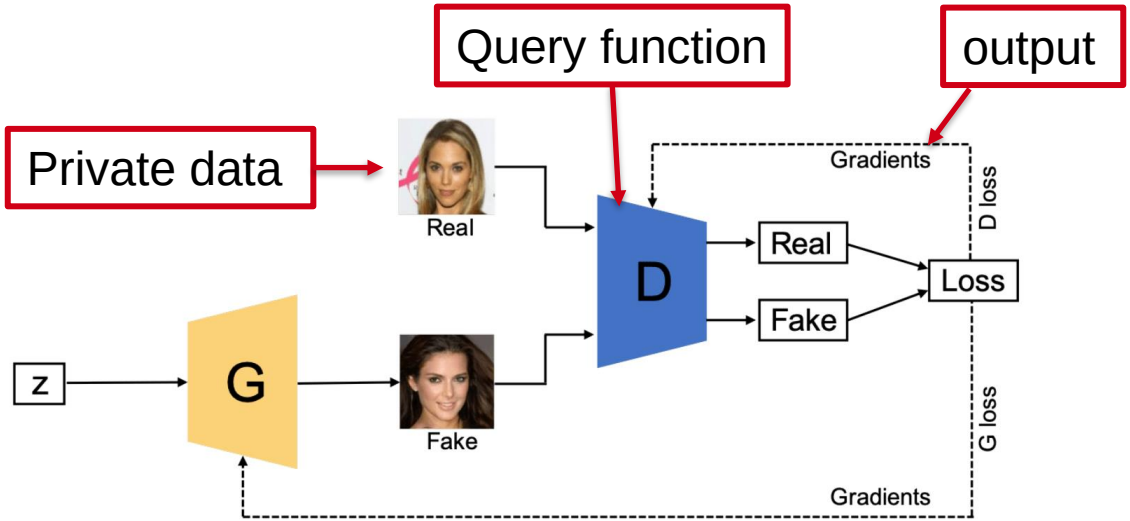
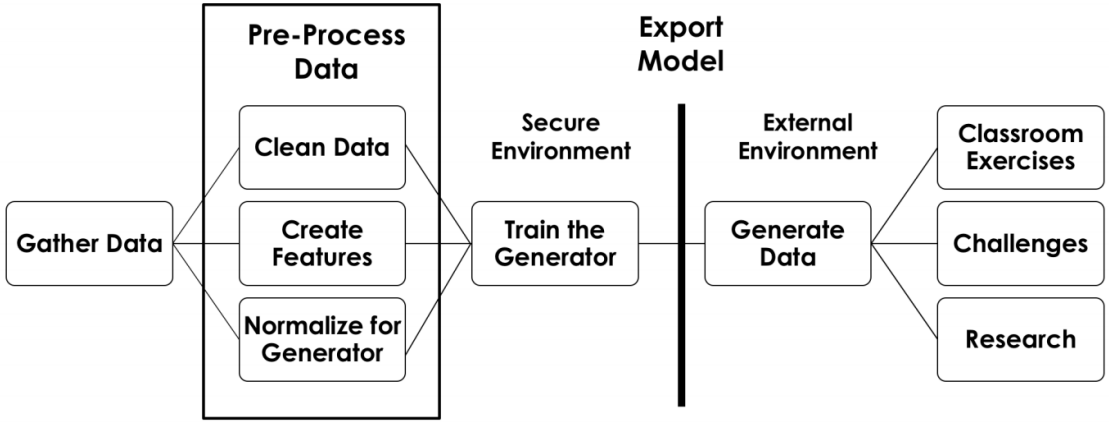


IMAGE SOURCE: Zhengwei Wang, Qi She, Tomas E. Ward. Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. arXiv:1906.01529v6 [cs.LG] 29 Dec 2020

For private deep learning, add noise to the gradient computed on each batch of private data.

SOURCE: Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pages 308–318, 2016.



- WGAN-GP
- Empirically Private
  - nearest neighbor Adversarial Accuracy
- Not Differentially Private

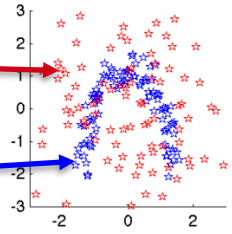
SOURCE: Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Privacy Preserving Synthetic Health Data. In ESANN 2019 - European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, April 2019

Nearest Neighbor Adversarial Accuracy Metric

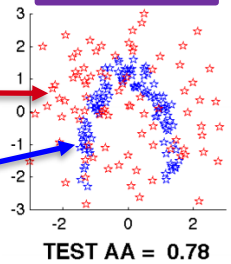
Fake data closer to other fake data

Real data closer to other real data

nnAA close to 1



under fit

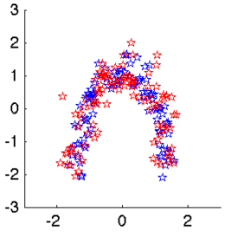


Fake data closer to other fake data

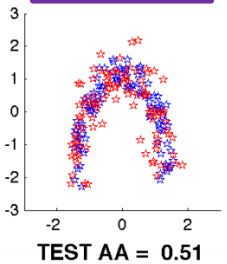
Real data closer to other real data

nnAA close to 1

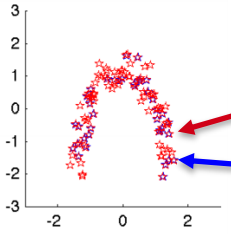
Privacy loss = 0.78 – 0.77 = **0.01**



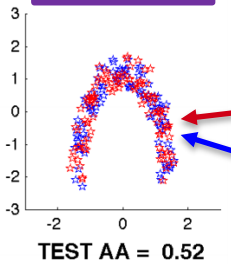
good fit



Privacy loss = 0.51 – 0.49 = **0.02**



over fit



Fake data closer to real data

Real data closer to fake data

nnAA = 0

Fake data about same distance

Real data about same distance

nnAA close to 0.5

Privacy loss = 0.52 – 0.00 = **0.52**

SOURCE: Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Privacy Preserving Synthetic Health Data. In ESANN 2019 - European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, April 2019

- Test robustness against privacy attack

- Membership Inference Attack – “given a machine learning model and a record, determine whether this record was used as part of the model’s training dataset or not.”

SOURCE: Reza Shokri and Marco Stronati and Congzheng Song and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. arXiv:1610.05820v2 [cs.CR] 31 Mar 2017

- Yeom et al. (2018) prove attribute inference, another type of attack in which the attacker tries to infer missing attributes given other attributes from a sample in the training data, is “harder” than membership inference

SOURCE: Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pages 268–282. IEEE, 2018.

Thesis Contributions

- **Designed an algorithm** WGAN-GP, provably differentially private, with good utility.
- **Improved on a metric** to estimate generative model privacy based on nearest neighbors.
- **Devised a methodology** to evaluate privacy empirically using membership inference attacks, with formal analysis of privacy in this methodology.
- **Designed a defense strategy** against membership attacks of an ideal attacker.

- Can better bounds be achieved for WGAN?
- Can WGAN-GP be DP without clipping weights or gradients?
- Does DP impact the performance of HealthGAN?
- Does DP make HealthGAN more private empirically?

Work in this section to appear in Springer Nature - Research Book Series: Transactions on Computational Science & Computational Intelligence by Joseph Pedersen and Kristin Bennett

- WGAN uses weight clipping to enforce Lipschitz constraint
- WGAN-GP uses a “gradient penalty” to enforce the Lipschitz constraint

$$L = \mathbb{E}_{\tilde{x}\sim p_g}[D(\tilde{x})] - \mathbb{E}_{x\sim p_{data}}[D(x)] + \lambda\, \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right] \qquad (2.10)$$

WGAN loss | gradient penalty

GRADIENT : Had a bound. I made it tight.† | Did not have a bound. I derived one.†

†For fully connected feedforward critic with ReLU or LeakyReLU.

SOURCE: Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs, 2017. arXiv preprint arXiv:1704.00028.

3.1.2 Derivation of Tight Bound on Gradient of WGAN

Theorem 3.1.1. *At each iteration of training, given the current weights and biases of a fully connected critic with ReLU or LeakyReLU activation functions, each component of the gradient of the critic can be bounded by bounds of the form:*

$$C_{0,k}^{\nabla_z} \leq [\nabla_z D_{W^{(t_c)}}(\mathbf{x})]_k \leq C_{1,k}^{\nabla_z} \tag{3.4}$$

- Provably tight
- Computed each iteration, obviating the need to clip weights
- Component-wise, permitting non-spherical Gaussian noise

Theorem 3.1.3. *The norm of the gradient of the gradient penalty is bounded by*

$$\|\nabla_{W^\ell}\left(\|\nabla_x D(x)\|_2 - 1\right)^2\|_F \leq 2\max\{\left(\|\nabla_x D(x)\|_2 - 1\right), 1\}\prod_{k\neq\ell}\|W^k\|_F \prod_{1\leq k\leq L}\|\phi'_{\alpha^k}\|_F$$

- Also computed by iteration
- Bound on norm, not component-wise

Analysis shows that it is many orders of magnitude more sensitive than the other term, preventing a low value of ε

Evaluation on Health Data from Published Medical Studies

Table 3.2: Characteristics of three MIMIC-III Datasets. + Rate refers to percentage of people who died/were readmitted in the training data.

Dataset	Data Points	Features	Label	+ Rate
1	2085	6 diagnostic scores/values 5 demographic categorical var.	Mortality	6.3%
2	2119	31 diagnostic scores 2 categorical var.	Mortality	23.4%
3	1719	6 diagnostic values	Readmission	3.03%

SOURCE of data: Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. Scientific data, 3(1):1–9, 2016

Noise Added for Differential Privacy Helped Mitigate Mode Collapse

	Dataset 1	Dataset 2	Dataset 3
training set	6.28	23.4	3.03
test set	6.04	25.9	2.97
$\sigma_{\text{dp}} = 0$ (HGAN)	4.46	17.8	2.60
$\sigma_{\text{dp}} = 10^{-7}$	**5.37**	19.1	2.39
$\sigma_{\text{dp}} = 10^{-6}$	5.23	**20.1**	2.68
$\sigma_{\text{dp}} = 10^{-5}$	4.22	19.6	**3.20**
$\sigma_{\text{dp}} = 10^{-4}$	4.89	18.9	2.97

Values indicate percent of rare class in the data.
Differential privacy tended to decrease mode collapse.

Synthetic Data with Least Mode Collapse Had Best AUROC

Measure	Dataset 1		Dataset 2		Dataset 3	
	HGAN	w/ D.P.	HGAN	w/ D.P.	HGAN	w/ D.P.
Utility – AUC	0.723	**0.724**	0.614	**0.657**	0.713	**0.720**
+Rate	4.5%	**5.4%**	17.8%	**20.2%**	2.6%	**3.2%**
TrResemblLoss	0.545	**0.539**	**0.856**	0.860	0.628	**0.627**
TeResemblLoss	**0.534**	0.550	**0.868**	0.872	**0.648**	0.649
PrivacyLoss	**-0.011**	0.012	**0.012**	**0.012**	0.022	**0.020**

Best values are in bold, but differences may not be statistically significant

Nearest Neighbor Adversarial Accuracy Metric

- Can the nearest neighbor Adversarial Accuracy metric be made into an unbiased estimator?
- Can that be done without making it too computationally expensive?
- Does it work with discrete and mixed distributions?
- Do both terms measure the same aspect of resemblance?

Original Definition is Biased: Expected Value ≠ 0.5

$$\mathcal{AA}_{TS} = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(d_{TS}(i) > d_{TT}(i)\right) + \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(d_{ST}(i) > d_{SS}(i)\right)\right) \tag{2.11}$$

$S_S = \left\{\left(\mathbf{x}_S^1, y_S^1\right), \ldots, \left(\mathbf{x}_S^n, y_S^n\right)\right\}$ is a sample from distribution P_S

$S_T = \left\{\left(\mathbf{x}_T^1, y_T^1\right), \ldots, \left(\mathbf{x}_T^n, y_T^n\right)\right\}$ is a sample from distribution P_T

$d_{TS}(i) = \min_j \|\mathbf{x}_T^i - \mathbf{x}_S^j\|$ is the distance from $\mathbf{x}_T^i \in S_T$ and its nearest neighbor in S_S

$d_{ST}(i) = \min_j \|\mathbf{x}_S^i - \mathbf{x}_T^j\|$ is the distance from $\mathbf{x}_S^i \in S_S$ and its nearest neighbor in S_T

Do not leave one out

$d_{TT}(i) = \min_{j, j\neq i} \|\mathbf{x}_T^i - \mathbf{x}_T^j\|$ is the distance from $\mathbf{x}_T^i \in S_T$ to its nearest neighbor in $S_T \setminus \{\mathbf{x}_T^i\}$

$d_{SS}(i) = \min_{j, j\neq i} \|\mathbf{x}_S^i - \mathbf{x}_S^j\|$ is the distance from $\mathbf{x}_S^i \in S_S$ to its nearest neighbor in $S_S \setminus \{\mathbf{x}_S^i\}$

Leave one out

SOURCE: Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Privacy Preserving Synthetic Health Data. In ESANN 2019 - European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, April 2019

Definition 4.1.1. *The unbiased nearest neighbor Adversarial Accuracy between two sets S and T, both of size n, is given by:*

$$\mathcal{AA}_{TS} = \frac{1}{2}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\mathbf{1}\left(d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)\right)\right) + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\mathbf{1}\left(d_{S(T\setminus\{\mathbf{x}_T^k\})}(i) > d_{SS}(i)\right)\right)\right] \tag{4.1}$$

Theorem 4.1.1. *For i.i.d. random samples S and T, both of the same size n, drawn from the same continuous distribution P on metric space $\mathcal{X}$, with probability density function $p(x)$,* $\mathbb{E}\left[\mathcal{AA}_{TS}\right] = 0.5$

- Made efficient implement using two nearest neighbors

Correction for Discrete and Mixed Distributions

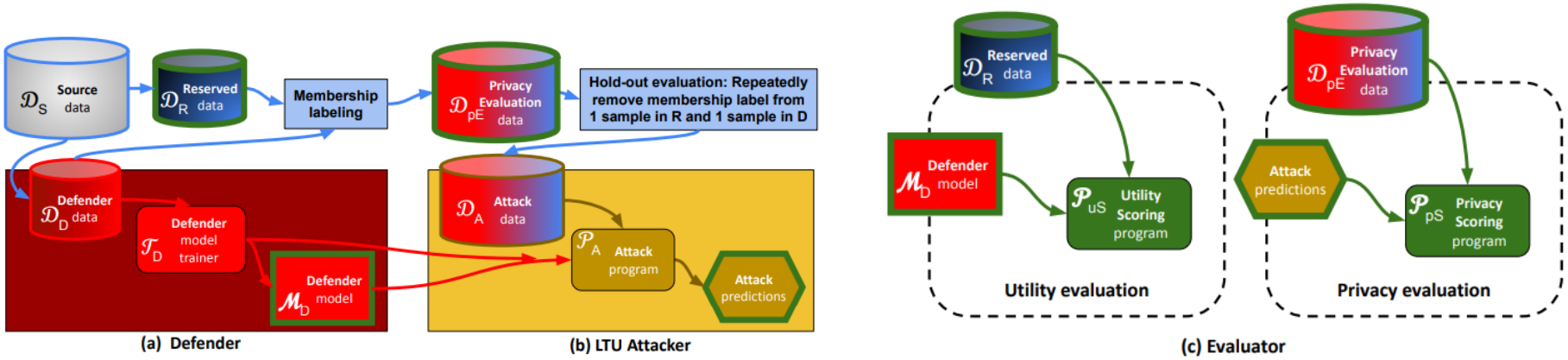
$$\mathcal{AA}_{TS} = \frac{1}{4n^2}\left[\sum_{i=1}^{n}\left(\sum_{k=1}^{n}\mathbf{1}\left(d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)\right)\right) + \sum_{i=1}^{n}\left(\sum_{k=1}^{n}\mathbf{1}\left(d_{S(T\setminus\{\mathbf{x}_T^k\})}(i) > d_{SS}(i)\right)\right)\right] + \frac{1}{4n^2}\left[\sum_{i=1}^{n}\left(\sum_{k=1}^{n}\mathbf{1}\left(d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) \geq d_{TT}(i)\right)\right) + \sum_{i=1}^{n}\left(\sum_{k=1}^{n}\mathbf{1}\left(d_{S(T\setminus\{\mathbf{x}_T^k\})}(i) \geq d_{SS}(i)\right)\right)\right] \quad (4.5)$$

- Return both terms, since each has expectation 0.5

- Can we create an easy to implement privacy evaluation framework to provide estimates of privacy risk for each sample, with a strong attack that does not need designed specifically to attack a particular model?
- Can we implement a general technique which increases privacy protection for the most exposed samples with only minor negative impact to utility?

Some work in this section presented in LTU Attacker for Membership Inference
at the Third AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-22)
by Joseph Pedersen and Isabelle Guyon and Jiangnan Huang and Rafael Muñoz-Gómez and Haozhe Sun and Wei-Wei

- Most knowledgeable attacker



A_{ltu} = the Attack accuracy over many rounds

$$Privacy = \min\{2(1 - A_{ltu}), 1\} \pm 2\sqrt{A_{ltu}(1 - A_{ltu})/N}$$

Black-box vs White-box

- Attacker access to Defender Model

Black box: can only submit query (submit input and receive output)
White box: complete access to the model, architecture, and parameter values.

- Attacker access to Defender Model Trainer

Black box: can submit input to create new model.
White box: has entire Defender Model Trainer, except random seeds used
Active: can attack learning algorithm during training.
Passive: no access during training.

Black-box Attack is Theoretically Optimal

Theorem 5.2.2 (No privacy protection by obscurity)**.** *For a fixed training set $d^{(n_1)}$, sample $u_i \in d^{(n_1)}$, trained model $\widetilde{m}$, and pair (u_1, u_2) for which the attacker knows that exactly one was in the training set, having only black-box access to $\widetilde{m}$ and the model trainer $\mathcal{T}_h$, with unlimited queries, is sufficient to perform the optimal attack.*

Proof. As shown already, the optimal attack is the Bayes' optimal classifier, with:

$$\Pr\left(u_1 \in d_{\text{train}}^{(n_1)} \middle| M = m\right) = \frac{\Pr(M_1 = m)}{\Pr(M_1 = m) + \Pr(M_2 = m)}$$

Theorem 5.2.1 (No learning without risk to privacy). *In the LTU framework, for a fixed training set* $d^{(n_1)}$ *and sample* $u_1 \in d^{(n_1)}$ *there is zero probability of the attacker having a better than random attack against* u_1 *if and only if for all* $u_i \sim \mathbb{P}_U$ *with* $\Pr(u_i) > 0$, *and defining* $d_i^{(n_1)} = d_{\text{train}}^{(n_1-1)} \cup \{u_i\}$ *and* $M_i = \mathcal{T}_h(Z, d_i^{(n_1)})$, *we have* $\Pr(M_1 = m) = \Pr(M_i = m)$ *for all* m.

- Only “risk”: the actual model learned may be private for u_1

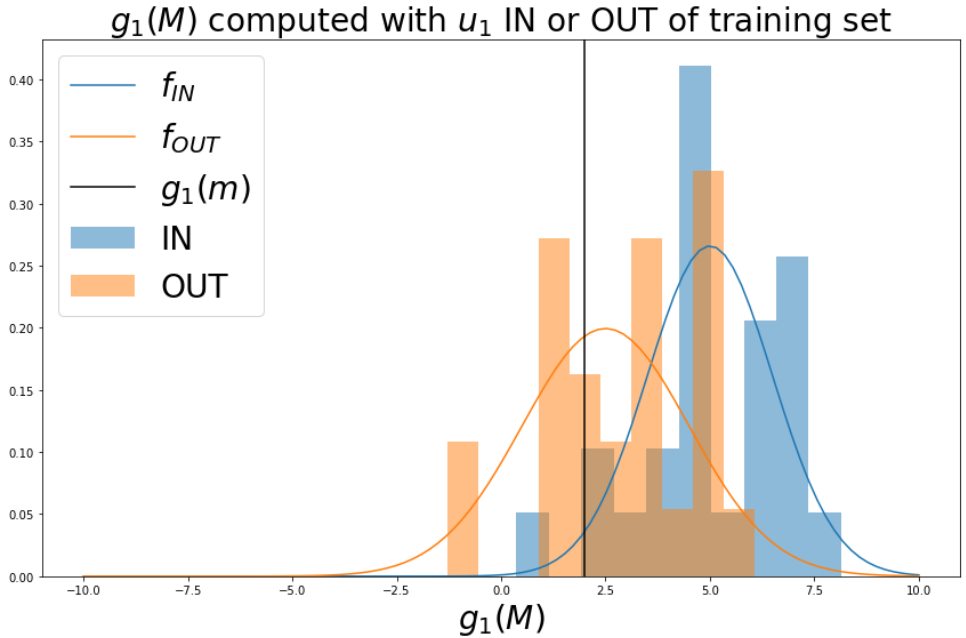
- In practice, estimating $\Pr(M_i = m)$ is infeasible

- Use projection: $g_i(m) = m(x_i) \cdot \vec{y}_i$ **Predicted Logits of Target Class**

- Use $\Pr(g_i(M_i) = g_i(m))$

- Attacking sample u_1

$$\Pr(u_1 \in D) = \frac{f_{IN}(g_1(m))}{f_{IN}(g_1(m)) + f_{OUT}(g_1(m))}$$



Motivation for Defense Strategy: Optimally Soft Labels

- Idea 1

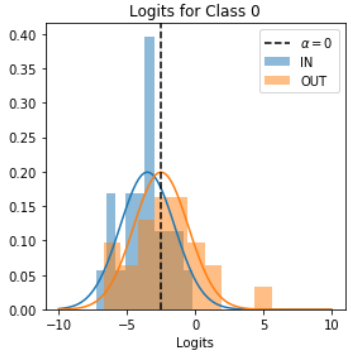
IN and OUT distributions identical $\Longrightarrow$ attack fails

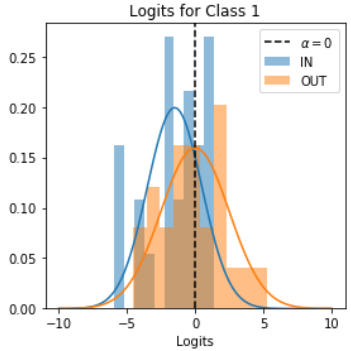
- Idea 2

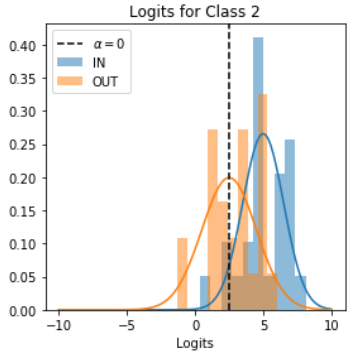
Mean of “Attacker OUT” distribution has correct argmax $\Longrightarrow$ using that as target may give good accuracy

Optimally Soft Labels: $\alpha = 0$

- Hypothetical example (target class is 2)
- Average of “OUT” has correct label
- Use it as target, $\alpha = 0$, (most private)

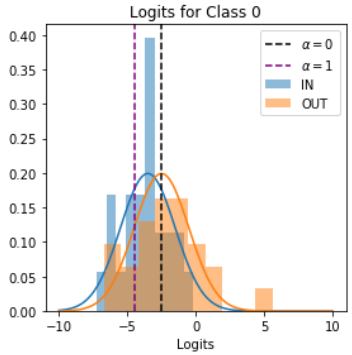


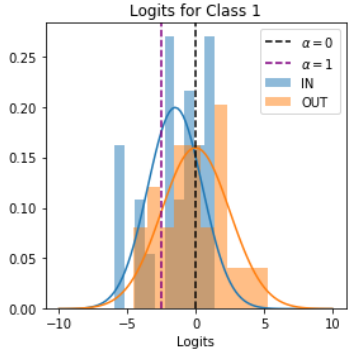


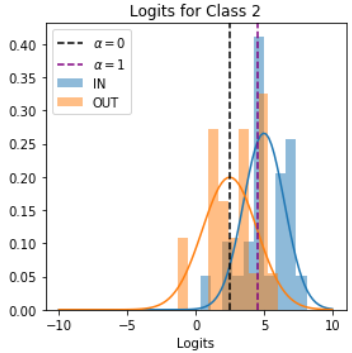


Optimally Soft Labels: $\alpha = 1$

- Hypothetical example (target class is 2)
- Or move α std. dev. toward $\vec{y}_i$
- $\alpha = 1$ is shown by purple line





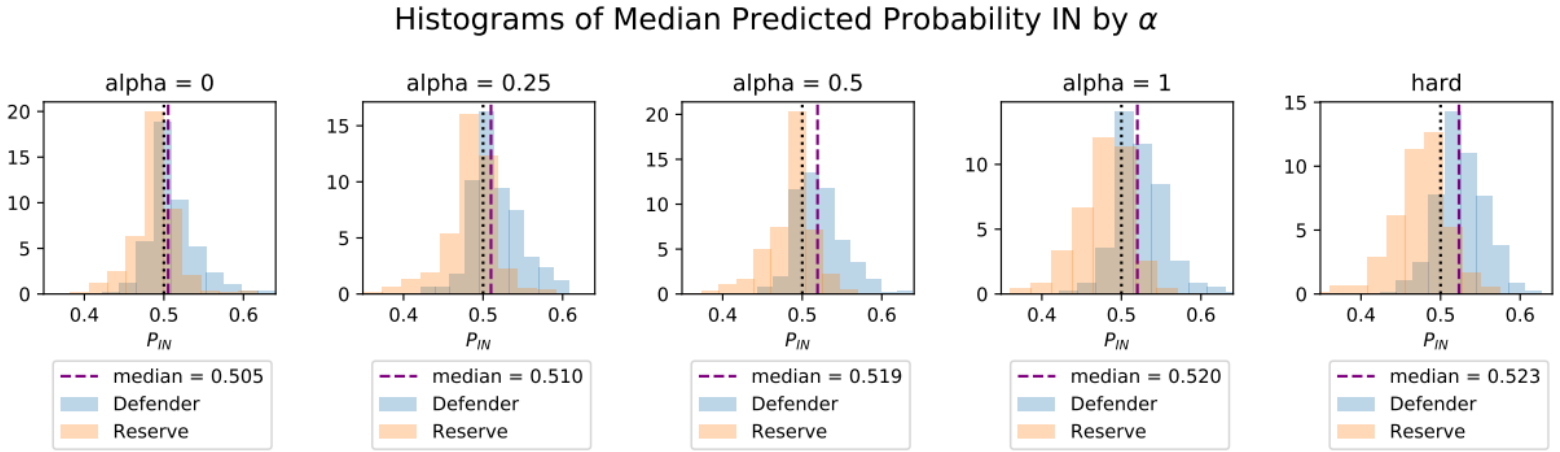


- To make the problem challenging:

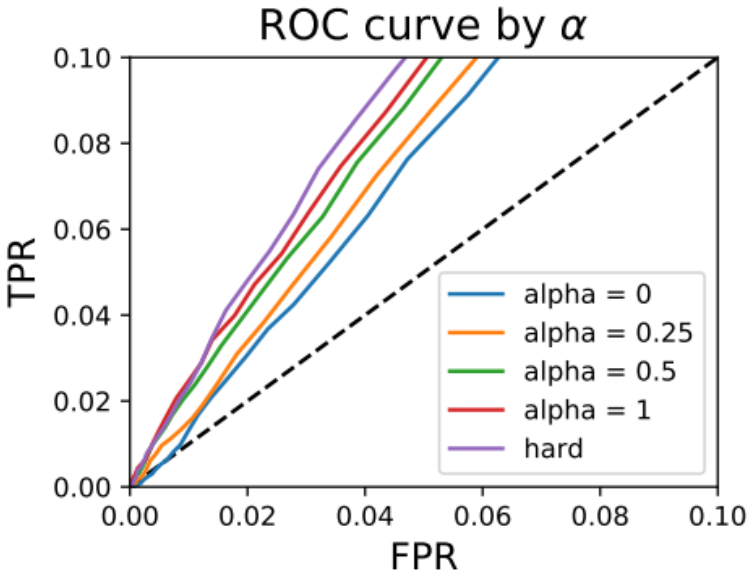
- We use a base model that is already well regularized, has differential privacy, and only trains for 6 epochs, making it already well defended. The Defender data was 400 (balanced) samples from CIFAR10, preprocessed with VGG19.

- We corrupt 20% of the labels, to guarantee that not all of the OUT scores will match the labels. Intuitively, these samples should be more effected by being IN or OUT, making them easier to attack, and hence harder to defend.

Statistically Significant Improvement to Privacy



Much More Protection in Low False Positive Regime



Optimally Soft Labels Increase Test Accuracy

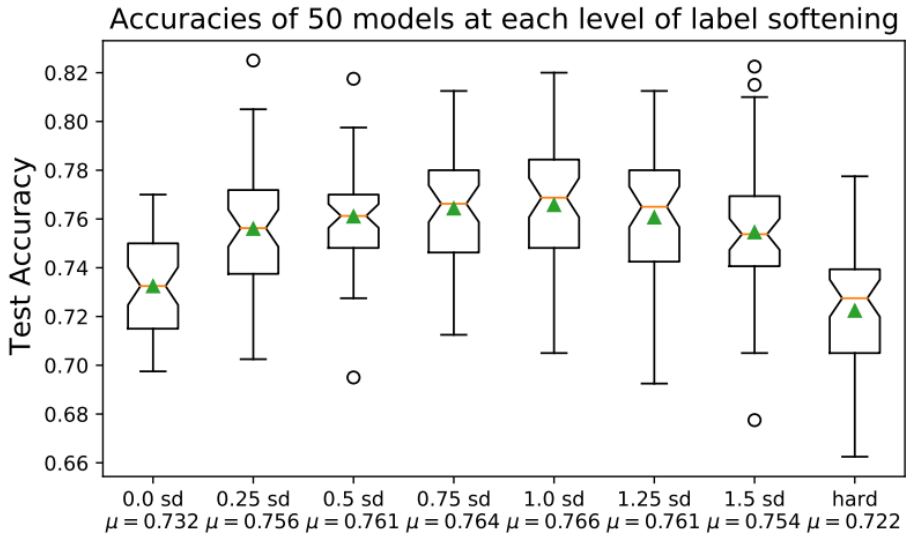


Figure 5.6: Boxplots of the test accuracy for 50 models trained on different levels of optimally soft labeling

Traditional Soft Labels DO NOT Increase Test Accuracy

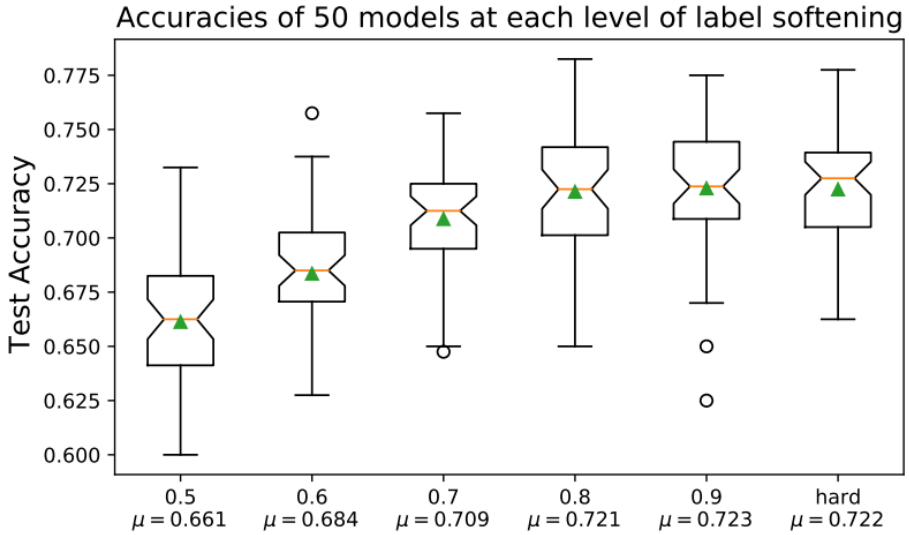
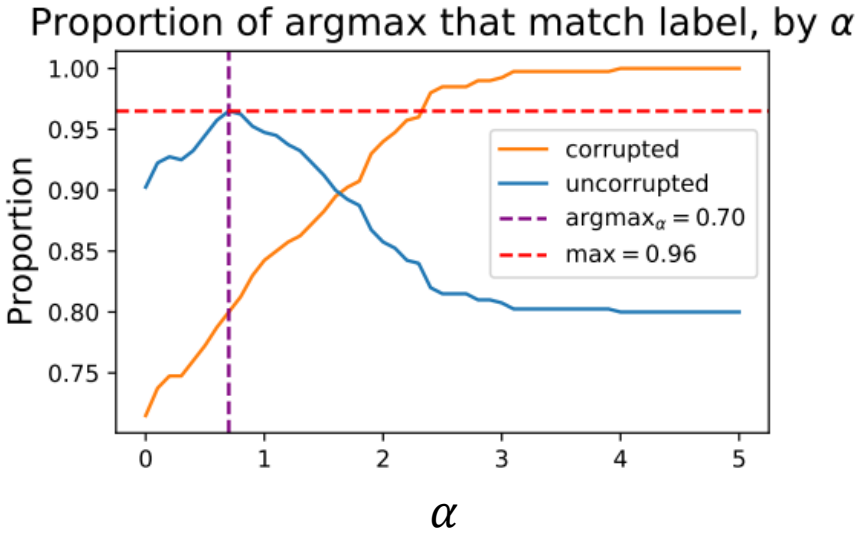
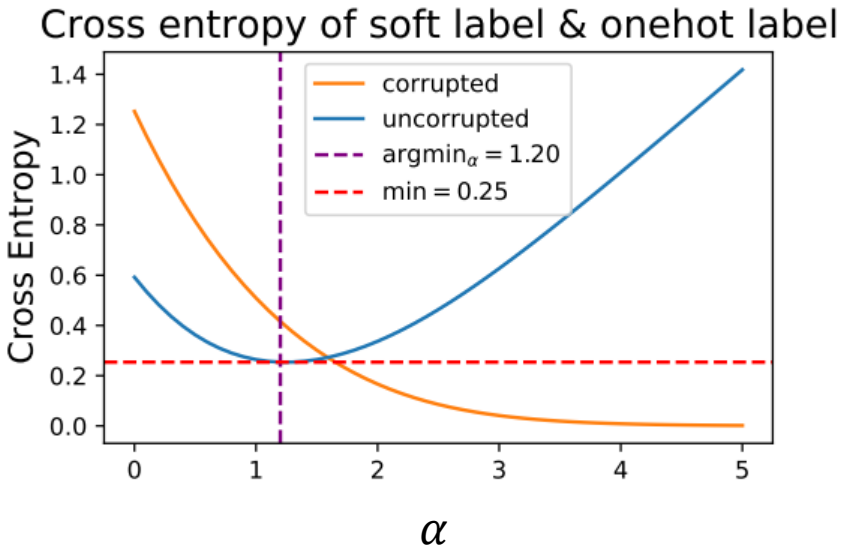


Figure 5.9: Boxplots of the test accuracy for 50 models trained on different levels of optimally soft labeling

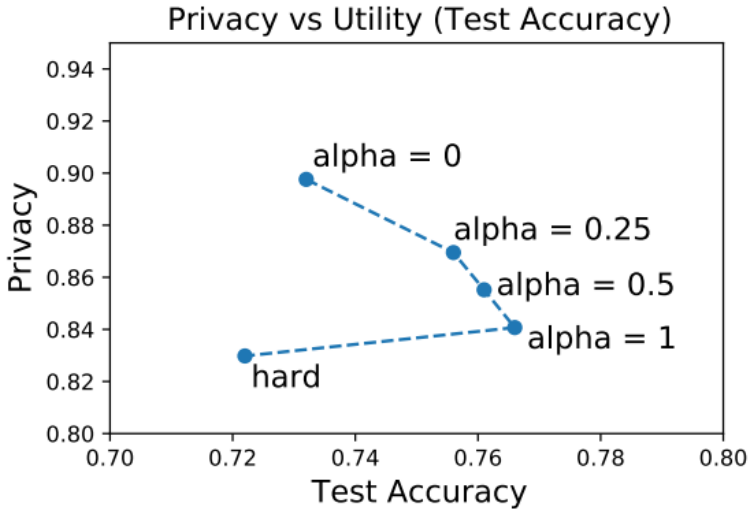
WHY Optimally Soft Labels Increase Accuracy



WHY Optimally Soft Labels Increase Accuracy



Privacy vs Test Accuracy



Thesis Contributions

- **Designed a DP algorithm** that uses provably tight bounds on WGAN, which improved the synthetic data for HealthGAN.
- **Improved metric** to estimate generative model privacy and resemblance to be provably unbiased and applicable to discrete and mixed distributions.
- **Devised a methodology** to evaluate privacy empirically using membership inference attacks, with formal analysis of privacy in this methodology.
- **Designed a defense strategy** against membership attacks of an ideal attacker that improved the privacy of an already well-defended model, providing more protection for the most exposed samples, while simultaneously increasing accuracy.

- Use our tight, component-wise bounds for WGAN with:
  - Different methods of enforcing Lipschitz constraint, such as **WGAN-SN**
  - Measure the impact of using **non-spherical Gaussian noise**
  - Measuring privacy preservation by the privacy loss parameter (ε), nearest neighbor Adversarial Accuracy, and against the LTU Attacker, and comparing them
- Test optimally soft labels against more models/datasets:
  - Try adjusting the value alpha individually on each sample
  - Try iterating the soft labels, perhaps only of the samples that need the most protection, to try to achieve more privacy
  - Try this technique with regression models

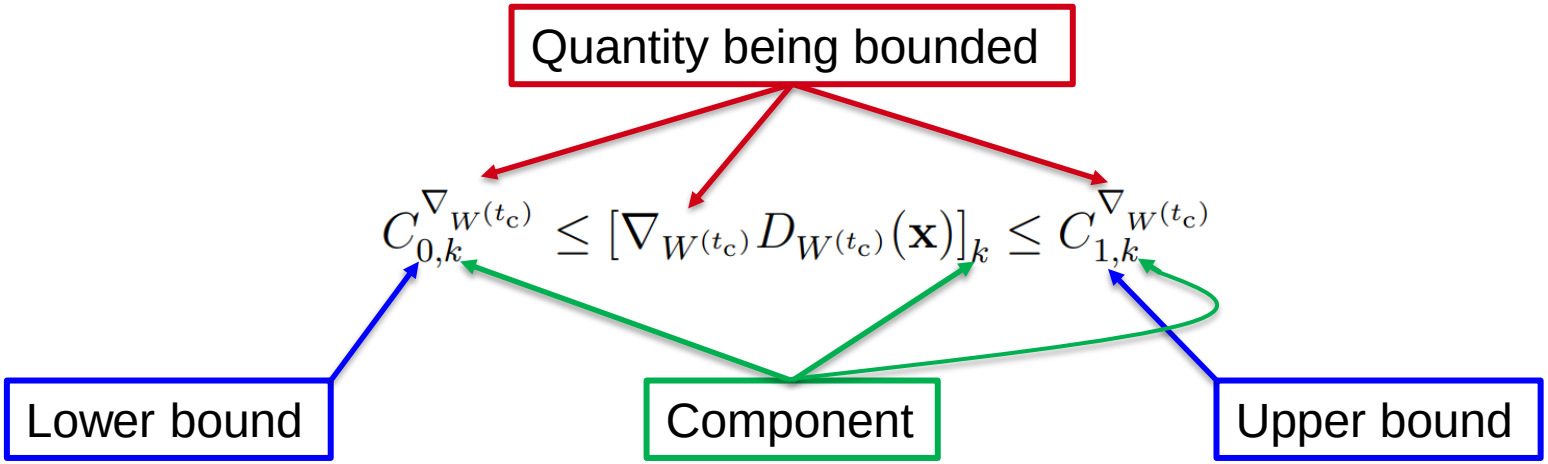
Questions

- Thank you for your attention.
- Please ask me any questions that you have
- My slides and other work are located here:
  - https://github.com/JosephMPedersen

Rensselaer

why not change the world?®

TIGHT WGAN BOUND DERIVATION



Fully Connected Feedforward

$$s_j^\ell = \left(\sum_i W_{ij}^\ell x_i^{\ell-1} \right) + b_j^\ell$$

$$x_j^\ell = \phi_{\alpha^\ell} \left(s_j^\ell \right)$$

Bounds Propagate Forward

$$C_{0,i}^{x^{\ell-1}} \leq x_i^{\ell-1} \leq C_{1,i}^{x^{\ell-1}}$$

$$C_{0,j}^{s^\ell} = \sum_{i:W_{ij}^\ell \geq 0} W_{ij}^\ell C_{0,i}^{x^{\ell-1}} + \sum_{j:W_{ij}^\ell < 0} W_{ij}^\ell C_{1,i}^{x^{\ell-1}}$$

$$C_{1,j}^{s^\ell} = \sum_{i:W_{ij}^\ell \geq 0} W_{ij}^\ell C_{1,i}^{x^{\ell-1}} + \sum_{j:W_{ij}^\ell < 0} W_{ij}^\ell C_{0,i}^{x^{\ell-1}}$$

$$C_{0,j}^{x^\ell} = \phi_{\alpha^\ell} \left(C_{0,j}^{s^\ell} \right) \leq x_j^\ell \leq \phi_{\alpha^\ell} \left(C_{1,j}^{s^\ell} \right) = C_{1,j}^{x^\ell}$$

Backprop Equations

$$\nabla_{b^\ell} e = \phi'_{\alpha^\ell}(s^\ell) \odot \nabla_{x^\ell} e$$

$$\nabla_{x^{\ell-1}} e = W^\ell \nabla_{b^\ell} e$$

$$\nabla_{W^\ell} e = (\nabla_{x^{\ell-1}} e)\,(\nabla_{b^\ell} e)^T$$

Bounds on vector/matrix operations

$$a_i \le y_i \le b_i \text{ and } c_j \le z_j \le d_j$$

$$\text{Let } M_{ij} = \max\{a_i c_j, a_i d_j, b_i c_j, b_i d_j\}, \text{ and}$$

$$m_{ij} = \min\{a_i c_j, a_i d_j, b_i c_j, b_i d_j\}, \text{ then}$$

$$m_{ii} \le [y \odot z]_i \le M_{ii}, \text{ if } y \odot z \text{ is defined}$$

$$m_{ij} \le \left[yz^T\right]_{ij} \le M_{ij}$$

TIGHT WGAN BOUND DERIVATION

Gradient of Bias

$$C_{0,j}^{\nabla_{b^\ell}} = \min\{C_{0,j}^{\nabla_{x^\ell}} C_{0,j}^{\phi'_{\alpha^\ell}}, C_{0,j}^{\nabla_{x^\ell}} C_{1,j}^{\phi'_{\alpha^\ell}}, C_{1,j}^{\nabla_{x^\ell}} C_{0,j}^{\phi'_{\alpha^\ell}}, C_{1,j}^{\nabla_{x^\ell}} C_{1,j}^{\phi'_{\alpha^\ell}}\}$$

$$C_{1,j}^{\nabla_{b^\ell}} = \max\{C_{0,j}^{\nabla_{x^\ell}} C_{0,j}^{\phi'_{\alpha^\ell}}, C_{0,j}^{\nabla_{x^\ell}} C_{1,j}^{\phi'_{\alpha^\ell}}, C_{1,j}^{\nabla_{x^\ell}} C_{0,j}^{\phi'_{\alpha^\ell}}, C_{1,j}^{\nabla_{x^\ell}} C_{1,j}^{\phi'_{\alpha^\ell}}\}$$

Gradient of Layer

$$C_{0,j}^{\nabla_{x^{\ell-1}}} = \sum_{i:W_{ij}^\ell \geq 0} W_{ij}^\ell C_{0,j}^{\nabla_{b^\ell}} + \sum_{j:W_{ij}^\ell < 0} W_{ij}^\ell C_{1,j}^{\nabla_{b^\ell}}$$

$$C_{1,j}^{\nabla_{x^{\ell-1}}} = \sum_{i:W_{ij}^\ell \geq 0} W_{ij}^\ell C_{1,j}^{\nabla_{b^\ell}} + \sum_{j:W_{ij}^\ell < 0} W_{ij}^\ell C_{0,j}^{\nabla_{b^\ell}}$$

Gradient of Weights

$$C_{0,j}^{\nabla_{W_{ij}^\ell}} = \min\{C_{0,j}^{\nabla_{x^{\ell-1}}} C_{0,j}^{\nabla_{b^\ell}}, C_{0,j}^{\nabla_{x^{\ell-1}}} C_{1,j}^{\nabla_{b^\ell}}, C_{1,j}^{\nabla_{x^{\ell-1}}} C_{0,j}^{\nabla_{b^\ell}}, C_{1,j}^{\nabla_{x^{\ell-1}}} C_{1,j}^{\nabla_{b^\ell}}\}$$

$$C_{1,j}^{\nabla_{W_{ij}^\ell}} = \max\{C_{0,j}^{\nabla_{x^{\ell-1}}} C_{0,j}^{\nabla_{b^\ell}}, C_{0,j}^{\nabla_{x^{\ell-1}}} C_{1,j}^{\nabla_{b^\ell}}, C_{1,j}^{\nabla_{x^{\ell-1}}} C_{0,j}^{\nabla_{b^\ell}}, C_{1,j}^{\nabla_{x^{\ell-1}}} C_{1,j}^{\nabla_{b^\ell}}\}$$

Using the chain rule, with a dummy variable z, and ignoring the non-differentiability at the origin, we have:

Gradient Penalty

$$\nabla_z \left(\|\nabla_x D(x)\|_2 - 1 \right)^2 = 2 \left[\left(\|\nabla_x D(x)\|_2 - 1 \right) \right] \nabla_z \|\nabla_x D(x)\|_2 \tag{3.24}$$

$$\nabla_z \|\nabla_x D(x)\|_2 = \nabla_z \left(\nabla_x D(x) \right) \nabla_x D(x) / \|\nabla_x D(x)\|_2 \tag{3.25}$$

Unit vector

$$\| \nabla_z \left(\|\nabla_x D(x)\|_2 \right) \|_F \leq \| \nabla_z \left(\nabla_x D(x) \right) \|_F \tag{3.26}$$

The gradient term $\nabla_x D(x)$ is computed using standard back propagation:

$$\frac{\partial D(x)}{\partial x_i^{\ell-1}} = \sum_j \left(\frac{\partial x_j^\ell}{\partial x_i^{\ell-1}}\right)\left(\frac{\partial D(x)}{\partial x_j^\ell}\right) = \sum_j \phi'_{\alpha^\ell}\left(s_j^\ell\right) W_{ij}^\ell \left(\frac{\partial D(x)}{\partial x_j^\ell}\right)$$

$$\nabla_{x^{\ell-1}} D(x) = \left(W^\ell\right)\left(\phi'_{\alpha^\ell}\right)\left(\nabla_{x^\ell} D(x)\right)$$

$$\nabla_{x^0} D(x) = \left(W^1\right)\left(\phi'_{\alpha^1}\right)\ldots\left(W^L\right)\left(\phi'_{\alpha^L}\right)\left(W^{L+1}\right)\left(\phi'_{\alpha^{L+1}}\right)$$

$$= \prod_{\ell=1}^{L+1}\left(W^\ell\right)\left(\phi'_{\alpha^\ell}\right)$$

- Since ReLU and LeakyReLU are piecewise linear
  - Second derivative is zero a.e. (by convention, everywhere)
  - Gradient w.r.t. bias is zero
  - Gradient w.r.t. weights is given by:

$$\nabla_{W^\ell}\left(\nabla_x D(x)\right)\nabla_x D(x)$$
$$= \left(\prod_{1\le k<\ell}\left(W^k\right)\left(\phi'_{\alpha^k}\right)\right)^T \nabla_x D(x)\left(\prod_{\ell<k\le L+1}\left(\phi'_{\alpha^{\ell-1}}\right)\left(W^\ell\right)\right)^T$$

- Thinking of the weights as matrices

$$\|\nabla_{W^\ell}\left(\nabla_x D(x)\right)\|_F \leq \prod_{k\neq\ell}\|W^k\|_F \prod_{1\leq k\leq L}\|\phi'_{\alpha^k}\|_F$$

- Combining yields the final form of the bound

$$\|\nabla_{W^\ell}\left(\|\nabla_x D(x)\|_2 - 1\right)^2\|_F \leq 2\max\{\left(\|\nabla_x D(x)\|_2 - 1\right), 1\}\prod_{k\neq\ell}\|W^k\|_F \prod_{1\leq k\leq L}\|\phi'_{\alpha^k}\|_F$$

Theorem 4.1.0.1. *For i.i.d. random samples S and T, both of the same size n, drawn from the same continuous distribution P on metric space $\mathcal{X}$, with probability density function $p(x)$,* $\mathbb{E}\left[\mathcal{AA}_{TS}\right] = 0.5$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\mathcal{AA}_{TS}\right] &= \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\mathbf{1}\left(d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)\right)\right]\right) \\
&+ \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\mathbf{1}\left(d_{S(T\setminus\{\mathbf{x}_T^k\})}(i) > d_{SS}(i)\right)\right]\right) \\
&= \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\Pr\left[d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)\right]\right) \\
&+ \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{k=1}^{n}\Pr\left[d_{S(T\setminus\{\mathbf{x}_T^k\})}(i) > d_{SS}(i)\right]\right)
\end{aligned}$$

NEAREST NEIGHBOR ADVERSARIAL ACCURACY PROOF UNBIASED

For any independently drawn random samples A and B, and any point $a_i \in A$, $d_{AB}(i)$ is a random variable taking values in $[0, \infty)$ with the conditional CDF $F(d_{AB}(i)|a_i)$ given by:

$$\begin{aligned}
\Pr\left[d_{AB}(i) \leq z|a_i\right] &= 1 - \Pr\left[d_{AB}(i) \geq z|a_i\right] \\
&= 1 - \prod_{j=1}^{n} \Pr\left[d(a_i, b_j) \geq z|a_i\right] \\
&= 1 - \left(\int_{\{\mathbf{x} \in \mathcal{X} | d(a_i, x) \geq z\}} p(\mathbf{x}) d\mathbf{x}\right)^n
\end{aligned}$$

Using this equation, we see that both $Z_1 = d_{T(S \setminus \{\mathbf{x}_S^k\})}(i)$ and $Z_2 = d_{TT}(i)$ have the same conditional CDF given x_T^i:

$$1 - \left(\int_{\{\mathbf{x} \in \mathcal{X} | d(x_T^i, \mathbf{x}) \geq z\}} p(\mathbf{x}) d\mathbf{x}\right)^{n-1}$$

NEAREST NEIGHBOR ADVERSARIAL ACCURACY PROOF UNBIASED

Furthermore, since $(S \setminus \{\mathbf{x}_S^k\})$ and $(T \setminus \{\mathbf{x}_T^i\})$ are both independent random samples, the random variables Z_1 and Z_2 are conditionally independent, given x_T^i. Therefore, their conditional joint PDF is the product of their conditional marginal PDFs: $f_{Z_1,Z_2}(z_1, z_2|x_T^i) = f_{Z_1}(z_1|x_T^i) f_{Z_2}(z_2|x_T^i)$. By symmetry, we see that:

$$\Pr\left[d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)|x_T^i\right] = 0.5$$

Since this is true for any x_T^i, we have:

$$\Pr\left[d_{T(S\setminus\{\mathbf{x}_S^k\})}(i) > d_{TT}(i)\right] = \int_{\mathcal{X}} (0.5)p(\mathbf{x})d\mathbf{x} = 0.5$$

- # Bounded Loss Function Attack

Theorem 4.2.2.2. *The Oracle Attacker can attain the same lower bound on accuracy connected to overfitting as the bounded loss function (BLF) adversary of [37].*

Proof. If pair order is random and $0 \leq f(x) \leq B$, then the Oracle Attacker could predict $u_1 \in \mathcal{D}_R$ with probability $f(u_1)/B$, and $u_1 \in \mathcal{D}_D$ otherwise, with accuracy:

Advantage

$$A_{ltu} = \frac{1}{2} \mathop{\mathbb{E}}_{u_1 \in \mathcal{D}_R} \left[\frac{f(u_1)}{B} \right] + \frac{1}{2} \mathop{\mathbb{E}}_{u_1 \in \mathcal{D}_D} \left[1 - \frac{f(u_1)}{B} \right] = \frac{1}{2} + \boxed{\frac{e_R - e_D}{2B}}$$

$$\text{where } e_R := \mathop{\mathbb{E}}_{u_1 \in \mathcal{D}_R} [f(u_1)] \text{ and } e_D := \mathop{\mathbb{E}}_{u_1 \in \mathcal{D}_D} [f(u_1)] \tag{4.3}$$

□

Similar to the BLF adversary of: Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pages 268–282. IEEE, 2018.

Proof. Denote the subset of $\mathcal{D}_A$ labeled "Defender" as $\mathcal{D}_D \setminus \{d\}$, and the two membership unlabeled points as u_1 and u_2. The attacker can use the Defender trainer with the same hyper-parameters on $(\mathcal{D}_D \setminus \{d\}) \cup \{u_1\}$ to produce model $\mathcal{M}_1$ and on $(\mathcal{D}_D \setminus \{d\}) \cup \{u_2\}$ to produce model $\mathcal{M}_2$.

By definition of the Oracle Attacker, the missing point d is either u_1 or u_2, and $\mathcal{D}_D \cap \mathcal{D}_R = \emptyset$, so $u_1 \neq u_2$.

There are two possible cases. If $u_1 = d$, then $\mathcal{D}_D = (\mathcal{D}_D \setminus \{d\}) \cup \{u_1\}$, so that $\mathcal{M}_1 = \mathcal{M}_D$, since $\mathcal{T}_D$ is deterministic and invariant to the order of the training data. However, $\mathcal{D}_D \neq (\mathcal{D}_D \setminus \{d\}) \cup \{u_2\}$, since $u_2 \neq u_1$, so $\mathcal{M}_2 \neq \mathcal{M}_D$, since $\mathcal{T}_D$ is also injective. Therefore, the Oracle Attacker can know, with no uncertainty, that u_1 has membership label "Defender" and u_2 has membership label "Reserved". The other case, for $u_2 = d$, has a symmetric argument. □

Theorem 3.1.2.1. *(Non-Spherical Gaussian Mechanism) Given any funtion* $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ *whose image is bounded by* $\forall i = 1, \ldots, k, a_i \leq f(D)_i \leq b_i$*, the following query mechanism* $q(D)$ *is* (ε, δ)*-differentially private, and the trace of the covariance matrix is the same as that for the Gaussian mechanism.*

$$q(D) = f(D) + \mathcal{N}_k(0, \Sigma), \text{ where}$$

$$\Sigma_{ij} = \begin{cases} k(b_i - a_i)^2 2 \log\left(\frac{5}{4\delta}\right) / \varepsilon^2 & i = j \\ 0 & i \neq j \end{cases}$$

Proof. Define $r_i = \frac{b_i - a_i}{2}$, $\mu_i = \frac{a_i + b_i}{2}$, and $\widetilde{f}(D) = \widetilde{\mathbf{Y}}$, where $\widetilde{Y}_i = (f(D)_i - \mu_i)/r_i$. Then $\forall i = 1, \ldots, k, -1 \leq \widetilde{Y}_i \leq 1$, so the ℓ_2-sensitivity of $\widetilde{f}$ is $\sqrt{\sum_{i=1}^{k}(2)^2} = 2\sqrt{k}$. Using the Gaussian mechanism, this means that $\widetilde{q}(D) = \widetilde{f}(D) + \mathcal{N}_k\left(0, 4k\left[2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2\right]\mathbf{I}_k\right)$ is (ε, δ)-differentially private.

By the immunity to post-processing proposition, if we define $q_2(D)$ such that: $q_2(D)_i = r_i\widetilde{q}(D)_i + \mu_i$, then $q_2(D)$ is also (ε, δ)-differentially private.

That equation makes $q_2(D)_i$ a random variable given by: $q_2(D)_i = f(D)_i + \mathcal{N}(0, k(b_i - a_i)^2 2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2)$, proving that $q(D)$ is equivalent to $q_2(D)$.

The ℓ_2-sensitivity of f is $\sqrt{\sum_{i=1}^{k}(b_i - a_i)^2}$, so the Gaussian mechanism on f would be given by: $f(D) + \mathcal{N}_k\left(0, \sum_{i=1}^{k}(b_i - a_i)^2 \left[2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2\right]\mathbf{I}_k\right)$.

The trace of that covariance matrix is given by:

$$\begin{aligned}\operatorname{tr}\left(\sum_{i=1}^{k}(b_i - a_i)^2\left[2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2\right]\mathbf{I}_k\right) &= \sum_{j=1}^{k}\sum_{i=1}^{k}(b_i - a_i)^2\left[2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2\right] \\ &= k\sum_{i=1}^{k}(b_i - a_i)^2\left[2\log\left(\frac{5}{4\delta}\right)/\varepsilon^2\right]\end{aligned}$$

□

- Overfitting Attack

- Any quantity (e.g. the loss function) that is overfit

$$p_R = \Pr_{\substack{u_1 \sim \mathcal{D}_R \\ u_2 \sim \mathcal{D}_D}} [f(u_1) > f(u_2)]$$

$$p_D = \Pr_{\substack{u_1 \sim \mathcal{D}_R \\ u_2 \sim \mathcal{D}_D}} [f(u_1) < f(u_2)]$$

Theorem 4.2.1. *If there is any function f for which $p_R > p_D$, an LTU Attacker exploiting that function can achieve an accuracy* $A_{ltu} \geq \frac{1}{2} + \frac{1}{2}(p_R - p_D)$

Advantage

- Deterministic Attack

- Two sources of randomness: random bits of algorithm, and order of the training data

Theorem 4.2.4. *If the Defender trainer $\mathcal{T}_D$ is deterministic, invariant to the order of the training data, and injective, then the LTU Attacker has an optimal attack strategy which achieves perfect accuracy.*

Optimally Soft Labels

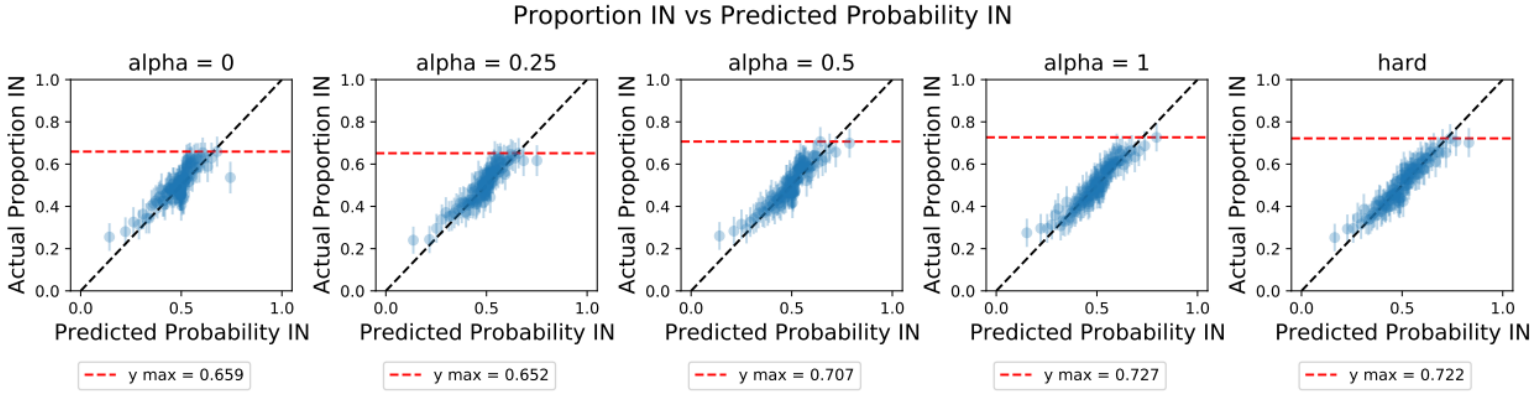


Figure 5.3: Scatter plots of the actual proportion of membership vs the predicted probability of membership, averaged over 100 groups of nearby predictions, which match well (near the 45 degree line). Using $\alpha = 0$ lowered the maximum by 6.3pp, corresponding to more privacy for the most exposed samples.

Optimally Soft Labels Increase Test Accuracy