Membership Inference Attacks for Generative Models

Josh Ward

University of California Los Angeles

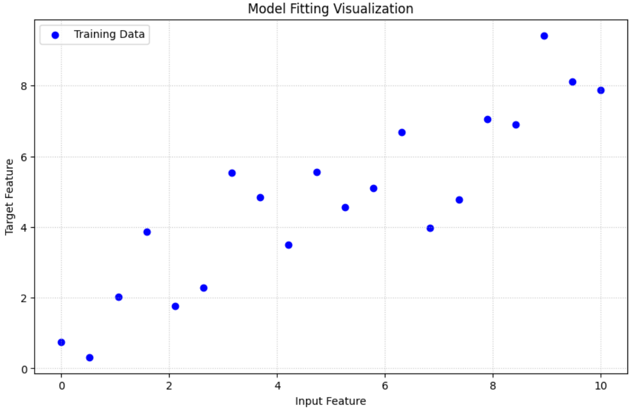
joshuaward@g.ucla.edu

March 2, 2025

Prologue: Model Overfitting

Generally, overfitting for regressors/ classifiers is well-understood

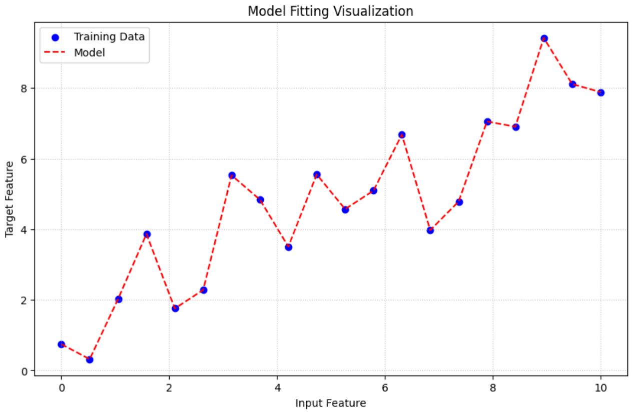
"A model that is fit too closely to patterns in the training set not present in the broader population"



Prologue: Model Overfitting

Generally, overfitting for regressors/ classifiers is well-understood

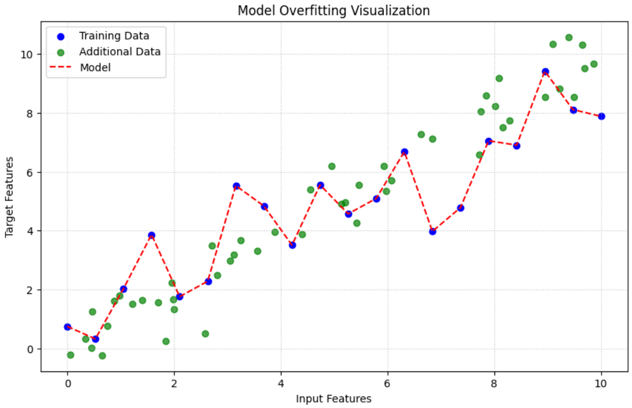
“A model that is fit too closely to patterns in the training set not present in the broader population”



Prologue: Model Overfitting

Generally, overfitting for regressors/ classifiers is well-understood

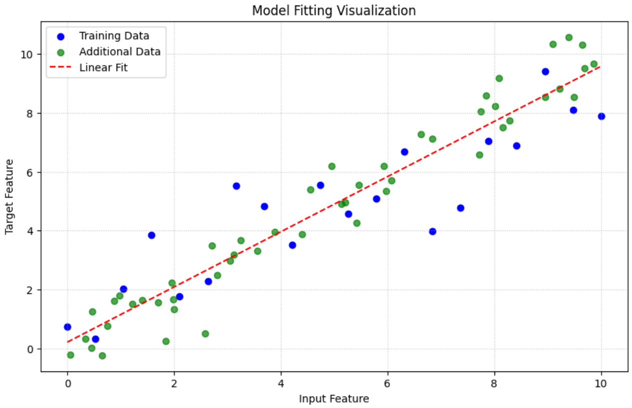
"A model that is fit too closely to patterns in the training set not present in the broader population"



Prologue: Model Overfitting

Generally, overfitting for regressors/ classifiers is well-understood

"A model that is fit too closely to patterns in the training set not present in the broader population"



Supervised Learning Model Overfitting

We have known methods for evaluating if our classifier/ regressor is "well-fit" (read not over-fit)

- Use holdout data to compare train/ test performance
- Cross validation designs
- Visual Inspection (in low dimensional cases)

Supervised Learning Model Overfitting

We have known methods for evaluating if our classifier/ regressor is "well-fit" (read not over-fit)

- Use holdout data to compare train/ test performance
- Cross validation designs
- Visual Inspection (in low dimensional cases)

This is great in the case of supervised learning algorithms because their outputs are generally "simple", computationally cheap to train, and sacrificing training size for holdout data usually doesn't significantly hurt model performance

Model Overfitting in Generative Models

Generative models are exactly the opposite:

Model Overfitting in Generative Models

Generative models are exactly the opposite:

- Their outputs are in the same space as the training data (high dimensional)

Model Overfitting in Generative Models

Generative models are exactly the opposite:

- Their outputs are in the same space as the training data (high dimensional)
- They are difficult to interpret (Deep Neural Networks)

Model Overfitting in Generative Models

Generative models are exactly the opposite:

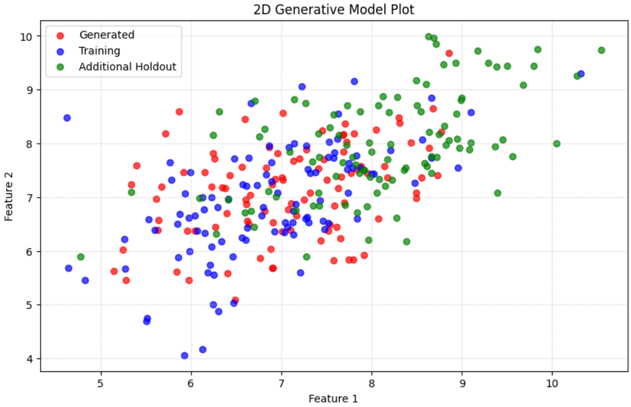
- Their outputs are in the same space as the training data (high dimensional)
- They are difficult to interpret (Deep Neural Networks)
- They are usually expensive to train (no cross validation, and even if you did, what would you evaluate?)

Model Overfitting in Generative Models

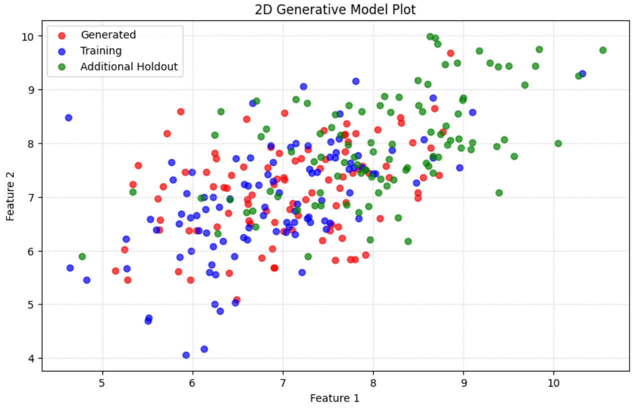
Generative models are exactly the opposite:

- Their outputs are in the same space as the training data (high dimensional)
- They are difficult to interpret (Deep Neural Networks)
- They are usually expensive to train (no cross validation, and even if you did, what would you evaluate?)
- They greatly benefit from additional training data

Generative Model Overfitting

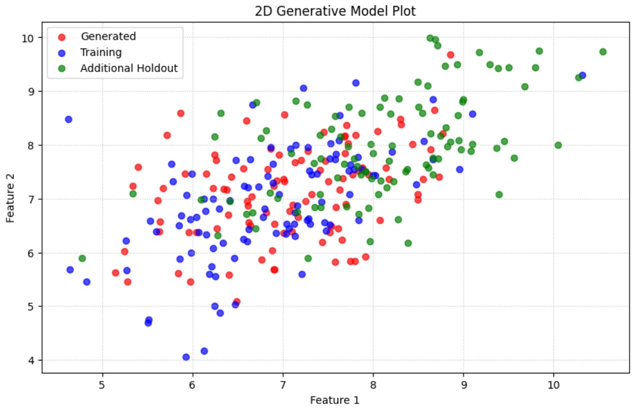


Generative Model Overfitting



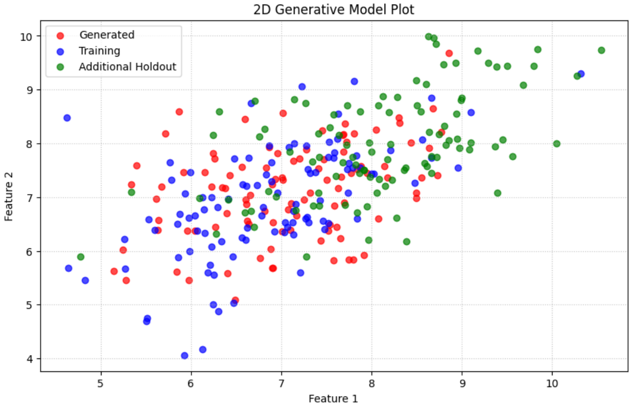
1. Is this generative model overfit?

Generative Model Overfitting



1. Is this generative model overfit?
2. Are there specific regions where the model is overfit?

Generative Model Overfitting



1. Is this generative model overfit?
2. Are there specific regions where the model is overfit?
3. What are even the problems with a generative model being overfit?

Spoiler

Spoiler: It's an active area of research for how to answer those questions.

Spoiler

Spoiler: It's an active area of research for how to answer those questions.

These are actually 3 multivariate gaussians with the same covariance matrices. The means of the training and generated data are the same, the holdout population data differs in mean by 1 unit.

Spoiler

Spoiler: It's an active area of research for how to answer those questions.

These are actually 3 multivariate gaussians with the same covariance matrices. The means of the training and generated data are the same, the holdout population data differs in mean by 1 unit.

We'll come back to this.

Talk Outline

Talk Outline:

1. Introduce Membership Inference Attacks as a framework for understanding overfitting/ privacy

Talk Outline

Talk Outline:

1. Introduce Membership Inference Attacks as a framework for understanding overfitting/ privacy
2. Distance based attacks: "Data Plagiarism Index"- Ward et al 2024a

Talk Outline

Talk Outline:

1. Introduce Membership Inference Attacks as a framework for understanding overfitting/ privacy
2. Distance based attacks: "Data Plagiarism Index"- Ward et al 2024a
3. Density based attacks: "Generative Likelihood Ratio Attack"- Ward et al 2024b

Talk Outline

Talk Outline:

1. Introduce Membership Inference Attacks as a framework for understanding overfitting/ privacy
2. Distance based attacks: "Data Plagiarism Index"- Ward et al 2024a
3. Density based attacks: "Generative Likelihood Ratio Attack"- Ward et al 2024b
4. Future Work

Introduction to Membership Inference Attacks

We'd like a framework to evaluate overfitting in generative models that has the following features:

Introduction to Membership Inference Attacks

We'd like a framework to evaluate overfitting in generative models that has the following features:

- It's easily interpretable (ideally to someone with little experience in machine learning)

Introduction to Membership Inference Attacks

We'd like a framework to evaluate overfitting in generative models that has the following features:

- It's easily interpretable (ideally to someone with little experience in machine learning)
- It's cheap to run

Introduction to Membership Inference Attacks

We'd like a framework to evaluate overfitting in generative models that has the following features:

- It's easily interpretable (ideally to someone with little experience in machine learning)
- It's cheap to run
- It highlights real, material risks as a consequence of overfitting

Introduction to Membership Inference Attacks

General Idea: Given the knowledge of a model defined by a *threat model*, how well can an adversary distinguish training data from non-training data?

Introduction to Membership Inference Attacks

General Idea: Given the knowledge of a model defined by a *threat model*, how well can an adversary distinguish training data from non-training data?

1. Sample a training dataset $T = {x_i}_{i=1}^{n}$ from the population distribution $x_i \sim \mathbb{P}$ and uses T to train a generative model $G \leftarrow \mathcal{T}(T)$.

Introduction to Membership Inference Attacks

General Idea: Given the knowledge of a model defined by a *threat model*, how well can an adversary distinguish training data from non-training data?

1. Sample a training dataset $T = {x_i}_{i=1}^n$ from the population distribution $x_i \sim \mathbb{P}$ and uses T to train a generative model $G \leftarrow \mathcal{T}(T)$.
2. The generative model G produces a synthetic dataset S.

Introduction to Membership Inference Attacks

General Idea: Given the knowledge of a model defined by a *threat model*, how well can an adversary distinguish training data from non-training data?

1. Sample a training dataset $T = {x_i}_{i=1}^{n}$ from the population distribution $x_i \sim \mathbb{P}$ and uses T to train a generative model $G \leftarrow \mathcal{T}(T)$.
2. The generative model G produces a synthetic dataset S.
3. Take a test observation $x^\star$ from the population distribution $\mathbb{P}$ OR from the training set T.

Introduction to Membership Inference Attacks

General Idea: Given the knowledge of a model defined by a *threat model*, how well can an adversary distinguish training data from non-training data?

1. Sample a training dataset $T = x_i{}_{i=1}^{n}$ from the population distribution $x_i \sim \mathbb{P}$ and uses T to train a generative model $G \leftarrow \mathcal{T}(T)$.
2. The generative model G produces a synthetic dataset S.
3. Take a test observation $x^{\star}$ from the population distribution $\mathbb{P}$ OR from the training set T.
4. An adversary $\mathcal{A}$ then, given some information about G and S, attempts to infer if x^{*} is an element of T

Membership Inference Attack Formalism

Mathematically, we express an MIA as:

$$\mathcal{A}(x^\star) = \mathbb{I}\left[f(x^\star) > \gamma\right] \tag{1}$$

where f is some scoring function and γ is some decision thresholding rule.

Membership Inference Attack Formalism

Mathematically, we express an MIA as:

$$\mathcal{A}(x^{\star}) = \mathbb{I}\left[f(x^{\star}) > \gamma\right] \tag{1}$$

where f is some scoring function and γ is some decision thresholding rule.

Notes:

- Our goal as an Adversary is to develop an f that best exploits the information provided to us from the threat model that leads to the most powerful classifier possible
- We can evaluate this classifier with traditional binary classification metrics (AUC-ROC, TPR, Accuracy, etc)
- The Advesary does not have labels in which to construct a classifier (Unsupervised Learning Problem)

Threat Models

The Threat Model describes the information $\mathcal{A}$ has in which to construct f.

Threat Models

The Threat Model describes the information $\mathcal{A}$ has in which to construct f.

- White-Box: The Adversary has access to model weights, hyperparameters, and S

Threat Models

The Threat Model describes the information $\mathcal{A}$ has in which to construct f.

- White-Box: The Adversary has access to model weights, hyperparameters, and S
- Shadow-Box: The Adversary has access to only S and a reference dataset $R \sim \mathbb{P}$

Threat Models

The Threat Model describes the information $\mathcal{A}$ has in which to construct f.

- White-Box: The Adversary has access to model weights, hyperparameters, and S
- Shadow-Box: The Adversary has access to only S and a reference dataset $R \sim \mathbb{P}$
- Black-Box: The Adversary has access to only S.

Shadow-Box Attacks

Intuition: Given the output of a model S and some reference dataset R, we wish to infer the membership of the training data. If the model is overfit, S will be "too much" like T relative to R and thus leak information about T. The better we are at inferring the membership of x^*, the more overfit our model is.

Shadow-Box Attacks

Intuition: Given the output of a model S and some reference dataset R, we wish to infer the membership of the training data. If the model is overfit, S will be "too much" like T relative to R and thus leak information about T. The better we are at inferring the membership of x^*, the more overfit our model is.

"A model that is fit too closely to patterns in the training set not present in the broader population."

Shadow-Box Attacks

Intuition: Given the output of a model S and some reference dataset R, we wish to infer the membership of the training data. If the model is overfit, S will be "too much" like T relative to R and thus leak information about T. The better we are at inferring the membership of x^*, the more overfit our model is.

"A model that is fit too closely to patterns in the training set not present in the broader population."

Further Comments on MIAs

Membership Inference Attacks frame overfitting/ data copying as a privacy risk and are easily interpretable with classificiation metrics.

Examples: Memorizing Hospital Records or Watermarks on Images

Training Data Privacy Leakage is a function of overfitting/ data memorization. While MIAs are a privacy auditing technique, they characterize how models compromise privacy **because of being overfit to** the training distribution.

Questions Before Continuing?

Previous Membership Inference Attacks

Remember, our goal is to construct an unsupervised classifier for x^* based on R and S.

Previous Membership Inference Attacks

Remember, our goal is to construct an unsupervised classifier for x^* based on R and S.

1. "Train a classifier on S and R, call it on x^*"- Logan et al 2017

Previous Membership Inference Attacks

Remember, our goal is to construct an unsupervised classifier for x^* based on R and S.

1. "Train a classifier on S and R, call it on x^*"- Logan et al 2017
2. "Compute the distance to the closest record in S from x^*- Chen et al 2020

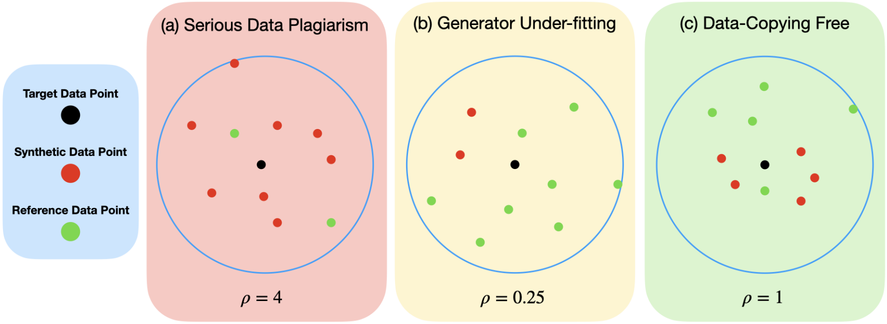
Previous Membership Inference Attacks

Remember, our goal is to construct an unsupervised classifier for x^* based on R and S.

1. "Train a classifier on S and R, call it on x^*"- Logan et al 2017
2. "Compute the distance to the closest record in S from x^*- Chen et al 2020
3. "Compute the density ratio of x^* over S and R- Beugal et al 2023

Data Plagiarism Index: Ward et al 2024a

Idea: Analyze the local neighborhood around x^*. If there are a disproportionate amount of observations from S relative to R, conclude that $x^* \in T$.



DPI Formalized

Denote $D(x^*)$ around x^* as the the K-Nearest Neighborhood on the space with R and S. Then:

$$f(x^*) \equiv \frac{\sum_{x_i \in D(x^*)} I(x_i^* \in S)}{\sum_{x_i \in D(x^*)} I(x_i \in R)} \tag{2}$$

DPI Formalized

Denote $D(x^*)$ around x^* as the the K-Nearest Neighborhood on the space with R and S. Then:

$$f(x^*) \equiv \frac{\sum_{x_i \in D(x^*)} I(x_i^* \in S)}{\sum_{x_i \in D(x^*)} I(x_i \in R)} \tag{2}$$

Notes:

DPI Formalized

Denote $D(x^*)$ around x^* as the the K-Nearest Neighborhood on the space with R and S. Then:

$$f(x^*) \equiv \frac{\sum_{x_i \in D(x^*)} I(x_i^* \in S)}{\sum_{x_i \in D(x^*)} I(x_i \in R)} \tag{2}$$

Notes:

1. The only hyperparameters are choice of K and measure of distance

DPI Formalized

Denote $D(x^*)$ around x^* as the the K-Nearest Neighborhood on the space with R and S. Then:

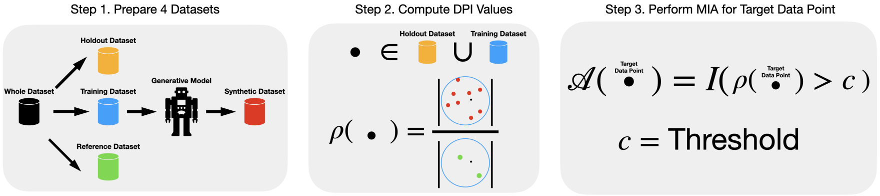
$$f(x^*) \equiv \frac{\sum_{x_i \in D(x^*)} I(x_i^* \in S)}{\sum_{x_i \in D(x^*)} I(x_i \in R)} \tag{2}$$

Notes:

1. The only hyperparameters are choice of K and measure of distance
2. Fast to run: for each x^*, K-closest neighbor lookup over S and R

Membership Inference Attacks in Practice

To actually implement the attack for auditing:



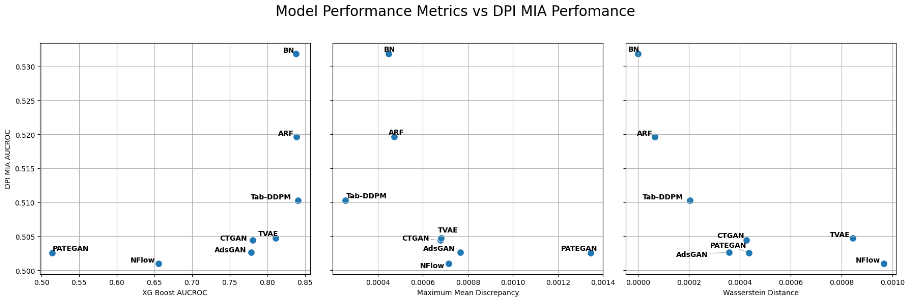
Membership Inference Attacks in Practice

Experiment:

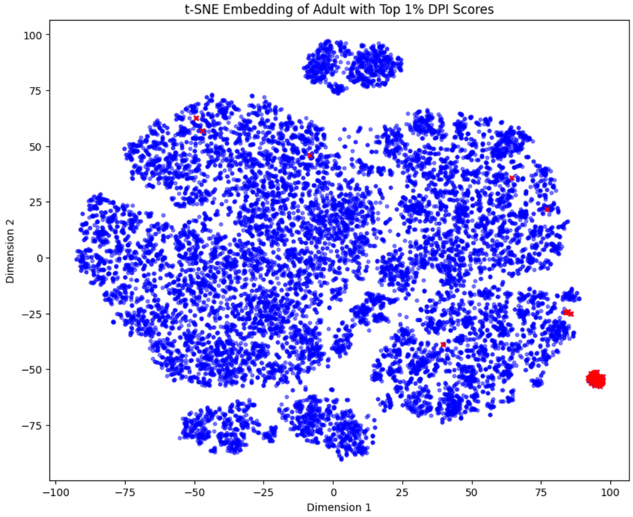
1. Adult Census: a dataset of demographic data describing adults
2. Split into equal size Training/ Reference/ Holdout sets of N=4000
3. Train different tabular generative models and evaluate DPI

Data Plagiarism Index Results

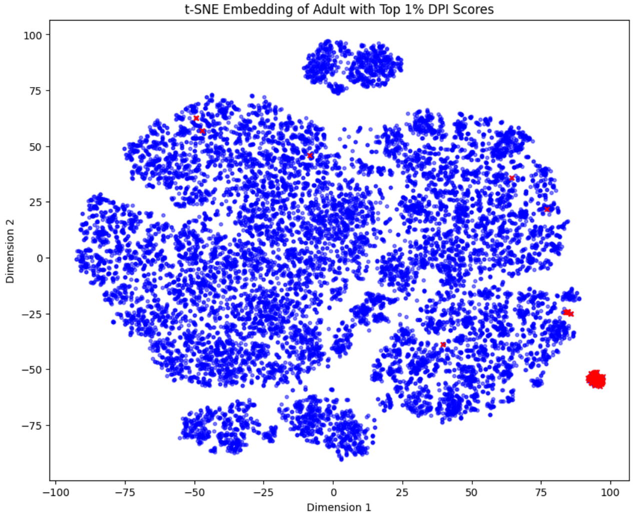
Overfitting/ Privacy Risk is correlated with higher performing models:



Data Plagiarism Index Results

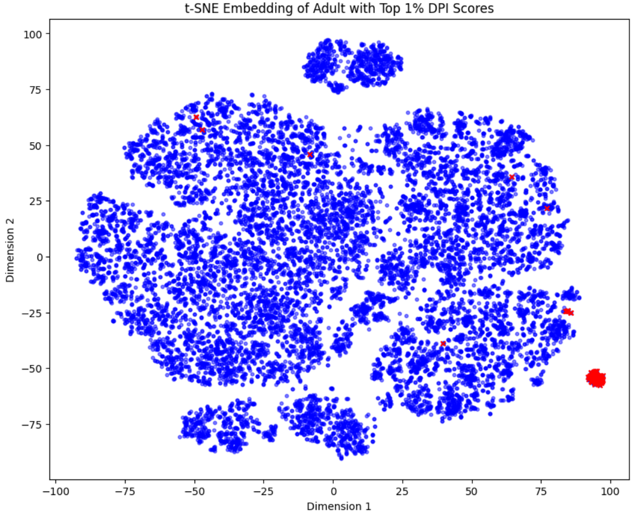


Data Plagiarism Index Results



1. In red, the top 1% of highest scored training data

Data Plagiarism Index Results



1. In red, the top 1% of highest scored training data
2. All of these observations are white, married, US, high investment returns, high income, men

DPI: Summary

DPI attacks generative models by analyzing a local neighborhood around candidate observations.

DPI: Summary

DPI attacks generative models by analyzing a local neighborhood around candidate observations.

Opportunities for Improvement:

DPI: Summary

DPI attacks generative models by analyzing a local neighborhood around candidate observations.

Opportunities for Improvement:

- Relies on hyperparameters to define a neighborhood

DPI: Summary

DPI attacks generative models by analyzing a local neighborhood around candidate observations.

Opportunities for Improvement:

- Relies on hyperparameters to define a neighborhood
- Fixed Choice of K might not be optimal for all x^*

DPI: Summary

DPI attacks generative models by analyzing a local neighborhood around candidate observations.

Opportunities for Improvement:

- Relies on hyperparameters to define a neighborhood
- Fixed Choice of K might not be optimal for all x^*
- Ad-hoc: there is not a strong theoretical backing for why this works.

Generative Likelihood Ratio Attack (Gen-LRA)

Idea: Treat Membership Inference as a hypothesis test

Generative Likelihood Ratio Attack (Gen-LRA)

Idea: Treat Membership Inference as a hypothesis test

The *null hypothesis* H_0 assumes that the synthetic data follows the population distribution $\mathbb{P}$, meaning that the generative model correctly models $\mathbb{P}$

$$H_0 : p(S|H_0) = \prod_{s \in S} p_{\mathbb{P}}(s) \tag{3}$$

In contrast, the *alternative hypothesis* H_1 assumes that the generative model overfits near $x^\star$, resulting in a modified probability distribution $p_{\mathbb{P} \cup \{x^\star\}}(s)$, which places additional weight on the vicinity of $x^\star$.

$$H_1 : p(S|H_1) = \prod_{s \in S} p_{\mathbb{P} \cup \{x^\star\}}(s) \tag{4}$$

Gen-LRA Formalized

By the Neyman-Pearson Lemma, we can write this as a Likelihood Ratio test:

$$\lambda_{\mathbb{P}}(S, x^\star) = \frac{\prod_{s \in S} p_{\mathbb{P} \cup \{x^\star\}}(s)}{\prod_{s \in S} p_{\mathbb{P}}(s)} \tag{5}$$

Or in the sample case because $R \sim \mathbb{P}$:

$$\lambda_R(S, x^\star) = \frac{\prod_{s \in S} p_{R \cup \{x^\star\}}(s)}{\prod_{s \in S} p_R(s)} \tag{6}$$

Gen-LRA Formalized

By the Neyman-Pearson Lemma, we can write this as a Likelihood Ratio test:

$$\lambda_{\mathbb{P}}(S, x^\star) = \frac{\prod_{s \in S} p_{\mathbb{P} \cup \{x^\star\}}(s)}{\prod_{s \in S} p_{\mathbb{P}}(s)} \tag{7}$$

Or in the sample case because $R \sim \mathbb{P}$:

$$\lambda_R(S, x^\star) = \frac{\prod_{s \in S} p_{R \cup \{x^\star\}}(s)}{\prod_{s \in S} p_R(s)} \tag{8}$$

Gen-LRA Density Estimation

In practice, we have to estimate the likelihood of S over $R \cup x^*$ and R with Kernel Density Estimators or Deep Neural Networks (Normalizing Flows).

$$\hat{p}_{R,K,h}(s) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{s - r_i}{h}\right) \tag{9}$$

- n is the number of samples in the reference dataset R
- h is the bandwidth parameter that controls the smoothness of the estimate
- r_i represent individual samples from the reference dataset R
- $K\left(\frac{s-r_i}{h}\right)$ is the kernel function applied to the scaled difference between the sample s and the reference sample r_i.

Gen-LRA Formalized

Thus through substitution:

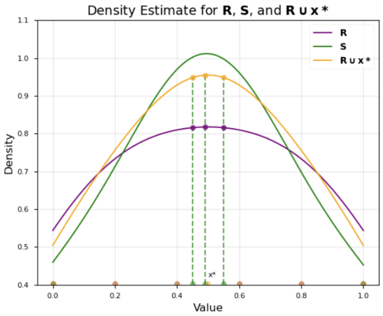
$$\lambda_{R,K}(S,x^\star) = \frac{\prod_{s\in S}\left(\frac{1}{(n+1)h}\left[\sum_{i=1}^{n} K\left(\frac{s-r_i}{h}\right) + K\left(\frac{s-x^\star}{h}\right)\right]\right)}{\prod_{s\in S}\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{s-r_i}{h}\right)\right)} \tag{10}$$

The likelihood ratio $f_{R,K}(S,x^\star)$, as computed from the KDE-based estimates, serves as our **scoring function** for membership prediction:

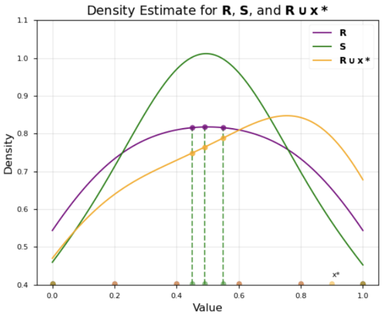
$$f(x^\star) \equiv \lambda_{R,K}(S,x^\star) \tag{11}$$

Gen-LRA Geometric Intuition

Intuition:



(a) Overfitting on x^*



(b) Underfitting on x^*

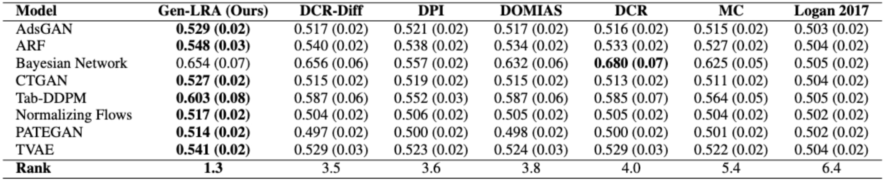
Gen-LRA Algorithm

Algorithm 1 Gen-LRA

```
Require:
 1: $\mathbf{X}_{\text{test}} \in \mathbb{R}^{n_{\text{test}} \times d}$: Test dataset
 2: $\mathbf{S} \in \mathbb{R}^{n_S \times d}$: Generated dataset
 3: $\mathbf{R} \in \mathbb{R}^{n_{\text{ref}} \times d}$: Reference dataset
 4: $k \in \mathbb{N}$: Number of closest points to compare
Ensure:
 5: $\mathbf{S}_{\text{scores}} \in \mathbb{R}^{n_{\text{test}}}$: Attack scores for test samples
 6: function GENLRATTACK($\mathbf{X}_{\text{test}}, \mathbf{S}, \mathbf{R}, k$)
 7:     $\mathbf{S}_{\text{scores}} \leftarrow \emptyset$        ▷ Initialize score array
 8:     $\text{DE}_{\mathbf{R}} \leftarrow \text{FitDensityEstimator}(\mathbf{R})$        ▷ Fit density estimator on $\mathbf{R}$
 9:     for $\mathbf{x} \in \mathbf{X}_{\text{test}}$ do
10:         $\mathbf{R}' \leftarrow \mathbf{R} \cup \{\mathbf{x}\}$        ▷ Insert $\mathbf{x}$ into reference set
11:         $\text{DE}_{\mathbf{R}'} \leftarrow \text{FitDensityEstimator}(\mathbf{R}')$        ▷ Fit density estimator on $\mathbf{R}'$
12:         $\mathbf{S}_{\text{close}} \leftarrow \text{FindKNearestNeighbors}(\mathbf{S}, \mathbf{x}, k)$        ▷ Find $k$ closest points in $\mathbf{S}$
13:         $\mathbf{L}_{\mathbf{R}'} \leftarrow \text{DE}_{\mathbf{R}'}(\mathbf{S}_{\text{close}})$        ▷ Compute likelihoods using $\text{DE}_{\mathbf{R}'}$
14:         $\mathbf{L}_{\mathbf{R}} \leftarrow \text{DE}_{\mathbf{R}}(\mathbf{S}_{\text{close}})$        ▷ Compute likelihoods using $\text{DE}_{\mathbf{R}}$
15:         $s \leftarrow \sum_{\mathbf{s} \in \mathbf{S}_{\text{close}}} \log(\mathbf{L}_{\mathbf{R}'}[\mathbf{s}]) - \sum_{\mathbf{s} \in \mathbf{S}_{\text{close}}} \log(\mathbf{L}_{\mathbf{R}}[\mathbf{s}])$        ▷ Compute log-likelihood difference
16:         $\mathbf{S}_{\text{scores}} \leftarrow \mathbf{S}_{\text{scores}} \cup \{s\}$
17:     end for
18:     return $\mathbf{S}_{\text{scores}}$
19: end function
```

Gen-LRA Results

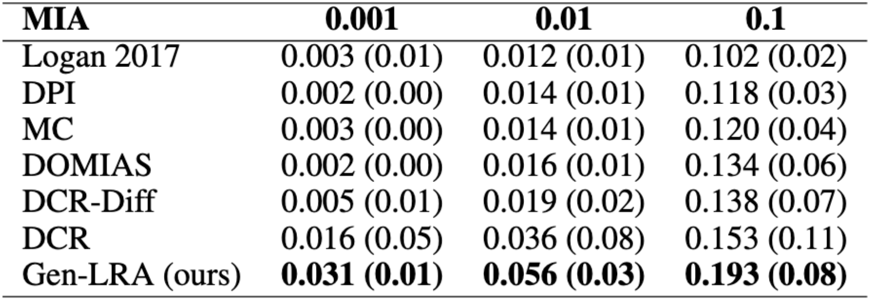
Across 15 tabular datasets and 8 generative model architectures, Gen-LRA has the highest ranked AUC-ROC



Model	**Gen-LRA (Ours)**	**DCR-Diff**	**DPI**	**DOMIAS**	**DCR**	**MC**	**Logan 2017**
AdsGAN	**0.529 (0.02)**	0.517 (0.02)	0.521 (0.02)	0.517 (0.02)	0.516 (0.02)	0.515 (0.02)	0.503 (0.02)
ARF	**0.548 (0.03)**	0.540 (0.02)	0.538 (0.02)	0.534 (0.02)	0.533 (0.02)	0.527 (0.02)	0.504 (0.02)
Bayesian Network	0.654 (0.07)	0.656 (0.06)	0.557 (0.02)	0.632 (0.06)	**0.680 (0.07)**	0.625 (0.05)	0.505 (0.02)
CTGAN	**0.527 (0.02)**	0.515 (0.02)	0.519 (0.02)	0.515 (0.02)	0.513 (0.02)	0.511 (0.02)	0.504 (0.02)
Tab-DDPM	**0.603 (0.08)**	0.587 (0.06)	0.552 (0.03)	0.587 (0.06)	0.585 (0.07)	0.564 (0.05)	0.505 (0.02)
Normalizing Flows	**0.517 (0.02)**	0.504 (0.02)	0.506 (0.02)	0.505 (0.02)	0.505 (0.02)	0.504 (0.02)	0.502 (0.02)
PATEGAN	**0.514 (0.02)**	0.497 (0.02)	0.500 (0.02)	0.498 (0.02)	0.500 (0.02)	0.501 (0.02)	0.502 (0.02)
TVAE	**0.541 (0.02)**	0.529 (0.03)	0.523 (0.02)	0.524 (0.03)	0.529 (0.03)	0.522 (0.02)	0.504 (0.02)
Rank	**1.3**	3.5	3.6	3.8	4.0	5.4	6.4

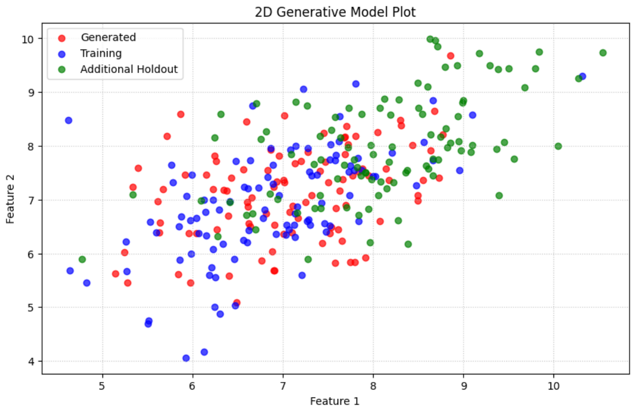
True Positive Rate @ Fixed False Positive Rate

True Positive Rate @ Fixed False Positive Rate:

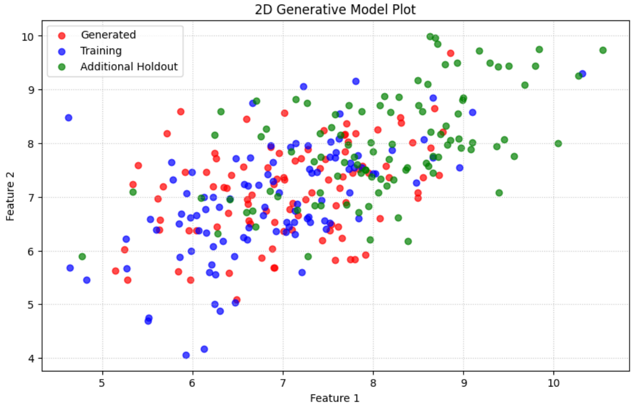


MIA	**0.001**	**0.01**	**0.1**
Logan 2017	0.003 (0.01)	0.012 (0.01)	0.102 (0.02)
DPI	0.002 (0.00)	0.014 (0.01)	0.118 (0.03)
MC	0.003 (0.00)	0.014 (0.01)	0.120 (0.04)
DOMIAS	0.002 (0.00)	0.016 (0.01)	0.134 (0.06)
DCR-Diff	0.005 (0.01)	0.019 (0.02)	0.138 (0.07)
DCR	0.016 (0.05)	0.036 (0.08)	0.153 (0.11)
Gen-LRA (ours)	**0.031 (0.01)**	**0.056 (0.03)**	**0.193 (0.08)**

Back to our Original Motivation...

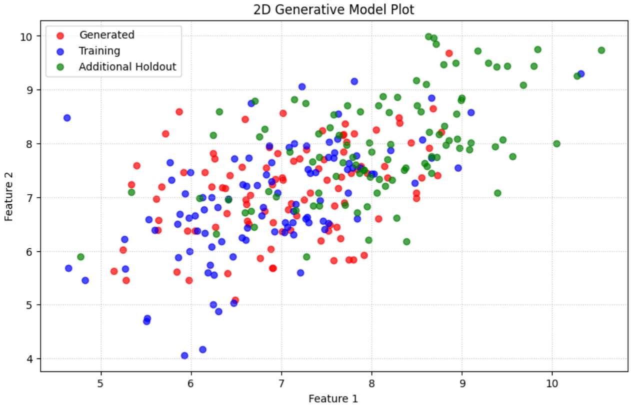


Back to our Original Motivation...



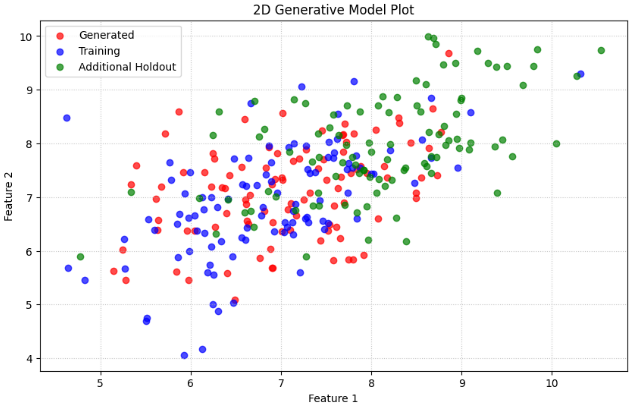
1. There was only a 1 unit difference in means between the training/generated and holdout sets

Back to our Original Motivation...



1. There was only a 1 unit difference in means between the training/ generated and holdout sets
2. Is this overfit?

Back to our Original Motivation...



1. There was only a 1 unit difference in means between the training/ generated and holdout sets
2. Is this overfit?
3. Gen-LRA scored a .73 AUC-ROC on this dataset!

MIA Takeaways

Conclusions for Practitioners:

MIA Takeaways

Conclusions for Practitioners:

- We can characterize overfitting in complex models by evaluating how much training data we can steal given limited information about said model.

MIA Takeaways

Conclusions for Practitioners:

- We can characterize overfitting in complex models by evaluating how much training data we can steal given limited information about said model.
- These techniques can be architecture, training parameter, data type agnostic.

MIA Takeaways

Conclusions for Practitioners:

- We can characterize overfitting in complex models by evaluating how much training data we can steal given limited information about said model.
- These techniques can be architecture, training parameter, data type agnostic.
- MIAs can satisfy our (Useful/ Interpretable/ Cheap) criteria for a good methodology.

MIA Takeaways

Conclusions for Practitioners:

- We can characterize overfitting in complex models by evaluating how much training data we can steal given limited information about said model.
- These techniques can be architecture, training parameter, data type agnostic.
- MIAs can satisfy our (Useful/ Interpretable/ Cheap) criteria for a good methodology.
- Generative Models are not magic and require thoughtful user practices to responsibly deploy them.

Future Work

Chi-Hua and I are actively looking for folks who want to help with research on MIAs!

- Deep Learning Density Ratio Estimation
- MIAs in multi-update models
- Characterizing High Risk Sub-groups in training data
- MIAs for Large Language Models

Questions?