CY 7790

Special Topics in Security and Privacy: Machine Learning Security and Privacy
Fall 2021

Alina Oprea
Associate Professor
Khoury College of Computer Science

November 8 2021

Membership Inference Attacks Against Machine Learning Models

By Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov

Presented by Cem Topcuoglu (11/8/21)

Membership Inference Attack

- Given a data record and black-box access to a model, determine if the record was in the model's training dataset or not

- What if you train a model that includes sensitive data (e.g., health-care) and shared this model publicly

Background

- Model inversion: It uses model’s output to infer something about this input or to extract features
- -> Does not produce an actual member or does not infer if the given record was in the training dataset.
- Identifying the presence of an individual’s data given some statistics about the pool
- -> Unlike this paper, requires some statistics

Privacy in ML

- Delanius desideratum: Nothing about an individual should be learnable from the database (model) that cannot be learned without access to the database (model).
- Not achievable by any useful model

1. Inference About Members of the Population

- If the model is based on statistical facts about the population, it may not be possible to prevent the privacy breach

- For example, high correlation between a person’s phenotype and genetic predisposition to a certain disease

2. Inference About Members of the Training Dataset

- Focus of this work

- What the model reveals about them beyond what it reveals about an arbitrary member of the population
- Goal is to measure the membership risk if the person allows their data to be used to train a model

Threat Model

- Adversary is limited to the black-box queries
- This queries return the model’s output (prediction vector) on a given input

- Two settings

1. Oracle access -> In this case the attacker doesn’t know the model’s structure or meta-parameters
2. No oracle access -> The attacker knows the type and architecture of the machine learning model

Threat Model

- The attacker may have some background knowledge about the population from which the target model's training dataset was drawn
- The attacker may know some general statistics about the population. For example, the marginal distribution of feature values

Methodology

- Train an attack model that distinguishes the target model's behavior from the training inputs from the inputs that it did not encounter during training
- The membership interference problem -> classification problem

- Inference techniques are generic and not based on any dataset or model type.

Attack

- The attacker is given a data record and black-box query access to the target model.
- Success -> If the attacker can determine whether this data record was part of the model’s training dataset or not.

Membership Inference Attack

- Machine learning models often behave differently on the data that they were trained on versus the data that they “see” for the first time.
- Overfitting is a common reason (not only)
- The objective of the attacker -> construct an attack model that can recognize such differences and use it to distinguish members from non-members.

Methodology

- Attack model is a collection of “shadow” models, one for each output class of the target model -> This increases accuracy of the attack

- We know whether a given record was in shadow models’ training dataset. By using this idea, we can teach how to distinguish

End-to-end attack process

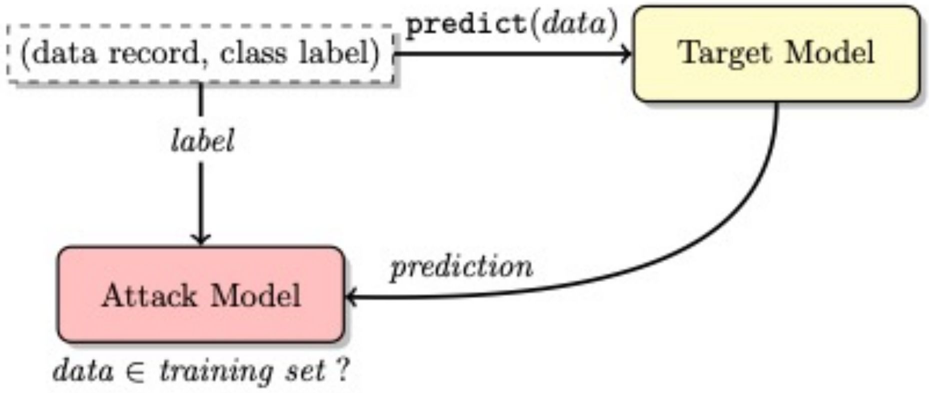


Fig. 1: Membership inference attack in the black-box setting. The attacker queries the target model with a data record and obtains the model's prediction on that record. The prediction is a vector of probabilities, one per class, that the record belongs to a certain class. This prediction vector, along with the label of the target record, is passed to the attack model, which infers whether the record was *in* or *out* of the target model's training dataset.

Challenge

- How to attack when the attacker has no information about the internal parameters of the target model (only limited to queries from the public API)

- Shadow training

1. Multiple “shadow models” are created -> using the generated training set
2. Train the attack model on the labeled inputs and outputs of the shadow models

Shadow Model Technique

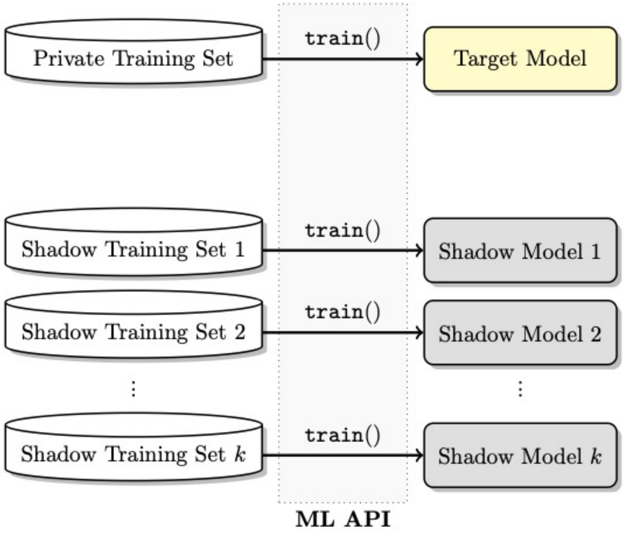


Fig. 2: Training shadow models using the same machine learning platform as was used to train the target model. The training datasets of the target and shadow models have the same format but are disjoint. The training datasets of the shadow models may overlap. All models' internal parameters are trained independently.

- They use the same ML API to train shadow models
- The accuracy of the attack is increasing with the number of Shadow models -> It learns the ones that it did encounter and ones that it did not

Generating Training Data for Shadow Models

- Should be distributed similarly to the targe model's training data.

1. Model-based synthesis

2. Statistics-based synthesis

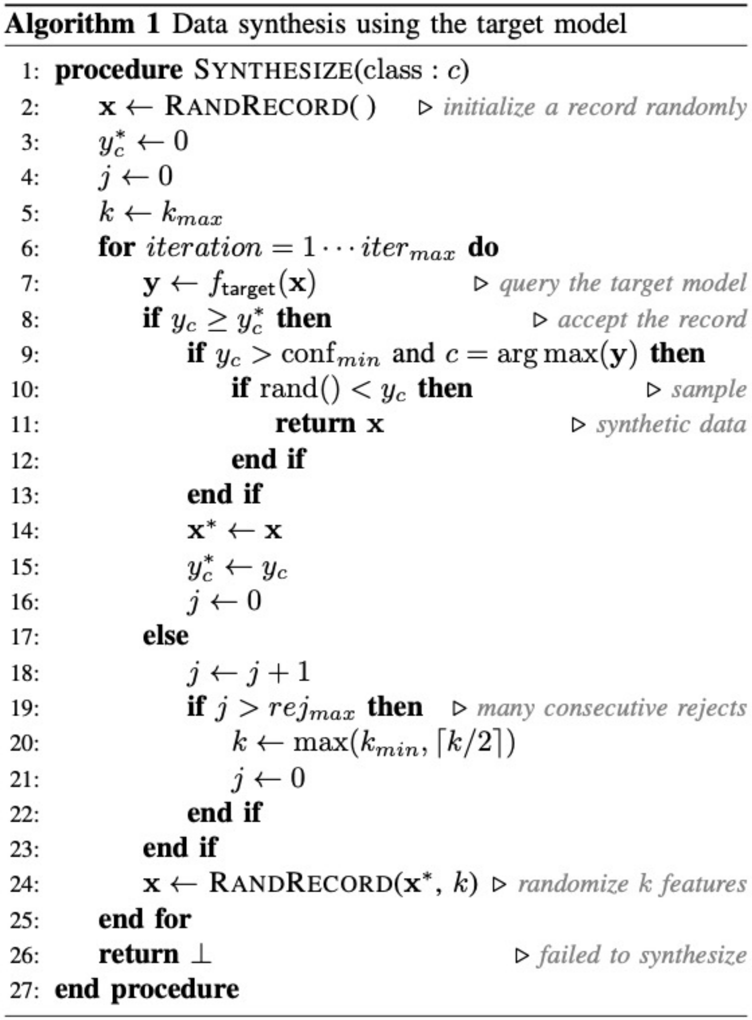
3. Noisy real data

1) Model-based synthesis

- No real data or any statistics
- Create by using the target model itself

- Intuition -> records that classified with high confidence should be statistically similar to the target’s training dataset.

Data Synthesis



Algorithm 1 Data synthesis using the target model

```
procedure SYNTHESIZE(class : $c$)
    $\mathbf{x} \leftarrow$ RANDRECORD( )    ▷ initialize a record randomly
    $y_c^* \leftarrow 0$
    $j \leftarrow 0$
    $k \leftarrow k_{max}$
    for $iteration = 1 \cdots iter_{max}$ do
        $\mathbf{y} \leftarrow f_{\text{target}}(\mathbf{x})$    ▷ query the target model
        if $y_c \geq y_c^*$ then    ▷ accept the record
            if $y_c > \text{conf}_{min}$ and $c = \arg\max(\mathbf{y})$ then
                if $\text{rand}() < y_c$ then    ▷ sample
                    return $\mathbf{x}$    ▷ synthetic data
                end if
            end if
            $\mathbf{x}^* \leftarrow \mathbf{x}$
            $y_c^* \leftarrow y_c$
            $j \leftarrow 0$
        else
            $j \leftarrow j + 1$
            if $j > rej_{max}$ then    ▷ many consecutive rejects
                $k \leftarrow \max(k_{min}, \lceil k/2 \rceil)$
                $j \leftarrow 0$
            end if
        end if
        $\mathbf{x} \leftarrow$ RANDRECORD($\mathbf{x}^*$, $k$)    ▷ randomize $k$ features
    end for
    return $\perp$    ▷ failed to synthesize
end procedure
```

The synthesis process runs in two phases:

1. Search using a hill-climbing algorithm
2. Sample synthetic data from these records. Repeat it until the training dataset for shadow models are full

Limitations

- Works only if the attacker can explore the space of possible inputs
- For example, may not work in high-resolution images

2) Statistics-based synthesis

- Some statistical information about the population

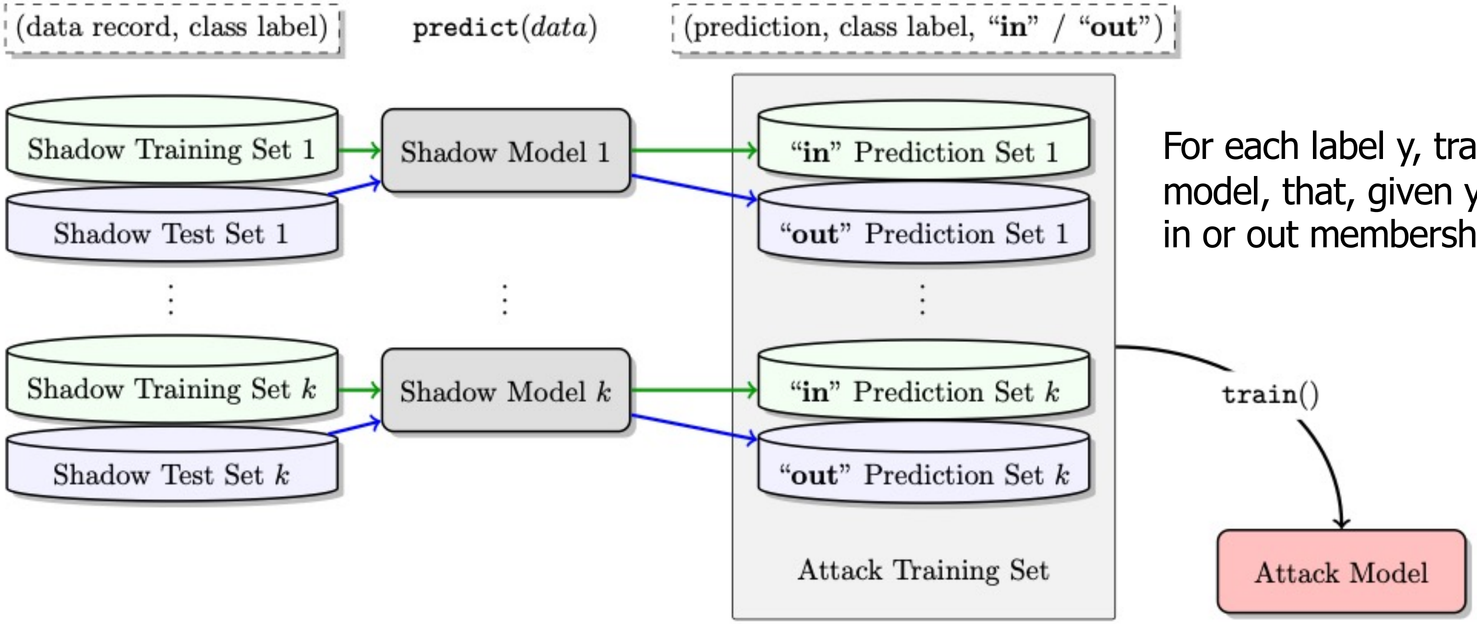
- In experiments, they sampled the value of each feature from its own marginal distribution

3) Noisy real data

- Attacker has access to some data that is similar to the target model's training data -> "noisy"

- They simulated by flipping the values of 10% or 20% randomly selected features

Training of Attack Models



For each label y, train a separate model, that, given y, predicts the in or out membership status for x

Fig. 3: Training the attack model on the inputs and outputs of the shadow models. For all records in the training dataset of a shadow model, we query the model and obtain the output. These output vectors are labeled "in" and added to the attack model's training dataset. We also query the shadow model with a test dataset disjoint from its training dataset. The outputs on this set are labeled "out" and also added to the attack model's training dataset. Having constructed a dataset that reflects the black-box behavior of the shadow models on their training and test datasets, we train a collection of c_{target} attack models, one per each output class of the target model.

Evaluation

- Data
  - CIFAR
  - Purchases -> Kaggle's acquire valued shoppers
  - Locations
  - Texas hospital stays
  - MNIST
  - UCI adult

Target Models

- Google Prediction API
  - Upload dataset and obtain an API for querying the resulting model
  - No configuration parameter
- Amazon ML
  - User can control few meta-parameters
    - Max number of passes ever the training data: controls convergence of model training
    - L2 regularization tunes the regularization in order to avoid overfitting
- Local Neural Network
  - CIFAR (only locally) and purchase datasets

Findings

- Increasing the number of shadow models would increase the accuracy of the attack but also its cost

- Accuracy of the attack can vary considerably for different classes
- Because each class have different composition

Overfitting

- Baseline accuracy (random guessing) is 0.5 since they used same number of members and non-members
- The test accuracy is 0.6 and 0.2 for CIFAR-10 and CIFAR-100

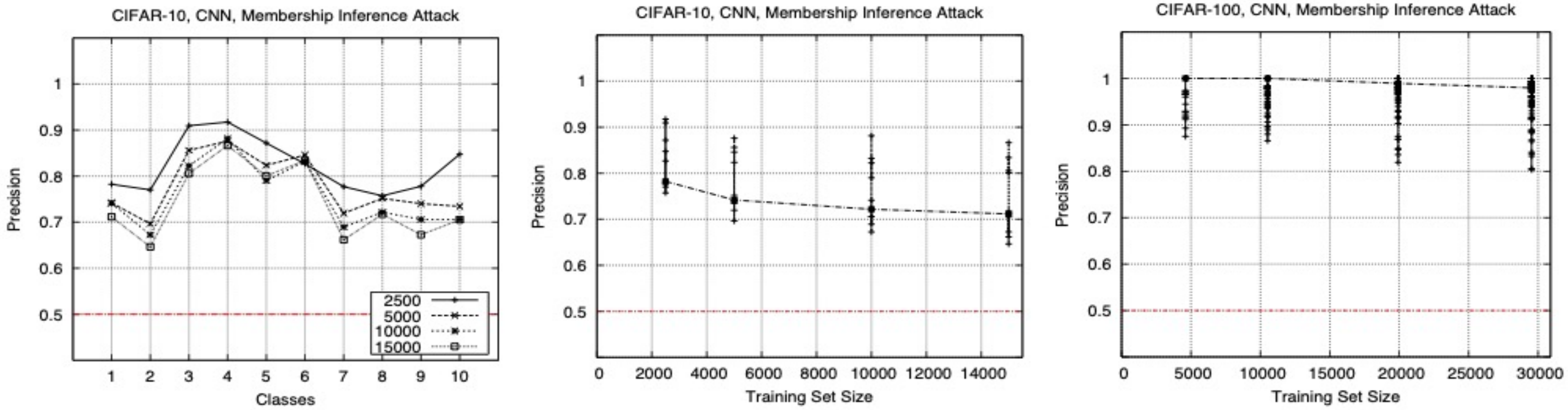
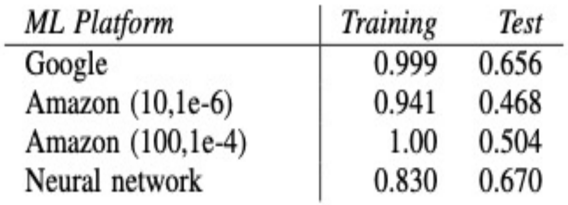


Fig. 4: Precision of the membership inference attack against neural networks trained on CIFAR datasets. The graphs show precision for different classes while varying the size of the training datasets. The median values are connected across different training set sizes. The median precision (from the smallest dataset size to largest) is $0.78, 0.74, 0.72, 0.71$ for CIFAR-10 and $1, 1, 0.98, 0.97$ for CIFAR-100. Recall is almost 1 for both datasets. The figure on the left shows the per-class precision (for CIFAR-10). Random guessing accuracy is 0.5.

Training – Test Accuracies



ML Platform	*Training*	*Test*
Google	0.999	0.656
Amazon (10,1e-6)	0.941	0.468
Amazon (100,1e-4)	1.00	0.504
Neural network	0.830	0.670

TABLE I: Training and test accuracy of the models constructed using different ML-as-a-service platforms on the purchase dataset (with 100 classes).

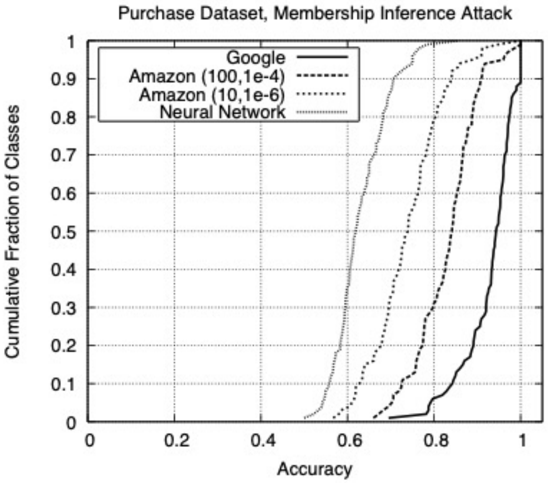


Fig. 7: Precision of the membership inference attack against models trained on the same datasets but using different platforms. The attack model is a neural network.

Using Noisy Data

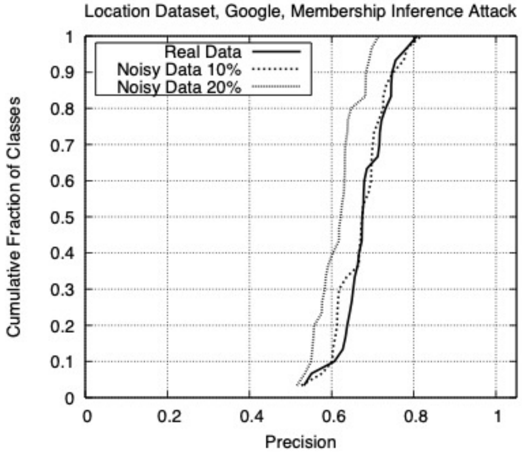
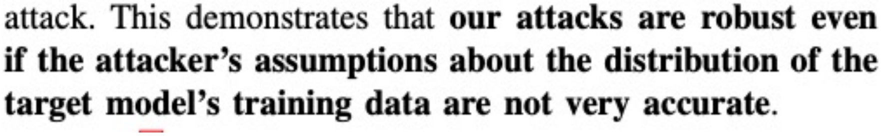
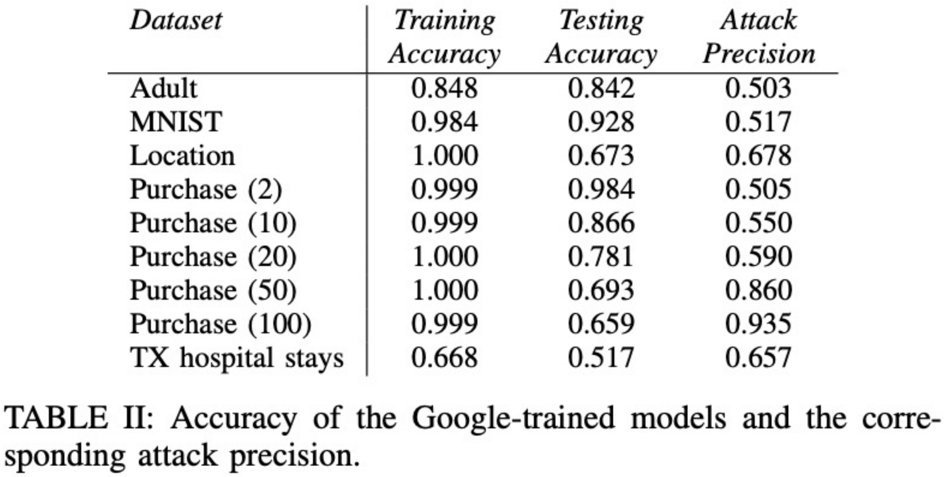


Fig. 8: Empirical CDF of the precision of the membership inference attack against the Google-trained model for the location dataset. Results are shown for the shadow models trained on real data and for the shadow models trained on noisy data with 10% and 20% noise (i.e., $x\%$ of features are replaced with random values). Precision of the attack over all classes is 0.678 (real data), 0.666 (data with 10% noise), and 0.613 (data with 20% noise). The corresponding recall of the attack is 0.98, 0.99, and 1.00, respectively.



attack. This demonstrates that **our attacks are robust even if the attacker's assumptions about the distribution of the target model's training data are not very accurate**.

Training-test Accuracy and Attack Precision



Dataset	*Training Accuracy*	*Testing Accuracy*	*Attack Precision*
Adult	0.848	0.842	0.503
MNIST	0.984	0.928	0.517
Location	1.000	0.673	0.678
Purchase (2)	0.999	0.984	0.505
Purchase (10)	0.999	0.866	0.550
Purchase (20)	1.000	0.781	0.590
Purchase (50)	1.000	0.693	0.860
Purchase (100)	0.999	0.659	0.935
TX hospital stays	0.668	0.517	0.657

TABLE II: Accuracy of the Google-trained models and the corresponding attack precision.

Train-test Accuracy Gap

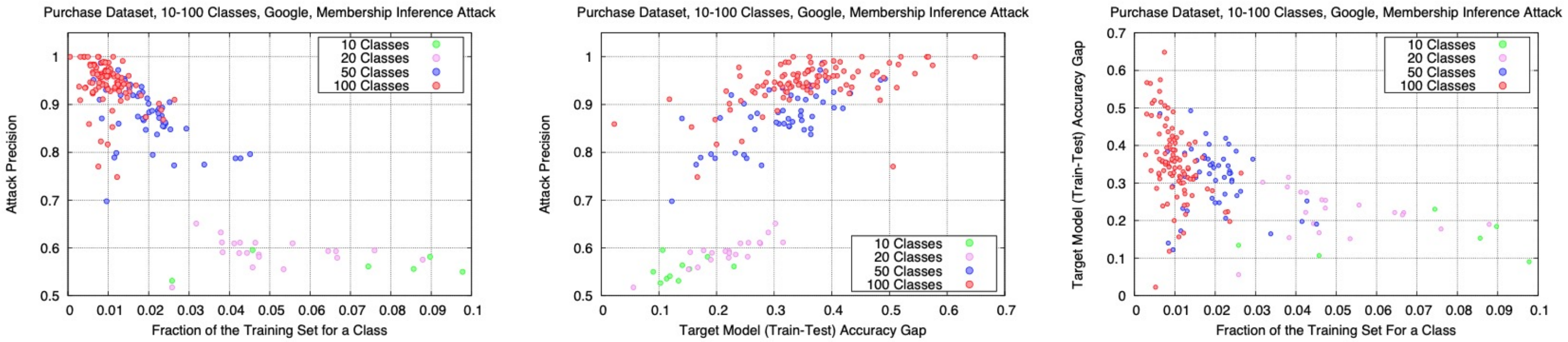


Fig. 11: Relationship between the precision of the membership inference attack on a class and the (train-test) accuracy gap of the target model, as well as the fraction of the training dataset that belongs to this class. Each point represent the values for one class. The (train-test) accuracy gap is a metric for generalization error [18] and an indicator of how overfitted the target model is.

Attack Success Based on Number of Classes

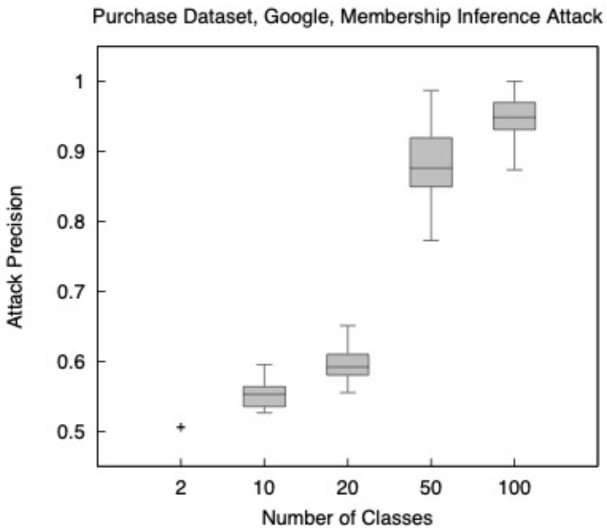


Fig. 10: Precision of the membership inference attack against different purchase classification models trained on the Google platform. The boxplots show the distribution of precision over different classification tasks (with a different number of classes).

- Models with fewer classes leak less information about their training inputs. As the number of classes increases, the model needs to extract more distinctive features from the data to be able to classify inputs with high accuracy.

Mitigation

- Regularization techniques such as dropout can help defeat overfitting and also strengthen privacy guarantees in neural networks
- Differentially private models are, by construction, secure against membership inference attacks of the kind developed in this paper because our attacks operate solely on the outputs of the model, without any auxiliary information
- differentially private models may significantly reduce the model’s prediction accuracy for small ε values

Mitigation strategies

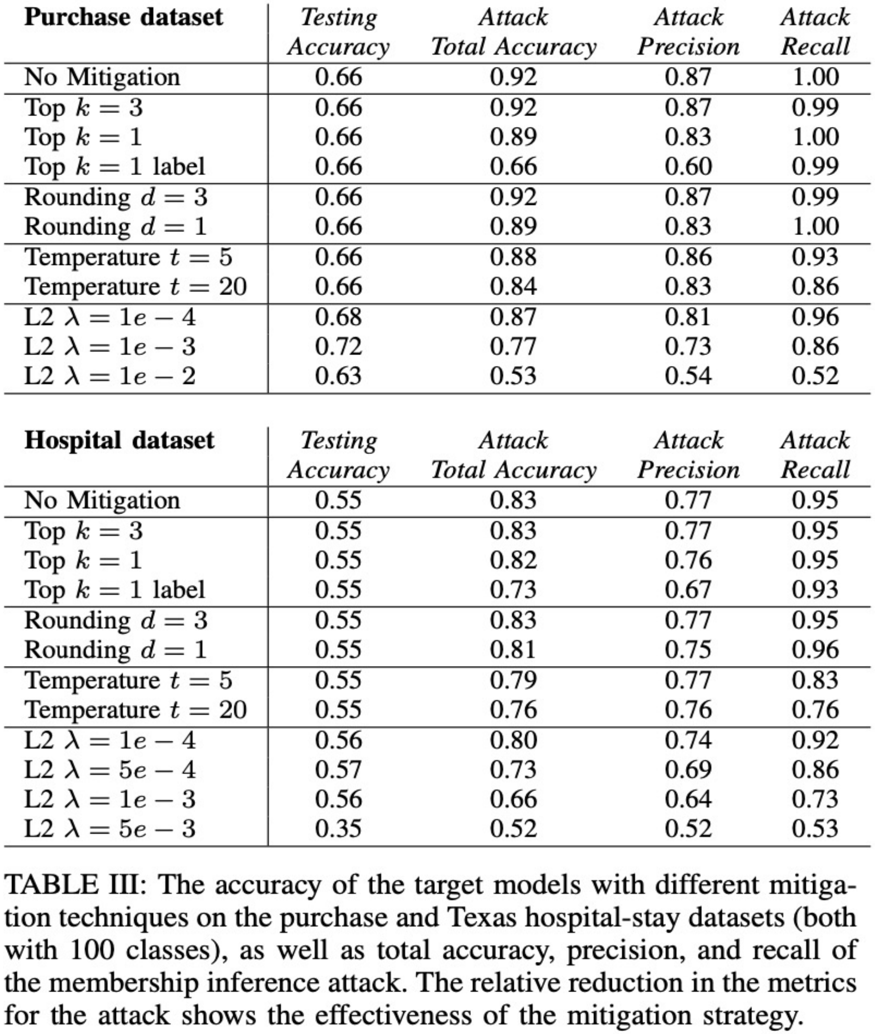
1. Restrict the prediction vector to top k classes.
   - Many classes may have very small probabilities in the model's prediction vector. The smaller k is, the less information the model leaks.
2. Coarsen precision of the prediction vector
   - Round the classification probabilities in the prediction vector down to d floating point digits. The smaller d is, the less information the model leaks.
3. Increase entropy of the prediction vector.
   - For neural-network models, modify (or add) the softmax layer and increase its normalizing temperature t > 0.

$$\frac{e^{z_i/t}}{\sum_j e^{z_j/t}}$$

4. Use regularization
   - L2-norm standard regularization

Evaluation of mitigation strategies

- Overall, the attack is robust against these mitigation strategies.



Purchase dataset	*Testing Accuracy*	*Attack Total Accuracy*	*Attack Precision*	*Attack Recall*
No Mitigation	0.66	0.92	0.87	1.00
Top $k = 3$	0.66	0.92	0.87	0.99
Top $k = 1$	0.66	0.89	0.83	1.00
Top $k = 1$ label	0.66	0.66	0.60	0.99
Rounding $d = 3$	0.66	0.92	0.87	0.99
Rounding $d = 1$	0.66	0.89	0.83	1.00
Temperature $t = 5$	0.66	0.88	0.86	0.93
Temperature $t = 20$	0.66	0.84	0.83	0.86
L2 $\lambda = 1e - 4$	0.68	0.87	0.81	0.96
L2 $\lambda = 1e - 3$	0.72	0.77	0.73	0.86
L2 $\lambda = 1e - 2$	0.63	0.53	0.54	0.52

Hospital dataset	*Testing Accuracy*	*Attack Total Accuracy*	*Attack Precision*	*Attack Recall*
No Mitigation	0.55	0.83	0.77	0.95
Top $k = 3$	0.55	0.83	0.77	0.95
Top $k = 1$	0.55	0.82	0.76	0.95
Top $k = 1$ label	0.55	0.73	0.67	0.93
Rounding $d = 3$	0.55	0.83	0.77	0.95
Rounding $d = 1$	0.55	0.81	0.75	0.96
Temperature $t = 5$	0.55	0.79	0.77	0.83
Temperature $t = 20$	0.55	0.76	0.76	0.76
L2 $\lambda = 1e - 4$	0.56	0.80	0.74	0.92
L2 $\lambda = 5e - 4$	0.57	0.73	0.69	0.86
L2 $\lambda = 1e - 3$	0.56	0.66	0.64	0.73
L2 $\lambda = 5e - 3$	0.35	0.52	0.52	0.53

TABLE III: The accuracy of the target models with different mitigation techniques on the purchase and Texas hospital-stay datasets (both with 100 classes), as well as total accuracy, precision, and recall of the membership inference attack. The relative reduction in the metrics for the attack shows the effectiveness of the mitigation strategy.

Discussion

Limitations

- Model based synthesis -> It may not work if the inputs are high-resolution images and the target model performs a complex image classification task

Strengths

- Not bounded to a particular dataset or model type
- Realistic classification tasks

Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, Somesh Jha

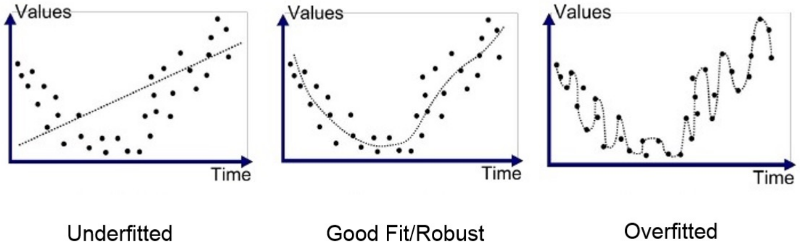
Problem Statement

Problem Statement

- Machine learning applications require sensitive personal data
  - Medical information
  - Behavioral patterns
  - Personally identifiable information
- ML Models have been shown to leak information about training data

Problem Statement

- Relationship between privacy risk and *overfitting*
- Relationship between privacy risk and *influence*
- **Characterize the effect that overfitting and influence have on the advantage of adversaries attempting to infer specific facts about the training set**



Background

Attack Types Considered

Membership Inference Attack

- Aim to determine whether a given data point was present in the training data used to build a model.

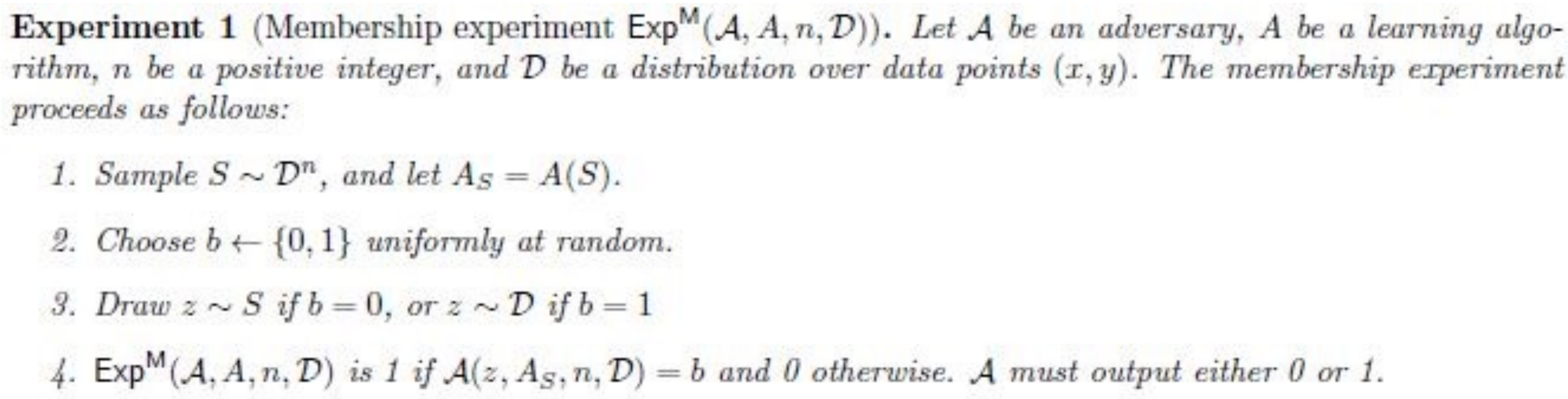
Attribute Inference Attack

- Adversary uses a machine learning model and incomplete information about a data point to infer the missing information for that point.

Stability and Generalization

- An algorithm is *stable* if a small change to its input causes limited change in its output (i.e., in machine learning, the replacement of a single data point in the training set of an ML algorithm).
- Stability is closely related to the notion of *differential privacy*.
- An algorithm overfits (loses *generality*) if its expected loss on samples drawn from the greater distribution of data is much greater than its expected loss on the training set

Membership Inference Attacks



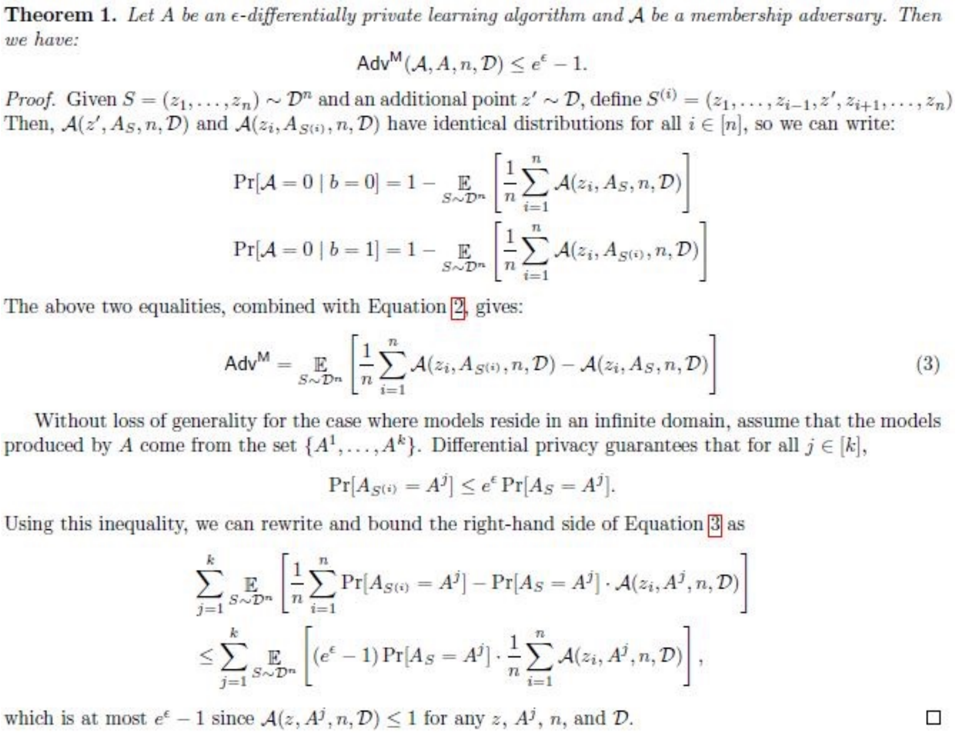
Experiment 1 (Membership experiment $\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$). *Let $\mathcal{A}$ be an adversary, A be a learning algorithm, n be a positive integer, and $\mathcal{D}$ be a distribution over data points (x, y). The membership experiment proceeds as follows:*

1. *Sample $S \sim \mathcal{D}^n$, and let $A_S = A(S)$.*
2. *Choose $b \leftarrow \{0, 1\}$ uniformly at random.*
3. *Draw $z \sim S$ if $b = 0$, or $z \sim \mathcal{D}$ if $b = 1$*
4. *$\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$ is 1 if $\mathcal{A}(z, A_S, n, \mathcal{D}) = b$ and 0 otherwise. $\mathcal{A}$ must output either 0 or 1.*

- *Membership advantage* describes how well an attacker can distinguish a point from the training set given a model

Bounds from differential privacy

- Differential privacy limits how much any one point in the training data may affect the outcome
- Differential privacy limits the success of membership inference attacks
  - A function of epsilon
- Bound on membership advantage when the loss function is convex and Lipschitz



Theorem 1. *Let A be an ϵ-differentially private learning algorithm and $\mathcal{A}$ be a membership adversary. Then we have:*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) \leq e^{\epsilon} - 1.$$

Proof. Given $S = (z_1, \ldots, z_n) \sim \mathcal{D}^n$ and an additional point $z' \sim \mathcal{D}$, define $S^{(i)} = (z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_n)$ Then, $\mathcal{A}(z', A_S, n, \mathcal{D})$ and $\mathcal{A}(z_i, A_{S^{(i)}}, n, \mathcal{D})$ have identical distributions for all $i \in [n]$, so we can write:

$$\Pr[\mathcal{A} = 0 \mid b = 0] = 1 - \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n} \mathcal{A}(z_i, A_S, n, \mathcal{D})\right]$$

$$\Pr[\mathcal{A} = 0 \mid b = 1] = 1 - \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n} \mathcal{A}(z_i, A_{S^{(i)}}, n, \mathcal{D})\right]$$

The above two equalities, combined with Equation 2, gives:

$$\mathsf{Adv}^{\mathsf{M}} = \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n} \mathcal{A}(z_i, A_{S^{(i)}}, n, \mathcal{D}) - \mathcal{A}(z_i, A_S, n, \mathcal{D})\right] \tag{3}$$

Without loss of generality for the case where models reside in an infinite domain, assume that the models produced by A come from the set $\{A^1, \ldots, A^k\}$. Differential privacy guarantees that for all $j \in [k]$,

$$\Pr[A_{S^{(i)}} = A^j] \leq e^{\epsilon} \Pr[A_S = A^j].$$

Using this inequality, we can rewrite and bound the right-hand side of Equation 3 as

$$\sum_{j=1}^{k} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\frac{1}{n}\sum_{i=1}^{n} \Pr[A_{S^{(i)}} = A^j] - \Pr[A_S = A^j] \cdot \mathcal{A}(z_i, A^j, n, \mathcal{D})\right]$$

$$\leq \sum_{j=1}^{k} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[(e^{\epsilon} - 1)\Pr[A_S = A^j] \cdot \frac{1}{n}\sum_{i=1}^{n} \mathcal{A}(z_i, A^j, n, \mathcal{D})\right],$$

which is at most $e^{\epsilon} - 1$ since $\mathcal{A}(z, A^j, n, \mathcal{D}) \leq 1$ for any z, A^j, n, and $\mathcal{D}$. □

Attribute Inference Attacks

- Adversary seeks to guess the value of sensitive feature of a data point given only some public knowledge about it and the model.
  - Adversary only has partial information about the data point in question

- *Advantage* in this case measures the amount of information about the target that the algorithm leaks specifically concerning the training data

Methodology

Adversaries

Adversary 1 (Membership Inference)

- Loss function bounded by some constant *B*
- Adversary predicts that *z* is not in the training set with probability proportional to the model's loss at *z*
- Membership advantage of this approach is proportional to the generalization error of *A*

- Advantage and generalization error are closely related.

Adversary 2 (Membership Inference)

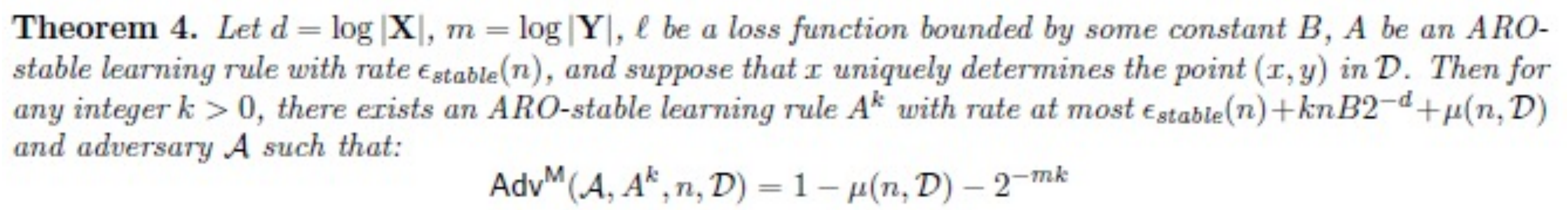
- Adversary knows the exact error distribution
  - Can compute which value of *b* most likely
- Regression problems
- Standard error published along with the release of the model

- Advantage and generalization error are closely related.

Adversary 3 (Membership Inference)

- Attacker can influence the training algorithm or substitute it with a malicious one
- Stable learning rule with a bounded loss function

- Overfitting is not necessary for membership advantage.
- Algorithm substitution attack
  - Colluding algorithm and membership inference



Theorem 4. *Let $d = \log|\mathbf{X}|$, $m = \log|\mathbf{Y}|$, ℓ be a loss function bounded by some constant B, A be an ARO-stable learning rule with rate $\epsilon_{\mathsf{stable}}(n)$, and suppose that x uniquely determines the point (x, y) in $\mathcal{D}$. Then for any integer $k > 0$, there exists an ARO-stable learning rule A^k with rate at most $\epsilon_{\mathsf{stable}}(n) + knB2^{-d} + \mu(n, \mathcal{D})$ and adversary $\mathcal{A}$ such that:*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A^k, n, \mathcal{D}) = 1 - \mu(n, \mathcal{D}) - 2^{-mk}$$

Adversary 4 (Attribute Inference)

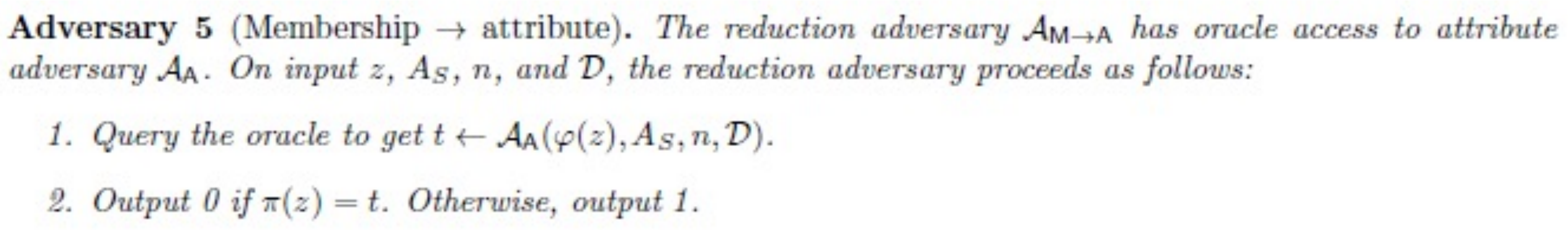
- Adversary can approximate the error distribution
- Adversary then can try all possible values for the sensitive attribute
- Picks the value of the sensitive attribute with the highest probability

- Model inversion
- Interested in the effect that generalization error has on advantage.
- *Functional Influence*

Adversary 5 (Membership -> Attribute)

- Adversary uses an *attribute oracle* to accomplish membership inference.

- Examines the connections between membership and attribute inference attacks.



Adversary 5 (Membership $\to$ attribute). *The reduction adversary $\mathcal{A}_{\mathsf{M}\to\mathsf{A}}$ has oracle access to attribute adversary $\mathcal{A}_{\mathsf{A}}$. On input z, A_S, n, and $\mathcal{D}$, the reduction adversary proceeds as follows:*

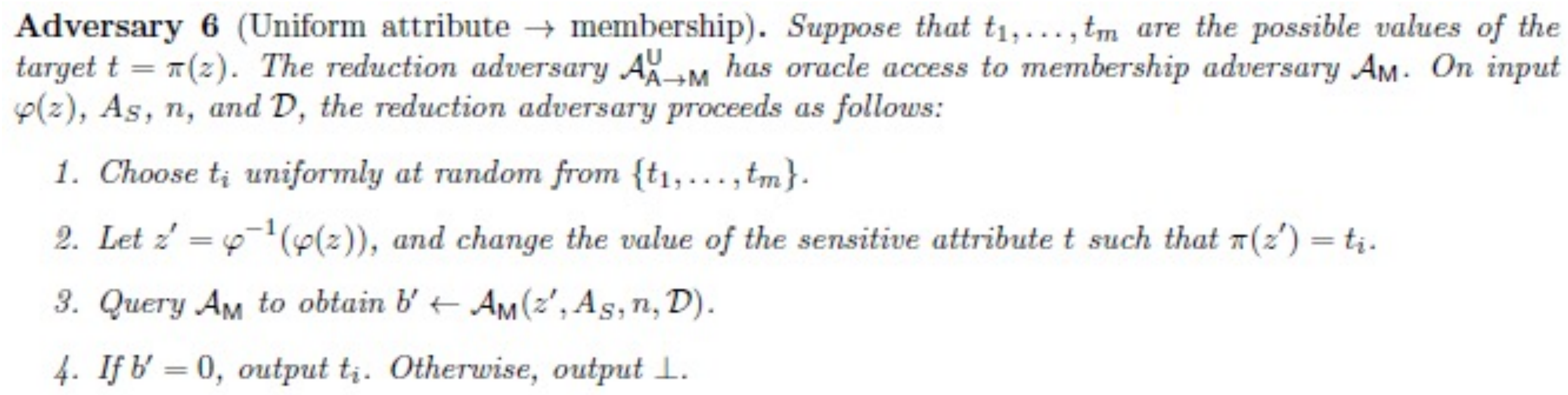
1. *Query the oracle to get $t \leftarrow \mathcal{A}_{\mathsf{A}}(\varphi(z), A_S, n, \mathcal{D})$.*

2. *Output 0 if $\pi(z) = t$. Otherwise, output 1.*

Adversary 6 (Attribute -> Membership)

- Adversary given partial information on *z* and reconstructs the entire point to query the membership oracle.

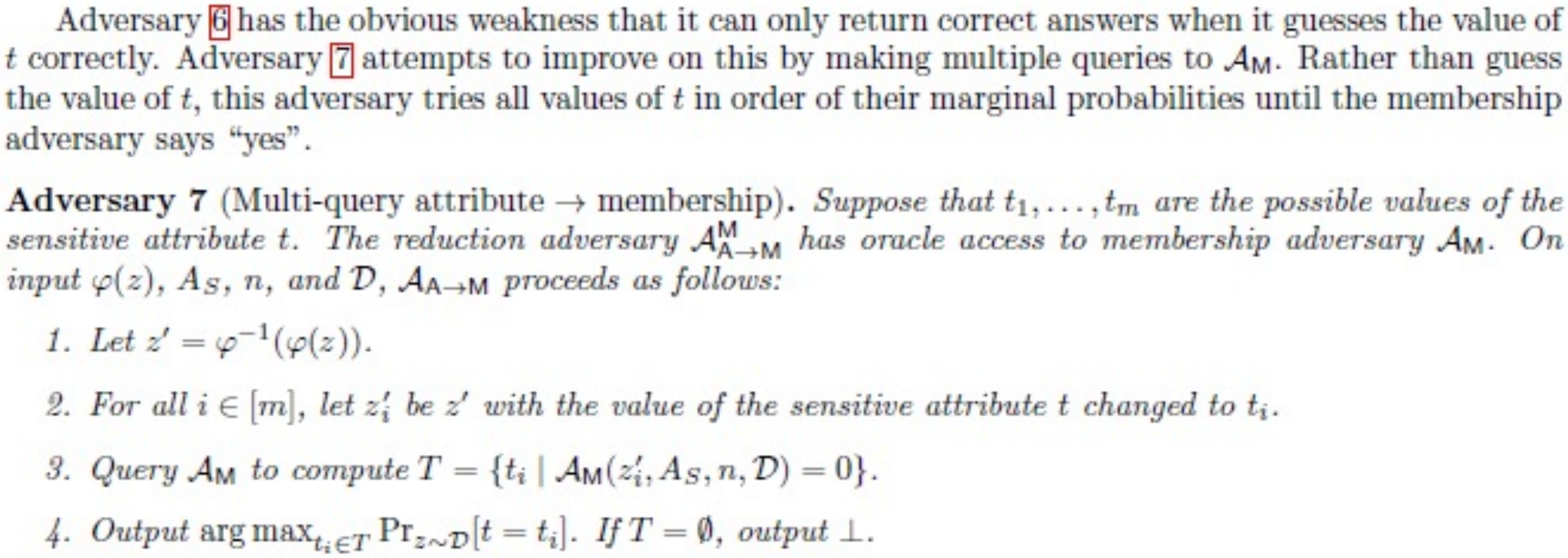
- Examines the connections between membership and attribute inference attacks.



Adversary 6 (Uniform attribute → membership). *Suppose that* $t_1, \ldots, t_m$ *are the possible values of the target* $t = \pi(z)$. *The reduction adversary* $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ *has oracle access to membership adversary* $\mathcal{A}_{\mathsf{M}}$. *On input* $\varphi(z)$, A_S, n, *and* $\mathcal{D}$, *the reduction adversary proceeds as follows:*

1. *Choose* t_i *uniformly at random from* $\{t_1, \ldots, t_m\}$.
2. *Let* $z' = \varphi^{-1}(\varphi(z))$, *and change the value of the sensitive attribute* t *such that* $\pi(z') = t_i$.
3. *Query* $\mathcal{A}_{\mathsf{M}}$ *to obtain* $b' \leftarrow \mathcal{A}_{\mathsf{M}}(z', A_S, n, \mathcal{D})$.
4. *If* $b' = 0$, *output* t_i. *Otherwise, output* $\perp$.

Adversary 7 (Multi-Query Attribute -> Member)



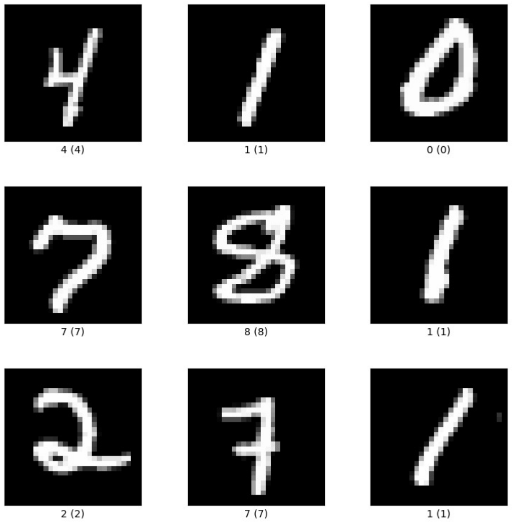
Adversary 6 has the obvious weakness that it can only return correct answers when it guesses the value of t correctly. Adversary 7 attempts to improve on this by making multiple queries to $\mathcal{A}_{\mathsf{M}}$. Rather than guess the value of t, this adversary tries all values of t in order of their marginal probabilities until the membership adversary says “yes”.

Adversary 7 (Multi-query attribute → membership). *Suppose that $t_1, \dots, t_m$ are the possible values of the sensitive attribute t. The reduction adversary $\mathcal{A}^{\mathsf{M}}_{\mathsf{A}\to\mathsf{M}}$ has oracle access to membership adversary $\mathcal{A}_{\mathsf{M}}$. On input $\varphi(z)$, A_S, n, and $\mathcal{D}$, $\mathcal{A}_{\mathsf{A}\to\mathsf{M}}$ proceeds as follows:*

1. *Let $z' = \varphi^{-1}(\varphi(z))$.*
2. *For all $i \in [m]$, let z'_i be z' with the value of the sensitive attribute t changed to t_i.*
3. *Query $\mathcal{A}_{\mathsf{M}}$ to compute $T = \{t_i \mid \mathcal{A}_{\mathsf{M}}(z'_i, A_S, n, \mathcal{D}) = 0\}$.*
4. *Output $\arg\max_{t_i \in T} \Pr_{z\sim\mathcal{D}}[t = t_i]$. If $T = \emptyset$, output $\perp$.*

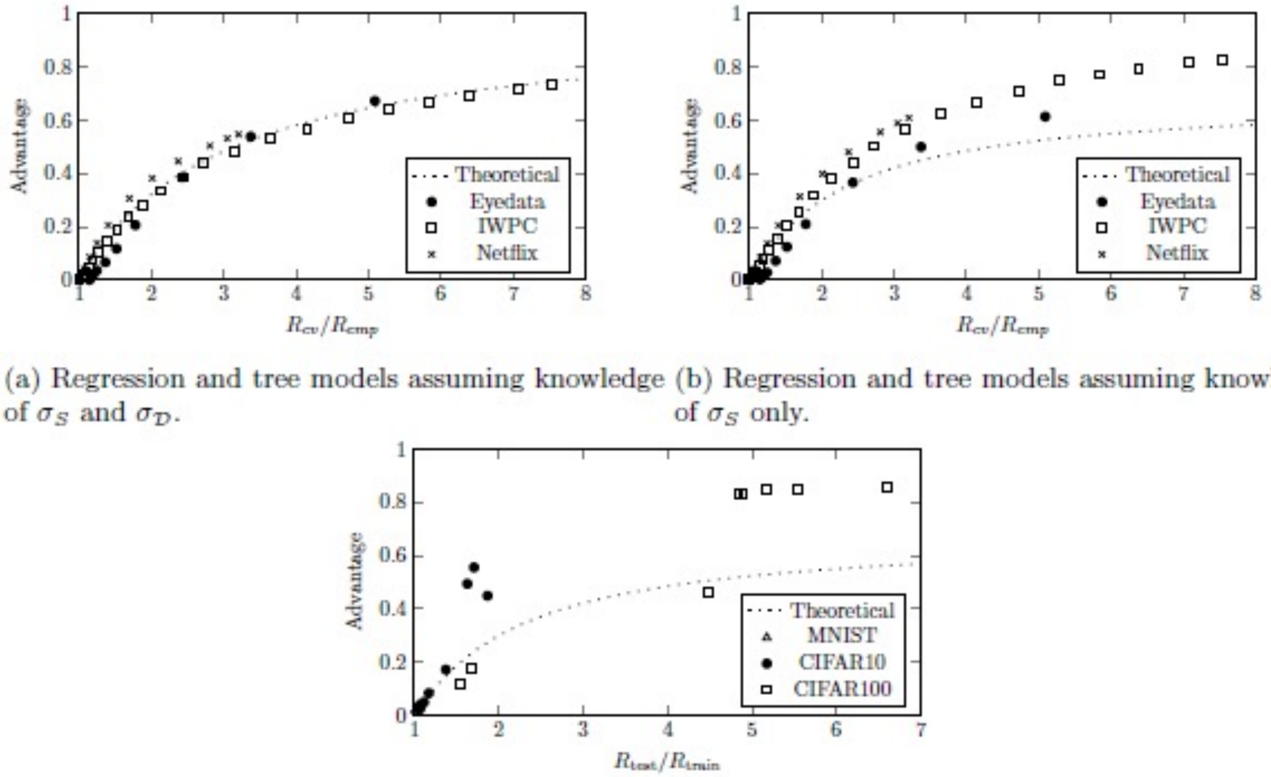
Methodology

- Linear and tree models
  - Eyedata
  - International Warfarin Pharmacogenetics Consortium
  - Netflix
- Deep CNN
  - MNIST
  - CIFAR-10
  - CIFAR-100



Evaluation

Membership Inference



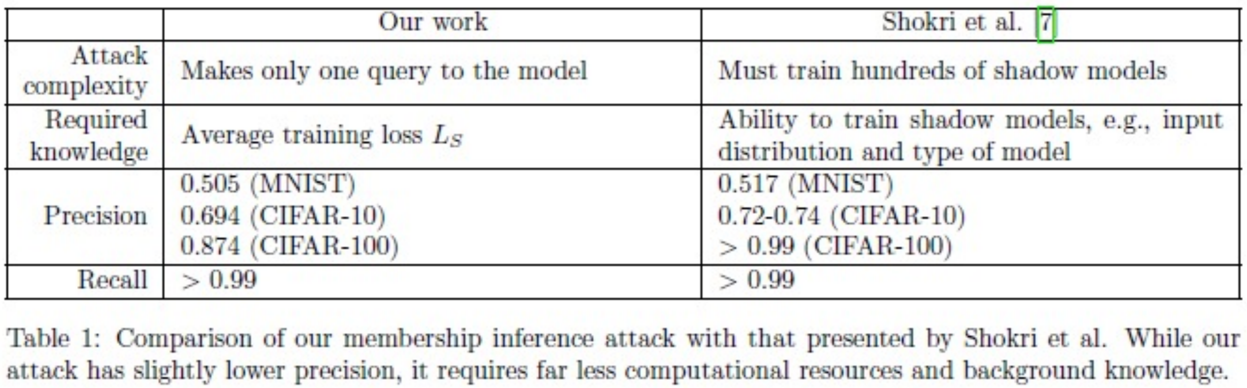
(a) Regression and tree models assuming knowledge of σ_S and $\sigma_{\mathcal{D}}$.

(b) Regression and tree models assuming knowledge of σ_S only.

(c) Deep CNNs assuming knowledge of average training loss L_S.

Figure 2: Empirical membership advantage of the threshold adversary (Adversary 2) given as a function of generalization ratio for regression, tree, and CNN models.

Membership Inference

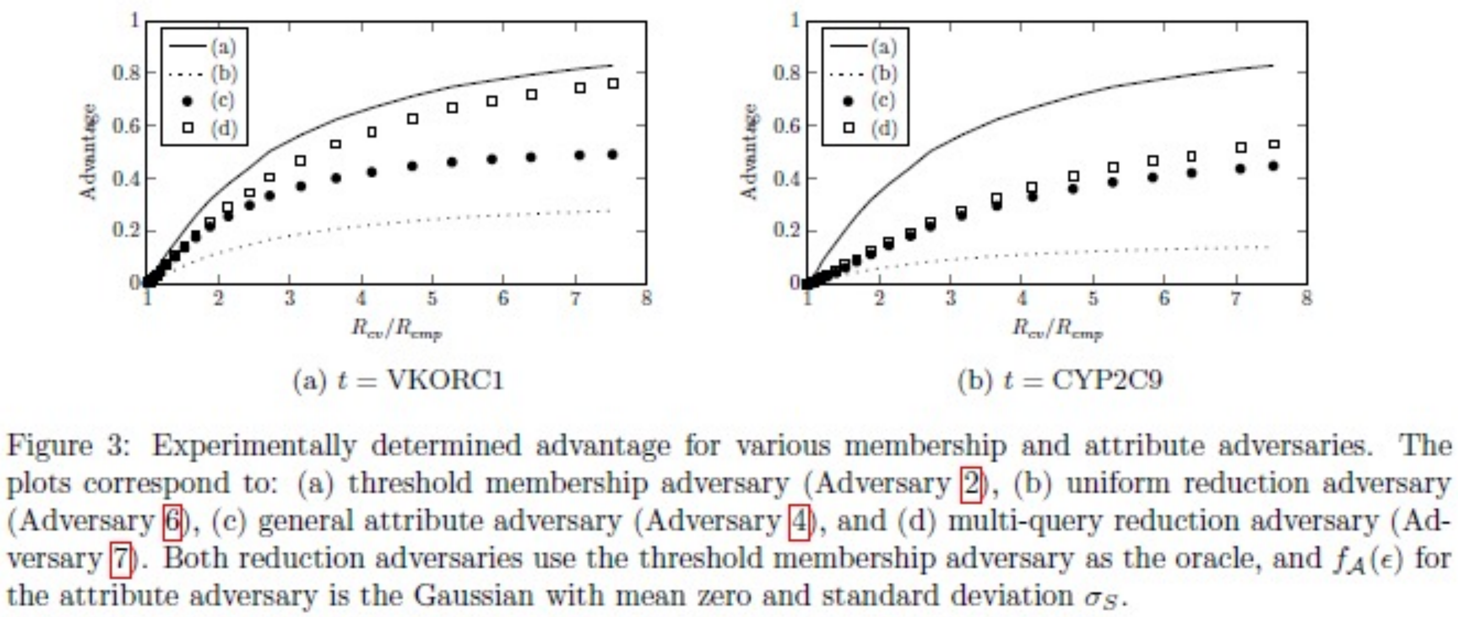


	Our work	Shokri et al. [7]
Attack complexity	Makes only one query to the model	Must train hundreds of shadow models
Required knowledge	Average training loss L_S	Ability to train shadow models, e.g., input distribution and type of model
Precision	0.505 (MNIST) 0.694 (CIFAR-10) 0.874 (CIFAR-100)	0.517 (MNIST) 0.72-0.74 (CIFAR-10) > 0.99 (CIFAR-100)
Recall	> 0.99	> 0.99

Table 1: Comparison of our membership inference attack with that presented by Shokri et al. While our attack has slightly lower precision, it requires far less computational resources and background knowledge.

- Similar performance of proposed attack, while drastically reducing the computational complexity required

Attribute Inference and Reduction

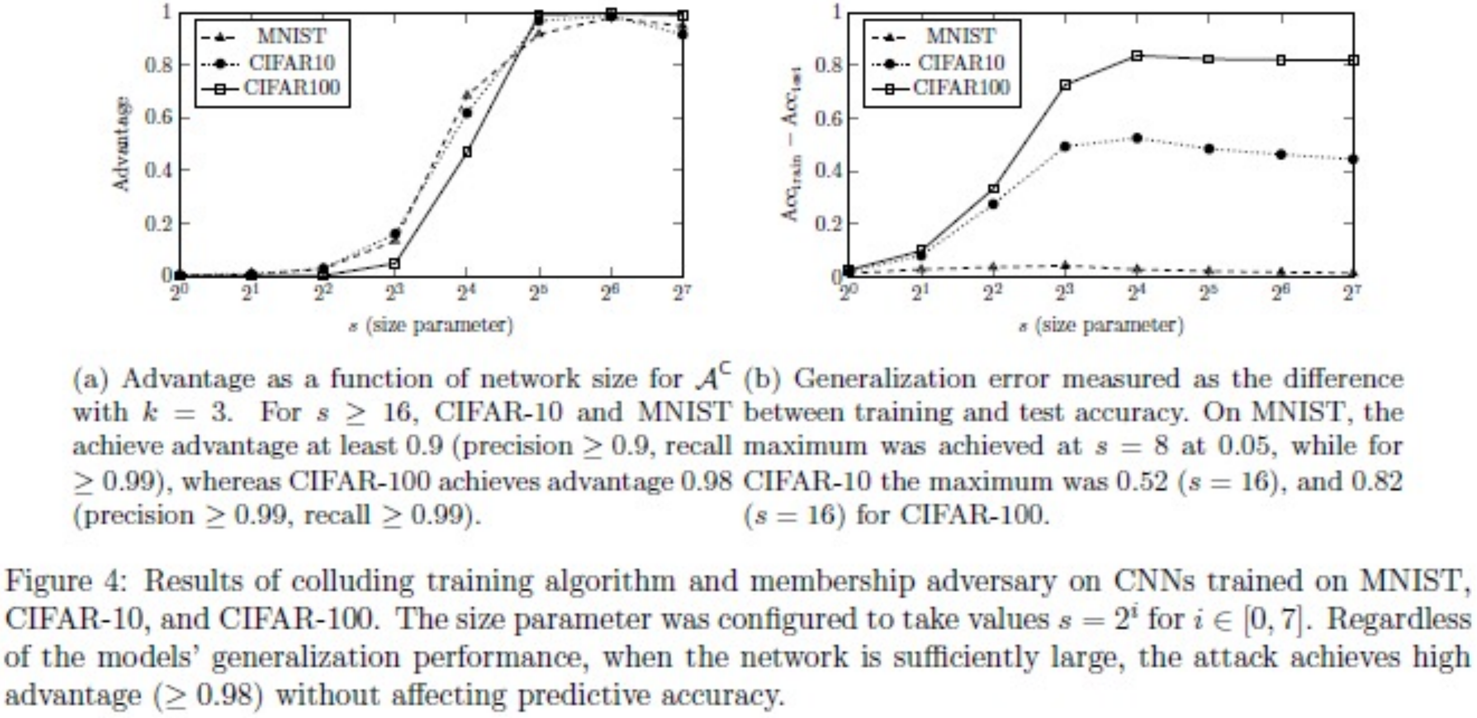


(a) t = VKORC1

(b) t = CYP2C9

Figure 3: Experimentally determined advantage for various membership and attribute adversaries. The plots correspond to: (a) threshold membership adversary (Adversary 2), (b) uniform reduction adversary (Adversary 6), (c) general attribute adversary (Adversary 4), and (d) multi-query reduction adversary (Adversary 7). Both reduction adversaries use the threshold membership adversary as the oracle, and $f_{\mathcal{A}}(\epsilon)$ for the attribute adversary is the Gaussian with mean zero and standard deviation σ_S.

Colluding Algorithm



(a) Advantage as a function of network size for $\mathcal{A}^{\mathsf{C}}$ with $k = 3$. For $s \geq 16$, CIFAR-10 and MNIST achieve advantage at least 0.9 (precision ≥ 0.9, recall ≥ 0.99), whereas CIFAR-100 achieves advantage 0.98 (precision ≥ 0.99, recall ≥ 0.99).

(b) Generalization error measured as the difference between training and test accuracy. On MNIST, the maximum was achieved at $s = 8$ at 0.05, while for CIFAR-10 the maximum was 0.52 ($s = 16$), and 0.82 ($s = 16$) for CIFAR-100.

Figure 4: Results of colluding training algorithm and membership adversary on CNNs trained on MNIST, CIFAR-10, and CIFAR-100. The size parameter was configured to take values $s = 2^i$ for $i \in [0, 7]$. Regardless of the models' generalization performance, when the network is sufficiently large, the attack achieves high advantage (≥ 0.98) without affecting predictive accuracy.

Results

- Theoretical and experimental results agree when the standard errors are known and decision boundary set accordingly.
- When the adversary does not know the standard error of the distribution, it performs better than theory predicts.
- Proposed attack offers nearly as strong as SOTA performance, with less computational requirements.

Results

- Attribute advantage is not as high as membership advantage
- **Advantage increases as overfitting increases.**
- The reduction adversaries are more effective than running the attribute inference attack directly.
- Collusion in membership inference increases advantage without decreasing the performance of the model.

Conclusion

Thoughts

- Dense read.
- Theoretical formalization supports the empirical results obtained from the experiments.
- Overfitting certainly plays a role in the advantage of an attacker in an inference attack but is not *necessary*.
- Proposed collusion algorithm is an intriguing step but seems unlikely to happen in the real world.