CY 7790, Lecture 15: Privacy risks in ML. Membership Inference

Pablo Kvitca, Zohair Shafi

November 8, 2021

The topic of the class today is model privacy attack, which are attacks against machine learning (ML) at testing time, to infer information about the model and its training data. We focused on privacy attack for training dataset inference, mainly membership inference and attribute inference. The two papers discussed today are a practical first approach to show membership inference attacks and a theoretical analysis of membership and attribute inference attacks and their connection to overfitting.

1 Shokri et al. Membership Inference Attacks Against Machine Learning Models

Membership Inference Attack Definition Membership inference refers to determining whether or not (binary) a given data record was part of a target model's training dataset.

Background Previous work has involved model inversion, however this technique cannot generate a specific input that was part of the training data.

Previous Work Previous work from the field of statistical analysis has focused on this topic. This work includes the concept o "Dalenius desideraum", which states that a model should not reveal more about the its input than what was already known. Note that this would not allow any useful model, since the purpose of the model is to apply some classification/prediction about its input.

Similarly, the model is expected to generalize on a population, so preventing the leak of privacy of general population members is not possible. So this is not considered a privacy breach. The current work focuses on whether models reveal unintended information about the members of its training dataset.

Threat Model This research considers a black-box access to the model (a white-box version that has access to the training data would be trivial). The adversary considered only has access to the output of the model. There are three variations of this adversary:

1. Can make multiple queries to the target model, used to synthesize data for the attack. Then a final query for the data record to do membership inference for.
2. Can use prior statistics and information about the population to synthesize data for the attack. Then a final query for the data record to do membership inference for.
3. Has access to a noisy version of the target model's dataset.

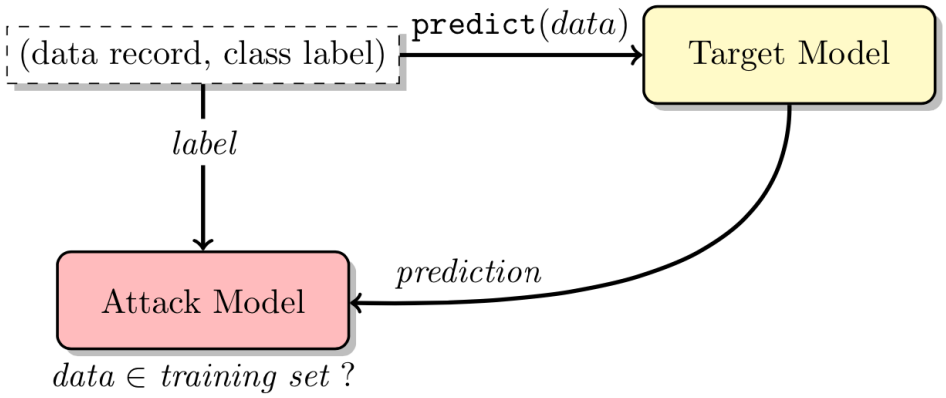


Figure 1: Attack Model

Attack Methodology The main methodology for the attack is training an attack model for the binary classification task of membership inference ("in"/"out" of the target model's training dataset). The input to the attack model is the output of the target model on the data record of interest. The success metric for the attack is whether the attacker can determine the membership for the given record.

Attack steps:

1. Synthesize shadow dataset (see below)
2. Train k shadow models
3. Get predictions from shadow models and generate an attack dataset with of the shadow model output and whether the record was part of the training data or not.
4. Split the attack model dataset into one per class ($cmodels, c = numberofclassesoftargetmodel$).
5. Train c attack models
6. Get the output for the data record from the target model. Apply the that output as input to the attack model (the one that corresponds to the class predicted by the target model).

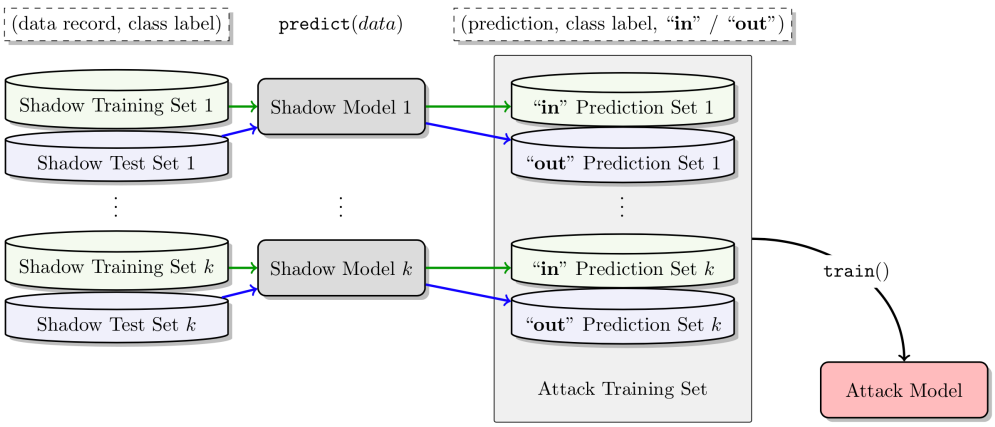


Figure 2: Attack Models

Intuition ML models behave differently on the input is data they were trained on and new data. Specially neural networks that seems to be have "memory" for inputs seen during training. Overfitting makes the classification confidence of the model to (usually) be higher for seen inputs. The prediction uncertainty for members/non-members increases by the number of classes from the entropy on the confidence values for each class. This gives the intuition towards developing the attack models to distinguish in/out.

Shadow Models These models are meant to behave similarly on seen/unseen data as the target model. These models have the same input shape and output shape as the target model. Their outputs are used to create the training dataset for the attack models. The shadow models are trained from the shadow datasets (which are synthesised to emulate the target model's data).

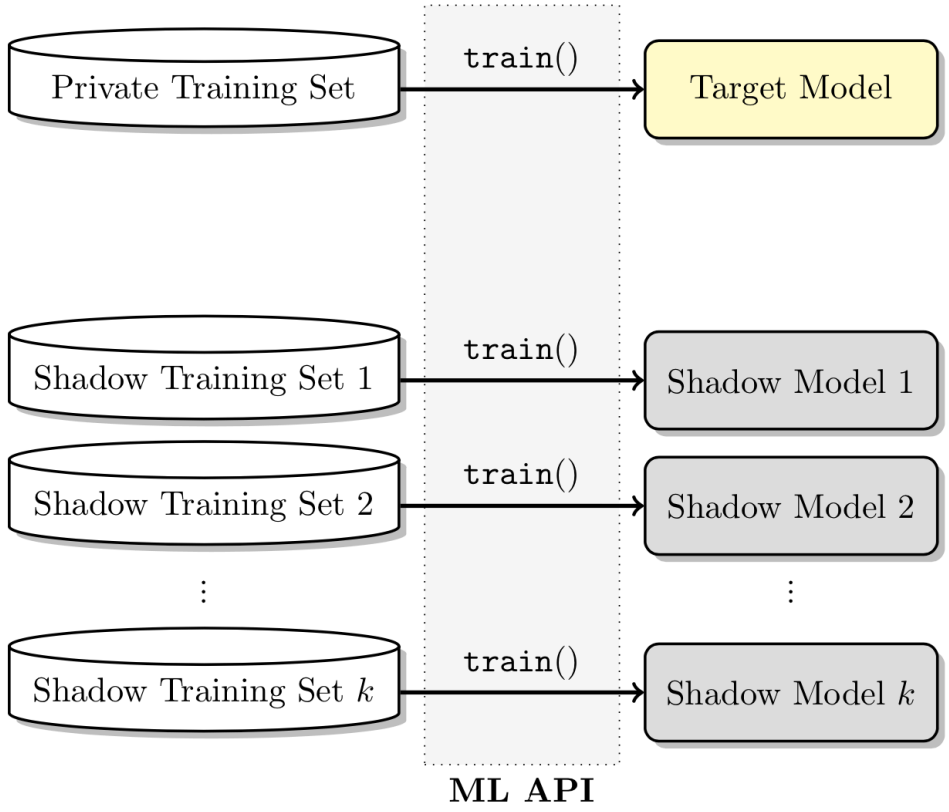


Figure 3: Shadow Models

Shadow Dataset Synthesis The data for training the shadow models should have a similar distribution as the data used by the target model. The authors present three variations for the shadow datasets.

1. **Model-based synthesis**: queries the target model to generate synthesized data. They use a hill-climbing algorithm where, for each class they take a randomly generated record, query the target model with it, then use hill-climbing to maximize the confidence towards the target class. This is repeated until the shadow dataset is filled.
2. **Statistics-based synthesis**: the shadow datasets are created by sampling from known statistics about the relevant population

3. **Noisy real data**: the attacker has access to data that is already similarly distributed to the data used by the target model. The authors simulate this by flipping values of randomly selected features for some percent of the data.

Evaluation and Target Models The attack was evaluated on models trained on the *CIFAR*, *Purchases*, *Locations*, *Texas Hospital Stays*, *MNIST*, and *UCI Adult* datasets. These were tested with models created: locally, the Google Prediction API, and Amazon ML.

Results

1. Increasing the number of shadow models would increase the performance of the attack.
2. Accuracy of the attack can vary considerably for different classes, because each class can have a different composition.
3. Models with more overfitting seems to be more "vulnerable", since it does not generalize well to inputs being its training data.

Mitigation

1. Regularization techniques such as dropout can help defeat overfitting and also strengthen privacy guarantees in neural networks
2. 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
3. Differentially private models may significantly reduce the model's prediction accuracy for small ϵ values

Limitations and Strengths

1. Limitation: Model based synthesis -¿ It may not work if the inputs are high-resolution images and the target model performs a complex image classification task
2. Strength: Not bounded to a particular dataset or model type
3. Strength: Realistic classification tasks

Class Discussion

Multiple Shadow Models The research suggests using k shadow models. They show the larger k models can improve the performance of the attack. k is NOT the number of classes of the original task, but actually the number of variations of the target model that are used for generating the attack models' training data. If the architecture/hyperparameters of the target model is known, all models would be fixed to that known information, but in the general case variation of architecture, methods, and hyperparameters for each shadow model might be useful. If it is known the target model was created using an ML-as-a-service platform, that same platform could be used for training the shadow models.

Shadow Models Input The shadow models are meant to simulate the target model on the original task, the input is the same feature vectors used by the target model, but from the synthesized dataset.

Multiple Attack Models The authors mention they train one attack model per class (of the target model), instead of one single (larger) attack model, then use the one that corresponds to the predicted class. This is done to "improve accuracy". The intuition behind this is each class can classify differently. Though there is no indication that a single attack model that is sufficiently large could not have the same performance.

Data Synthesis Newer approaches for synthesising the data for the shadow models could be used, such as GAN models.

Unbalanced datasets The distribution of the shadow datasets should be similar to the distribution of the target model's training. If there is a certain proportion of samples for each class for the target model, this should be known (and reflected on the shadow datasets).

Prediction Histograms Could the prediction histograms be used directly to perform some type of membership inference after querying the black box model multiple times?

Shadow Model Architecture Do the shadow models all have the same architecture? Given this can be understood as some form of a transfer learning task (given a completely black box setting), would it make more sense to have multiple architectures across the shadow models?

2 Yeom et al. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting

Problem Statement ML models can leak information about training data. Explore the relationship between privacy risk and *overfitting* between privacy risk and *influence*.

ML application require sensitive private information and they have been shown to leak that information about their training data once deployed. The author's goal is to characterize the effect of overfitting and influence on the level of advantage the adversaries have at attempting to infer specific facts about the training set.

Background The authors focus on two types of attacks: **Membership Inference**, whether a given record was used on the training data, and **Attribute Inference**, getting information about specific attributes of an individual record using the model.

Stability An algorithm is stable if for a small change on the input there is a limited change on the output. In the case of ML model training, whether a small change in the training dataset results in limited changes to the resulting model. This notion is related to $differential\ privacy$.

Overview of Adversaries We give a brief overview of the methods used by the authors. Throughout the paper, they present 7 types of adversaries with different capabilities and attack methodologies. **Adversary 1** assumes some knowledge about the loss function and then uses the threshold on the loss function to compute membership inference. **Adversary 2**, assumes that the error distribution is known. Once known, it queries the model to get the errors of the attack points to determine membership. **Adversary 3** is a bit of a departure from the previous two and assumes an adversary that colludes with the algorithm. The training set is augmented with modified data points that leak membership information. This adversary can be thought of also as a way of showing that overfitting is not the only mandatory condition for privacy leakage. **Adversary 4** attempts attribute inference. Similar to **Adversary 2**, **Adversary 4** attempts to examine the error generated by different attribute values and then picks the attribute that maximises the likelihood of the error distribution. **Adversary 5** assumes access to an attribute oracle that returns if a particular attribute of a data point is correct or not. This information is then used to infer membership. **Adversary 6** looks at the reverse case when access to an oracle with membership information is used to infer attribute information. A single attribute is uniformly sampled and the reconstructed data point is then passed into the oracle to check for membership. **Adversary 7** follows from **Adversary 6** where instead of a single target attribute, multiple are checked with multiple queries to the oracle. Each of these adversaries is defined in more detail below.

Membership Inference Attacks determine whether a given data point was part of the training data set for a target model

Experiment 1 Membership experiment

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.*

Figure 4: Experiment 1

Definition 4 (Membership advantage)**.** *The* membership advantage *of $\mathcal{A}$ is defined as*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) = 2\Pr[\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) = 1] - 1,$$

where the probabilities are taken over the coin flips of $\mathcal{A}$, the random choices of S and b, and the random data point $z \sim S$ or $z \sim \mathcal{D}$.

Equivalently, the right-hand side can be expressed as the difference between $\mathcal{A}$'s true and false positive rates

$$\mathsf{Adv}^{\mathsf{M}} = \Pr[\mathcal{A} = 0 \mid b = 0] - \Pr[\mathcal{A} = 0 \mid b = 1], \tag{2}$$

where $\mathsf{Adv}^{\mathsf{M}}$ is a shortcut for $\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$.

Figure 5: Definition 4

Membership Advantage Definition

Bounds from Differential Privacy Differential Privacy limits how much any one point in the training data may affect the outcome of the model, which limits the success of membership inference attacks. There is an upper bound on membership advantage, when the model's 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:*

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

Figure 6: Theorem 1

Theorem 1

Attribute Inference Attacks determine specific attributes about a data record from the output of a target model. In this case, "advantage" instead measure how much information about the target individual the algorithm leaks, for individual records in the training data.

Adversaries The authors define seven different 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 1 (Bounded loss function)**.** *Suppose $\ell(A_S, z) \leq B$ for some constant B, all $S \sim \mathcal{D}^n$, and all z sampled from S or $\mathcal{D}$. Then, on input $z = (x, y)$, A_S, n, and $\mathcal{D}$, the membership adversary $\mathcal{A}$ proceeds as follows:*

1. *Query the model to get $A_S(x)$.*
2. *Output 1 with probability $\ell(A_S, z)/B$. Else, output 0.*

Figure 7: Adversary 1

Adversary 2 - Membership Inference Adversary knows the exact error distribution, can compute which value of b most likely. Used on regression problems. Standard error published along with the release of the model. *Advantage and generalization error are closely related.*

Adversary 2 (Threshold)**.** *Suppose $f(\epsilon \mid b = 0)$ and $f(\epsilon \mid b = 1)$, the conditional probability density functions of the error, are known in advance. Then, on input $z = (x, y)$, A_S, n, and $\mathcal{D}$, the membership adversary $\mathcal{A}$ proceeds as follows:*

1. *Query the model to get $A_S(x)$.*
2. *Let $\epsilon = y - A_S(x)$. Output $\arg\max_{b \in \{0,1\}} f(\epsilon \mid b)$.*

Figure 8: Adversary 2

Adversary 3 - Membership Inference Attacker can influence the training algorithm or substitute it with a malicious one. Has a stable learning rule with a bounded loss function. *Overfitting is not necessary for membership advantage*. This is a algorithm substitution attack.

Adversary 3 (Colluding adversary $\mathcal{A}^{\mathsf{C}}$)**.** *Let $F_K : \mathbf{X} \mapsto \mathbf{X}$, $G_K : \mathbf{X} \mapsto \mathbf{Y}$ and $K_1, \ldots, K_k$ be the functions and keys used by A^{C}, and $A_{S'}$ be the product of training with A^{C} with those keys. On input $z = (x, y)$, the adversary $\mathcal{A}^{\mathsf{C}}$ proceeds as follows:*

1. *For $j \in [k]$, let $y'_j \leftarrow A_{S'}(F_{K_j}(x))$.*
2. *Output 0 if $y'_j = G_{K_j}(x)$ for all $j \in [k]$. Else, output 1.*

Figure 9: Adversary 3

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 $\epsilon_{stable}(n)$, 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 $\epsilon_{stable}(n) + knB2^{-d} + \mu(n, \mathcal{D})$ and adversary $\mathcal{A}$ such that:*

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

Figure 10: Theorem 4

Adversary 4 - Attribute Inference Adversary can approximate the error distribution of the target model. The adversary then can try all possible values for the sensitive attribute, then picks the value of the sensitive attribute with the highest probability. This is related to *model inversion*.

Adversary 4 (General)**.** *Let $f_{\mathcal{A}}(\epsilon)$ be the adversary's guess for the probability density of the error $\epsilon = y - A_S(x)$. On input v, y, A_S, n, and $\mathcal{D}$, the adversary proceeds as follows:*

1. *Query the model to get $A_S(v, t_i)$ for all $i \in [m]$.*
2. *Let $\epsilon(t_i) = y - A_S(v, t_i)$.*
3. *Return the result of $\arg\max_{t_i}(\Pr_{z\sim\mathcal{D}}[t = t_i] \cdot f_{\mathcal{A}}(\epsilon(t_i)))$.*

Figure 11: Adversary 4

Adversary 5 - Membership to Attribute The adversary can use an attribute oracle to accomplish membership inference. This adversary examines the *connections between the membership and attribute inference attacks*.

Adversary 5 (Membership $\rightarrow$ attribute)**.** *The reduction adversary $\mathcal{A}_{\mathsf{M}\rightarrow\mathsf{A}}$ 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.*

Figure 12: Adversary 5

Adversary 6 - Attribute to Membership The adversary is given some partial information on z and reconstructs the entire point to query the membership oracle. This adversary 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 $\perp$.*

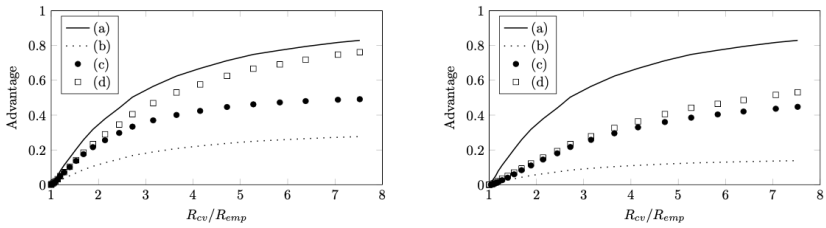
Figure 13: Adversary 6

Adversary 7 - Multi-Query Attribute to Membership Similar as Adversary 6 but allows multiple queries to the oracle.

Adversary 7 (Multi-query attribute → membership)**.** *Suppose that $t_1, \ldots, t_m$ 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 $\perp$.*

Figure 14: Adversary 7



(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.

comparison.png

Figure 15: Comparison of Adversaries 2, 6, 4 and 7

Results and Observations

- Both the theoretical and experimental results match, when the standard errors are known and the decision boundary for the models is set accordingly.
- In the case the adversary does not know the standard error of the model's distribution, it actually performs better than expected.
- The newly proposed attack method performs almost as well as the state-of-the-art attack and with less computational costs.

- In general, attribute advantage is not as high as membership advantage
- Advantage increases on models that overfit more, but they show with Adversary 3 that overfitting is not mandatory and membership information can leak with a provably stable model
- The the adversaries based on reduction are more effective than the direct attribute inference attack
- The collusion approach for membership inferences increases attack performance without decreasing the model's performance.

Discussion

Game-Style Adversaries The different adversaries are presented as "games" where different knowledge and capabilities are allowed for the attack. These similar to what is used in cryptography research.

Other than overfitting Overfitting is not the only factor that causes the ML to leak information about the training dataset. They introduce Adversary 3 to show provably stable models can leak information too.