Datamodel Distance: A New Metric for Privacy

Paul Lintilhac[1], **Henry Scheible**[1], **Nathaniel D. Bastian**[2]

[1] Thayer School of Engineering and Department of Computer Science, Dartmouth College, Hanover, NH 03755 USA
[2]Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY 10996 USA
paul.s.lintilhac.th@dartmouth.edu, henry.j.scheible.26@dartmouth.edu, nathaniel.bastian@westpoint.edu

Abstract

Recent work developing Membership Inference Attacks (MIA) has demonstrated that certain points in the dataset are often intrinsically easier to attack than others. In this paper, we introduce a new pointwise metric, the Datamodel Distance, and show that it is empirically correlated to and establishes a theoretical lower bound for the success probability for a point under the LiRA MIA. This establishes an explicit connection between the related concepts of Datamodels and Membership Inference via surrogate models, and also gives new intuitive explanations for why certain points are more susceptible to attack than others. We then use datamodels as a lens through which to investigate the Privacy Onion Effect.

Introduction

There has been great recent interest in membership inference attacks (MIA) that have high power, in that they can not only have a certain guaranteed maximum false positive rate, but that the true positive rate will be high even at low false positive rates. This is important, as the value of identifying a member in a dataset is usually much higher than the value of confirming that a target point is not a member. The approach of Carlini et al. (2022a) took this goal seriously, and developed a maximum-power hypothesis test for membership inference. However, even their test does not give any explicit guarantee of true positive rate under any natural assumptions. Ideally, one would like to know that, if a dataset is contaminated enough, there is a high probability of being detected as such by our MIA. To approach this problem, we take inspiration from the body of work on property testing (Goldreich, Goldwasser, and Ron 1998), a methodology that has recently seen some applications for evaluating a bounding an ML model's privacy properties (Jha and Raskhodnikova 2011) (Gilbert and McMillan 2019) as well as verifying robustness properties (Lintilhac, Ackerman, and Cybenko 2024). Following this general framing of the problem as one of property testing, the goal of this paper is to lower bound the probability of a particular membership inference attack using the assumption that the ML model is at least $\epsilon-$far from a certain property, under a certain natural definition of distance.

In order to find a test with a guaranteed true positive rate for a target point, we can make use of an assumption about how "far" the dataset is from being "clean" of this data point. This kind of quantitative metric for dataset membership is desirable, not only because it may give us some hope of having a guaranteed true positive rate, but because it is useful in general to measure not only whether a dataset contains a point, but also to what extent it is influenced by that point. A commonly used heuristic for the level of contamination of a dataset by a point is the number of repetitions of that point in the training set. One might consider this to be a proxy for how contaminated a dataset is with a particular point, and reasonably assume that the more times a data point is repeated in the training set, the more likely a MIA is to succeed. But while this has been demonstrated empirically (Duan et al. 2024), unfortunately there is no clear path to proving any such guarantee. Instead, we look to the work on datamodeling by Ilyas et al. (2022) for inspiration in defining a distance metric that accurately predicts true positives of the Likelihood Ratio Attack (LiRA) offline attack.

Background

LiRA: The Online Variant

In the online version of the LiRA attack, the target model is trained on a random subset of the full CIFAR dataset using $\alpha-$fraction of the training points, and then several so-called "shadow models" are trained using independently sampled subsets of the training data of the same size, some which contain the target point, and some which don't. In the most basic version of their online attack, the distribution of confidences (correct-class logits) for training sets that contain x and exclude x are then estimated using an assumed normal distribution. We let $\mathbf{Q}_{in}$ and $\mathbf{Q}_{out}$ be the normal approximations of confidence distributions for shadow models trained with and without the target point, respectively. Let C be the actual confidence of the target model, μ_{in}, σ_{in} the mean and standard deviation of the normal approximation of the in-distribution of confidences, and μ_{out}, σ_{out} the mean and standard deviation of the normal approximation of the out-distribution. Then $\mathbf{Q}_{in} = \mathcal{N}(\mu_{in}, \sigma_{in})$ and $\mathbf{Q}_{out} = \mathcal{N}(\mu_{out}, \sigma_{out})$. C is then compared to these two distributions using a likelihood ratio test, which rejects the null hypothesis when the ratio of the likelihood of C un-

der $\mathbf{Q}_{in}$ and $\mathcal{Q}_{out}$ exceeds a fixed threshold, z^*. This test achieves maximum power by the Neyman-Pearson Lemma.

While our main analysis in this paper relates to the offline attack, we pause to highlight an intuitive if somewhat tautological definition of distance. If we make the limiting assumption that the variances of the in and out distributions are equal, it can be easily shown that, under the assumption of normality of $\mathbf{Q}_{in}$ and $\mathbf{Q}_{out}$, the probability of inference is $log\left(\frac{P_{in}}{P_{out}}\right) = \frac{1}{\sigma^2}(C-\bar{\mu})(\mu_{in}-\mu_{out})$, where $\bar{\mu} := \frac{\mu_{in}+\mu_{out}}{2}$ is the average of the means of the two distributions. This definition therefore captures one of two key terms that gives the test power. One term $\frac{\mu_{in}-\mu_{out}}{\sigma}$, which is close to the definition of d, captures the property of being an "outlier". The other term, $\frac{C-\bar{\mu}}{\sigma}$, captures the notion of being "easy-to-fit".

LiRA: The Offline Variant

While the online LiRA attack has the benefit that it is the maximum-power test at a given false positive rate (FPR), this variant of the attack is not as practical to implement in practice, as it requires the training of models using the target point in advance, and therefore can't be deployed in real-time after receiving first knowledge of the target point. Hence we opt to focus instead on analyzing the more tractable and nearly-as-powerful offline LiRA algorithm.

In the offline variant of the LiRA attack, the attacker is not assumed to have the ability to train shadow models containing the target point, and therefore only the out-distribution of confidences is estimated. The most basic experiment performed in the original paper uses a similar setup as the online attack, where we train shadow models on several subsets of the training data that are randomly selected. However, this time we discard any subsets that contain the target point in order to estimate $\mathbf{Q}_{out}$ only, and then perform a one-sided hypothesis test that calculates the likelihood (based on a "z-score" of the confidence) of the observed confidence of the target model, C, under $\mathbf{Q}_{out}$. We again infer membership if this z-score exceeds some threshold, z^*. Note that every distinct FPR on the ROC curve corresponds to a contiguous interval of threshold values, and thus choosing this threshold corresponds to setting the FPR.

Connection with Datamodels

First introduced in (Ilyas et al. 2022), datamodels are used to predict a model's output (or some function thereof) for a single target point – or one at a time at least, using a separate model for each target – based on the particular subset of the data universe used for training. In this work the authors made two key empirical observations. First, they showed that simple linear predictors often model the output of a target point well, given an inclusion vector of which training points $S \subset D$ were used for training. In other words,

$$c(S, x) \approx \Theta^T \mathbf{1}_S + \theta_0$$

tends to do a good job of predicting the **correct class margin** of the function when trained on S. We argue below that the same is true when modeling the **confidence** of the model as a function of the training subset, a modification to the response variable which is directly applicable to the shadow models used in the LiRA attack. Second, they observe that the coefficients of the optimal weight vector Θ tend to encode similarity of the training points, in that the largest coefficients will tend to be for points that are most similar to the target. To fit a single datamodel function on CIFAR-10, the authors train 300,000 models on different size-25,000 subsets of the training data. Finally, a LASSO regression is performed to obtain the regularized coefficient values.

Analyzing the Offline Attack with Finite Data Universe

We now introduce the mathematical model used to connect the LiRA attack and datamodels. Definitions:

- $D = \{y_j\}, j \in [|D|]$: Full Dataset
- S_i: Shadow Model Training Dataset, so $S_i \subseteq D$. S_i are assumed to be uniformly random subset of size N from $D \setminus \{x\}$
- S': The dataset that the target model was trained on. S' is assumed to be the union of $\{x\}$ and a uniformly random subset of size $N-1$ from $D \setminus \{x\}$
- N:=$|S_i| = |S'|$, the size of the shadow target models' datasets.
- $x \in D$: target data point. For ease of notation, we assume that the index of x is 0, i.e. $x = y_0$
- Let $C = c(S', x)$ be the confidence of the target model trained on S'.
- Let $\tilde{c}(S', x) = \theta_x + \Theta^T 1_{S' \setminus x}$, where Θ is a vector of coefficients for each y_j, i.e. $\Theta = \{\theta_x, \theta_{y_1}, ..., \theta_{y_{|D|}}\}$. We assume that with probability at most δ_ϵ, $\tilde{c}(S', x)$ has error exceeding ϵ_d, i.e. $Pr[|c(S', x) - \tilde{c}(S', x)| \leq \epsilon_d] < \delta_\epsilon$.
- Let $c(S_i, x)$ be the confidence of the model when trained on shadow model dataset S_i.
- Let $\tilde{c}(S_i, x) = \Theta^T 1_{S_i}$
- Let $\bar{\theta} = \frac{1}{T}\sum_{y \in D \text{ s.t. } y \neq x} \theta_y$ be the global mean datamodel coefficient (excluding θ_x)
- Let $\hat{\theta}_{S_i} = \frac{1}{N}\sum_{y \in S_i \text{ s.t. } y \neq x} \theta_y$ be the sample mean datamodel coefficient (excluding θ_x)
- Let $C = c(S_i, x)$ be the confidence of the model under attack on x,
- Let μ_{out}, σ_{out} be the average and standard deviation, respectively, of $c(S_i, x)$ over M independently sampled datasets S_i of size N from $D \setminus \{x\}$.
- Let $\tilde{\mu}_{\text{out}}$, $\tilde{\sigma}_{\text{out}}$ be the average and standard deviation, respectively, of $\tilde{c}(S_i, x)$ over M independently sampled datasets S_i of size N from $D \setminus \{x\}$.

In the original work by Carlini et. al. (Carlini et al. 2022a) that introduced datamodels, θ_x was constrained to always be zero, however, relaxing this constraint is useful for our purposes, and we fit the LASSO regression using all training points, including the target point, which usually has a the largest coefficient. We will sometimes refer to θ_x as the "self-coefficient" or "diagonal coefficient", while we refer to $\bar{\theta}$ as simply the "mean similarity".

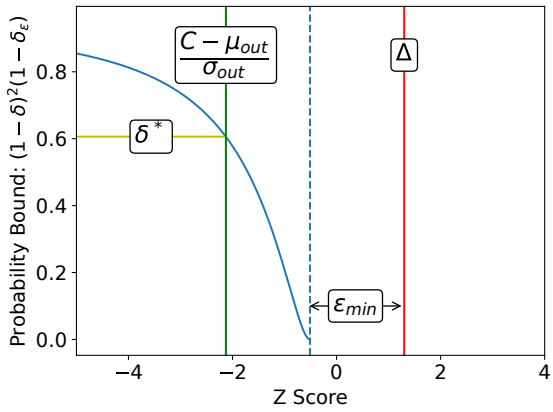

Figure 1: Example values of $\frac{\Delta}{\sqrt{N}}$, $\frac{C-\mu_{\text{out}}}{\sigma_{out}}$ and the RHS of Equation 2. The y-value of the blue curve indicates the probability (according to our bound) that $\frac{C-\mu_{\text{out}}}{\sigma_{out}}$ is at least the corresponding x-value. We denote δ^* specifically as the probability (according to our bound) that $\frac{C-\mu_{\text{out}}}{\sigma_{out}}$ is at least what we empirically observed it to be. See Equation 3 for a definition of ϵ_{min}.

In the offline attack, we construct M different shadow models from the out-distribution, each using N training points from D. This reduces the error in estimating μ_{out} and σ_{out}. In order to lower bound the offline LiRA attack success rate, we start by lower bounding the quantity $z := \frac{C-\mu_{out}}{\sigma_{out}}$ in terms of a certain statistic of the datamodel. The intuition behind the following calculations is that μ_{out} is well-approximated as an average of M sums of N random coefficients that are sampled from the same distribution, while C is well-approximated as the sum of $N-1$ random coefficients and a special self-coefficient. We can therefore use concentration arguments to put high-probability bounds on z.

Definition 1. *We denote the datamodel distance by the quantity,*

$$\Delta := \frac{\theta_x - \bar{\theta}}{\sigma_\theta} \tag{1}$$

We show in Appendix A that the LiRA attack z-score can not be too far below $\frac{\Delta}{\sqrt{N}}$. More precisely, we show that with probability at least $(1-\delta)^2(1-\delta_\epsilon)$,

$$z > \frac{\Delta}{\sqrt{N}} - \frac{2\epsilon_d}{\sqrt{N\sigma_\theta^2}} - \sqrt{\frac{1}{M\delta}} - \sqrt{\frac{1}{\delta}} \tag{2}$$

where δ_ϵ is the probability of satisfying the error bound on the datamodel approximation, which considered to be fixed. δ is our free confidence parameter which controls the tightness of the bound, which will ultimately turn out to be our lower bound on the TPR for outliers.

To intuitively understand this bound, observe a sketch of it in Figure 1. In this figure, we pick one target point and model, run the LiRA attack to compute a Z score $\frac{C-\mu_{\text{out}}}{\sigma_{out}}$, then compare that score to the bound derived above, which is plotted in blue. Note that if $\delta = 1$, then this reduces to

$$z > \frac{\Delta}{\sqrt{N}} - \underbrace{\left(\frac{2\epsilon_d}{\sqrt{N\sigma_\theta^2}} + \sqrt{\frac{1}{M}} + 1\right)}_{\epsilon_{\text{min}}}. \tag{3}$$

This means that no matter how low of a probability we ask for, our bound can be no tighter than some fixed $\epsilon_{\text{min}} \geq 1$. This is an intrinsic problem of the Chebyshev Inequality (see Appendix for details) being vacuous within one standard deviation of the mean.

Note that there are two different scenarios in which we can use the above inequality to give us a bound on the probability of $\frac{C-\mu_{out}}{\sigma_{out}}$ exceeding some threshold, z^*. In the first case where x is an outlier, $\Delta > z^*\sqrt{N}$, so we can adjust δ so that the lower bound matches with $z^*\sqrt{N}$. In that case, $1-\delta$ gives us a lower bound on the probability that $C - \mu_{out} > \sigma_{out} z^*$. The upper bound in this case tells us nothing about the probability of exceeding z^* (since the upper bound must always be greater than Δ, and thus can never coincide with z^*). In the second case, x is an inlier, i.e. $\Delta < z^*$, and we can adjust δ so that the upper bound on $(C - \mu_{out})$ matches with $\sigma_{out} z^*$. In this case, δ gives us an upper bound on the probability of membership inference using LiRA. While it is not the focus of this work, there could be some utility in conditionally providing guarantees of privacy to inliers. For example, in a scenario where each input represents a unique person whose data is included in training, guarantees for some individuals may still have some value to those individuals, even if such a guarantee isn't available to everyone. Conversely, the lower bound tells us nothing. In either case, we can only ever use a one-sided bound for a given target.

We can see from Equation 1 that $\frac{C-\mu_{out}}{\sigma_{out}}$ is lower bounded by the scaled datamodel distance $\frac{\Delta}{\sqrt{N}}$ less a variation term that depends on δ, M, N, and σ_θ. We can therefore think of Δ as a distance to the property of the joint model and training data system having memorized x, which itself is a more fine-grained substitute for the binary membership indicator traditionally considered to be the key metric in membership inference. In order to compute a lower bound on the probability of success of the LiRA, we set

$$z^* = \frac{\Delta}{\sqrt{N}} - \frac{2\epsilon_d}{\sqrt{N\sigma_\theta^2}} - \sqrt{\frac{1}{M\delta^*}} - \sqrt{\frac{1}{\delta^*}}$$

and solve for δ^*. The probability that $\frac{C-\mu_{out}}{\sigma_{out}} > z^*$ is therefore at least $(1-\delta_\epsilon)(1-\delta^*)^2$.

To frame this in a way that is in line with the formalism of property testing, we define the following property. We say that x is a $C-$inlier with respect to the target model if $\Delta < C$. We define the distance to this property as follows: the target model is $\epsilon-$far from being a $C-$inlier if $(\Delta - C)_+ > \epsilon$. Letting $C = \frac{z^*}{\sqrt{N}}$, this distance assumption combined with our result immediately implies that $\delta > O\left(\frac{N}{\epsilon^2}\right)$, where we have ommitted the dependency on ϵ_d, σ_θ, and M.

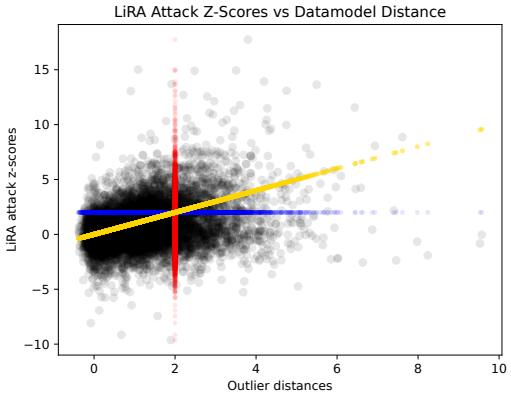

Figure 2: A scatterplot of the relationship between the LiRA attack's z-score and the datamodel distance. The yellow line shows the baseline of equal $\frac{\Delta}{\sqrt{N}} = z$. Points to the right of the vertical red line are outliers: they have $\frac{\Delta}{\sqrt{N}} > z^*$. They apparently carry a higher chance of having a LiRA attack score above z^*.

Bounding TPR for an Entire Dataset

Thus far, we have defined a property of a model's behavior on a specific target point known as the "datamodel distance", and we have shown that the datamodel distance can be used to lower bound the TPR. For a single example, the TPR is averaged over the randomness of the shadow model dataset selection, as well as the randomness of the training algorithm. To lower bound the TPR for an entire dataset, we can define the "aggregate datamodel distance" which lower bounds the probability of a successful inference for a randomly chosen point. Denoting by Ξ the event of a positive inference, we have:

$$(Pr\left[\Xi\right]) > (Pr\left[\Xi \wedge \Delta_y > z^*\right])$$

$$= (Pr\left[\Delta_y > z^*\right])(Pr\left[\Xi | \Delta_y > z^*\right])$$

$$= \frac{1}{N} \sum_{y \in S'} \delta(\Delta_y) I\left[\Delta_y > z^*\right],$$

where $\delta(\Delta_y)$ is our lower bound on the probability of positive inference for target $y \in S'$.

Figure 2 demonstrates that many of the positive inferences are not in fact from datamodel outliers. Since our bound does not lower bound such points, it will likely not be useful to bound the TPR of the entire training set using a very small FPR. While it may be possible to lower bound the z-score using an anti-concentration argument, such analysis is beyond the scope of this paper. Rather, if we decrease our threshold significantly, to say $z^* = 1.5$, this would include the majority of positive inferences. Therefore, we conclude that our method for bounding risk on an entire dataset is likely only practical for bounding the TPR on an entire dataset at a relatively high FPR, corresponding to a low z^*.

Experiments

Luckily, both the Datamodels paper by Ilyas et. al. (2022) and the LiRA attack paper by Carlini et. al. (2021) perform some of their tests using the CIFAR-10 dataset, creating an overlap in their experimental settings. We therefore use this dataset as our primary means of analyzing the relationship between datamodels and the success rates of the LiRA attack. While both papers used CIFAR-10, the LiRA paper used a version from Tensorflow datasets, while the dataset used in the datamodels paper used a dataset from torchvision. Since these datasets did not match precisely, we replaced the dataset in the LiRA attack to be the exact one used when training in the datamodels.

In our first experiment, we consider the exact same setup as the online LiRA attack, where the target models in question are trained on some random subset of the same training dataset as the shadow models. While this is not a realistic scenario (as openly admitted by the authors of the LiRA paper), it allows us to confirm the above relationship established theoretically in a controlled experimental setting. The only modification to this experimental setting was to use the CIFAR-10 test set as the training set, in order to cut down on computation time. Both the datamodels and the LiRA shadow models were trained using approximately 50% of all training points; however, the training masks used for the datamodels are selected as independent bernoulli random variables with $p = \frac{1}{2}$, and were therefore not always size-5000, but rather closely concentrated around this mean. On the other hand, the masks used for the LiRA shadow model training were selected uniformly from the set of all size-5000 subsets, as in the original LiRA attack paper. We trained 26,000 models in order to train our datamodels on 10,000 regression variables.

We argue that the Datamodel Distance metric encodes the degree to which a point is an outlier in the training set. This agrees with the intuition that the inclusion/exclusion of outliers has a greater effect on the outputs of the trained model than the inclusion/exclusion of inliers does. For a concrete example, refer to Figure 3. The left column (which shows the highest ranked cats in CIFAR-10 by Datamodel Distance) are noticeably weirder than the cats in the right column (which shows the lowest ranked datamodel distances). Both our empirical and theoretical results indicate that these "weird ones" are more vulnerable to the LiRA attack.

The Privacy Onion Effect, Through a New Lens

In a recent paper entitled "The Privacy Onion Effect: Memorization is Relative" by Carlini et al. (2022b), it was shown empirically that one cannot effectively reduce a dataset's exposure to membership inference attacks like LiRA simply by removing the points that are the greatest outliers. Instead, they found that when such points are removed from the training set, the remaining points that were previously "safe" from privacy attacks will then become outliers, and the success rate of the MIA on those points will increase. While the paper did offer some theoretical explanation for this effect via the example of the SVM, we can shed further light on this phenomenon by again leveraging datamodels.

Figure 3: This shows five highest (left column) and lowest (right column) ranked examples with respect to the Datamodel Distance metric in the "cat" class in CIFAR-10.

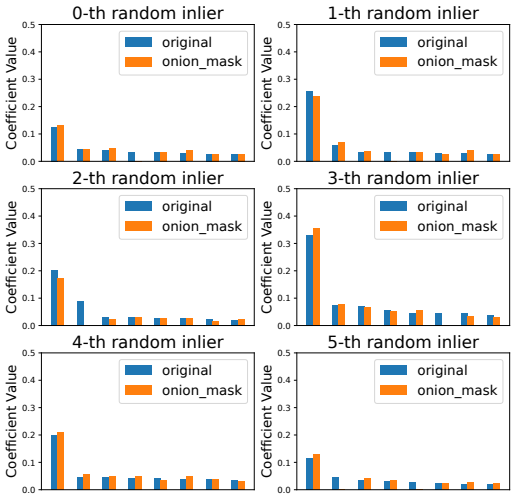

Figure 4: This plot shows the changes in the top datamodel coefficients for random inliers. We can see that the top coefficient (which is always the self-coefficient) doesn't always increase after removing the outliers from the training of the datamodels.

We first confirm the relevance of such a perspective by checking that an analagous effect is observed with "datamodel outliers". Specifically, if we create datamodels for the full training set, remove the top 5% of outliers as defined by our datamodel distance, and then retrain the datamodels with those points ommitted, does the datamodel distance of the remaining points increase? In particular, do the remaining points which already have a large distance increase disproportionately as compared to the points that are firmly inliers?

The answer to both of these questions turns out to be yes. Figures inliers,layer shows the results of our experiment using the CIFAR-10 test set. We found that, after removing the top 5% of datamodel outliers, the total datamodel distance for the next top-500 outliers (the next "layer" in the onion) experienced a median increase in distance of 19%, while the typical remaining 9,000 points increased only 6%. while 99% of the next layer outliers increased in distance, only 62% of the inliers' distance increased.

Why does this datamodel distance increase happen? There are three possible culprits: an increase in the diagonal coefficients θ_x in the numerator, a decrease in the mean similarity $\bar{\theta}$, or a decrease in the variance σ_θ. We found that, in these "next layer" outliers, $\bar{\theta}$ had virtually no change. However, θ_x increased by around 13% for next-layer outliers (and around 8% for inliers). σ_θ had a median decrease of 5% (and a 2 increase % for inliers), contributing to the increase in distance.

We conducted an investigation of why self-coefficients of the outliers in the second layer increased after removing the outermost 10%. Our first hypothesis was that the LASSO regression was causing ommitted variable bias when some im-

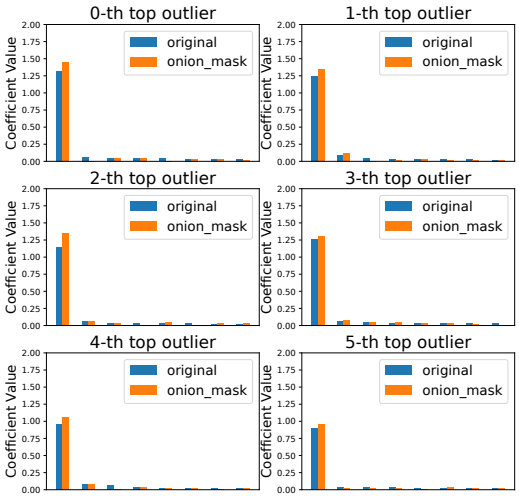

Figure 5: This plot shows the changes in the top datamodel coefficients for the top outliers. We can see that the top coefficient (which is always the self-coefficient) always tends to increase after removing the outliers from the training of the datamodels.

portant regressors were removed. While the conditions under which omitted variable bias (OVB) occurs in LASSO regressions are unclear to us, we re-ran the datamodel regression using simple OLS and found the same effect. OVB should not happen for uncorrelated predictors, but while our empirical correlations were low, the inherent randomness of the sampling process left many correlations in the 1%-5% range. This small amount of correlation could have caused other predictors' coefficients to increase overall. Since the diagonal coefficient θ_x is already the largest by far, this may have caused a disproportionate increase in θ_x, and thus an increase in the distance. This OVB may have been compounded by l1-regularization, which further amplifies the most significant coefficients while suppressing others.

Our investigation into why the variance term decreased after removing outliers was more conclusive. The coefficients corresponding to those outliers tended to be more extreme than other coefficients. Figure 4 shows the distribution of datamodel coefficients with and without the outliers removed to demonstrate this effect. Figure 6 shows that the standard deviation of datamodel coefficients corresponding to outliers is markedly higher than the standard deviation of non-outlier coefficients. As a result of σ_θ decreasing, the standard deviation of the shadow model's confidences also decreases, thus increasing the datamodel distance (and by implication, the LiRA success rate).

While there are still unanswered questions as to why the top coefficients increase, we believe the datamodels perspective on this phenomenon sheds further light on why the privacy onion effect is observed in practice.

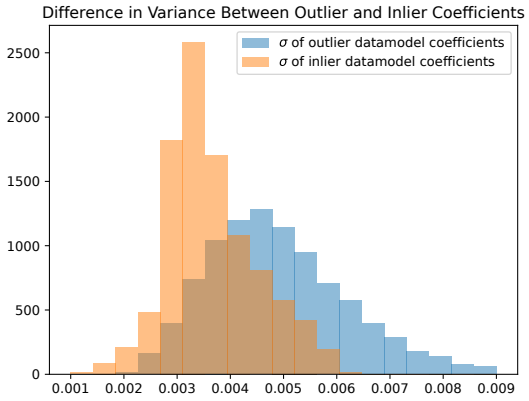

Figure 6: This plot shows the standard deviation for datamodel coefficients corresponding to outliers versus the standard deviation of coefficients for inliers. Each sample uses a single datamodel and calculates the standard deviation of these two disjoint subsets.

Future Work

We note that there is a term in our bound that can never be taken to 0. An explanation of this comes from the fact that, unlike the shadow models which can approximate the global distribution of confidences arbitrarily well, there is only one target model, and therefore some noise around this confidence is inevitable. Another way to look at this is that the Chebyshev inequality always has a gap of at least one standard deviation unit, since the bound becomes vacuous for $k < 1$. We investigated other bounds such as Hoeffding and found that they were even less tight, but it is likely that a different concentration inequality could give us a tighter bound. We leave this investigation for future work.

A technique known as "outlier privacy" was developed by Lui et. al. (2015), which seeks finds a better trade-off between privacy and utility by ensuring differential privacy only to those points that are considered outliers. The theoretical guarantees around our datamodel distance suggest that this metric could be ideal for this purpose.

Our work also relates to recent efforts to detect benchmark contamination in training data. In those cases, we may have a set of points $\{y^1, ..., y^R\}$ such that we want to test for inclusion of any of them in the training data. For these purposes, we may with to lower bound the probability that any of the LiRA attack z-scores exceeds some threshold Λ.

$$Pr[z^1 > \Lambda \vee z^2 > \Lambda ... \vee z^R > \Lambda] = 1 - Pr[z^1 \leq \Lambda \wedge z^2 \leq \Lambda ... \wedge z^R \leq \Lambda] \quad (4)$$

It is easy to show using the second-order Bonferroni inequality that

$$Pr[\cup_{i=1}^{R}(z^i > \Lambda)] \geq \sum_i (1-\delta_i) - \sum_{i,j}(1-\delta_i)(1-\delta_j) + \rho_{ij}\sqrt{\delta_i\delta_j(1-\delta_i))(1-\delta_j))}, \quad (5)$$

where each δ_i is the probability of inference for a single target. This can be further simplified by assuming that the correlations of inferences on pairs of points ρ_{ij} is either small or large compared to the probabilities of inference $(1-\delta_i)$. An interesting area of future research would be to upper bound the correlation of the TPR between pairs of target points based on the similarity of their datamodel embeddings, and thus lower bound the probability of inference for this "one of many" attack.

One limitation of our paper is that it assumes a finite data universe from which the datamodel is estimated, and from which the target model's shadow models are sub-sampled. This means that it does not account for the possibility of inferring membership on a point that was not in the superset at all, but is an exact copy (or close match) of one that the target model was trained on. For example, if our target model is invariant under some transformation of each input, we might ask whether any input in the equivalence class defined by this transformation is in the training data. Unfortunately, our datamodel distance as currently defined has nothing to say about the probability of inferring membership for such a transformed input.

To handle this concern, we could shift our paradigm to that of a continuous input data distribution, and we assume that the attacker can sample from this distribution. Given an unseen input and our set of models used for datamodeling, we can re-run the LASSO regression to obtain a datamodel for the new point. In order to define an analogous outlier distance that we can use to lower bound the probability of inferring membership in this setting, we could instead look at the quantity $\frac{\theta_{max}-\bar{\theta}}{\sigma_{out}}$, noting that the value of the self-coefficient has been replaced by the *maximum* coefficient. Our definition of $\bar{\theta}$ is now the average of the datamodel coefficients with the maximum coefficient excluded.

In order to prove an analagous lower bound in this setting, we can consider a set of "shadow data models" (which may not actually be created in practice, but rather hypothesized as a theoretical tool), trained on a disjoint but identically distributed set of data. As long as the number of training points for the shadow models is the same as those used for datamodeling, the distribution of datamodel coefficients for the shadow models should be very similar to that of the defender's datamodel coefficients. We see this type of analysis as tractable and worth pursuing, in order to extend our results to unseen inputs that have such a closely matching input in the training data.

Conclusion

Our work makes explicit the connection between datamodels and shadow models in a privacy attack. We see the development of datamodels as a fundamental tool that can be used for many different purposes, from measuring train-test leakage, to predicting counterfactuals, and measuring similarity of training points. Our work adds an additional utility to datamodels by using them to define a formal property of the model that provably relates to its privacy. Other properties such as differential privacy can also be characterized as properties of the function mapping datasets to outputs.

We showed that the more of an outlier a point is in terms of our datamodel distance, the more power the LiRA attack has. We argued that Carlini's attack is a property test for a point being an inlier, a global property of the datamodel function on the entire training data universe, which can be lower bounded through local testing – in this case subsampling and creating shadow models. While simply being included in the dataset does not give us any guarantee of a successful attack, having a certain datamodel distance does. If the LiRA attack fails, this implies the point is likely an inlier. This may occur naturally, or because of a process of differential privacy applied during training.

Appendix A: Details of Analyzing the Offline Attack with Finite Data Universe

In this section, we will seek a lower bound on $\frac{C-\mu_{out}}{\sigma_{out}}$, the "z-score" term from the LiRA test. We start with the same assumptions on the data model error not exceeding ϵ_d with probability at least $(1-\delta_\epsilon)$ to arrive at the following inequality:

$$\begin{aligned}&(E[\tilde{c}(S',x)] - E[\tilde{\mu}_{out}]) - (C-\mu_{out}) \\ &< 2\epsilon_d + (E[\tilde{c}(S',x)] - \tilde{c}(S',x)) + (E[\tilde{\mu}_{out}] - \tilde{\mu}_{out})\end{aligned} \tag{6}$$

The next step is to divide both sides of this equation by σ_{out}. We note that $E[\tilde{c}(S',x)] = (N-1)\bar{\theta} + \theta_x + \theta_0$, and $E[\tilde{\mu}_{out}] = N\bar{\theta} + \theta_0$. We define $\Delta = \frac{\theta_x - \bar{\theta}}{\sigma_\theta}$, and note that $\sigma_{out} = \sqrt{N\sigma_\theta^2}$. Our bound thus becomes:

$$\begin{aligned}&\frac{C-\mu_{out}}{\sigma_{out}} > \frac{\Delta}{\sqrt{N}} - \\ &\frac{2\epsilon_d + (E[\tilde{c}(S',x)] - \tilde{c}(S',x)) + (E[\tilde{\mu}_{out}] - \tilde{\mu}_{out})}{\sigma_{out}}\end{aligned} \tag{7}$$

By Chebyshev's inequality, we have:

$$Pr\left[\frac{(E[\tilde{c}(S',x)] - \tilde{c}(S',x))}{\sqrt{Var(\tilde{c}(S',x))}} > k\right] < \frac{1}{k^2}$$

Letting $\delta_c = \frac{1}{k^2}$, we know that $k = \sqrt{\frac{1}{\delta_c}}$. It follows that, with probability at least $(1-\delta_c)$,

$$\frac{(E[\tilde{c}(S',x)] - \tilde{c}(S',x))}{\sqrt{(N-1)\sigma_\theta^2}} < \sqrt{\frac{1}{\delta_c}},$$

where the variance is over the random selection of $(N-1)$ coefficients from the datamodel. Similarly, for the difference between the expected and empirical mean of the out-distribution, with probability at least $(1-\delta_\mu)$,

$$\frac{(E[\tilde{\mu}_{out}] - \tilde{\mu}_{out})}{\sqrt{\frac{N}{M}\sigma_\theta^2}} < \sqrt{\frac{1}{\delta_\mu}},$$

where the variance is of the average of M shadow models, each summing over N randomly selected coefficients. Taking N to be large, we arrive at the final bound:

$$\frac{C - \mu_{out}}{\sigma_{out}} > \frac{\Delta}{\sqrt{N}} - \frac{2\epsilon_d}{\sqrt{N\sigma_\theta^2}} - \sqrt{\frac{1}{M\delta_\mu}} - \sqrt{\frac{1}{\delta_\mu}} \quad (8)$$

Again assuming each concentration bound holds with probability $(1 - \delta) = (1 - \delta_c) = (1 - \delta_\mu)$, the probability that both bounds hold simultaneously is at least $(1-\delta_\epsilon)(1-\delta)^2$.

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement No. HR00112420351 and the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory under Support Agreement No. USMA21050. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

References

Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; and Tramer, F. 2022a. Membership Inference Attacks From First Principles. arXiv:2112.03570.

Carlini, N.; Jagielski, M.; Zhang, C.; Papernot, N.; Terzis, A.; and Tramer, F. 2022b. The Privacy Onion Effect: Memorization is Relative. arXiv:2206.10469.

Duan, M.; Suri, A.; Mireshghallah, N.; Min, S.; Shi, W.; Zettlemoyer, L.; Tsvetkov, Y.; Choi, Y.; Evans, D.; and Hajishirzi, H. 2024. Do Membership Inference Attacks Work on Large Language Models? arXiv:2402.07841.

Gilbert, A.; and McMillan, A. 2019. Property Testing for Differential Privacy. arXiv:1806.06427.

Goldreich, O.; Goldwasser, S.; and Ron, D. 1998. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4): 653–750.

Ilyas, A.; Park, S. M.; Engstrom, L.; Leclerc, G.; and Madry, A. 2022. Datamodels: Predicting Predictions from Training Data. arXiv:2202.00622.

Jha, M.; and Raskhodnikova, S. 2011. Testing and Reconstruction of Lipschitz Functions with Applications to Data Privacy. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, 433–442.

Lintilhac, P.; Ackerman, J.; and Cybenko, G. 2024. Research Report: Testing and Evaluating Artificial Intelligence Applications. *Langsec: Language Theoretic Security*.