AM207 Final Paper

AM 207
December 17, 2021 Max Guo (interested in Research), Wenqi Chen, Xingyu Liu, Yang Xiang
Github Repository: https://github.com/charlotte12l/am207_fisher

1 Problem Statement

Machine learning models may implicitly leak the data they are trained on, either through the model itself or through their predictions. In other words, a malicious adversary may have the ability to find training data information just given the trained machine learning model. Obviously this is not a desirable property for privacy reasons, especially because the training data may contain sensitive attributes. The question of how to quantify the information leakage is the primary problem examined in this paper. This paper tries to propose a method, i.e. Fisher Information Loss, to quantify the data leakage of the models.

2 Context/Scope

Data leakage tends to happen since machine learning models trained on sensitive data are usually made public. Even if the models are not publicly accessible, data can still be implicitly leaked through model predictions. Without mitigating measures, adversaries may be able to infer training set membership and extract sensitive attributes.

Data privacy is of vital importance to users and keeping private data and sensitive information safe is paramount. It can be dangerous if items like healthcare information, financial data, and other personal consumer data get into the wrong hands. The lack of access control regarding personal information can put individuals at risk for fraud and identity theft. Additionally, on the government level, a data breach may risk the security of entire countries.

Therefore, it is really important to take the data leakage into consideration while building models. And how to quantify such leakage will be a vital problem.

3 Existing Work

Developed in the past two decades, a commonly used technique to protect data privacy is differential privacy [1]. Differential privacy works by adding noise to the data set before using the data for model training. In this way, it is more difficult to infer the data through model or model predictions.

There also exist other methods to measure data leakage. Farokhi and Kaafar [2] propose an information theoretic measure of the vulnerability of individual examples to membership inference attacks. Carlini et al. [3] propose a heuristic, which is able to infer the susceptibility of data to model inversion attacks.

More recently there have been more studies using specifically Fisher information as a measure of privacy. Farokhi and Sandberg [4] investigated the relationship of privacy with Fisher information to estimate error through the use of Cramer-Rao bound. Additionally, Farokhi and Sandberg [4] use Fisher information as a practical alternative for differential privacy in protecting data privacy specifically in the case of household smart meters.

4 Contribution

Although differential privacy is a common method for protecting data privacy, there are two disadvantages with this method and the existing work on quantifying data leakage. Firstly, differential privacy can only provide a worst-case guarantee of data privacy, instead of measuring leakage with respect to specific examples. Secondly, differential privacy implicitly degrades when correlations exist in the dataset.

To address the above issues, this paper proposed an example-specific and correlation-aware measure of data leakage using Fisher information. The proposed metric, Fisher information loss (FIL), can measure leakage with respect to specific examples, attributes, or sub-populations within the dataset. This is a large contribution to the research of quantifying data leakage because the previous work does not identify vulnerability at the level of the individual or sub-population. And unlike differential privacy, FIL does not implicitly degrade when data is correlated. Work from this paper also builds on prior studies of Fisher information as a measure of privacy in several directions, including a broader application to generalized linear models as well as the use of Fisher information loss in an algorithm, Iteratively Reweighted Fisher Information Loss, to provide fair protection against privacy attacks for different subgroups.

5 Technical Content

In this section we will cover the mathematical/algorithmic results from the paper. Section 5.1 gives a high level summary of the technical definitions and results from the paper. We cover notation in Section 5.2. We go over the definitions of the Fisher Information Matrix and Fisher Information Loss and we discuss their properties in Section 5.3 and Section 5.4. We give an overview of some of the mathematical derivations behind Fisher Information Loss for the specific cases of linear and logistic regression in Section 5.5. Finally, we finish with a discussion on Iteratively Reweighted FIL in Section 5.6.

5.1 High Level Overview

Intuitively, for a random variable dependent in some way on a unknown parameter, the Fisher Information of that particular parameter is a measure of how much information the observed values of the random variable contains about the parameter. For example, if we consider two different Normal random variables with different (known) variances and unknown means, observations from the Normal random variable with less variance will tell you more about what its mean parameter is, so the Fisher information for the unknown mean parameter of the smaller variance Normal random variable will be smaller than the corresponding Fisher information for the larger variance Normal.

In this paper, we treat the model as a random variable (for example, the parameters of the model follow some multivariate distribution) that depends on the parameters which are the training data. The Fisher Information Matrix and Fisher Information Loss then naturally define a some measure of how much information the provided model (the random variable) contains about the training data (the unknown parameters).

The benefit of this approach is that there is a very strong theoretical result involving the variance of any unbiased estimator and Fisher Information. The Cramer-Rao Bound lower bounds the variance of any unbiased estimator in terms of the Fisher Information. Thus, given the Fisher Information Matrix of the training data for a specific model, we would have a theoretical guarantee on the variance of any unbiased estimator an adversary might use to estimate the training data. This paper, through many experimental results, attempts to validate this theory.

5.2 Notation

We take the notation from the paper. Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the training dataset of the model ($\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$). $\mathcal{A}$ is some random learning algorithm that takes the training dataset and outputs a hypothesis $h \in \mathcal{H}$, the hypothesis space. $p_{\mathcal{A}}(h|\mathcal{D})$ is the conditional probability of obtaining h given $\mathcal{D}$ under the randomness of $\mathcal{A}$.

For example, in the case of a linear or logistic regression with loss function l, we let our randomized learning algorithm be $\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + \mathbf{b}$, where $\mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, and our hypothesis is completely characterized by parameters w:

$$w^* = f(D) = \text{argmin}_w \sum_{i=1}^{n} l(w^T \mathbf{x}_i, y_i) + \frac{n\lambda}{2} \|w\|_2^2 \tag{1}$$

Furthermore, we let f_i be f as a function of training data point $(\mathbf{x}_i, y_i)$ (keeping other training data points fixed).

5.3 Definitions

Definition 5.1 (Fisher Information Matrix). We define our Fisher Information Matrix (FIM) as

$$\mathcal{I}_h(\mathcal{D}) = -\mathbb{E}_h[\nabla^2_{\mathcal{D}} \log p_{\mathcal{A}}(h|\mathcal{D})] \tag{2}$$

(See Section 7.1 for our evaluation of this notation).

Definition 5.2 (Fisher Information Loss). We say that $h \sim \mathcal{A}(\mathcal{D})$ has Fisher information loss (FIL) of η with respect to $\mathcal{D}$ if

$$||\mathcal{I}_h(\mathcal{D})||_2 \leq \eta^2,$$

where $||\mathcal{I}_h(\mathcal{D})||_2$ denotes the 2-norm, or largest singular value, of the FIM. A smaller η means h contains less Fisher Information about the training data.

5.4 Properties

Now we discuss some of the properties of the Fisher Information Matrix and Fisher Information Loss mentioned in the paper. We do not go into deep technical detail because none of these properties particularly affect the experimental results or their interpretations.

- **Subsets**. We can take subsets of the full FIM in order to obtain the FIM for a certain example (including all attributes), or for a certain attribute (including all examples).
- **Composition**. Given an evaluation of k unique but independent randomized algorithms $\{h_i \sim \mathcal{A}_i(\mathcal{D}) | i = 1, \ldots, k\}$ with FIM of $\mathcal{I}_{h_i}(\mathcal{D})$, the FIM with respect to all of the h_i's is given by $\mathcal{I}_{h_1,\ldots,h_k}(\mathcal{D}) = \sum_{i=1}^{k} \mathcal{I}_{h_i}(\mathcal{D})$. If $||\mathcal{I}_{h_i}(\mathcal{D})||_2 \leq \eta_i^2$, then combined FIL is $\leq (\sum_{i=1}^{k} \eta_i^2)^{1/2}$. Essentially, we have a bound on the FIL in the case that we train multiple models on the same dataset and release all of them to the adversary, in terms of the FIL of each individual model.
- **Closed under post-processing**. If $||\mathcal{I}_h(\mathcal{D})||_2 \leq \eta^2$, then for any function $g(h)$ we have $||\mathcal{I}_{g(h)}(\mathcal{D})||_2 \leq \eta^2$. In other words, there is no way to increase the FIL by post processing the data.

5.5 Computing FIL

Consider the linear setting as defined in Section 5.2. In this scenario, we have that the FIM is:

$$\mathcal{I}_{w^*}(D) = \frac{1}{\sigma^2} J_f^T J_f \tag{3}$$

where J_f is the Jacobian of $f(D)$ with respect to data D. Then the FIL (Fisher Information Loss) is given by

$$\eta = \frac{1}{\sigma} \|J_f\|_2 \tag{4}$$

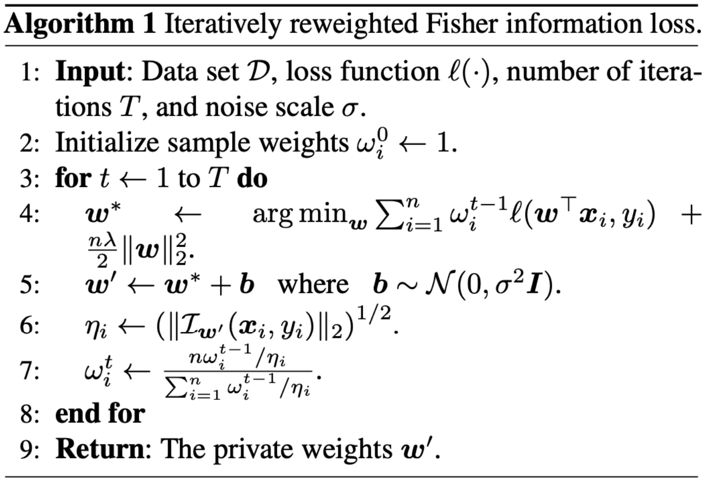
Since we want to compute the example specific FIL, we need to compute the Jacobian with respect to a specific data point (x_i, y_i), which is J_{f_i} given by

$$J_{f_i}|_{x_i,y_i} = -H_{w^*}^{-1}\nabla_{x,y}\nabla_w l({w^*}^T x_i, y_i) \tag{5}$$

The paper goes on to provide derivations of FIL for Linear Regression and Logistic Regression. Essentially, we need to compute the Hessian $\nabla_w^2 l$ and concatenate $\nabla_x \nabla_w l$ and $\nabla_y \nabla_w l$ to calculate J_{f_i}. For the sake of brevity we do not reproduce all the mathematical equations here, but they are written explicitly in the paper and implemented in our code.

5.6 Iteratively Reweighted FIL (IRFIL)

A related problem to protecting a machine learning model from information leakage is to provide equitable protection for different subgroups of data within the entire training dataset. Under the premise that example-specific FIL provides a good measure of how much leakage a model has for any example, this problem motivates the idea of constructing a model that has similar example-specific FIL for all examples. The paper provides an algorithm for constructing such a model in the linear/logistic regression regime, which it calls Iteratively Reweighted FIL (IRFIL). As seen in Figure 5.1, the algorithm continuously updates weights ω_i of each sample until convergence of the weights, which the paper shows empirically leads to a model with equal per-example FIL.



Algorithm 1 Iteratively reweighted Fisher information loss.

```
**Input**: Data set $\mathcal{D}$, loss function $\ell(\cdot)$, number of iterations $T$, and noise scale $\sigma$.
Initialize sample weights $\omega_i^0 \leftarrow 1$.
**for** $t \leftarrow 1$ to $T$ **do**
    $\boldsymbol{w}^* \leftarrow \arg\min_{\boldsymbol{w}} \sum_{i=1}^n \omega_i^{t-1}\ell(\boldsymbol{w}^\top \boldsymbol{x}_i, y_i) + \frac{n\lambda}{2}\|\boldsymbol{w}\|_2^2$.
    $\boldsymbol{w}' \leftarrow \boldsymbol{w}^* + \boldsymbol{b}$ where $\boldsymbol{b} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.
    $\eta_i \leftarrow (\|\mathcal{I}_{\boldsymbol{w}'}(\boldsymbol{x}_i, y_i)\|_2)^{1/2}$.
    $\omega_i^t \leftarrow \frac{n\omega_i^{t-1}/\eta_i}{\sum_{i=1}^n \omega_i^{t-1}/\eta_i}$.
**end for**
**Return**: The private weights $\boldsymbol{w}'$.
```

Figure 5.1: IRFIL Algorithm (taken from paper)

6 Experiments

In this section we detail our approaches to experimentation for the paper. We implement FIL for two toy datasets for both linear and logistic regression as pedagogical examples demonstrating successfully the ability of FIL to characterize data points that are more prone to leakage in Section 6.1. Then we give instances of successes and failures of applying FIL for the MNIST dataset in Section 6.2. As a comparison, we also repeat the MNIST experiments in the paper in the Appendix, in Section 10.1. Then we compare our implementation of IRFIL to the paper's implementation of IRFIL using the aforementioned toy dataset in Section 6.3. In Section 6.4 we briefly discuss a portion of the paper on adversarial attacks that we did not spend so much effort on and provide our reasoning for why. Finally, we provide novel experimentation in Section 6.5 that demonstrates our own ideas for different applications of FIL.

We note that, due to the parallel nature of groupwork, some portions of this experimentation section used the original paper's implementations. This includes the entirety of Section 6.5 and Section 6.4, as well as other sections in which the original paper's code and results are used in comparison to our own. However, we were using them for novel and original experiments in Section 6.5. Our own implementations are used in Section 6.1 and Section 6.2. Our GitHub repository contains both our own implementation of FIL as well as one heavily based on the paper's implementation.

6.1 Validating FIL on Toy Datasets

In this section we generated toy datasets for linear regression and logistic regression and found the per-example FIL for each.

6.1.1 Linear Regression

For the linear toy datasets in , we generate 9 datapoints $y = 5x + \varepsilon$ for $x \in \{1, 2, \dots, 9\}$, where $\varepsilon \sim N(0, 1)$. For and an outlier with $x = 0, y = -8$. After fitting the data with linear regression model and calculate per-sample fisher information loss, we get Figure 6.1. As we expected, the outlier has a much larger Fisher Information Loss than the remainder of the points. This makes sense because, given information of the model (the line) and values of the remainder of the data points, an adversary can infer the outlier point the most. However, it is not completely true that the closer the data points are to the line, the lower the Fisher Information Loss.

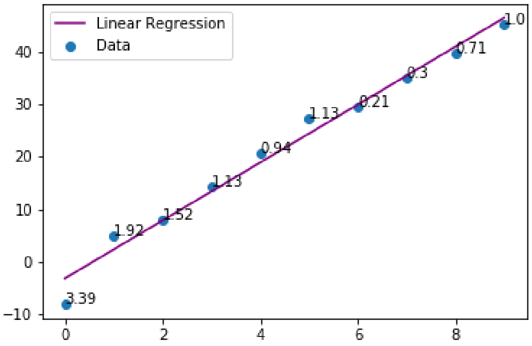


Figure 6.1: Linear Regression for Toy Data

6.1.2 Logistic Regression

For the Logistic Regression Toy Dataset, we generate 2D dataset (x_1, x_2), where if label $y = 1$, $x_1 = x_2$; if label $y = 0$, $x_1 = x_2 + 1$. We insert an data point with label $y = 1$ in a region where it is considered an outlier, as seen in Figure 6.2. This figure also shows us fitting the data with logistic regression model and calculate per-sample fisher information loss. Note again how the outlier data point has a much higher FIL than the rest.

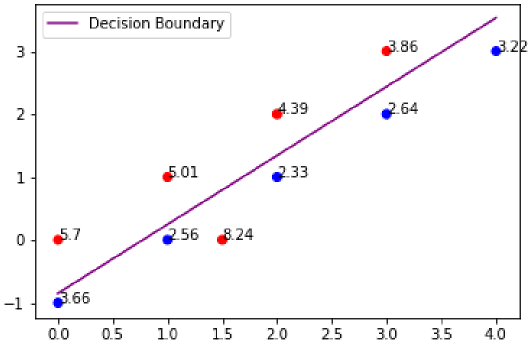


Figure 6.2: Logistic Regression for Toy Data. Blue denotes $y = 0$ and Red denotes $y = 1$.

6.2 Validating FIL in Logistic Regression - MNIST Dataset

Next, we decided to attempt to replicate the MNIST experiments from the paper using our implementation of FIL. However, we did not agree with the paper's choice to use linear regression for MNIST because MNIST is used inherently as a classification dataset, so we chose to only replicate the logistic regression portion. We normalized the data and then used PCA to project onto the first 20 principal components. Then we decided to test on more digits other than 0/1, which goes further than the paper's tests.

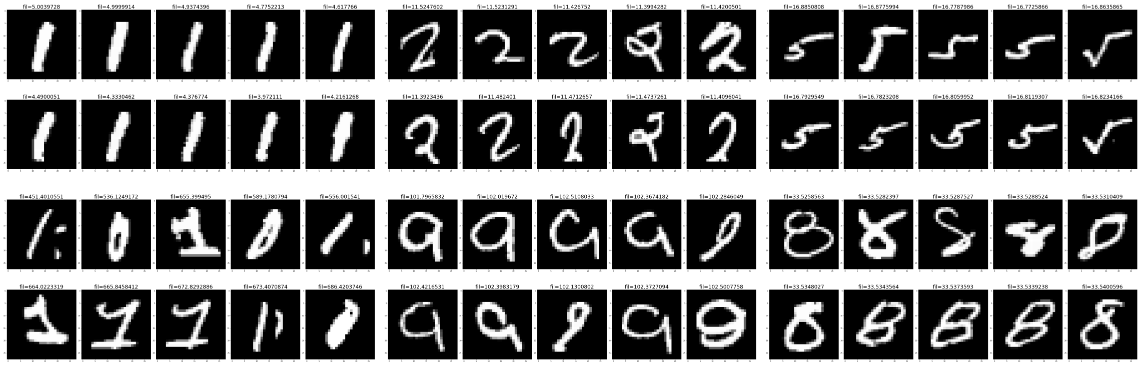


Figure 6.3: Three experiments of a Logistic Regression Classifier on MNIST datasets. The first two compared blocks (the first column) come from the experiment with LR trained on 0/1. The second column comes from the experiment with LR trained on 2/9. And the last column comes from the experiment trained on 5/8. The FIL above each picture is the corresponding FIL value for each picture. These three columns show the images associated with the lowest 10 (first row) and highest 10 (second row) FILs of the respective training datasets.

From Figure 6.3 we can see the FIL results on different digits from logistic regression. As is also illustrated in the paper, in the experiment on digits 0/1, larger FIL represents 'outliers'. The pictures on the right half will be more irregular and they have larger FIL. But the experiments on 2/9 and 5/8 do not reveal similar trends. The FIL values for the digit pictures are similar although they have irregular shapes, and we do not have a comparison metric for the idiosynchratic nature of any particular image (other than FIL itself and our intuition). Therefore, it may be reasonable to conclude that FIL may not be as valid as is stated in the paper for all the scenarios.

6.3 Testing Iterative Reweighted FIL

Next, we decided to implement the Iterative Reweighted FIL on the toy linear and logistic datasets from before. We chose parameters of 50 steps and 0.1 noise (standard deviation) for both, and the results are shown in Figure 6.4 and Figure 6.5.

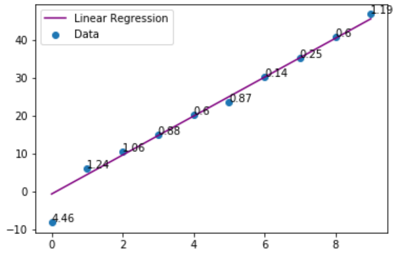


Figure 6.4: IRFIL Linear Regression Toy Data

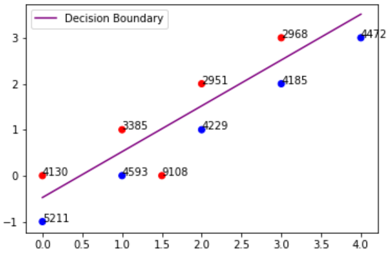


Figure 6.5: IRFIL Logistic Regression Toy Data

As evident from these images, IRFIL did not actually output a model that equalizes the per-sample FIL. Moreover, the FIL values are largely dependent on the number of iterations, as seen in Figure 6.5. These results did not match the paper's, so we also ran the paper's code on the datasets and show the result from Linear and Logistic Regression in Figure 6.6 and Figure 6.7.

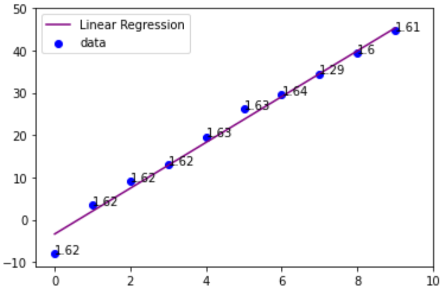


Figure 6.6: IRFIL Linear Regression Toy Data, Paper's Implementation

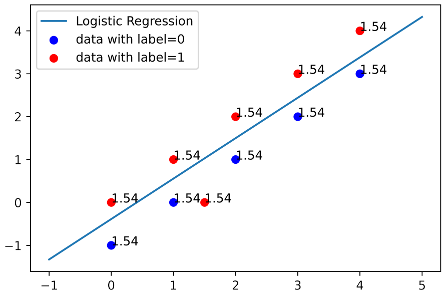


Figure 6.7: IRFIL Logistic Regression Toy Data, Paper's Implementation

Note in both our's and the paper's implementation, the model largely disregards the outlier(s) in its training, which makes sense because the weight for the outliers gets lower and lower over each iteration because they have large FIL's (see Figure 5.1). However, the paper's implementation also weights the FIL's of each data point, which explains the difference between the results of our implementation and the paper's. We offer criticism of this approach in Section 7.3.

6.4 Adversarial Attacks

The original paper implements a few adversarial attackers and tests the relationship between mean FIL per example and the ability of each attacker to invert and obtain the training data. However, these attackers must all satisfy strict conditions. First, the adversarial attacks assume that all of the regularity conditions hold such that the FIM exists and the Cramer Rao Lower Bound applies (the authors assert these conditions are satisfied for our models). Our adversary is also limited to unbiased estimators of the unknown data. Finally, if our adversary is trying to estimate some of the data, they have access to the remainder of the data. For the white-box attack, the adversary has access to the complete training dataset except the target attribute of the example under attack. For the black-box attack, the adversary has access to the target example except the value of the target attribute.

We do not focus much on this portion of the paper because we felt the primary contribution of this paper was for proposing a new measure of how much information a model contained about its training parameter, and the ability for any specific adversary to invert the training data was more dependent on the adversary's capabilities than the FIL's. In particular, the restrictions on the types of adversaries that the paper assumes and the implemented adversaries of the paper may not generalize to all potential adversaries. However, for the sake of checking the paper's code and seeing if the results are consistent, we include code in our GitHub repository which includes the paper's implementation of adversarial attacks. The results show that both the white-box and black-box accuracies degrade as the mean Fisher Information Loss decreases. This means that the larger the FIL loss is, the more likely the adversarial attacks happen, which is consistent with the meaning of FIL.

6.5 Extensions

In this section we propose novel uses of FIL for two problems that we've discussed extensively in class - out of distribution data and overfitting.

6.5.1 Using FIL to detect Out of Distribution Data

We now propose to use FIL as an indicator of Out-of-Distribution (OOD) data. This is a natural suggestion as we've seen how outliers tend to have larger FIL than the normal data points. Now our idea is that we can apply FIL in a reversed way to regard it as an indicator of such outliers. We conduct several experiments to validate our assumptions.

Note that we are slightly abusing the proper definition of Fisher Information Loss here, since properly FIL is a function of the training data only. Howevere, here we are now considering FIL as a function of any possible data point, and from the following experiments we feel as though this generalization may be useful.

From homework 7 of AM207, we investigated a dataset of OOD in the task of classification. On the homework, we tried to implement Bayesian methods to probe whether a data point is OOD through measuring its epistemic uncertainty, which is the variance of the distribution over classification probabilities estimated from Bayesian posterior probabilities. However, revealing epistemic uncertainty turns out to be a difficult task, especially when the data points are far from the decision boundary. With FIL, the larger the FIL is, the more out of distribution the corresponding data may be, so we implement it as an indicator of such OOD data.

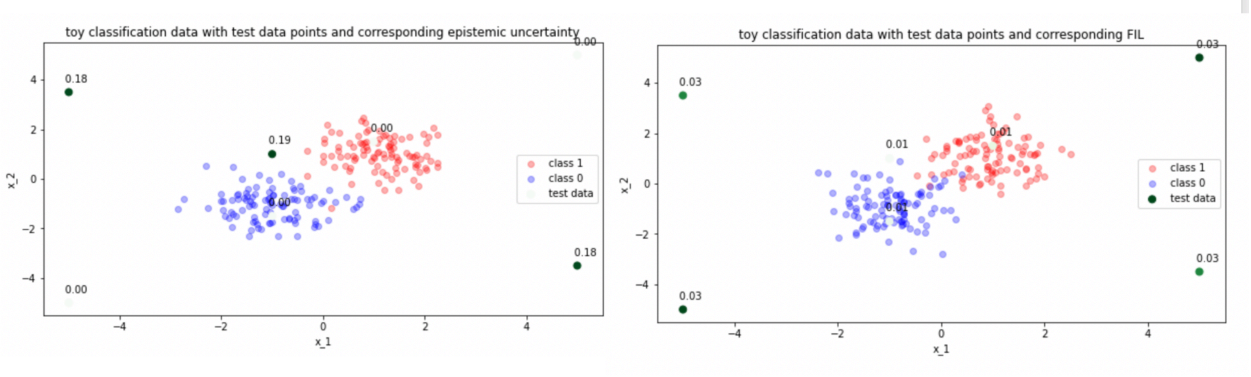


Figure 6.8: Toy classification data with test points and corresponding epidemic uncertainty and FIL. The left figure contains the test points with epistemic uncertainty as labels, which is the variance of the distribution over classification probabilities estimated from Bayesian posterior probabilities in HW7, and the right figure contains the test points with FIL as labels. For the test points, the darker the green is, the larger the corresponding indicator is. The average FIL of training data is 0.01 and the average epistemic uncertainty of the training data is 0.007.

From Figure 6.8, it can be seen that FIL is a more promising indicator as it can discriminate the points that are OOD (i.e., the points at the corners) from the points in the distribution. Note also that the in-distribution points have similar FIL as the average of training points.

Another experiment we conduct for OOD data is computing the FIL on the MNIST dataset *training and testing on different digits*. We train the logistic model on 0/1 dataset and test on other digits, regarding the other numbers as OOD data. As the results in Table 1 shows, the average FIL of such data is significantly larger than that of training dataset. This also indicates that FIL can work as an indicator of OOD or outliers.

	0	**1**	2	3	4	5	6	7	8	9
Mean of FIL	**0.05**	**0.04**	0.11	0.12	0.11	0.11	0.11	0.1	0.09	0.1

Table 1: FIL results of MNIST dataset trained on 0/1. The first row represents the digits of the picture.

In order to see whether the claim will also hold for linear regression model, we use the toy dataset to investigate the FIL distribution of data points on the map based on a toy dataset. Although the general assumption still holds that the larger the distances between the model and the data points, the larger FIL will be, there is still low FIL in the middle area where the training points are missing. We conjecture that FIL may not be a good indicator of OOD in the region where the model has high confidence (where confidence can be defined by Bayesian or MLE confidence intervals, as we discussed in class).

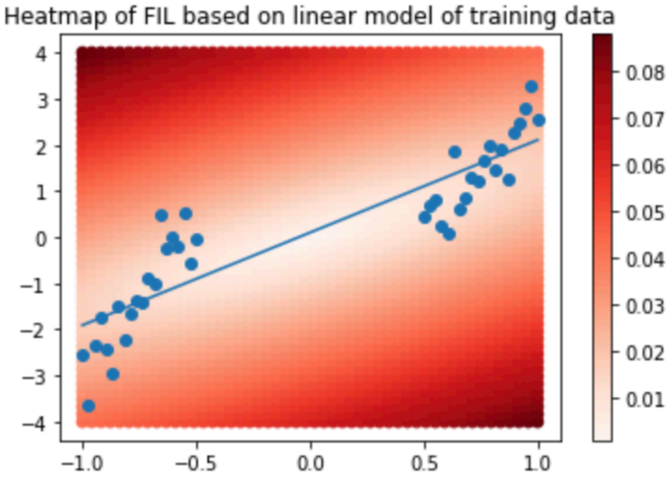


Figure 6.9: Heatmap of FIL on toy data with linear regression model. The blue points are training data and the blue line is the fitted linear model. The red heatmap represents the FIL distribution of corresponding points.

From these experiments, we conclude that FIL has a lot of potential to be applied as an indicator of OOD data but may not perform well in all settings (e.g. in a region where the model has high confidence).

6.5.2 Using FIL to detect Overfitting

We also propose that FIL can act as an indicator of overfitting to select appropriate models. This is because FIL reveals how much information about training data is carried by the model. In other words, we can probe whether the model is too dependent on the training data through FIL.

In order to validate our assumptions, we test different degrees of polynomial regression on the toy dataset from Figure 6.9 (the regression task mentioned in the OOD section). From the results in Figure 6.10, we can see that FIL can represent the reversed effect of validation MSE. Generally, we will use cross-validation to observe whether the current model will overfit and choose the corresponding best model. But with FIL, we can directly try to figure out whether it is overfitting with only the training data. It will be especially beneficial when we do not have enough data to conduct the validation test.

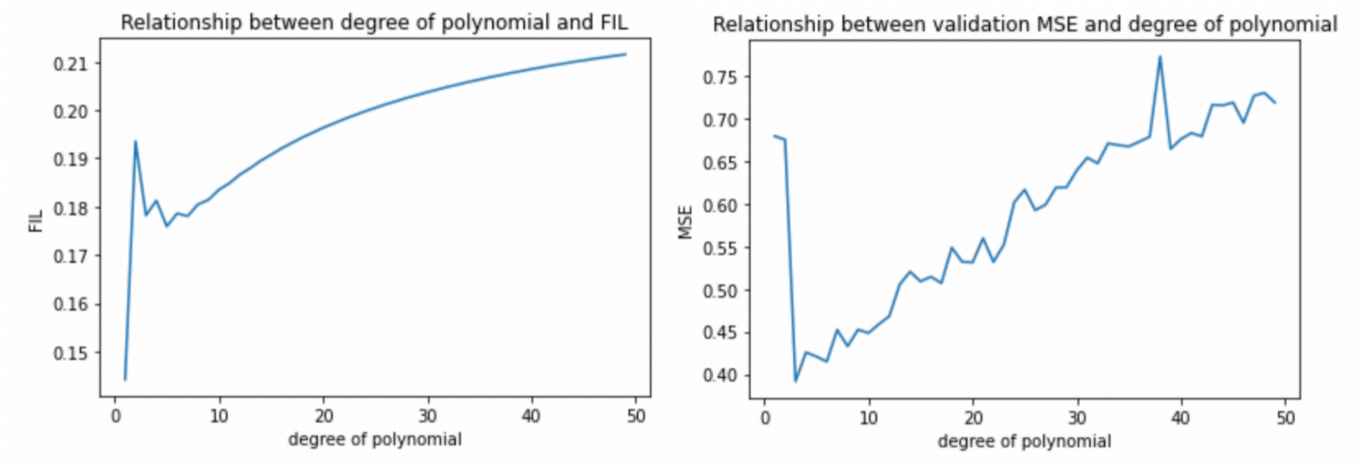


Figure 6.10: Results of the polynomial regression on toy dataset. The left figure represents the relationship between degree of polynomial and average of FIL over all train data points. The right figure represents the relationship between the validation MSE and degree of polynomial

We hypothesize that, with larger degree polynomials that tend to overfit the data, each individual data point leaks more and more information about the peculiarities of the model, so the FIL's increase with increasing degree polynomials. This suggests FIL as a potential metric for measuring overfitting.

7 Evaluation

Overall, we believe that Fisher Information Loss is a very promising metric for how much information a model tells us about a particular example. We believe that all of the mathematical derivations in the paper are sound. We also think that, given our experiments, it is promising in some other fields as well, such as OOD detection and overfitting detection.

However, we do have some criticisms of the paper. We already discussed some in the previous Experiments section, but we elaborate on more specific ones here.

7.1 Notational Concerns

Recall the exact definition in the paper of the Fisher Information Matrix is:

$$\mathcal{I}_h(\mathcal{D}) = -\mathbb{E}_h[\nabla^2_{\mathcal{D}} \log p_{\mathcal{A}}(h|\mathcal{D})]$$

However, we note that, if the right side's expectation is taken with respect to the randomness of $h \sim \mathcal{A}(\mathcal{D})$, then $\mathcal{I}_h(\mathcal{D})$ should not be parametrized by h, since there is no dependence of the right hand expectation on h. This is supported by the definition of Fisher Information, which is most definitely not parametrized by any one observation of the random variable. The only reasonable interpretation is that $\mathcal{I}_h(\mathcal{D})$ implicitly denotes $\mathcal{I}_{h^*}(\mathcal{D})$, where $h^* = \mathbb{E}[h]$ is the "optimal" model returned by $\mathcal{A}$, assuming $\mathcal{A}$ is unbiased.

In particular, the definition of Fisher Information Loss $\mathcal{I}(\mathcal{D})$ should not be dependent on $h \sim \mathcal{A}(\mathcal{D})$, and instead the definition should be for a singular $h = \mathbb{E}[\mathcal{A}(\mathcal{D})]$.

7.2 Per-Example FIL vs. Dataset FIL

In the paper, the author defines two kinds of Fisher Information Loss. One is for the FIL per sample which is the type of FIL we have primarily been discussing, and one is for the FIL of the whole dataset, which is defined as the 2-norm or the largest singular value, of entire FIM. However, although the paper defines such dataset FIL, it is not used anywhere in the paper, nor is it implemented. We believe the original idea of an entire-dataset FIL is to evaluate the whether the whole dataset is sensitive to attack compared with others. But after some experiments we argue that there is no concrete mathematical trend which can be understood in the difference between the dataset FIL in different scenarios. That may be the reason why the paper did not implement it and use it in its experiments.

7.3 Relevance of Reweighted FIL

From the comparison in Section 6.3, we can see that for IRFIL, the results of our implementation is quite different from the paper's implementation regarding the values of the final per-example FIL, but is actually quite similar regarding the final fitted model. That is because in the IRFIL algorithm, each iteration, both we and the authors will fitted a new model with updated weights, but the authors also choose to compute the new FIL with the newly fitted model and then reweight each sample's FIL. However, we think it makes more sense to just compute the new FIL with the newly fitted model. Manually weighting samples to get a more evenly distributed FIL across the dataset seems trivial and unhelpful.

Therefore, after digging into the algorithm of reweighted FIL, we found that it actually cannot be significantly beneficial to privacy protection in actual applications. The key idea of reweighted FIL algorithm is to even out the FIL of each data point for the whole dataset. But the algorithm is only changing the weights used to calculate FIL while the model do not change much. In other words, the consequence of equalizing FIL's in the training data will only happen if the attacker uses the same trick as weighted FIL to calculate which data point exposes more information, which is almost definitely not the case in the actual scenario. Therefore, we conclude that the reweighted FIL algorithm is not as promising as the paper claims. It would be more useful if the algorithm is implemented in a way to focus more on revising the model.

8 Future Work

In this paper, we've suggested two novel ways of applying FIL in the broader scenarios of machine learning: probing out-of-distribution data points and overfitting models. Although we have implemented experiments for several cases, there is still remaining extra work we will do in the future to validate these novel ideas.

Firstly, we have identified a problem in probing OOD data with FIL: FIL will not perform as expected in regions where the model has high confidence. In order to solve this problem, we may try to introduce Bayesian uncertainties to complement the drawbacks of FIL in these scenarios. We will try to propose a modified FIL so that it can be also effective when the model has low variance.

In addition, the effectiveness of probing overfitting with FIL will be an important point to be focused on. We have verified that it can perform well in polynomial regression. However, limited to the methods of computing FIL proposed in the paper, it is difficult for us to implement it in more complicated models, such as tree-based models and even neural networks. In the future, we will try to modify FIL so that it can have the capability to handle the complicated models. If the assumption still holds, we will be able to come up with a powerful indicator of overfitting, which will be vital for the research of machine learning.

9 Broader Impact

FIL is extremely useful to quantify data leakage, and we have shown its possibilities for being effective in probing out-of-distribution data and overfitting models. Therefore, FIL can be utilized as a superb probe for practitioners of machine learning to evaluate the model in terms of data privacy and overfitting before launching it to the real application. Instead of adjusting the models after getting the results from real application, the practitioners may be able to know its effects on data leakage and overfitting which are generally hard to probe with only training data in the beginning. This may contribute to saving large amount of money and preventing the dangerous consequences caused after launching the models in the real world.

However, we should still be careful about the usage of FIL, which may be used in a harmful way. For example, if the model designer completely relies on FIL to probe the possibility of data leakage, the model may still be at risk for leaking data. After all, FIL can only provide a comparative approach of evaluating which model will be more dangerous to leakage issues instead of acting as a criteria of safety. In other words, we can only use it to compare different models and adjust them accordingly instead of knowing whether the model is safe or not. The misuse of FIL will cause terrible consequences. The affected communities will be the organization or company of applying the models and most importantly, the users who has data stored in the company or organization. Therefore, the precise understanding of the scope of application of FIL will be vital in the real world scenarios.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[2] F. Farokhi and M. A. Kaafar, "Modelling and quantifying membership information leakage in machine learning," 2020.

[3] N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," 2019.

[4] F. Farokhi and H. Sandberg, "Fisher information as a measure of privacy: Preserving privacy of households with smart meters using batteries," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4726–4734, 2018.

[5] A. Hannun, C. Guo, and L. van der Maaten, "Measuring data leakage in machine-learning models with fisher information," in *Conference on Uncertainty in Artificial Intelligence*, 2021.

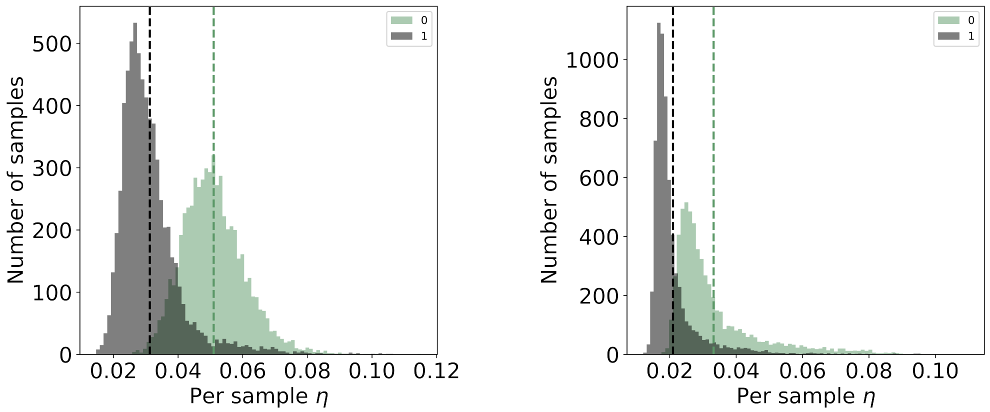
10 Appendix

10.1 Repeating the paper's MNIST experiments

We now describe our repetition of the paper's work. The paper uses MNIST to perform binary classification of the digits 0 and 1 using a training dataset of 12,665 examples. They also normalize all inputs, and then project each input using PCA onto the top twenty principal components for the dataset.

For linear regression, we didn't apply L2 regularization. Besides, we transformed label $\{0, 1\}$ to $\{-1, 1\}$ to get a better results as the orignal paper[5] indicated. Figure 1(a) shows the histograms of the per-example η with linear model, and we can see that digit 0 has a overall larger fisher information loss than digit 1, which implies that the model in general contains more information about images of 0 than of 1. Also, in Figure 2(a) we visualized the eight images with the largest and smallest η to do sanity check. We can see that in the second row, these correspond to the digit 1 written in a very typical manner and has smaller FIL, while those in the first row are more idiosyncratic and thus has larger FIL. The results are well-aligned with the original paper's [5] results and also intuitively make sense: the more idiosyncratic a sample is, larger the fisher information loss of that sample is.

For logistic regression, we set L2=8e-4 as the original paper[5] did, and used limited-memory BFGS to compute the minimizer. Figure 1(b) shows the histograms of the per-example η with logistic model, and we can see that similar to linear model, digit 0 also has a overall larger fisher information loss than digit 1, which implies that the logistic

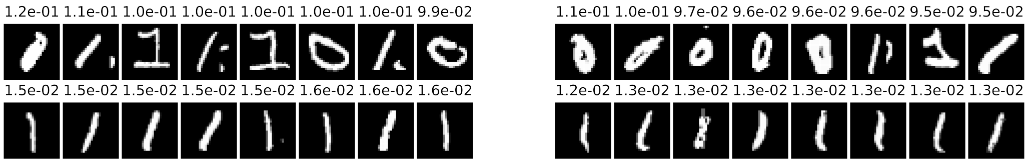


(a) Linear, MNIST

(b) Logistic, MNIST

Figure 10.1: Histograms of per-example η separated by class label for the MNIST training sets for linear and logistic regression. Each class label's mean η is denoted by the dashed vertical line.

model contains more information about images of 0 than of 1. Also, in Figure 2(b) we visualized the eight images with the largest and smallest η, and the assumption about idiosyncratic yield larger FIL still holds.


1.2e-01 1.1e-01 1.0e-01 1.0e-01 1.0e-01 1.0e-01 1.0e-01 9.9e-02
1.5e-02 1.5e-02 1.5e-02 1.5e-02 1.5e-02 1.6e-02 1.6e-02 1.6e-02
1.1e-01 1.0e-01 9.7e-02 9.6e-02 9.6e-02 9.6e-02 9.5e-02 9.5e-02
1.2e-02 1.3e-02 1.3e-02 1.3e-02 1.3e-02 1.3e-02 1.3e-02 1.3e-02

(a) Linear, MNIST

(b) Logistic, MNIST

Figure 10.2: The eight images with the smallest and largest η over the MNIST training sets for linear and logistic regression. The number above each individual image is the corresponding η.

Table 2: The mean $\bar{\eta}$($\pm$ standard deviation) of the example-level η and test accuracy before and after IRFIL.

Model	$\bar{\eta}$	Accuracy
Linear	0.040 ± 0.014	100
+IRFIL	0.047 ± 0.000	99.8
Logistic	0.027 ± 0.012	99.8
+IRFIL	0.024 ± 0.000	99.7

Finally, we applied IRFIL algorithm on the MNIST dataset in Figure 10.3, which plots the standard deviation of the per-example η against the number of reweighting iterations. After 15 iterations, both Linear and Logistic Regression model's per-sample η converge to a similar low level. Table 2 shows the mean and standard deviation of η, as well as the test accuracy for models trained with and without IRFIL. We can see that after applying IRFIL, the test accuracy and average FIL $\bar{\eta}$ do not change much, while the standard deviation in η decreases significantly, which means the information leakage are more equally-distributed. Overall, IRFIL achieves fairness in privacy loss with little change in accuracy or average privacy loss.

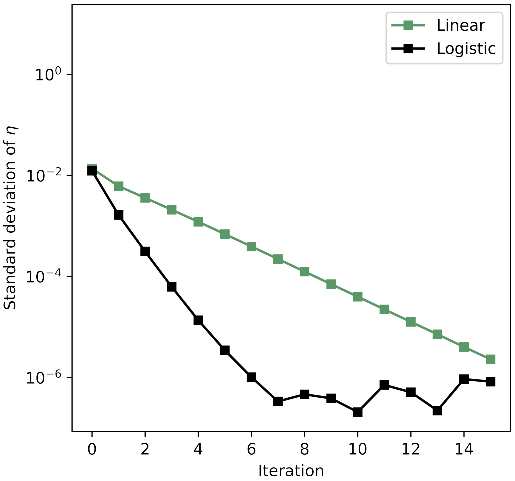


Figure 10.3: The standard deviation of the example-level η over iterations of the IRFIL algorithm.